CHAPTER FOUR

Solution Methods of Finite Difference Equations

In this chapter, solution methods for elliptic, parabolic, hyperbolic equations, and Burgers' equations are presented. These equations do not represent actual fluid dynamics problems, but the methods discussed in this chapter will form the basis for solving incompressible and compressible flow problems which are presented in Chapters 5 and 6, respectively. Although the computational schemes for these equations have been in existence for many years and are well documented in other text books, they are summarized here merely for the sake of completeness and for references in later chapters.

4.1 ELLIPTIC EQUATIONS

Elliptic equations represent one of the fundamental building blocks in fluid mechanics. Steady heat conduction, diffusion processes in viscous, turbulent, and boundary layer flows, as well as chemically reacting flows are characterized by the elliptic nature of the governing equations. Various difference schemes for the elliptic equations and some solution methods are also presented in this chapter.

4.1.1 FINITE DIFFERENCE FORMULATIONS

Consider the Laplace equation which is one of the typical elliptic equations,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{4.1.1}$$

The five-point and nine-point finite differences for the Laplace equation are, respectively,

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} = 0 \tag{4.1.2}$$

$$\frac{-u_{i-2,j} + 16u_{i-1,j} - 30u_{i,j} + 16u_{i+1,j} - u_{i+2,j}}{12\Delta x^2} + \frac{-u_{i,j-2} + 16u_{i,j-1} - 30u_{i,j} + 16u_{i,j+1} - u_{i,j+2}}{12\Delta y^2} = 0 \tag{4.1.3}$$

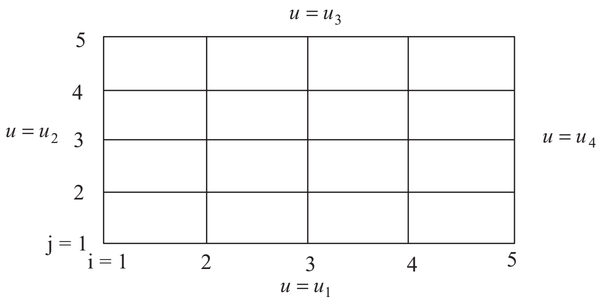


Figure 4.1.1 Finite difference grids with Dirichlet boundary conditions specified at all boundary nodes.

as discussed in Chapter 3. For illustration, let us consider the five-point scheme (4.1.2) for the geometry given in Figure 4.1.1.

$$u_{i+1,j} + u_{i-1,j} + \beta^2 u_{i,j+1} + \beta^2 u_{i,j-1} - 2(1+\beta^2)u_{i,j} = 0 \tag{4.1.4}$$

where β is defined as $\beta = \Delta x/\Delta y$. For Dirichlet boundary conditions, the values of u at all boundary nodes are given. Thus, writing (4.1.4) at all interior nodes and setting

$$\gamma = -2(1+\beta^2)$$

we obtain for the discretization as shown in Figure 4.1.1,

$$\begin{bmatrix} \gamma & 1 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 \\ 1 & \gamma & 1 & 0 & \beta^2 & 0 & 0 & 0 & 0 \\ 0 & 1 & \gamma & 0 & 0 & \beta^2 & 0 & 0 & 0 \\ \beta^2 & 0 & 0 & \gamma & 1 & 0 & \beta^2 & 0 & 0 \\ 0 & \beta^2 & 0 & 1 & \gamma & 1 & 0 & \beta^2 & 0 \\ 0 & 0 & \beta^2 & 0 & 1 & \gamma & 0 & 0 & \beta^2 \\ 0 & 0 & 0 & \beta^2 & 0 & 0 & \gamma & 1 & 0 \\ 0 & 0 & 0 & 0 & \beta^2 & 0 & 1 & \gamma & 1 \\ 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 1 & \gamma \end{bmatrix} \begin{bmatrix} u_{2,2} \\ u_{3,2} \\ u_{4,2} \\ u_{2,3} \\ u_{3,3} \\ u_{4,3} \\ u_{2,4} \\ u_{3,4} \\ u_{4,4} \end{bmatrix} = \begin{bmatrix} -u_{1,2} - \beta^2 u_{2,1} \\ -\beta^2 u_{3,1} \\ -u_{5,2} - \beta^2 u_{4,1} \\ -u_{1,3} \\ 0 \\ -u_{5,3} \\ -u_{1,4} - \beta^2 u_{2,5} \\ -\beta^2 u_{3,5} \\ -u_{5,4} - \beta^2 u_{4,5} \end{bmatrix} \tag{4.1.5}$$

Notice that the matrix on the left-hand side is always pentadiagonalized for the five-point scheme. The nine-point schemes given by (4.1.3), although more complicated, can be written similarly as in (4.1.5).

There are two types of solution methods for the linear algebraic equations of the form (4.1.5). The first kind includes the direct methods such as Gauss elimination, Thomas algorithm, Chelosky method, etc. The second kind includes the iterative methods such as Jacobi iteration, point Gauss-Seidel iteration, line Gauss-Seidel iteration, point-successive over-relaxation (PSOR), line successive over-relaxation (LSOR), alternating direction implicit (ADI), and so on.

The disadvantage of the direct methods is that they are more time consuming than iterative methods. Additionally, direct methods are susceptible to round-off errors which, in large systems of equations, can be catastrophic. In contrast, errors in each step of an iterative method are corrected in the subsequent step, thus round-off errors are usually not a concern. We elaborate on some of the iterative methods in Section 4.1.2, and a direct method of Gaussian elimination in Section 4.1.3. Other methods will be presented in later chapters, including conjugate gradient methods (CGM) (Section 10.3.1) and generalized minimal residual (GMRES) algorithm (Section 11.5.3).

4.1.2 ITERATIVE SOLUTION METHODS

Jacobi Iteration Method

In this method, the unknown u at each grid point is solved in terms of the initial guess values or previously computed values. Thus, from (4.1.4), we compute a new value of $u_{i,j}$ at the new iteration $k+1$ level as

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^k + u_{i-1,j}^k + \beta^2\left(u_{i,j+1}^k + u_{i,j-1}^k\right)\right] \tag{4.1.6}$$

where k represents the previously computed values or the initial guesses for the first round of computations. The computation is carried out until a specified convergence criterion is achieved.

We may use the newly computed values of the dependent variables to compute the neighboring points when available. This process leads to efficient schemes such as the Gauss-Seidel method.

Point Gauss-Seidel Iteration Method

In this method, the current values of the dependent variables are used to compute neighboring points as soon as they are available. This will increase the convergence rate. The solution for the independent variables is obtained as

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^k + u_{i-1,j}^{k+1} + \beta^2\left(u_{i,j+1}^k + u_{i,j-1}^{k+1}\right)\right] \tag{4.1.7}$$

The $k+1$ level on the right-hand side of (4.1.7) indicates that the solution process takes advantage of the values at $i-1$ and $j-1$ which have just been calculated in the previous step.

Line Gauss-Seidel Iteration Method

Equation (4.1.5) may be solved for the three unknowns at $(i-1, j)$, (i, j), $(i+1, j)$, as follows:

$$u_{i-1,j}^{k+1} - 2(1+\beta^2)u_{i,j}^{k+1} + u_{i+1,j}^{k+1} = -\beta^2\left(u_{i,j+1}^k + u_{i,j-1}^{k+1}\right) \tag{4.1.8}$$

which leads to a tridiagonal matrix. Note that $u_{i,j-1}^{k+1}$ is known at the $k+1$ level, whereas $u_{i,j+1}^k$ was determined at the kth level. This method converges faster than the point Gauss-Seidel method, but it takes more computer time per iteration. The line iteration technique is useful when the variable changes more rapidly in the direction of the iteration because of the use of the updated values.

 https://doi.org/10.1017/CBO9780511780066.008

Point Successive Over-Relaxation Method (PSOR)

Convergence of the point Gauss-Seidel method can be accelerated by rearranging (4.1.7),

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}\right) - 2(1+\beta^2)u_{i,j}^{k}\right] \tag{4.1.9}$$

The idea is to make $u_{i,j}^{k}$ approach $u_{i,j}^{k+1}$ faster. To this end, we introduce the relaxation parameter, ω, to be multiplied to the terms with brackets on the right-hand side of (4.1.9),

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

or

$$u_{i,j}^{k+1} = (1-\omega)u_{i,j}^{k} + \frac{\omega}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}\right)\right] \tag{4.1.10}$$

where we choose $1 < \omega < 2$ for convergence. This is known as the point successive over-relaxation procedure. For certain problems, however, a better convergence may be achieved by under-relaxation, where the relaxation parameter is chosen as $0 < \omega < 1$. Note that for $\omega = 1$ we recover the Gauss-Seidel iteration method.

For a rectangular domain subjected to Dirichlet boundary conditions with constant step size, we obtain the optimum relaxation parameter

$$\omega_{opt} = \frac{2-\sqrt{1-a}}{a} \tag{4.1.11}$$

with

$$a = \left[\frac{\cos\left(\dfrac{\pi}{IM-1}\right) + \beta^2\cos\left(\dfrac{\pi}{JM-1}\right)}{1+\beta^2}\right]^2 \tag{4.1.12}$$

where IM and JM refer to the maximum numbers of i and j, respectively. Further details are found in Wachspress [1966] and Hageman and Young [1981].

Line Successive Over-Relaxation Method (LSOR)

The idea of relaxation may also be applied to the line Gauss-Seidel method,

$$\omega u_{i-1,j}^{k+1} - 2(1+\beta^2)u_{i,j}^{k+1} + \omega u_{i+1,j}^{k+1} = -(1-\omega)\lfloor 2(1+\beta^2)\rfloor u_{i,j}^{k} - \omega\beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}\right) \tag{4.1.13}$$

where an optimum relaxation parameter ω can be determined experimentally, or by (4.1.11).

Alternating Direction Implicit (ADI) Method

In this method, a tridiagonal system is solved for rows first and then followed by columns, or vice versa. Toward this end, we recast (4.1.8) into two parts:

$$u_{i-1,j}^{k+\frac{1}{2}} - 2(1+\beta^2)u_{i,j}^{k+\frac{1}{2}} + u_{i+1,j}^{k+\frac{1}{2}} = -\beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+\frac{1}{2}}\right) \tag{4.1.14a}$$

and

$$\beta^2 u_{i,j-1}^{k+1} - 2(1+\beta^2)u_{i,j}^{k+1} + \beta^2 u_{i,j+1}^{k+1} = -\left(u_{i+1,j}^{k+\frac{1}{2}} + u_{i-1,j}^{k+1}\right) \tag{4.1.14b}$$

Here (4.1.14a) and (4.1.14b) are solved implicitly in the x-direction and y-direction, respectively. The relaxation parameter ω may be introduced to accelerate the convergence,

$$\omega u_{i-1,j}^{k+\frac{1}{2}} - 2(1+\beta^2)u_{i,j}^{k+\frac{1}{2}} + \omega u_{i+1,j}^{k+\frac{1}{2}} = -(1-\omega)[2(1+\beta^2)]u_{i,j}^{k} - \omega\beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+\frac{1}{2}}\right) \tag{4.1.15a}$$

and

$$\omega\beta^2 u_{i,j-1}^{k+1} - 2(1+\beta^2)u_{i,j}^{k+1} + \omega\beta^2 u_{i,j+1}^{k+1} = -(1-\omega)[2(1+\beta^2)]u_{i,j}^{k+\frac{1}{2}} - \omega\left(u_{i+1,j}^{k+\frac{1}{2}} + u_{i-1,j}^{k+1}\right) \tag{4.1.15b}$$

with the optimum ω being determined experimentally as appropriate for different physical problems.

4.1.3 DIRECT METHOD WITH GAUSSIAN ELIMINATION

Consider the simultaneous equations resulting from the finite difference approximation of (4.1.2) in the form

$$\begin{aligned}
&k_{11}u_1 + k_{12}u_2 + \cdots = g_1 \\
&k_{21}u_1 + k_{22}u_2 + \cdots = g_2 \\
&\vdots \\
&k_{n1}u_n \cdot \qquad\qquad \cdots = g_n
\end{aligned} \tag{4.1.16}$$

Here, our objective is to transform the system into an upper triangular array. To this end, we choose the first row as the "pivot" equation and eliminate the u_1 term from each equation below it. To eliminate u_1 from the second equation, we multiply the first equation by k_{21}/k_{11} and subtract it from the second equation. We continue similarly until u_1 is eliminated from all equations. We then eliminate $u_2, u_3, \ldots$ in the same manner until we achieve the upper triangular form,

$$\begin{bmatrix} k_{11} & k_{12} & \cdot & \cdot \\ & k'_{22} & \cdot & \cdot \\ & & \cdot & \cdot \\ & & & k'_{nn} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ u_n \end{bmatrix} = \begin{bmatrix} g_1 \\ g'_2 \\ \cdot \\ g'_n \end{bmatrix} \tag{4.1.17}$$

It is seen that backsubstitution will determine all unknowns.

An example for the solution of a typical elliptical equation is shown in Section 4.7.1.

4.2 PARABOLIC EQUATIONS

The governing equations for some problems in fluid dynamics, such as unsteady heat conduction or boundary layer flows, are parabolic. The finite difference representation

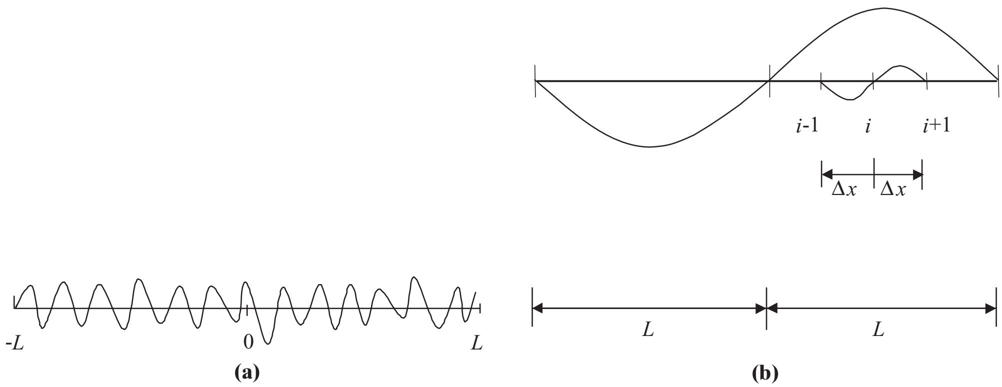


Figure 4.2.1 Fourier representation of the error on interval $(-L, L)$. **(a)** Error distribution. **(b)** Maximum and minimum wavelength.

of these equations may be represented in either explicit or implicit schemes, as illustrated below.

4.2.1 EXPLICIT SCHEMES AND VON NEUMANN STABILITY ANALYSIS

Forward-Time/Central-Space (FTCS) Method

A typical parabolic equation is the unsteady diffusion problem characterized by

$$\frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2} = 0 \tag{4.2.1}$$

An explicit finite difference equation scheme for (4.2.1) may be written in the forward difference in time and central difference in space (FTCS) as (see Figure 4.2.1a)

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\alpha\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)}{\Delta x^2} + O(\Delta t, \Delta x^2) \tag{4.2.2a}$$

or

$$u_i^{n+1} = u_i^n + d\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \tag{4.2.2b}$$

where d is the diffusion number

$$d = \frac{\alpha \Delta t}{\Delta x^2} \tag{4.2.3}$$

By definition, (4.2.2) is explicit because u_i^{n+1} at time step $n+1$ can be solved *explicitly* in terms of the known quantities at the previous time step n, thus called an *explicit scheme*.

In order to determine the stability of the solution of finite difference equations, it is convenient to expand the difference equation in a Fourier series. Decay or growth of an amplification factor indicates whether or not the numerical algorithm is stable. This is known as the von Neumann stability analysis [Ortega and Rheinbolt, 1970]. Assuming that at any time step n, the computed solution u_i^n is the sum of the exact solution $\bar{u}_i^n$ and error ε_i^n

$$u_i^n = \bar{u}_i^n + \varepsilon_i^n \tag{4.2.4}$$

and substituting (4.2.4) into (4.2.2a), we obtain

$$\frac{\bar{u}_i^{n+1} - \bar{u}_i^n}{\Delta t} + \frac{\varepsilon_i^{n+1} - \varepsilon_i^n}{\Delta t} = \frac{\alpha}{(\Delta x)^2}\left(\bar{u}_{i+1}^n - 2\bar{u}_i^n + \bar{u}_{i-1}^n\right) + \frac{\alpha}{(\Delta x)^2}\left(\varepsilon_{i+1}^n - 2\varepsilon_i^n + \varepsilon_{i-1}^n\right) \tag{4.2.5}$$

or

$$\frac{\varepsilon_i^{n+1} - \varepsilon_i^n}{\Delta t} = \frac{\alpha}{(\Delta x)^2}\left(\varepsilon_{i+1}^n - 2\varepsilon_i^n + \varepsilon_{i-1}^n\right) \tag{4.2.6}$$

Writing (4.2.4) – (4.2.6) for the entire domain leads to

$$\mathbf{U}^n = \bar{\mathbf{U}}^n + \boldsymbol{\varepsilon}^n \tag{4.2.7}$$

with

$$\boldsymbol{\varepsilon}^n = \begin{bmatrix} \cdot \\ \varepsilon_{i-1}^n \\ \varepsilon_i^n \\ \varepsilon_{i+1}^n \\ \cdot \end{bmatrix} \tag{4.2.8}$$

$$\bar{\mathbf{U}}^{n+1} + \boldsymbol{\varepsilon}^{n+1} = \mathbf{C}(\bar{\mathbf{U}}^n + \boldsymbol{\varepsilon}^n) \tag{4.2.9}$$

$$\boldsymbol{\varepsilon}^{n+1} = \mathbf{C}\boldsymbol{\varepsilon}^n \tag{4.2.10}$$

with

$$C = 1 + d(E - 2 + E^{-1}) = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ d & (1-2d) & d & 0 & 0 \\ \cdot & d & (1-2d) & d & 0 \\ \cdot & 0 & d & (1-2d) & d \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \tag{4.2.11}$$

If the boundary conditions are considered as periodic, the error ε^n can be decomposed into a Fourier series in space at each time level n. The fundamental frequency in a one-dimensional domain between $-L$ and L (Figure 4.2.1) corresponds to the maximum wave length of $\lambda_{\max} = 2L$. The wave number $k = 2\pi/\lambda$ becomes minimum as $k_{\min} = \pi/L$, whereas the maximum wave number $k_{\max}$ is associated with the shortest wavelength λ on a mesh with spacing Δx corresponding to $\lambda_{\min} = 2\Delta x$, leading to $k_{\max} = \pi/\Delta x$. Thus, the harmonics on a finite mesh are

$$k_j = jk_{\min} = j\pi/L = j\pi/(N\Delta x), \quad j = 0, 1, \ldots N \tag{4.2.12}$$

with $\Delta x = L/N$. The highest value of j is equal to the number of mesh intervals N. Any finite mesh function, such as ε_i^n or the full solution u_i^n, can be decomposed into a Fourier series

$$\varepsilon_i^n = \sum_{j=-N}^{N} \bar{\varepsilon}_j^n \, e^{Ik_j(i\Delta x)} = \sum_{j=-N}^{N} \bar{\varepsilon}_j^n \, e^{Iji\pi/N} \tag{4.2.13}$$

with $I = \sqrt{-1}$, $\bar{\varepsilon}_j^n$ being the amplitude of the j^{th} harmonic, and the spatial phase angle ϕ is given as

$$\phi = k_j \Delta x = j\pi/N \tag{4.2.14}$$

with $\phi = \pi$ corresponding to the highest frequency resolvable on the mesh, namely the frequency of the wavelength $2\Delta x$. Thus

$$\varepsilon_i^n = \sum_{j=-N}^{N} \bar{\varepsilon}_j^n \, e^{Ii\phi} \tag{4.2.15}$$

Substituting (4.2.15) into (4.2.6) yields

$$\frac{\bar{\varepsilon}^{n+1} - \bar{\varepsilon}^n}{\Delta t} e^{Ii\phi} = \frac{\alpha}{\Delta x^2}\left(\bar{\varepsilon}^n e^{I(i+1)\phi} - 2\bar{\varepsilon}^n e^{Ii\phi} + \bar{\varepsilon}^n e^{I(i-1)\phi}\right)$$

or

$$\bar{\varepsilon}^{n+1} - \bar{\varepsilon}^n - d\bar{\varepsilon}^n(e^{I\phi} - 2 + e^{-I\phi}) = 0 \tag{4.2.16}$$

The computational scheme is said to be stable if the amplitude of any error harmonic $\bar{\varepsilon}^n$ does not grow in time, that is, if the following ratio holds:

$$|g| = \left|\frac{\bar{\varepsilon}^{n+1}}{\bar{\varepsilon}^n}\right| \leq 1 \quad \text{for all } \phi \tag{4.2.17}$$

where $g = \bar{\varepsilon}^{n+1}/\bar{\varepsilon}^n$ is the amplification factor, and is a function of time step Δt, frequency, and the mesh size Δx. It follows from (4.2.16) that

$$g = 1 + d(e^{I\phi} - 2 + e^{-I\phi}) \tag{4.2.18a}$$

or

$$g = 1 - 2d(1 - \cos\phi) \tag{4.2.18b}$$

Thus, the stability condition is

$$g \leq 1 \tag{4.2.19}$$

or

$$1 - 2d(1 - \cos\phi) \geq -1 \tag{4.2.20}$$

Since the maximum of $1 - \cos\phi$ is 2, we arrive at, for stability,

$$0 \leq d \leq 1/2 \tag{4.2.21}$$

The von Neumann stability analysis shown above can be used to determine the computational stability properties of other finite difference schemes to be discussed subsequently.

OTHER EXPLICIT SCHEMES

Richardson Method

If the diffusion equation (4.2.1) is modeled by the form

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \frac{\alpha\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)}{\Delta x^2}, \quad O(\Delta t^2, \Delta x^2) \tag{4.2.22}$$

This is known as the Richardson method and is unconditionally unstable.

Dufort-Frankel Method

The finite difference equation for this method is given by

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \frac{\alpha\left(u_{i+1}^n - 2\dfrac{u_i^{n+1} + u_i^{n-1}}{2} + u_{i-1}^n\right)}{\Delta x^2} \tag{4.2.23a}$$

or

$$(1 + 2d)u_i^{n+1} = (1 - 2d)u_i^{n-1} + 2d\left(u_{i+1}^n + u_{i-1}^n\right), \quad O(\Delta t^2, \Delta x^2, (\Delta t/\Delta x)^2) \tag{4.2.23b}$$

This scheme can be shown to be unconditionally stable by the von Neumann stability analysis.

4.2.2 IMPLICIT SCHEMES

Laasonen Method

Contrary to the explicit schemes, the solution for *implicit schemes* involves the variables at more than one nodal point for the time step $(n + 1)$. For example, we may write the difference equation for (4.2.1a) in the form

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\alpha\left(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right)}{\Delta x^2}, \quad O(\Delta t, \Delta x^2) \tag{4.2.24}$$

This equation is written for all grid points at $n + 1$ time step, leading to a tridiagonal form. The scheme given by (4.2.24) is known as the Laasonen method. This is unconditionally stable.

Crank-Nicolson Method

An alternative scheme of (4.2.24) is to replace the diffusion term by an average between n and $n + 1$,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\alpha}{2}\left[\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} + \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}\right], \quad O(\Delta t^2, \Delta x^2) \tag{4.2.25}$$

This may be rewritten as

$$A + B = C + D \tag{4.2.26}$$

where

$$A = \frac{u_i^{n+\frac{1}{2}} - u_i^n}{\Delta t/2}, \qquad B = \frac{u_i^{n+1} - u_i^{n+\frac{1}{2}}}{\Delta t/2}, \qquad C = \frac{\alpha\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right)}{(\Delta x)^2},$$

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

Note that $A = C$ and $B = D$ represent explicit and implicit scheme, respectively. This scheme is known as the Crank-Nicolson method. It is seen that $A = C$ is solved explicitly for the time step $n + 1/2$ and the result is substituted into $B = D$. The scheme is unconditionally stable.

β-Method

A general form of the finite difference equation for (4.2.1) may be written as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha\left[\frac{\beta\left(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right)}{(\Delta x)^2} + \frac{(1-\beta)\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)}{(\Delta x)^2}\right] \tag{4.2.27}$$

This is known as the β-method. For $1/2 \leq \beta \leq 1$, the method is unconditionally stable. For $\beta = 1/2$, equation (4.2.27) reduces to the Crank-Nicolson scheme, whereas $\beta = 0$ leads to the FTCS method.

A numerical example for the solution of a typical parabolic equation characterized by Couette flow is presented in Section 4.7.2.

4.2.3 ALTERNATING DIRECTION IMPLICIT (ADI) SCHEMES

Let us now examine the solution of the two-dimensional diffusion equation,

$$\frac{\partial u}{\partial t} - \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0 \tag{4.2.28}$$

with the forward difference in time and the central difference in space (FTCS). We write an explicit scheme in the form

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \alpha\left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta y^2}\right), \quad O(\Delta t, \Delta x^2, \Delta y^2) \tag{4.2.29}$$

It can be shown that the system is stable if

$$d_x + d_y \leq \frac{1}{2} \tag{4.2.30}$$

Here, diffusion numbers d_x and d_y are defined as

$$d_x = \frac{\alpha \Delta t}{\Delta x^2}, \qquad d_y = \frac{\alpha \Delta t}{\Delta y^2} \tag{4.2.31}$$

For simplicity, let $d_x = d_y = d$ for $\Delta x = \Delta y$. This will give $d \leq 1/4$ for stability, which is twice as restrictive. To avoid this restriction, consider an implicit scheme

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \alpha\left(\frac{u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2}\right) \tag{4.2.32}$$

or

$$d_x u_{i+1,j}^{n+1} + d_x u_{i-1,j}^{n+1} - (2d_x + 2d_y + 1)u_{i,j}^{n+1} + d_y u_{i,j+1}^{n+1} + d_y u_{i,j-1}^{n+1} = -u_{i,j}^{n} \tag{4.2.33}$$

This leads to a pentadiagonal system.

An alternative is to use the alternating direction implicit scheme, by splitting (4.2.25) into two equations:

$$\frac{u_{i,j}^{n+\frac{1}{2}} - u_{i,j}^{n}}{\Delta t/2} = \alpha\left(\frac{u_{i+1,j}^{n+\frac{1}{2}} - 2u_{i,j}^{n+\frac{1}{2}} + u_{i-1,j}^{n+\frac{1}{2}}}{\Delta x^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2}\right) \tag{4.2.34a}$$

and

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n+\frac{1}{2}}}{\Delta t/2} = \alpha\left(\frac{u_{i+1,j}^{n+\frac{1}{2}} - 2u_{i,j}^{n+\frac{1}{2}} + u_{i-1,j}^{n+\frac{1}{2}}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2}\right) \tag{4.2.34b}$$

This scheme is unconditionally stable. These two equations can be written in a tridiagonal form as follows:

$$\underbrace{-d_1 u_{i+1,j}^{n+\frac{1}{2}} + (1+2d_1)u_{i,j}^{n+\frac{1}{2}} - d_1 u_{i-1,j}^{n+\frac{1}{2}}}_{\text{implicit in } x\text{-direction}} = \underbrace{d_2 u_{i,j+1}^{n} + (1-2d_2)u_{i,j}^{n} + d_2 u_{i,j-1}^{n}}_{\text{explicit in } y\text{-direction}} \tag{4.2.35a}$$

$$\underbrace{-d_2 u_{i,j+1}^{n+1} + (1+2d_2)u_{i,j}^{n+1} - d_2 u_{i,j-1}^{n+1}}_{\text{unknown}} = \underbrace{d_1 u_{i+1,j}^{n+\frac{1}{2}} + (1-2d_1)u_{i,j}^{n+\frac{1}{2}} + d_1 u_{i-1,j}^{n+\frac{1}{2}}}_{\text{known}} \tag{4.2.35b}$$

where

$$d_1 = \frac{1}{2}d_x = \frac{1}{2}\frac{\alpha \Delta t}{\Delta x^2}$$

$$d_2 = \frac{1}{2}d_y = \frac{1}{2}\frac{\alpha \Delta t}{\Delta y^2}$$

Note that (4.2.35a) is implicit in the x-direction and explicit in the y-direction, known as the x-sweep. The solution of (4.2.35a) provides the data for (4.2.35b) so that the y-sweep can be carried out in which the solution is implicit in the y-direction and explicit in the x-direction.

4.2.4 APPROXIMATE FACTORIZATION

The ADI formulation can be shown to be an approximate factorization of the Crank-Nicolson scheme. To this end, let us write the Crank-Nicolson scheme for (4.2.25) in the form

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}=\frac{\alpha}{2}\left[\begin{array}{l}\dfrac{u_{i+1,j}^{n+1}-2u_{i,j}^{n+1}+u_{i-1,j}^{n+1}}{\Delta x^2}+\dfrac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x^2}\\+\dfrac{u_{i,j+1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j-1}^{n+1}}{\Delta y^2}+\dfrac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{\Delta y^2}\end{array}\right],$$

$$O(\Delta t^2,\Delta x^2,\Delta y^2) \quad (4.2.36)$$

Introducing a compact notation,

$$\delta_x^2 u_{i,j}=u_{i+1,j}-2u_{i,j}+u_{i-1,j}$$
$$\delta_y^2 u_{i,j}=u_{i,j+1}-2u_{i,j}+u_{i,j-1}$$

we may rewrite (4.2.36) as

$$\left[1-\frac{1}{2}\left(d_x\delta_x^2+d_y\delta_y^2\right)\right]u_{i,j}^{n+1}=\left[1+\frac{1}{2}\left(d_x\delta_x^2+d_y\delta_y^2\right)\right]u_{i,j}^{n} \quad (4.2.37)$$

To compare (4.2.37) with the ADI formulation, we use (4.2.36) to rewrite the ADI equations as

$$\frac{u_{i,j}^{n+\frac{1}{2}}-u_{i,j}^{n}}{\frac{\Delta t}{2}}=\alpha\left(\frac{\delta_x^2u_{i,j}^{n+\frac{1}{2}}}{\Delta x^2}+\frac{\delta_y^2u_{i,j}^{n}}{\Delta y^2}\right) \quad (4.2.38a)$$

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n+\frac{1}{2}}}{\frac{\Delta t}{2}}=\alpha\left(\frac{\delta_x^2u_{i,j}^{n+\frac{1}{2}}}{\Delta x^2}+\frac{\delta_y^2u_{i,j}^{n+1}}{\Delta y^2}\right) \quad (4.2.38b)$$

Rearranging (4.2.38a,b)

$$\left(1-\frac{1}{2}d_x\delta_x^2\right)u_{i,j}^{n+\frac{1}{2}}=\left(1+\frac{1}{2}d_y\delta_y^2\right)u_{i,j}^{n} \quad (4.2.39a)$$

$$\left(1-\frac{1}{2}d_y\delta_y^2\right)u_{i,j}^{n+1}=\left(1+\frac{1}{2}d_x\delta_x^2\right)u_{i,j}^{n+\frac{1}{2}} \quad (4.2.39b)$$

and eliminating $u_{i,j}^{n+\frac{1}{2}}$ between (4.2.39a) and (4.2.39b),

$$\left(1-\frac{1}{2}d_x\delta_x^2\right)\left(1-\frac{1}{2}d_y\delta_y^2\right)u_{i,j}^{n+1}=\left(1+\frac{1}{2}d_x\delta_x^2\right)\left(1+\frac{1}{2}d_y\delta_y^2\right)u_{i,j}^{n} \quad (4.2.40)$$

or

$$\left[1-\frac{1}{2}\left(d_x\delta_x^2+d_y\delta_y^2\right)+\frac{1}{4}d_xd_y\delta_x^2\delta_y^2\right]u_{i,j}^{n+1}=\left[1+\frac{1}{2}\left(d_x\delta_x^2+d_y\delta_y^2\right)+\frac{1}{4}d_xd_y\delta_x^2\delta_y^2\right]u_{i,j}^{n} \quad (4.2.41)$$

We note that, compared to (4.2.37), the additional term in (4.2.41)

$$\frac{1}{4}d_xd_y\delta_x^2\delta_y^2\left(u_{i,j}^{n+1}-u_{i,j}^{n}\right)$$

is smaller than the truncation error of (4.2.37). Thus, it is seen that the ADI formulation is an approximate factorization of the Crank-Nicolson scheme.

4.2.5 FRACTIONAL STEP METHODS

An approximation of multidimensional problems similar to ADI or approximate factorization schemes is also known as the method of fractional steps. This method splits the multidimensional equations into a series of one-dimensional equations and solves them sequentially. For example, consider a two-dimensional equation

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{4.2.42}$$

The Crank-Nicolson scheme for (4.2.36) can be written in two steps:

$$\frac{u_{i,j}^{n+\frac{1}{2}} - u_{i,j}^{n}}{\frac{\Delta t}{2}} = \frac{\alpha}{2}\left[\frac{u_{i+1,j}^{n+\frac{1}{2}} - 2u_{i,j}^{n+\frac{1}{2}} + u_{i-1,j}^{n+\frac{1}{2}}}{\Delta x^2} + \frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2}\right] \tag{4.2.43a}$$

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n+\frac{1}{2}}}{\frac{\Delta t}{2}} = \frac{\alpha}{2}\left[\frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2} + \frac{u_{i,j+1}^{n+\frac{1}{2}} - 2u_{i,j}^{n+\frac{1}{2}} + u_{i,j-1}^{n+\frac{1}{2}}}{\Delta y^2}\right] + O(\Delta t^2, \Delta x^2, \Delta y^2) \tag{4.2.43b}$$

This scheme is unconditionally stable.

4.2.6 THREE DIMENSIONS

The ADI method can be extended to three-space dimensions for the time intervals $n, n+1/3, n+2/3$, and $n+1$. Consider the unsteady diffusion problem,

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \tag{4.2.44}$$

The three-step FDM equations are written as

$$\frac{u_{i,j,k}^{n+\frac{1}{3}} - u_{i,j,k}^{n}}{\Delta t/3} = \alpha\left(\frac{\delta_x^2 u_{i,j,k}^{n+\frac{1}{3}}}{\Delta x^2} + \frac{\delta_y^2 u_{i,j,k}^{n}}{\Delta y^2} + \frac{\delta_z^2 u_{i,j,k}^{n}}{\Delta z^2}\right) \tag{4.2.45a}$$

$$\frac{u_{i,j,k}^{n+\frac{2}{3}} - u_{i,j,k}^{n+\frac{1}{3}}}{\Delta t/3} = \alpha\left(\frac{\delta_x^2 u_{i,j,k}^{n+\frac{1}{3}}}{\Delta x^2} + \frac{\delta_y^2 u_{i,j,k}^{n+\frac{2}{3}}}{\Delta y^2} + \frac{\delta_z^2 u_{i,j,k}^{n+\frac{1}{3}}}{\Delta z^2}\right) \tag{4.2.45b}$$

$$\frac{u_{i,j,k}^{n+1} - u_{i,j,k}^{n+\frac{2}{3}}}{\Delta t/3} = \alpha\left(\frac{\delta_x^2 u_{i,j,k}^{n+\frac{2}{3}}}{\Delta x^2} + \frac{\delta_y^2 u_{i,j,k}^{n+\frac{2}{3}}}{\Delta y^2} + \frac{\delta_z^2 u_{i,j,k}^{n+1}}{\Delta z^2}\right), \quad O(\Delta t, \Delta x^2, \Delta y^2, \Delta z^2) \tag{4.2.45c}$$

This method is conditionally stable with $(d_x + d_y + d_z) \le 3/2$. A more efficient method may be derived using the Crank-Nicolson scheme.

$$\begin{aligned}
\frac{u^{*}_{i,j,k} - u^{n}_{i,j,k}}{\Delta t} &= \alpha\left[\frac{1}{2}\frac{\delta_x^2 u^{*}_{i,j,k} + \delta_x^2 u^{n}_{i,j,k}}{\Delta x^2} + \frac{\delta_y^2 u^{n}_{i,j,k}}{\Delta y^2} + \frac{\delta_z^2 u^{n}_{i,j,k}}{\Delta z^2}\right] \\
\frac{u^{**}_{i,j,k} - u^{n}_{i,j,k}}{\Delta t} &= \alpha\left[\frac{1}{2}\frac{\delta_x^2 u^{*}_{i,j,k} + \delta_x^2 u^{n}_{i,j,k}}{\Delta x^2} + \frac{1}{2}\frac{\delta_y^2 u^{**}_{i,j,k} + \delta_y^2 u^{n}_{i,j,k}}{\Delta y^2} + \frac{\delta_z^2 u^{n}_{i,j,k}}{\Delta z^2}\right] \\
\frac{u^{n+1}_{i,j,k} - u^{n}_{i,j,k}}{\Delta t} &= \alpha\left[\frac{1}{2}\frac{\delta_x^2 u^{*}_{i,j,k} + \delta_x^2 u^{n}_{i,j,k}}{\Delta x^2} + \frac{1}{2}\frac{\delta_y^2 u^{**}_{i,j,k} + \delta_y^2 u^{n}_{i,j,k}}{\Delta y^2} + \frac{1}{2}\frac{\delta_z^2 u^{n+1}_{i,j,k} + \delta_z^2 u^{n}_{i,j,k}}{\Delta z^2}\right]
\end{aligned} \tag{4.2.46}$$

In this scheme, the final solution $u^{n+1}_{i,j,k}$ is obtained in terms of the intermediate steps $u^{*}_{i,j,k}$ and $u^{**}_{i,j,k}$.

4.2.7 DIRECT METHOD WITH TRIDIAGONAL MATRIX ALGORITHM

Consider the implicit FDM discretization for the transient heat conduction equation in the form,

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{\alpha}{\Delta x^2}\left(T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}\right) \tag{4.2.47}$$

This may be rewritten as

$$a_i T_{i-1}^{n+1} + b_i T_i^{n+1} + c_i T_{i+1}^{n+1} = g_i \tag{4.2.48}$$

with

$$a_i = c_i = -\frac{\alpha \Delta t}{\Delta x^2}, \qquad b_i = 1 + \frac{2\alpha\Delta t}{\Delta x^2}, \qquad g_i = T_i^n \tag{4.2.49}$$

If Dirichlet boundary conditions are applied to this problem, we obtain the following tridiagonal form, known as tridiagonal matrix algorithm (TDMA) or Thomas algorithm [Thomas, 1949]:

$$\begin{bmatrix}
b_1 & c_1 & 0 & \cdot & \cdot & \cdot & \cdot \\
a_2 & b_2 & c_2 & 0 & \cdot & \cdot & \cdot \\
0 & a_3 & b_3 & c_3 & 0 & \cdot & \cdot \\
\cdot & \cdot & * & * & * & \cdot & \cdot \\
\cdot & \cdot & \cdot & * & * & * & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & c_{NI-1} \\
0 & \cdot & \cdot & \cdot & \cdot & a_{NI} & b_{NI}
\end{bmatrix}
\begin{bmatrix} T_1^{n+1} \\ T_2^{n+1} \\ T_3^{n+1} \\ * \\ * \\ * \\ T_{NI}^{n+1} \end{bmatrix}
=
\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ * \\ * \\ * \\ g_{NI} \end{bmatrix} \tag{4.2.50}$$

An upper triangular form of the tridiagonal matrix may be obtained as follows:

$$b_i = b_i - \frac{a_i}{b_{i-1}} c_{i-1} \quad i = 2, 3, \ldots NI$$
$$g_i = g_i - \frac{a_i}{b_{i-1}} g_{i-1} \quad i = 2, 3, \ldots NI$$
$$T_{NI} = \frac{g_{NI}}{b_{NI}}$$
$$T_j = \frac{g_j - c_j T_{j+1}}{b_j} \quad j = NI - 1, \quad NI - 2, \ldots, 1$$

It should be noted that Neumann boundary conditions can also be accommodated into this algorithm with the tridiagonal form still maintained.

4.3 HYPERBOLIC EQUATIONS

Hyperbolic equations, in general, represent wave propagation. They are given by either first order or second order differential equations, which may be approximated in either explicit or implicit forms of finite difference equations. Various computational schemes are examined below.

4.3.1 EXPLICIT SCHEMES AND VON NEUMANN STABILITY ANALYSIS

Euler's Forward Time and Forward Space (FTFS) Approximations

Consider the first order wave equation (Euler equation) of the form

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \quad a > 0 \tag{4.3.1}$$

The Euler's forward time and forward space approximation of (4.3.1) is written in the FTFS scheme as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a\frac{u_{i+1}^n - u_i^n}{\Delta x} \tag{4.3.2}$$

It follows from (4.2.15) and (4.3.2) that the amplification factor assumes the form

$$g = 1 - C(e^{I\phi} - 1) = 1 - C(\cos\phi - 1) - IC\sin\phi = 1 + 2C\sin^2\frac{\phi}{2} - IC\sin\phi \tag{4.3.3}$$

with C being the Courant number or CFL number [Courant, Friedrichs, and Lewy, 1967],

$$C = \frac{a\Delta t}{\Delta x}$$

and

$$|g|^2 = g\,g^* = \left(1 + 2C\sin^2\frac{\phi}{2}\right)^2 + C^2\sin^2\phi = 1 + 4C(1 + C)\sin^2\frac{\phi}{2} \geq 1 \tag{4.3.4}$$

where g^* is the complex conjugate of g. Note that the criterion $|g| \leq 1$ for all values of ϕ can not be satisfied ($|g|$ lies outside the unit circle for all values of ϕ, Figure 4.3.1). Therefore, the explicit Euler scheme with FTFS is unconditionally unstable.

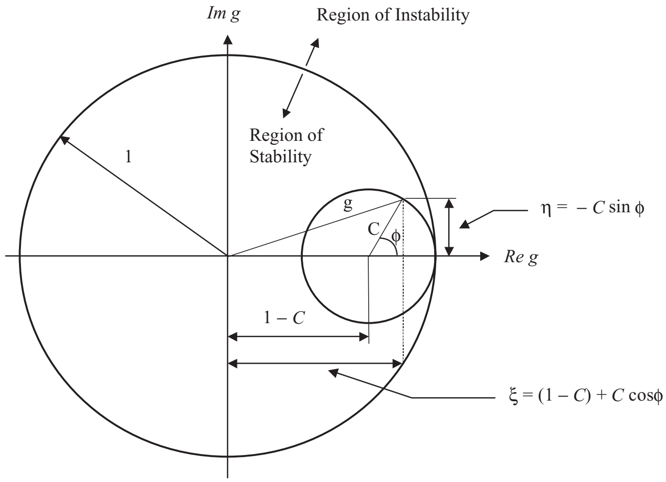


Figure 4.3.1 Complex g plane for upwind scheme with unit circle representing the stability region.

Euler's Forward Time and Central Space (FTCS) Approximations

In this method, Euler's forward time and central space approximation of (4.3.1) is used:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a\frac{\left(u_{i+1}^n - u_{i-1}^n\right)}{2\Delta x}, \quad \mathrm{O}(\Delta t, \Delta x) \tag{4.3.5}$$

The von Neumann analysis shows that this is also unconditionally unstable.

Euler's Forward Time and Backward Space (FTBS) Approximations – First Order Upwind Scheme

The Euler's forward time and backward space approximations (also known as upwind method) is given by

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a\frac{u_i^n - u_{i-1}^n}{\Delta x}, \quad \mathrm{O}(\Delta t, \Delta x) \tag{4.3.6}$$

The amplification factor takes the form

$$\begin{aligned} g &= 1 - C(1 - e^{-I\phi}) = 1 - C(1 - \cos\phi) - IC\sin\phi \\ &= 1 - 2C\sin^2\frac{\phi}{2} - IC\sin\phi \end{aligned} \tag{4.3.7}$$

or

$$g = \xi + I\eta, \qquad |g| = \left[1 - 4C(1-C)\sin^2\frac{\phi}{2}\right]^{1/2} \tag{4.3.8a,b}$$

with

$$\begin{aligned} \xi &= 1 - 2C\sin^2\frac{\phi}{2} = (1 - C) + C\cos\phi \\ \eta &= -C\sin\phi \end{aligned}$$

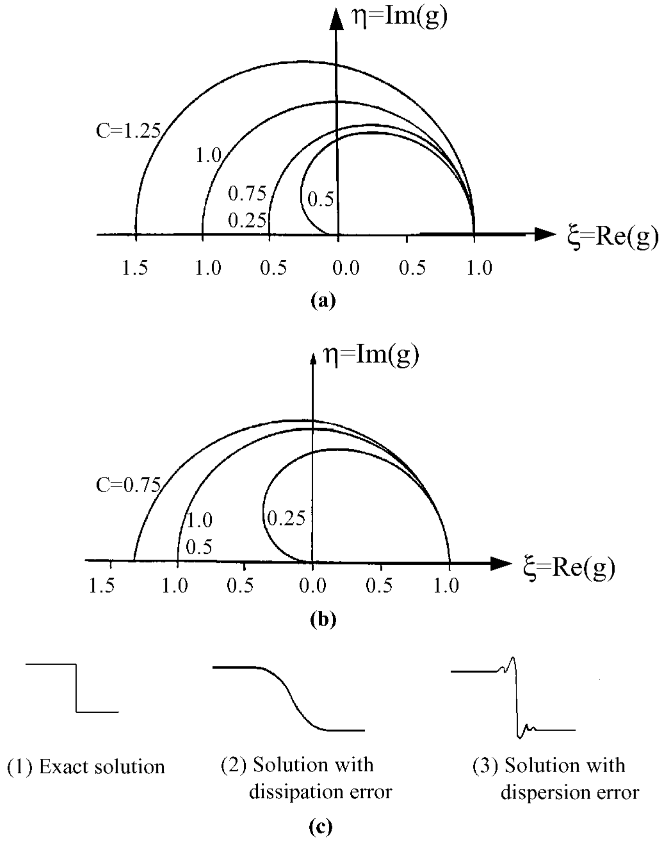


Figure 4.3.2 Dissipation and dispersion errors compared to exact solution. **(a)** Dissipation error (amplification factor modulus $|g|$). **(b)** Dispersion error (relative phase error, $\Phi/\tilde{\Phi}$). **(c)** Comparison of exact solution with dissipation error and dispersion error for shock tube problem.

which represents the parametric equation of a unit circle centered on the real axis ξ at $(1 - C)$ with radius C (Figure 4.3.1), whereas the modulus of the amplification factor, $|g|$, for various values of C are shown in Figure 4.3.2a.

In this complex plane of g, the stability condition (4.3.7) states that the curve representing g for all values of $\phi = k\Delta x$ should remain within the unit circle. It is seen that the scheme is stable for

$$0 < g < 1 \tag{4.3.9}$$

Hence, the scheme (4.3.6) is conditionally stable. Equation (4.3.9) is known as the Courant-Friedrich-Lewy (CFL) condition.

We have so far discussed the amplification factor g which represents dissipation error (Figure 4.3.2a). In numerical solutions of finite difference equations, we are also concerned with dispersion (phase) error as shown in Figure 4.3.2b. The phase Φ as determined by the adopted numerical scheme is given by the arctangent of the ratio of imaginary and real parts of g,

$$\Phi = \tan^{-1}\frac{\mathrm{Im}(g)}{\mathrm{Re}(g)} = \tan^{-1}\frac{\eta}{\xi} = \tan^{-1}\frac{-C\sin\phi}{1 - C + C\cos\phi} \tag{4.3.10}$$

The phase angle $\tilde{\Phi}$ is

$$\tilde{\Phi} = ka\Delta t = C\phi \tag{4.3.11}$$

The dispersion error or relative phase error is defined as

$$\varepsilon_\phi = \frac{\Phi}{\tilde{\Phi}} = \frac{\tan^{-1}[(-C\sin\phi)/(1 - C + C\cos\phi)]}{C\phi} \tag{4.3.12a}$$

or

$$\varepsilon_\phi \approx 1 - \frac{1}{6}(2C^2 - 3C + 1)\phi^2 \tag{4.3.12b}$$

As shown in Figure 4.3.2b, the dispersion error is said to be "leading" for $\varepsilon_\phi > 1$.

The dissipation error and dispersion error for a shock tube problem can be compared to the exact solution. This is demonstrated in Figure 4.3.2c. Here, we must choose computational schemes such that dissipation and dispersion errors are as small as possible. To this end, we review the following well-known methods.

Lax Method

In this method, an average value of u_i^n in the Euler's FTCS is used:

$$u_i^{n+1} = \frac{1}{2}\left(u_{i+1}^n + u_{i-1}^n\right) - \frac{C}{2}\left(u_{i+1}^n - u_{i-1}^n\right) \tag{4.3.13}$$

The von Neumann stability analysis shows that this scheme is stable for $C \le 1$.

Midpoint Leapfrog Method

Central differences for both time and spaces are used in this method:

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = -\frac{a\left(u_{i+1}^n - u_{i-1}^n\right)}{2\Delta x}, \quad O(\Delta t^2, \Delta x^2) \tag{4.3.14}$$

This scheme is stable for $C \le 1$. It has a second order accuracy, but requires two sets of initial values when the starter solution can provide only one set of initial data. This may lead to two independent solutions which are inaccurate.

Lax-Wendroff Method

In this method, we utilize the finite difference equation derived from Taylor series,

$$u(x, t + \Delta t) = u(x, t) + \frac{\partial u}{\partial t}\Delta t + \frac{1}{2!}\frac{\partial^2 u}{\partial t^2}\Delta t^2 + O(\Delta t^3) \tag{4.3.15a}$$

or

$$u_i^{n+1} = u_i^n + \frac{\partial u}{\partial t}\Delta t + \frac{1}{2!}\frac{\partial^2 u}{\partial t^2}\Delta t^2 + O(\Delta t^3) \tag{4.3.15b}$$

Differentiating (4.3.1) with respect to time yields

$$\frac{\partial^2 u}{\partial t^2} = -a\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\right) = a^2\frac{\partial^2 u}{\partial x^2} \tag{4.3.16}$$

Substituting (4.3.1) and (4.3.16) into (4.3.15b) leads to

$$u_i^{n+1} = u_i^n + \Delta t\left(-a\frac{\partial u}{\partial x}\right) + \frac{\Delta t^2}{2}\left(a^2\frac{\partial^2 u}{\partial x^2}\right) \tag{4.3.17}$$

Using central differencing of the second order for the spatial derivative, we obtain

$$u_i^{n+1} = u_i^n - a\Delta t\left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right) + \frac{1}{2}(a\,\Delta t)^2\left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}\right), \quad O(\Delta t^2, \Delta x^2) \tag{4.3.18}$$

This method is stable for $C \leq 1$.

4.3.2 IMPLICIT SCHEMES

Implicit schemes for approximating (4.3.1) are unconditionally stable. Two representative implicit schemes are Euler's FTCS method and the Crank-Nicolson method.

Euler's FTCS Method

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{-a}{2\Delta x}(u_{i+1}^{n+1} - u_{i-1}^{n+1}), \quad O(\Delta t, \Delta x^2) \tag{4.3.19}$$

or

$$\frac{C}{2}u_{i-1}^{n+1} - u_i^{n+1} - \frac{C}{2}u_{i+1}^{n+1} = -u_i^n \tag{4.3.20}$$

Crank-Nicolson Method

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{a}{2}\left[\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x} + \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right], \quad O(\Delta t^2, \Delta x^2) \tag{4.3.21}$$

or

$$\frac{C}{4}u_{i-1}^{n+1} - u_i^{n+1} - \frac{C}{4}u_{i+1}^{n+1} = -\frac{C}{4}u_{i-1}^n - u_i^n + \frac{C}{4}u_{i+1}^n \tag{4.3.22}$$

Examples of the numerical solution procedure for a typical first order hyperbolic equation using the explicit and implicit schemes are shown in Section 4.7.3.

4.3.3 MULTISTEP (SPLITTING, PREDICTOR-CORRECTOR) METHODS

Computational stability, convergence, and accuracy may be improved using multistep (intermediate step between n and $n+1$) schemes, such as Richtmyer, Lax-Wendroff, and McCormack methods. The two-step schemes for these methods are shown below.

Richtmyer Multistep Scheme

Step 1

$$\frac{u_i^{n+\frac{1}{2}} - \frac{1}{2}(u_{i+1}^n + u_{i-1}^n)}{\Delta t/2} = -a\frac{(u_{i+1}^n - u_{i-1}^n)}{2\Delta x} \tag{4.3.23a}$$

Step 2

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a \frac{\left(u_{i+1}^{n+\frac{1}{2}} - u_{i-1}^{n+\frac{1}{2}}\right)}{2\Delta x} \tag{4.3.23b}$$

These equations can be rearranged in the form

Step 1

$$u_i^{n+\frac{1}{2}} = \frac{1}{2}\left(u_{i+1}^n + u_{i-1}^n\right) - \frac{C}{4}\left(u_{i+1}^n - u_{i-1}^n\right) \tag{4.3.24a}$$

Step 2

$$u_i^{n+1} = u_i^n - \frac{C}{2}\left(u_{i+1}^{n+\frac{1}{2}} - u_{i-1}^{n+\frac{1}{2}}\right), \quad \mathrm{O}(\Delta t^2, \Delta x^2) \tag{4.3.24b}$$

This scheme is stable for $C \le 2$.

Lax-Wendroff Multistep Scheme

Step 1

$$u_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}\left(u_{i+1}^n + u_i^n\right) - \frac{C}{2}\left(u_{i+1}^n - u_i^n\right), \quad \mathrm{O}(\Delta t^2, \Delta x^2) \tag{4.3.25a}$$

Step 2

$$u_i^{n+1} = u_i^n - C\left(u_{i+\frac{1}{2}}^{n+\frac{1}{2}} - u_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right), \quad \mathrm{O}(\Delta t^2, \Delta x^2) \tag{4.3.25b}$$

The stability condition is $C \le 1$. Note that substitution of (4.3.25a) into (4.3.25b) recovers the original Lax-Wendroff equation (4.3.18). The same result is obtained with (4.3.24a) and (4.3.24b).

MacCormack Multistep Scheme

Here we consider an intermediate step u_i^* which is related to $u_i^{n+\frac{1}{2}}$:

$$u_i^{n+\frac{1}{2}} = \frac{1}{2}\left(u_i^n + u_i^*\right) \tag{4.3.26}$$

Step 1

$$\frac{u_i^* - u_i^n}{\Delta t} = -a \frac{\left(u_{i+1}^n - u_i^n\right)}{\Delta x} \tag{4.3.27a}$$

Step 2

$$\frac{u_i^{n+1} - u_i^{n+\frac{1}{2}}}{\Delta t/2} = -a \frac{\left(u_i^* - u_{i-1}^*\right)}{\Delta x} \tag{4.3.27b}$$

Substituting (4.3.26) into (4.3.27b) yields

Predictor

$$u_i^* = u_i^n - C\left(u_{i+1}^n - u_i^n\right) \tag{4.3.28a}$$

Corrector

$$u_i^{n+1} = \frac{1}{2}\left[\left(u_i^n + u_i^*\right) - C\left(u_i^* - u_{i-1}^*\right)\right], \quad O(\Delta t^2, \Delta x^2) \tag{4.3.28b}$$

with the stability criterion of $C \leq 1$.

The MacCormack multistep method is well suited for nonlinear problems. It becomes equivalent to the Lax-Wendroff method for linear problems.

4.3.4 NONLINEAR PROBLEMS

A classical nonlinear first order hyperbolic equation is the Euler's equation

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} \tag{4.3.29}$$

which in conservation form may be written as

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) \tag{4.3.30a}$$

or

$$\frac{\partial u}{\partial t} = -\frac{\partial F}{\partial x} \quad \text{with } F = \left(\frac{u^2}{2}\right) \tag{4.3.30b}$$

The solution of (4.3.30b) may be obtained by several methods: Lax method, Lax-Wendroff method, MacCormack method, and Beam-Warming implicit method. These are described below.

Lax Method

In this method, the FTCS differencing scheme is used.

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{F_{i+1}^n - F_{i-1}^n}{2\Delta x}, \quad O(\Delta t, \Delta x^2) \tag{4.3.31}$$

To maintain stability, we replace u_i^n by its average,

$$u_i^{n+1} = \frac{1}{2}\left(u_{i+1}^n + u_{i-1}^n\right) - \frac{\Delta t}{2\Delta x}\left(F_{i+1}^n - F_{i-1}^n\right) \tag{4.3.32}$$

or

$$u_i^{n+1} = \frac{1}{2}\left(u_{i+1}^n + u_{i-1}^n\right) - \frac{\Delta t}{4\Delta x}\left[\left(u_{i+1}^n\right)^2 - \left(u_{i-1}^n\right)^2\right] \tag{4.3.33}$$

The solution will be stable if

$$\left|\frac{\Delta t}{\Delta x}u_{\max}\right| \leq 1 \tag{4.3.34}$$

Lax-Wendroff Method

In this method, the finite difference equation is derived from the Taylor series expansion,

$$u_i^{n+1} = u_i^n + \frac{\partial u}{\partial t}\Delta t + \frac{1}{2!}\frac{\partial^2 u}{\partial t^2}\Delta t^2 + \cdots \tag{4.3.35}$$

Using (4.3.30b) we have

$$\frac{\partial^2 u}{\partial t^2} = -\frac{\partial}{\partial t}\left(\frac{\partial F}{\partial x}\right) = -\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial t}\right) \tag{4.3.36}$$

where

$$\frac{\partial F}{\partial t} = \frac{\partial F}{\partial u}\frac{\partial u}{\partial t} = \frac{\partial F}{\partial u}\left(-\frac{\partial F}{\partial x}\right) = -A\frac{\partial F}{\partial x} \tag{4.3.37}$$

with A being the Jacobian.

$$A = \frac{\partial F}{\partial u} = \frac{\partial}{\partial u}\left(\frac{u^2}{2}\right) = u \tag{4.3.38}$$

Thus

$$\frac{\partial^2 u}{\partial t^2} = -\frac{\partial}{\partial x}\left(-A\frac{\partial F}{\partial x}\right) = \frac{\partial}{\partial x}\left(A\frac{\partial F}{\partial x}\right) \tag{4.3.39}$$

Substituting (4.3.39) and (4.3.30b) into (4.3.35) yields

$$u_i^{n+1} = u_i^n + \left(-\frac{\partial F}{\partial x}\right)\Delta t + \frac{\partial}{\partial x}\left(A\frac{\partial F}{\partial x}\right)\frac{\Delta t^2}{2} + \mathrm{O}(\Delta t^3)$$

or

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{\partial F}{\partial x} + \frac{\partial}{\partial x}\left(A\frac{\partial F}{\partial x}\right)\frac{\Delta t}{2} + \mathrm{O}(\Delta t^2)$$

Approximating the spatial derivatives by central differencing of order 2,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{F_{i+1}^n - F_{i-1}^n}{2\Delta x} + \frac{\Delta t}{2\Delta x}\left[\left(A\frac{\partial F}{\partial x}\right)_{i+\frac{1}{2}}^n - \left(A\frac{\partial F}{\partial x}\right)_{i-\frac{1}{2}}^n\right] \tag{4.3.40}$$

The last term above is approximated as

$$\frac{\left(A\frac{\partial F}{\partial x}\right)_{i+\frac{1}{2}}^n - \left(A\frac{\partial F}{\partial x}\right)_{i-\frac{1}{2}}^n}{\Delta x} = \frac{A_{i+\frac{1}{2}}^n \frac{F_{i+1}^n - F_i^n}{\Delta x} - A_{i-\frac{1}{2}}^n \frac{F_i^n - F_{i-1}^n}{\Delta x}}{\Delta x}$$
$$= \frac{\frac{1}{2\Delta x}\left(A_{i+1}^n + A_i^n\right)\left(F_{i+1}^n - F_i^n\right) - \frac{1}{2\Delta x}\left(A_i^n + A_{i-1}^n\right)\left(F_i^n - F_{i-1}^n\right)}{\Delta x} \tag{4.3.41}$$

For $A = u$, we obtain

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{2\Delta x}\left(F_{i+1}^n - F_{i-1}^n\right)$$
$$+ \frac{1}{4}\frac{\Delta t^2}{\Delta x^2}\left[\left(u_{i+1}^n + u_i^n\right)\left(F_{i+1}^n - F_i^n\right) - \left(u_i^n + u_{i-1}^n\right)\left(F_i^n - F_{i-1}^n\right)\right] \tag{4.3.42}$$

This is second order accurate with the stability requirement,

$$\left|\frac{\Delta t}{\Delta x}u_{\max}\right| \le 1$$

MacCormack Method

In this method, the multilevel scheme is used as given by

$$u_i^* = u_i^n - \frac{\Delta t}{\Delta x}(F_{i+1}^n - F_i^n) \tag{4.3.43a}$$

$$u_i^{n+1} = \frac{1}{2}\left[u_i^n + u_i^* - \frac{\Delta t}{\Delta x}(F_i^* - F_{i-1}^*)\right] \tag{4.3.43b}$$

Because of the two-level splitting, the solution performs better than the Lax method or the Lax-Wendroff method. One of the most widely used implicit schemes is the Beam-Warming method, discussed below.

Beam-Warming Implicit Method

Let us consider the Taylor series expansion,

$$u(x,t+\Delta t) = u(x,t) + \left.\frac{\partial u}{\partial t}\right|_{x,t}\Delta t + \left.\frac{\partial^2 u}{\partial t^2}\right|_{x,t}\frac{\Delta t^2}{2} + \mathrm{O}(\Delta t^3) \tag{4.3.44}$$

and

$$u(x,t) = u(x,t+\Delta t) - \left.\frac{\partial u}{\partial t}\right|_{x,t+\Delta t}\Delta t + \left.\frac{\partial^2 u}{\partial t^2}\right|_{x,t+\Delta t}\frac{\Delta t^2}{2!} + \mathrm{O}(\Delta t^3) \tag{4.3.45}$$

Subtracting (4.3.45) from (4.3.44)

$$\begin{aligned} 2u(x,t+\Delta t) \;=\; & 2u(x,t) + \left.\frac{\partial u}{\partial t}\right|_{x,t}\Delta t + \left.\frac{\partial u}{\partial t}\right|_{x,t+\Delta t}\Delta t \\ & + \left.\frac{\partial^2 u}{\partial t^2}\right|_{x,t}\frac{\Delta t^2}{2!} - \left.\frac{\partial^2 u}{\partial t^2}\right|_{x,t+\Delta t}\frac{\Delta t^2}{2!} + \mathrm{O}(\Delta t^3) \end{aligned}$$

or

$$u_i^{n+1} = u_i^n + \frac{1}{2}\left[\left(\frac{\partial u}{\partial t}\right)_i^n + \left(\frac{\partial u}{\partial t}\right)_i^{n+1}\right]\Delta t + \frac{1}{2}\left[\left(\frac{\partial^2 u}{\partial t^2}\right)_i^n - \left(\frac{\partial^2 u}{\partial t^2}\right)_i^{n+1}\right]\frac{\Delta t^2}{2!} + \mathrm{O}(\Delta t^3)$$

where

$$\left(\frac{\partial^2 u}{\partial t^2}\right)_i^{n+1} = \left(\frac{\partial^2 u}{\partial t^2}\right)_i^n + \frac{\partial}{\partial t}\left(\frac{\partial^2 u}{\partial t^2}\right)_i^n \Delta t + \mathrm{O}(\Delta t^2)$$

Thus, we arrive at

$$u_i^{n+1} = u_i^n + \frac{1}{2}\left[\left(\frac{\partial u}{\partial t}\right)_i^n + \left(\frac{\partial u}{\partial t}\right)_i^{n+1}\right]\Delta t + \mathrm{O}(\Delta t^3) \tag{4.3.46}$$

For the model equation

$$\frac{\partial u}{\partial t} = -\frac{\partial F}{\partial x} \tag{4.3.47}$$

Using (4.3.46) in (4.3.47), we obtain

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{1}{2}\left[\left(\frac{\partial F}{\partial x}\right)_i^n + \left(\frac{\partial F}{\partial x}\right)_i^{n+1}\right] + \mathrm{O}(\Delta t^2) \tag{4.3.48}$$

This indicates that (4.3.48) leads to the second order accuracy.

Recall that the nonlinear term $F = u^2/2$ was applied at the known time level n, and the resulting FDE in explicit form was linear. The resulting FDE in implicit formulation is nonlinear, and therefore a procedure is used to linearize the FDE. To this end, we write a Taylor series for $F(t + \Delta t)$ in the form

$$\begin{aligned} F(t+\Delta t) &= F(t) + \frac{\partial F}{\partial t}\Delta t + \mathrm{O}(\Delta t^2) \\ &= F(t) + \frac{\partial F}{\partial u}\frac{\partial u}{\partial t}\Delta t + \mathrm{O}(\Delta t^2) \end{aligned}$$

or

$$F^{n+1} = F^n + \frac{\partial F}{\partial u}\left(\frac{u^{n+1} - u^n}{\Delta t}\right)\Delta t + \mathrm{O}(\Delta t^2) \tag{4.3.49}$$

Taking a partial derivative of (4.3.49) yields

$$\left(\frac{\partial F}{\partial x}\right)^{n+1} = \left(\frac{\partial F}{\partial x}\right)^n + \frac{\partial}{\partial x}[A(u^{n+1} - u^n)] \tag{4.3.50}$$

Combining (4.3.48) and (4.3.50) gives

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{1}{2}\left\{\left(\frac{\partial F}{\partial x}\right)_i^n + \left(\frac{\partial F}{\partial x}\right)_i^n + \frac{\partial}{\partial x}\left[A\left(u_i^{n+1} - u_i^n\right)\right]\right\}$$

or

$$u_i^{n+1} = u_i^n - \frac{1}{2}\Delta t\left\{2\left(\frac{\partial F}{\partial x}\right)_i^n + \frac{\partial}{\partial x}\left[A\left(u_i^{n+1} - u_i^n\right)\right]\right\} \tag{4.3.51}$$

Using a second order central differencing for the terms with A on the right-hand side of (4.3.51) and linearizing, we obtain

$$\begin{aligned} u_i^{n+1} &= u_i^n - \frac{1}{2}\Delta t\left[\frac{2\left(F_{i+1}^n - F_{i-1}^n\right)}{2\Delta x} + \frac{A_{i+1}^n u_{i+1}^{n+1} - A_{i-1}^n u_{i-1}^{n+1}}{2\Delta x}\right. \\ &\qquad \left. - \frac{A_{i+1}^n u_{i+1}^n - A_{i-1}^n u_{i-1}^n}{2\Delta x}\right] \end{aligned} \tag{4.3.52}$$

Modifying (4.3.52) to a tridiagonal form

$$\begin{aligned} &-\frac{\Delta t}{4\Delta x}A_{i-1}^n u_{i-1}^{n+1} + u_i^{n+1} + \frac{\Delta t}{4\Delta x}A_{i+1}^n u_{i+1}^{n+1} \\ &\quad = u_i^n - \frac{1}{2}\frac{\Delta t}{\Delta x}\left(F_{i+1}^n - F_{i-1}^n\right) + \frac{\Delta t}{4\Delta x}A_{i+1}^n u_{i+1}^n - \frac{\Delta t}{4\Delta x}A_{i-1}^n u_{i-1}^n + D \end{aligned} \tag{4.3.53}$$

This scheme is second order accurate, unconditionally stable, but dispersion errors may arise. To prevent this, a fourth order smoothing (damping) term is explicitly added:

$$D = -\frac{\omega}{8}\left(u_{i+2}^n - 4u_{i+1}^n + 6u_i^n - 4u_{i-1}^n + u_{i-2}^n\right),$$

with $0 < \omega < 1$. Since the added damping term is of fourth order, it does not affect the second order accuracy of the method.

4.3.5 SECOND ORDER ONE-DIMENSIONAL WAVE EQUATIONS

Let us consider the second order one-dimensional wave equation,

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2} \tag{4.3.54}$$

Here we require two sets of initial conditions,

$$u(x,0) = f(x)$$

$$\frac{\partial u}{\partial t}(x,0) = g(x)$$

and two sets of boundary conditions,

$$u(0,t) = h_1(t)$$

$$u(L,t) = h_2(t)$$

We may use the midpoint leapfrog method for this problem,

$$u_i^{n+1} = 2u_i^n - u_i^{n-1} + C^2\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right) \tag{4.3.55}$$

If we choose $\dfrac{\partial u(x,0)}{\partial t} = 0$, then

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = 0$$

or

$$u_i^{n+1} = u_i^{n-1}$$

Thus, from (4.3.55), we obtain

$$u_i^{n+1} = u_i^n + \frac{1}{2}C^2\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right) \tag{4.3.56}$$

This is called the midpoint leapfrog method. An example problem for the second order hyperbolic equation is demonstrated in Section 4.7.4.

4.4 BURGERS' EQUATION

The Burgers' equation is a special form of the momentum equation for irrotational, incompressible flows in which pressure gradients are neglected. It is informative to study this equation in the one-dimensional case before we launch upon full-scale CFD problems.

Consider the Burgers' equation written in various forms:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \tag{4.4.1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \tag{4.4.2}$$

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \tag{4.4.3}$$

with $F = 1/2\ u^2$. These equations are mixed hyperbolic, elliptic, and parabolic types. If steady state is considered, then they become mixed hyperbolic and elliptic equations. Because of these special properties, various solution schemes have been tested extensively for the Burgers' equations. In what follows, we shall examine some of the well-known numerical schemes.

4.4.1 EXPLICIT AND IMPLICIT SCHEMES

FTCS Explicit Scheme

In this scheme (FTCS), approximations of forward differences in time and central differences in space are used:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = \nu\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \tag{4.4.4}$$

where the truncation error is $O(\Delta t, \Delta x^2)$. The central difference for the convective term tends to introduce significant damping.

FTBS Explicit Scheme

This is the same as in FTCS except that backward differences are used for the convective term,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\frac{u_i^n - u_{i-1}^n}{\Delta x} = \nu\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \tag{4.4.5}$$

Here the first order approximation of the convective term may introduce an excessive dissipation error. A compromise is to use higher order schemes such as (3.2.20) for the second order. With (3.2.1) modified for four points, the third order scheme may be written as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\left(\frac{11u_i - 18u_{i-1} + 9u_{i-2} - 2u_{i-3}}{6\Delta x}\right) = \nu\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \tag{4.4.6}$$

DuFort-Frankel Explicit Scheme

In this scheme, we use second order central differences for all derivatives,

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

$$O\left(\Delta t^2, \Delta x^2, \left(\frac{\Delta t}{\Delta x}\right)^2\right) \tag{4.4.7a}$$

or

$$u_i^{n+1} = \left(\frac{1-2d}{1+2d}\right)u_i^{n-1} + \left(\frac{C+2d}{1+2d}\right)u_{i-1}^n - \left(\frac{C-2d}{1+2d}\right)u_{i+1}^n \tag{4.4.7b}$$

This is stable for $C \le 1$.

MacCormack Explicit Scheme

The two-step or predictor-corrector scheme is written as

Step 1

$$u_i^* = u_i^n - a\frac{\Delta t}{\Delta x}\left(u_{i+1}^n - u_i^n\right) + \nu\frac{\Delta t}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \tag{4.4.8a}$$

Step 2

$$u_i^{n+1} = \frac{1}{2}\left[u_i^n + u_i^* - a\frac{\Delta t}{\Delta x}(u_i^* - u_{i-1}^*) + \nu\frac{\Delta t}{\Delta x^2}(u_{i+1}^* - 2u_i^* + u_{i-1}^*)\right] \tag{4.4.8b}$$

This method is second order accurate with the stability requirement

$$\Delta t \le \frac{1}{\dfrac{a}{\Delta x} + \dfrac{2\nu}{\Delta x^2}} \tag{4.4.9}$$

The following alternate form may be used:

Step 1

$$\begin{aligned}\Delta u_i^n &= -a\frac{\Delta t}{\Delta x}\left(u_{i+1}^n - u_i^n\right) + \frac{\nu\Delta t}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)\\ u_i^* &= u_i^n + \Delta u_i^n\end{aligned} \tag{4.4.10a}$$

Step 2

$$\begin{aligned}\Delta u_i^* &= -a\frac{\Delta t}{\Delta x}(u_i^* - u_{i-1}^*) + \frac{\nu\Delta t}{\Delta x^2}(u_{i+1}^* - 2u_i^* + u_{i-1}^*)\\ u_i^{n+1} &= \frac{1}{2}(u_i^n + u_i^* + \Delta u_i^*)\end{aligned} \tag{4.4.10b}$$

MacCormack Implicit Scheme

One of the most frequently used implicit schemes is the MacCormack scheme.

Step 1

$$\begin{aligned}\left(1 + \lambda\frac{\Delta t}{\Delta x}\right)\delta u_i^* &= \Delta u_i^n + \lambda\frac{\Delta t}{\Delta x}\delta u_{i+1}^*\\ u_i^* &= u_i^n + \delta u_i^*\end{aligned} \tag{4.4.11a}$$

Step 2

$$\begin{aligned}\left(1 + \lambda\frac{\Delta t}{\Delta x}\right)\delta u_i^{n+1} &= \Delta u_i^* + \lambda\frac{\Delta t}{\Delta x}\delta u_{i-1}^{n+1}\\ u_i^{n+1} &= \frac{1}{2}(u_i^n + u_i^* + \delta u_i^{n+1})\end{aligned} \tag{4.4.11b}$$

where

$$\lambda \ge \max\left[\frac{1}{2}\left(|a| + \frac{2\nu}{\Delta x} - \frac{\Delta x}{\Delta t}\right), 0\right] \tag{4.4.12}$$

Note that equations (4.4.11a,b) form a tridiagonal system. The method is unconditionally stable and second order accurate as long as the diffusion number, $d = \nu\Delta t / \Delta x^2$, is bounded for the limiting process for which Δt and Δx approach zero.

4.4.2 RUNGE-KUTTA METHOD

The transient nonlinear inviscid Burgers' equation can be written as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

or

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0, \quad F = \frac{u^2}{2}$$

For nonlinear transient problems, the Runge-Kutta method is known to be efficient and has been used extensively. This method is briefly introduced below.

Let us consider an equation of the type

$$\frac{\partial u}{\partial t} = R(u) \tag{4.4.13}$$

One of the popular approaches is the fourth order Runge-Kutta scheme written as

Step 1

$$u^{(1)} = u^n + \frac{\Delta t}{2} R^n$$

Step 2

$$u^{(2)} = u^n + \frac{\Delta t}{2} R^{(1)}$$

Step 3

$$u^{(3)} = u^n + \Delta t R^{(2)}$$

Step 4

$$u^{n+1} = u^n + \frac{\Delta t}{6}\left(R^n + 2R^{(1)} + 2R^{(2)} + R^{(3)}\right) \tag{4.4.14}$$

with

$$R^{(1)} = R\left(t^{n+1/2}, u^{(1)}\right)$$
$$R^{(2)} = R\left(t^{n+1/2}, u^{(2)}\right)$$
$$R^{(3)} = R\left(t^{n}, u^{(3)}\right)$$

It is seen that higher order Runge-Kutta schemes require more steps for the evaluation of $R(u)$, resulting in additional computer time requirements.

An example of the solution procedure for the nonlinear Burgers' equation is presented in Section 4.7.5.

4.5 ALGEBRAIC EQUATION SOLVERS AND SOURCES OF ERRORS

4.5.1 SOLUTION METHODS

As a result of FDM formulations, we obtain linear or nonlinear simultaneous algebraic equations which must be solved. As we discussed in previous sections, either direct methods or iterative methods may be used. Recall that, as direct methods, we examined the Gaussian elimination in Section 4.1.3 and the Thomas algorithm (tridiagonal matrix algorithm, TDMA) in Section 4.2.7. We also discussed the Runge-Kutta method in Section 4.4.2 for the nonlinear time dependent equations.

In general, the number of arithmetic operations of a direct method can be very high particularly for a large system of equations – much larger than the total number of operations in an iterative method. Therefore, for fluid mechanics problems with nonlinear sparse matrices, it is more convenient, and often necessary, to work with iterative methods.

There are many iterative methods other than those already introduced in the earlier sections of this chapter. They include conjugate gradient method, generalized minimum residual (GMRES) algorithm, and multigrid method. These methods are well documented in the literature. Among them are Varga [1962], Wachspress [1966], Dahlquist and Bjork [1974], and Saad [1996].

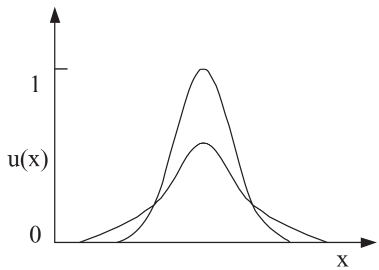
Some of these advanced iterative methods will be presented in Parts Three and Four. Conjugate gradient method, generalized minimum residual method, and multigrid method are presented in Sections 10.3.1, 11.5.2, and 20.2, respectively. This is because of the convenience of presentation as appropriate to the topical arrangements of this book. Namely, the iterative solution methods are included in Part Three since the element-by element method of FEM assembly requires special treatments of iterative solution procedures, whereas the multigrid method is included in Part Four as it is related to other topics including automatic grid generation. Newton-Raphson methods for nonlinear algebraic equations are discussed in Section 11.5.1. Thus, the reader may find it useful in visiting these sections as needed for his/her studies in FDM, Part Two.

4.5.2 EVALUATION OF SOURCES OF ERRORS

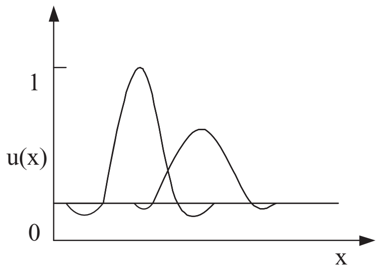
Recall that computational errors were discussed in terms of an amplification factor g in Sections 4.2 and 4.3. For $g < 1$, the result is numerical diffusion (sometimes known as numerical damping or numerical dissipation). On the other hand, for $g > 1$, the result is numerical instability. Both of these cases lead to amplitude errors as shown in Figure 4.5.1, which may be equivalent to the severely damped shock wave as depicted in Figure 4.3.2c(2).

If waves of different wavelengths travel in a medium, such a phenomenon is known as dispersion. The dispersion arises from discrete spatial approximations and results in a numerical error, called the numerical dispersion or phase error as shown in Figure 4.5.1b or Figure 4.3.2c(3). The dispersion error occurs in convection or wave equations, but not in diffusion equations.

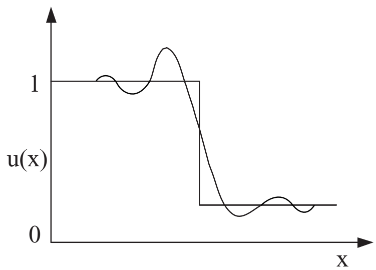
In numerical simulations, the so-called Gibb's phenomenon occurs due to discretization of the domain by a limited number of nodal points (Figure 4.5.1c). They appear as overshoots and undershoots near the steep gradients, similar to the diffusion errors.



(a) Dissipation error



(b) Dispersion error



(c) Gibb's error

Figure 4.5.1 Various numerical errors.

Next we shall discuss these errors, which are associated with the diffusion transport and convection transport equations.

Diffusive Transport

Parabolic equations represent the diffusion process associated with both spatial and temporal variations. A general form of (4.2.2a) may be written in the form

$$\frac{u^{n+1}{}_i - u_i^n}{\Delta t} = \frac{\alpha\theta}{\Delta x^2}\left(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right) + \frac{\alpha(1-\theta)}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \tag{4.5.1}$$

with $0 \le \theta \le 1$.

The method is fully explicit for $\theta = 0$ and partially implicit for $0 < \theta < 1$, with $\theta = 1$ being fully implicit. The scheme with $\theta = 1/2$, known as the centered scheme, provides reasonably stable and accurate solutions in general.

Using the definitions given in (4.2.12–4.2.15), the analytical solution of the diffusion equation (4.2.1) may be written in the form

$$u(x,t) = u(t)e^{Ikx} \tag{4.5.2}$$

or

$$u(x,t) = u_0 e^{-\alpha k^2 t} e^{Ikx} \tag{4.5.3}$$

Substituting (4.5.2) into (4.5.1) and using the definition of the amplification factor (4.2.17), we obtain the amplification factor for various values of θ,

$$|g|_\theta = \frac{[1 - d(1-\theta)(1-\cos(k\Delta x))]^{1/d}}{[1 + d\theta(1-\cos(k\Delta x))]^{1/d}} \tag{4.5.4}$$

The amplification factors for explicit scheme (E), centered scheme (C), and fully implicit scheme (I) for $\theta = 0$, $\theta = 1/2$, $\theta = 1$ are shown in Figure 4.5.2. It is seen that the explicit and centered schemes behave irregularly for high values of diffusion number, whereas the fully implicit scheme is stable.

For multidimensional problems, the implicit method requires the inversion of large and sparse matrix equations and is computationally expensive. Although the solution may be stable with large time steps, numerical diffusion becomes excessive, resulting in inaccuracy. On the other hand, the explicit scheme is less expensive, but small time steps are necessary in order to achieve accuracy. The amplitude errors are significant in the diffusive transport equations, whereas dispersion errors and Gibb's errors dominate in the convective transport equations.

Convective Transport

Hyperbolic equations represent convection and wave phenomena. A typical convection equation may be written in the finite difference form

$$\frac{u_i^{n+1}-u_i^n}{\Delta t} = -a\left[\theta\frac{\left(u_{i+1}^{n+1}-u_{i-1}^{n+1}\right)}{2\Delta x} + \left((1-\theta)\frac{\left(u_{i+1}^{n}-u_{i-1}^{n}\right)}{2\Delta x}\right)\right] \tag{4.5.5}$$

or in terms of the Courant number $C = a\Delta t/\Delta x$.

$$u_i^{n+1} + \frac{\theta C}{2}\left(u_{i+1}^{n+1}-u_{i-1}^{n+1}\right) = u_i^n - \frac{(1-\theta)C}{2}\left(u_{i+1}^{n}-u_{i-1}^{n}\right) \tag{4.5.6}$$

Note that the values of u at $n+1$ for $\theta > 0$ (implicit scheme) are calculated in terms of the values at n, but are involved in three different spatial locations, resulting in a

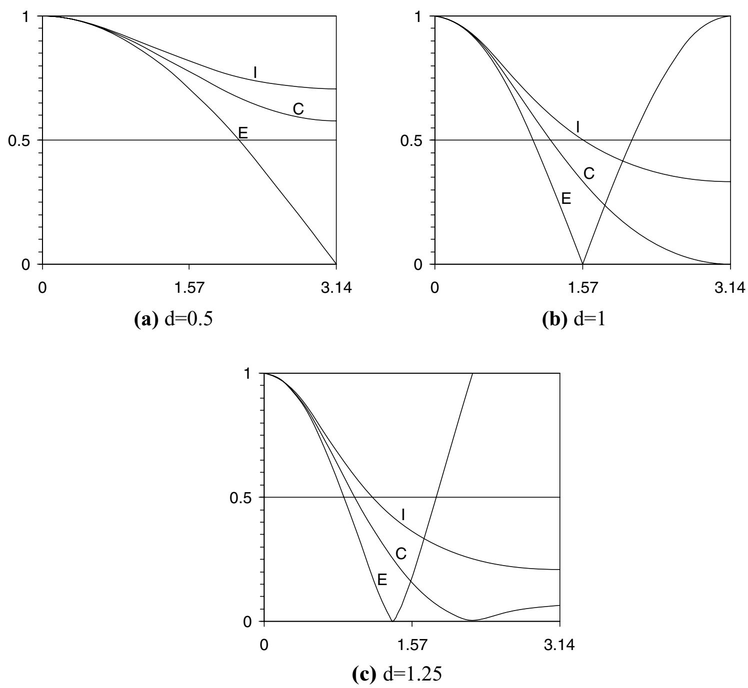


Figure 4.5.2 Amplification factors for diffusion equation, E = explicit, C = centered, I = fully implicit.

tridiagonal matrix. Although the explicit scheme ($\theta = 0$) reduces to a simple algebraic equation, computational difficulties in stability and accuracy are likely to occur.

In view of (4.2.15), (4.3.3), and (4.5.6), the amplification factors for $\theta = 0$, $\theta = 1/2$, and $\theta = 1$ can be written in the form, respectively,

$$|g|_E = |1 - IC\sin(k\Delta x)| \tag{4.5.7}$$

$$|g|_C = \left|\frac{1 - I\dfrac{C}{2}\sin(k\Delta x)}{1 + I\dfrac{C}{2}\sin(k\Delta x)}\right| \tag{4.5.8}$$

$$|g|_I = \left|\frac{1}{1 + IC\sin(k\Delta x)}\right| \tag{4.5.9}$$

Similarly, using (4.3.18), the amplification factor for the Lax-Wendroff scheme is derived in the form,

$$|g|_L = |1 - C^2(1 - \cos(k\Delta x)) - IC\sin(k\Delta x)| \tag{4.5.10}$$

These results (Figure 4.5.3) show that the explicit scheme performs poorly in the region $k\Delta x = \pi/2$, whereas the Lax-Wendroff scheme behaves quite satisfactorily in the high wave number region.

As seen in other schemes studied in Section 3, computational errors including amplitude errors, dispersion errors, and Gibb's errors must be carefully examined, particularly in multidimensional problems. Some of the schemes used in one-dimensional problems may be extended to multidimensional problems, although the conclusions reached for one-dimensional problems discussed here are by no means universally applicable. In order to deal with more complicated geometries and physical aspects in CFD, many other schemes and methodologies will be explored in Chapters 5 and 6 (incompressible flows and compressible flows, respectively) and in FEM, Part Three.

4.6 COORDINATE TRANSFORMATION FOR ARBITRARY GEOMETRIES

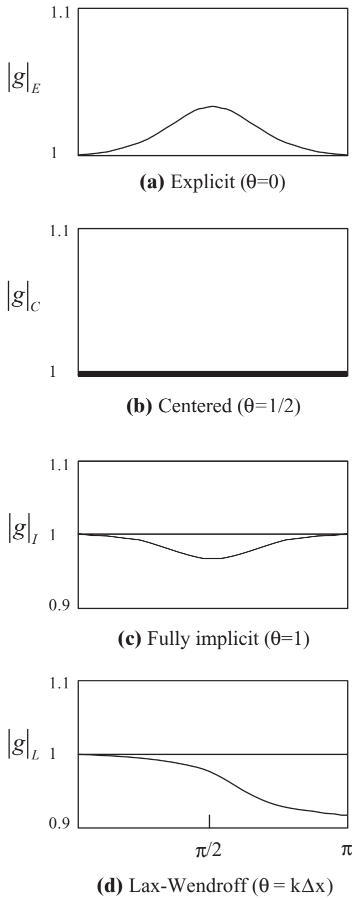
Finite difference formulas developed in Chapter 3 and finite difference solution schemes discussed so far are applicable only to rectangular cartesian coordinates. If grids are oriented in arbitrary directions of 2-D or 3-D geometries, then it is necessary to transform the arbitrary physical domain into the computational domain of a rectangular cartesian system so that finite difference equations can be written in orthogonal directions. Such transformations are possible as long as the entire grid system is structured.

4.6.1 DETERMINATION OF JACOBIANS AND TRANSFORMED EQUATIONS

Let us consider for simplicity a two-dimensional coordinate system of the physical domain (x, y), and the computational domain (ξ and η) as shown in Figure 4.6.1. We begin with spatial derivatives of any variable with respect to ξ and η as

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

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



$$|g|_E = |1 + IC\sin(k\Delta x)|$$

$$|g|_C = \left| \frac{1 - I\frac{C}{2}\sin(k\Delta x)}{1 + I\frac{C}{2}\sin(k\Delta x)} \right|$$

$$|g|_I = \left| \frac{1}{1 + IC\sin(k\Delta x)} \right|$$

$$|g|_L = |1 - C^2(1 - \cos(k\Delta x)) - IC\sin(k\Delta x)|$$

Figure 4.5.3 Amplification factors.

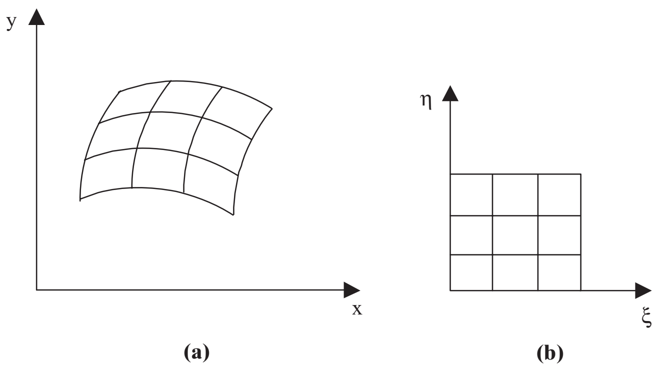


Figure 4.6.1 Transformation from curvilinear grid system into rectangular grid system. **(a)** Original curvilinear grid. **(b)** Transformed cartesian grid.

or

$$\begin{bmatrix} \dfrac{\partial}{\partial \xi} \\ \dfrac{\partial}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{bmatrix} = [J] \begin{bmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{bmatrix} \tag{4.6.1}$$

where $[J]$ is the Jacobian matrix

$$[J] = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix}$$

Thus

$$\begin{bmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{bmatrix} = \frac{1}{|J|} \begin{bmatrix} \dfrac{\partial y}{\partial \eta} & \dfrac{-\partial y}{\partial \xi} \\ \dfrac{-\partial x}{\partial \eta} & \dfrac{\partial x}{\partial \xi} \end{bmatrix} \begin{bmatrix} \dfrac{\partial}{\partial \xi} \\ \dfrac{\partial}{\partial \eta} \end{bmatrix} \tag{4.6.2}$$

Second derivatives of (4.6.2) are given by

$$\begin{aligned} \frac{\partial^2}{\partial x^2} = {} & \frac{1}{|J|^2}\left[\left(\frac{\partial y}{\partial \eta}\right)^2 \frac{\partial^2}{\partial \xi^2} - 2\frac{\partial y}{\partial \eta}\frac{\partial y}{\partial \xi}\frac{\partial^2}{\partial \xi \partial \eta} + \left(\frac{\partial y}{\partial \xi}\right)^2 \frac{\partial^2}{\partial \eta^2} \right. \\ & \left. + \left(\frac{\partial y}{\partial \eta}\frac{\partial^2 y}{\partial \xi \partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial^2 y}{\partial \eta^2}\right)\frac{\partial}{\partial \xi} + \left(\frac{\partial y}{\partial \xi}\frac{\partial^2 y}{\partial \xi \partial \eta} - \frac{\partial y}{\partial \eta}\frac{\partial^2 y}{\partial \xi^2}\right)\frac{\partial}{\partial \eta}\right] \\ & - \frac{1}{|J|^3}\left[\left(\frac{\partial y}{\partial \eta}\right)^2 \frac{\partial |J|}{\partial \xi}\frac{\partial}{\partial \xi} - \frac{\partial y}{\partial \eta}\frac{\partial y}{\partial \xi}\frac{\partial |J|}{\partial \xi}\frac{\partial}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial |J|}{\partial \eta}\frac{\partial}{\partial \xi} + \left(\frac{\partial y}{\partial \xi}\right)^2 \frac{\partial |J|}{\partial \eta}\frac{\partial}{\partial \eta}\right] \end{aligned} \tag{4.6.3a}$$

$$\begin{aligned} \frac{\partial^2}{\partial y^2} = {} & \frac{1}{|J|^2}\left[\left(\frac{\partial x}{\partial \eta}\right)^2 \frac{\partial^2}{\partial \xi^2} - 2\frac{\partial x}{\partial \eta}\frac{\partial x}{\partial \xi}\frac{\partial^2}{\partial \xi \partial \eta} + \left(\frac{\partial x}{\partial \xi}\right)^2 \frac{\partial^2}{\partial \eta^2} \right. \\ & \left. + \left(\frac{\partial x}{\partial \eta}\frac{\partial^2 x}{\partial \xi \partial \eta} - \frac{\partial x}{\partial \xi}\frac{\partial^2 x}{\partial \eta^2}\right)\frac{\partial}{\partial \xi} + \left(\frac{\partial x}{\partial \xi}\frac{\partial^2 x}{\partial \xi \partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial^2 x}{\partial \xi^2}\right)\frac{\partial}{\partial \eta}\right] \\ & - \frac{1}{|J|^3}\left[\left(\frac{\partial x}{\partial \eta}\right)^2 \frac{\partial |J|}{\partial \xi}\frac{\partial}{\partial \xi} - \frac{\partial x}{\partial \eta}\frac{\partial x}{\partial \xi}\frac{\partial |J|}{\partial \xi}\frac{\partial}{\partial \eta} - \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta}\frac{\partial |J|}{\partial \eta}\frac{\partial}{\partial \xi} + \left(\frac{\partial x}{\partial \xi}\right)^2 \frac{\partial |J|}{\partial \eta}\frac{\partial}{\partial \eta}\right] \end{aligned} \tag{4.6.3b}$$

where

$$\begin{aligned} \frac{\partial |J|}{\partial \xi} &= \frac{\partial}{\partial \xi}\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}\right) \\ &= \frac{\partial^2 x}{\partial \xi^2}\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \xi}\frac{\partial^2 y}{\partial \xi \partial \eta} - \frac{\partial^2 y}{\partial \xi^2}\frac{\partial x}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial^2 x}{\partial \xi \partial \eta} \\ \frac{\partial |J|}{\partial \eta} &= \frac{\partial}{\partial \eta}\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}\right) \\ &= \frac{\partial^2 x}{\partial \xi \partial \eta}\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \xi}\frac{\partial^2 y}{\partial \eta^2} - \frac{\partial^2 y}{\partial \eta \partial \xi}\frac{\partial x}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial^2 x}{\partial \eta^2} \end{aligned}$$

Consider the governing equations in the form

$$\frac{\partial \mathbf{U}}{\partial t} + u\frac{\partial \mathbf{U}}{\partial x} + \mathrm{v}\frac{\partial \mathbf{U}}{\partial y} - \nu\left(\frac{\partial^2 \mathbf{U}}{\partial x^2} + \frac{\partial^2 \mathbf{U}}{\partial y^2}\right) - \mathbf{f} = 0 \tag{4.6.4}$$

with

$$\mathbf{U} = \begin{bmatrix} u \\ \mathrm{v} \end{bmatrix}, \qquad \mathbf{f} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Applying (4.6.3) to (4.6.4) yields

$$\frac{\partial \mathbf{U}}{\partial t} + \bar{u}\frac{\partial \mathbf{U}}{\partial \xi} + \bar{\mathrm{v}}\frac{\partial \mathbf{U}}{\partial \eta} - \nu\left[\frac{1}{|J|^2}\left(a\frac{\partial^2 \mathbf{U}}{\partial \xi^2} - 2b\frac{\partial^2 \mathbf{U}}{\partial \xi\partial \eta} + c\frac{\partial^2 \mathbf{U}}{\partial \eta^2}\right) + p\frac{\partial \mathbf{U}}{\partial \xi} + q\frac{\partial \mathbf{U}}{\partial \eta}\right] - f = 0 \tag{4.6.5}$$

where

$$\bar{u} = \frac{1}{|J|}\left(u\frac{\partial y}{\partial \eta} - \mathrm{v}\frac{\partial x}{\partial \eta}\right)$$

$$\bar{\mathrm{v}} = \frac{1}{|J|}\left(\mathrm{v}\frac{\partial x}{\partial \xi} - u\frac{\partial y}{\partial \xi}\right)$$

$$p = \frac{1}{|J|^3}\left[-\frac{\partial y}{\partial \eta}\left(a\frac{\partial^2 x}{\partial \xi^2} - 2b\frac{\partial^2 x}{\partial \xi\partial \eta} + c\frac{\partial^2 x}{\partial \eta^2}\right) + \frac{\partial x}{\partial \eta}\left(a\frac{\partial^2 y}{\partial \xi^2} - 2b\frac{\partial^2 y}{\partial \xi\partial \eta} + c\frac{\partial^2 y}{\partial \eta^2}\right)\right]$$

$$q = \frac{1}{|J|^3}\left[\frac{\partial y}{\partial \xi}\left(a\frac{\partial^2 x}{\partial \xi^2} - 2b\frac{\partial^2 x}{\partial \xi\partial \eta} + c\frac{\partial^2 x}{\partial \eta^2}\right) - \frac{\partial x}{\partial \xi}\left(a\frac{\partial^2 y}{\partial \xi^2} - 2b\frac{\partial^2 y}{\partial \xi\partial \eta} + c\frac{\partial^2 y}{\partial \eta^2}\right)\right]$$

$$a = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2$$

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

$$c = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2$$

4.6.2 APPLICATION OF NEUMANN BOUNDARY CONDITIONS

Neumann boundary conditions are applied in the transformed coordinates based on the same procedure described above. For example, let us consider the gradient of **U** with respect to η.

$$\frac{\partial \mathbf{U}}{\partial \eta} = \frac{\partial \mathbf{U}}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial \mathbf{U}}{\partial y}\frac{\partial y}{\partial \eta} \tag{4.6.6}$$

Using a first order backward difference for $\frac{\partial \mathbf{U}}{\partial \eta}$, $\frac{\partial x}{\partial \eta}$ and $\frac{\partial y}{\partial \eta}$, we have

$$\frac{\mathbf{U}_{i,j} - \mathbf{U}_{i,j-1}}{\Delta\eta} = \frac{\partial \mathbf{U}}{\partial x}\frac{x_{i,j} - x_{i,j-1}}{\Delta\eta} + \frac{\partial \mathbf{U}}{\partial y}\frac{y_{i,j} - y_{i,j-1}}{\Delta\eta}$$

or

$$\mathbf{U}_{i,j} = \mathbf{U}_{i,j-1} + \frac{\partial \mathbf{U}}{\partial x}\Delta x + \frac{\partial \mathbf{U}}{\partial y}\Delta y \tag{4.6.7}$$

4.6.3 SOLUTION BY MACCORMACK METHOD

The transformed governing equations (4.6.5) may be solved using the MacCormack method as follows:

Predictor

$$\begin{aligned}\mathbf{U}^{*}_{i,j} = \mathbf{U}^{n}_{i,j} + \Delta t\Bigg\{&-\left(\bar{u}\frac{\partial \mathbf{U}}{\partial \xi} + \bar{v}\frac{\partial \mathbf{U}}{\partial \eta}\right)^{n}_{i,j}\\ &+ \nu\Delta t\left[\frac{1}{J^2}\left(a\frac{\partial^2 \mathbf{U}}{\partial \xi^2} - 2b\frac{\partial^2 \mathbf{U}}{\partial \xi \partial \eta} + c\frac{\partial^2 \mathbf{U}}{\partial \eta^2}\right) + p\frac{\partial \mathbf{U}}{\partial \xi} + q\frac{\partial \mathbf{U}}{\partial \eta}\right]^{n}_{i,j} + \mathbf{f}^{n}_{i,j}\Bigg\}\end{aligned} \tag{4.6.8a}$$

Corrector

$$\begin{aligned}\mathbf{U}^{n+1}_{i,j} = \frac{1}{2}(\mathbf{U}^{*}_{i,j} + \mathbf{U}^{n}_{i,j}) &+ \frac{\Delta t}{2}\left[-\left(\bar{u}\frac{\partial \mathbf{U}}{\partial \xi} + \bar{v}\frac{\partial \mathbf{U}}{\partial \eta}\right)^{*}_{i,j}\right]\\ &+ \frac{\nu\Delta t}{2}\left[\frac{1}{J^2}\left(a\frac{\partial^2 \mathbf{U}}{\partial \xi^2} - 2b\frac{\partial^2 \mathbf{U}}{\partial \xi \partial \eta} + c\frac{\partial^2 \mathbf{U}}{\partial \eta^2}\right) + p\frac{\partial \mathbf{U}}{\partial \xi} + q\frac{\partial \mathbf{U}}{\partial \eta}\right]^{*}_{i,j} + \frac{\Delta t}{2}\mathbf{f}^{n+1}_{i,j}\end{aligned} \tag{4.6.8b}$$

It is now clear that the solution of the governing equation (4.6.4) is replaced by the solution of transformed equation (4.6.5) in which finite difference formulas of Chapter 3 can be used using the grid system of Figure 4.6.1b. This cumbersome procedure can be avoided if finite volume methods (Chapter 7) or finite element methods (Part Three) are used.

4.7 EXAMPLE PROBLEMS

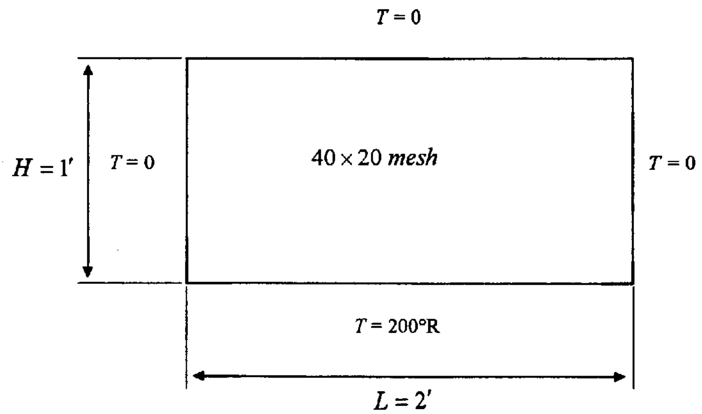
The purpose of this chapter was to list or summarize the existing numerical schemes for later references in forthcoming chapters. Thus, examples shown in this section are limited to simple problems for the benefit of the uninitiated reader.

4.7.1 ELLIPTIC EQUATION (HEAT CONDUCTION)

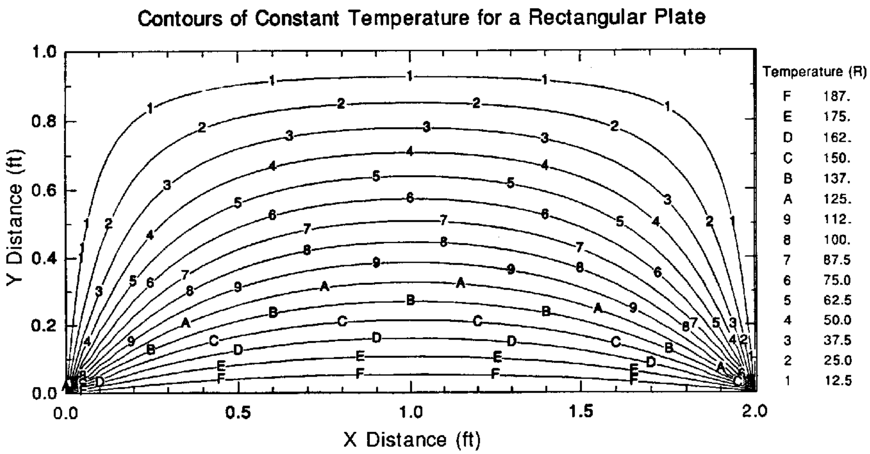
In this example, we demonstrate the solution of steady state heat conduction,

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

with the geometry and boundary conditions as shown in Figure 4.7.1.1a. The analytical



(a)



(b)

Figure 4.7.1.1 Heat conduction problem. (a) Geometry and discretization (40 × 20 mesh). (b) Computed results.

solution is given by

$$T = 200\left[2\sum_{n=1}^{N}\frac{1-(-1)^n}{n\pi}\frac{\sinh\dfrac{n\pi(H-y)}{L}}{\sinh\dfrac{n\pi H}{L}}\sin\frac{n\pi x}{L}\right]$$

Required: Solve using the point successive over-relaxation (PSOR).

Solution: The results for 40×20 mesh are shown in Figure 4.7.1.1b. The optimum relaxation parameter in this case is $\omega = 1.7$. The average error is approximately 0.5% as compared with the analytical solution ($N = 100$).

Remarks: For this simple problem, all methods introduced in this section will provide similar results.

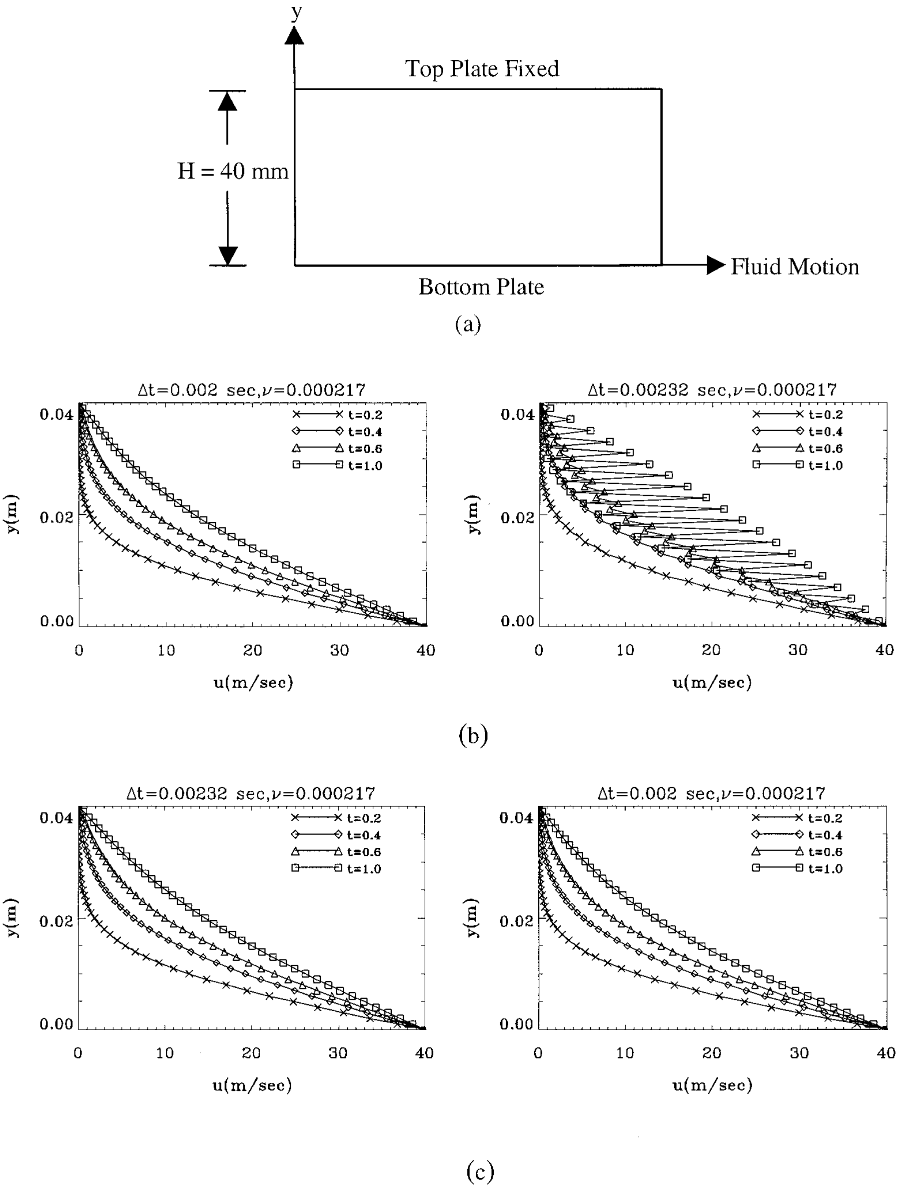


Figure 4.7.2.1 Couette flow. (**a**) Couette flow geometry. (**b**) Velocity profiles for FTCS explicit method (40 elements). (**c**) Velocity profiles for Crank-Nicolson method (40 elements).

4.7.2 PARABOLIC EQUATION (COUETTE FLOW)

Consider the Couette flow characterized by the parabolic equation,

$$\frac{\partial u}{\partial t} - \nu \frac{\partial^2 u}{\partial y^2} = 0, \qquad \nu = 0.000217\,\mathrm{m}^2/\mathrm{s}$$

with the geometry given in Figure 4.7.2.1a and

Initial conditions at $t = 0$ $\begin{cases} u = u_0 = 40\,\text{m/s}, & y = 0 \\ u = 0, \quad 0 < y \leq h \end{cases}$

Boundary conditions at $t > 0$ $\begin{cases} u = u_o = 40\,\text{m/s}, & y = 0 \\ u = 0, & y = h \end{cases}$

Required: Solve by FTCS and Crank-Nicolson methods with the initial and boundary conditions as shown below.

Solution: The results are shown in Figure 4.7.2.1b. As expected, FTCS for $d = .5034 > 1/2$ is unstable whereas Crank-Nicolson gives stable results regardless of diffusion number ranges.

4.7.3 HYPERBOLIC EQUATION (FIRST ORDER WAVE EQUATION)

The governing equation is given by

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \qquad a = 300\frac{m}{s}$$

with

Initial conditions at $t = 0$ $\begin{cases} u(x) = 0 & 0 \leq x \leq 50 \\ u(x) = 100 \sin \pi \dfrac{(x-50)}{60} & 50 \leq x \leq 110 \\ u(x) = 0 & 110 \leq x \leq 300 \end{cases}$

Boundary conditions at $t > 0$ $\begin{cases} u(x) = 0 & x = 0 \\ u(x) = 0 & x = L \end{cases}$

Explicit Schemes

Required: Solve by explicit schemes, (a) first order upwind scheme (FTBS), (b) Lax-Wendroff scheme, and FTCS implicit scheme.

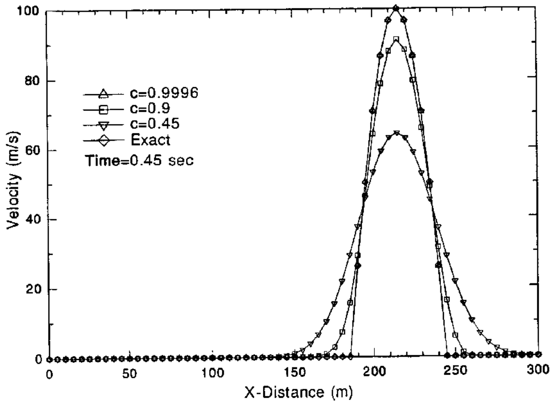
$$\begin{array}{llll} \Delta x = 5, & \Delta t = 0.01666 & (C = 0.9996) & \text{(CFL number)} \\ \Delta x = 5, & \Delta t = 0.015 & (C = 0.9) & \\ \Delta x = 5 & \Delta t = 0.0075 & (C = 0.45) & \end{array}$$

Solution: The results are as shown in Figure 4.7.3.1. Note that the exact solution is obtained for both methods for $C = 1$. However, as C decreases, FTBS becomes dissipative, whereas the Lax-Wendroff scheme (second order accurate) becomes dispersive.

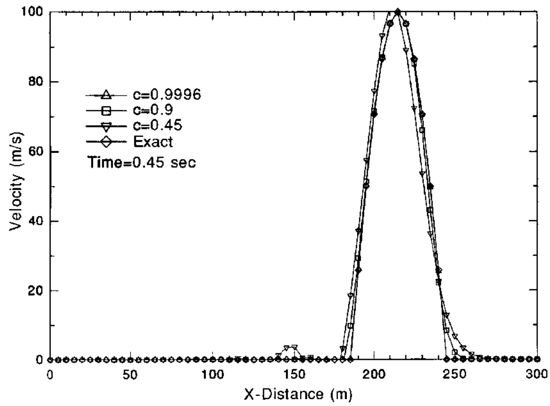
Implicit Schemes

Required: Solve by implicit scheme (FTCS).

Solution: The results are shown in Figure 4.7.3.2. This scheme is very dissipative at high C values. Although unconditionally stable, the results are poor, particularly with large time steps (large Courant number).



(a)



(b)

Figure 4.7.3.1 Solutions of first order wave equation by FTBS and Lax-Wendroff schemes, 60 nodes. **(a)** First order upwind (FTBS). **(b)** Lax-Wendroff scheme.

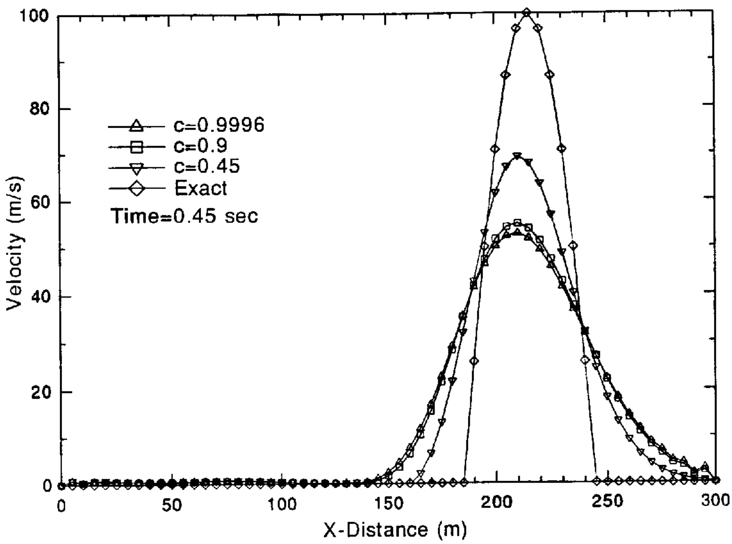


Figure 4.7.3.2 Solution of first order wave equation by FTCS implicit scheme.

4.7.4 HYPERBOLIC EQUATION (SECOND ORDER WAVE EQUATION)

The second order wave equation is considered in this example.

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}$$

Two sets of initial conditions are required:

Initial conditions

$$\text{(a) at } t = 0 \quad \begin{cases} u(x) = 0 & 0 \le x \le 100 \\ u(x) = 100 \sin\left[\dfrac{\pi(x-100)}{120}\right] & 100 \le x \le 220 \\ u(x) = 0 & 220 \le x \le 300 \end{cases}$$

$$\text{(b) at } t = 0 \quad \frac{\partial u(x)}{\partial t} = 0$$

Boundary conditions

$$t = 0 \quad \begin{cases} u(x) = 0 & x = 0 \\ u(x) = 0 & x = L \end{cases}$$

Required: Solve by the midpoint leapfrog scheme.

Solution: The results (Figure 4.7.4.1) are obtained at $t = 0.28$ seconds. The best solution occurs for $C = 1$. Note that dispersion errors occur for C less than 1.

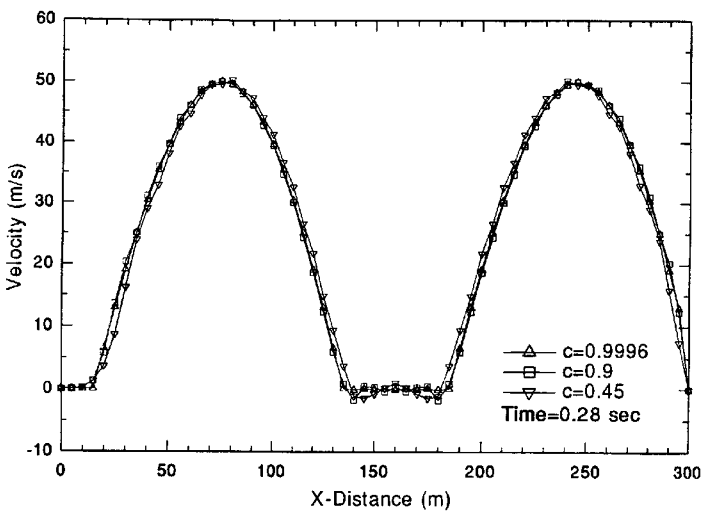


Figure 4.7.4.1 Solution of second order wave equation by midpoint leapfrog scheme, 60 nodes.

4.7.5 NONLINEAR WAVE EQUATION

Consider the nonlinear wave equation in the form

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

or

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0 \quad \text{with } F = \frac{1}{2}u^2$$

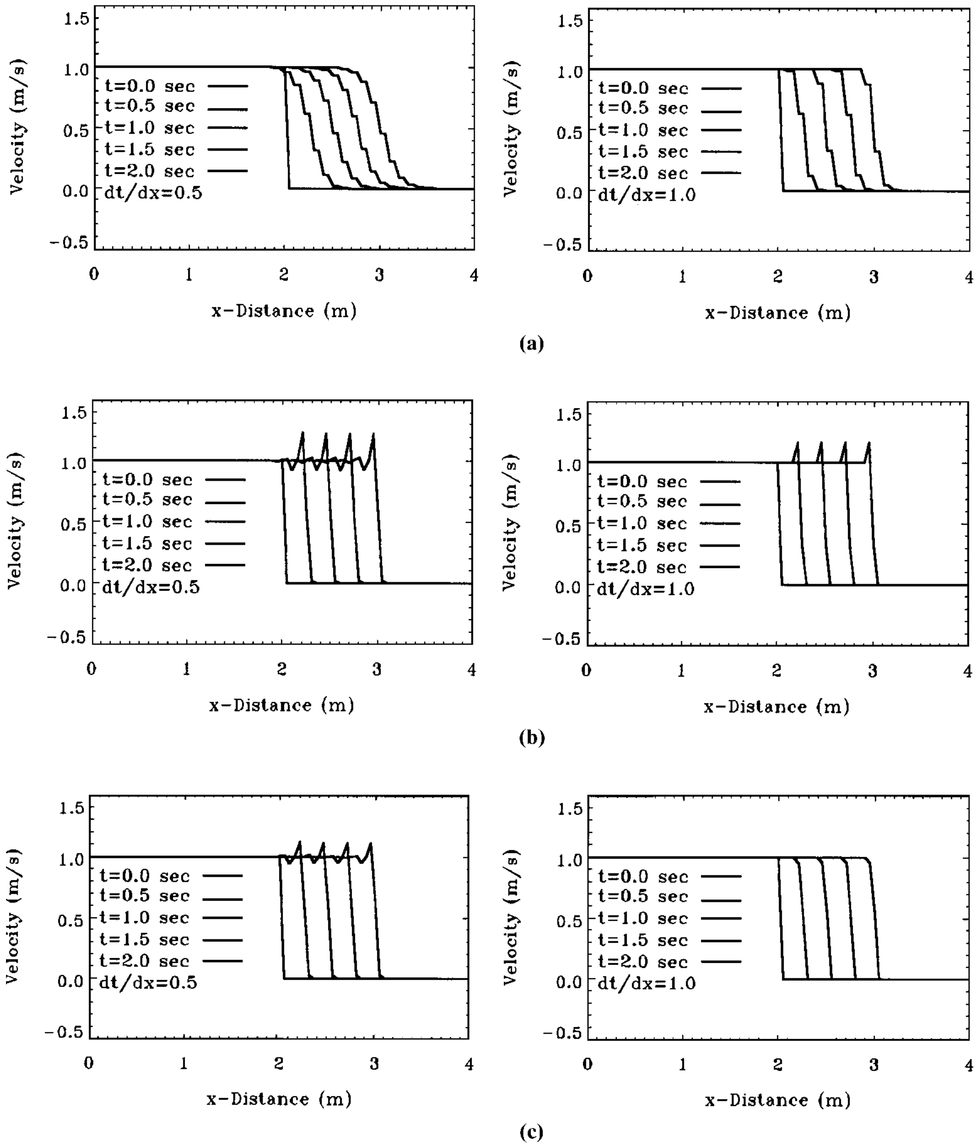


Figure 4.7.5.1 Solution of nonlinear wave equation by various methods. **(a)** Lax method (80 elements). **(b)** Lax-Wendroff method (80 elements). **(c)** MacCormack method (80 elements).

The following initial and boundary conditions are to be used:

$$u(x, 0) = 1 \quad 0 \leq x \leq 2$$

$$u(x, 0) = 0 \quad 2 \leq x \leq 4$$

Required: Solve by (a) Lax method, (b) Lax-Wendroff method, and (c) MacCormack method.

Solution: The results are obtained with $\Delta t/\Delta x = 1$ and $\Delta t/\Delta x = 0.5$. Referring to Figure 4.7.5.1, the Lax method is dissipative, whereas the Lax-Wendroff method is dispersive. This trend is worse when the Courant number is smaller. The MacCormack method gives better results particularly with Courant number near 1. It is still dispersive at lower Courant number, but better than the Lax-Wendroff scheme.

4.8 SUMMARY

In this chapter, FDM schemes for typical elliptic, parabolic, and hyperbolic partial differential equations and Burgers' equation have been presented. These equations do not represent complete fluid dynamics phenomena, but the computational schemes described herein do constitute the basis for computations involved in incompressible and compressible flows. Concepts of explicit and implicit schemes with von Neumann stability analyses are expected to play significant roles in all aspects of computational methods in fluid dynamics and heat transfer.

Although most of the computational schemes for FDM presented in this chapter are in terms of one-dimensional applications, their extensions to multidimensions including noncartesian orientations of physical domain can be accomplished by transformation into the cartesian computational domain.

In practical applications, most physical phenomena in fluid mechanics and heat transfer are multidimensional. Thus, significant modifications and improvements over the simple approaches introduced in this chapter are required in dealing with incompressible and compressible flows, which are the subjects of the subsequent chapters.

REFERENCES

Courant, R., Friedrichs, K. O., and Lewy, H. [1962]. On the partial differential equations of mathematical physics. *IBM J. Res. Dev.* 11, 215–24.

Demel, J. W. [1997]. *Applied Numerical Linear Algebra*. Philadelphia, PA: SIAM.

Dahlquist, G. and Bjork, A. [1974]. *Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall.

Hageman, P. and Young, D. M. [1981]. *Applied Iterative Methods*. New York: Academic Press.

Ortega, J. M. and Rheinboldt, W. C. [1970]. *Iterative Solutions of Non-Linear Equations in Several Variables*. London: Wiley.

Saad, Y. [1996]. *Iterative Methods for Sparse Linear Systems*. Boston: PWS Publishing.

Thomas, L. H. [1949]. Elliptic problems in linear differential equations over a network. Watson Sci. Comp. Lab., Columbia University, NY.

Varga, R. S. [1962]. *Matrix Iterative Analysis*. Englewood Cliffs, NJ: Prentice-Hall.

Wachspress, E. L. [1966]. *Iterative Solution of Elliptic Systems*. Englewood Cliffs, NJ: Prentice-Hall.