

## Chapter Four

### Parabolic Equations

#### 4.1 Overview

Equations of motion in fluid mechanics are frequently reduced to parabolic formulations. Boundary layer equations as well as Navier-Stokes equations are examples of such formulations. In addition, the unsteady heat conduction equation is also parabolic equation.

**4-2 One-dimensional Heat conduction Equation:**  
The unsteady heat conduction equation in one-space dimension can be expressed as

$$\frac{\partial U}{\partial t} = \alpha \frac{\partial^2 U}{\partial X^2} \quad (4-1)$$

where  $\alpha$  is assumed constant.

Various finite difference approximations can be used to represent the derivatives in Equation (4-1). The resulting finite difference equations are presented in the following sections.

### 4-2-1 Explicit Methods

This section introduces some of the commonly used explicit methods for solving parabolic equations.

4-2-1.1 The forward time/central space (FTCS) method:  
In this method, the time derivative,  $\frac{\partial u}{\partial t}$ , will be represented by a forward difference approximation which is of order  $\Delta t$ :

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{\Delta t} + O(\Delta t) \quad (4-2)$$

Using the second-order central differencing of order  $(\Delta x)^2$  for the space derivative in Eq(4-1):

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} + O((\Delta x)^2) \quad (4-3)$$

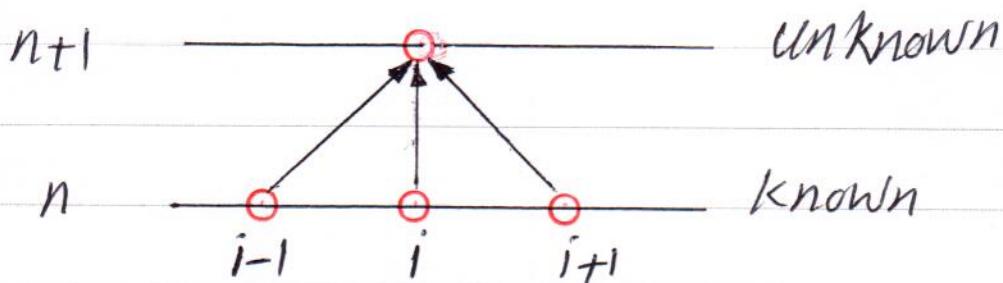
For the model equation (4-1), the resulting FDE is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} \quad (4-4)$$

In Eq(4-4),  $U_i^{n+1}$  is the only unknown and therefore, it can be computed from the following:

$$U_i^{n+1} = U_i^n + \frac{\alpha(\Delta t)}{(\Delta x)^2} (U_{i+1}^n - 2U_i^n + U_{i-1}^n) \quad (4-5)$$

Thus, the second-order PDE has been replaced by an algebraic equation. Graphical representation of the grid points in Eq(4-5) is shown in Figure (4-1).



Fig(4-1) Grid points for the explicit formulation.

Eq(4-5) is of order  $[(\Delta t), (\Delta x)^2]$ . The solution is stable for  $[\alpha \Delta t / (\Delta x)^2 \leq 0.5]$ .

**4-2-1-2 The DuFort-Frankel method:**  
In this formulation, the time derivative is approxima-

ted by a central differencing which is of order  $(\Delta t)^2$ . The second-order space derivative is also approximated by a central differencing of order  $(\Delta x)^2$ . However, due to stability considerations,  $u_i^n$  in the diffusion term is replaced by the average value of  $u_i^{n+1}$  and  $u_i^{n-1}$ . The resulting FDE is

$$\frac{u_i^{n+1} - u_i^{n-1}}{2 \Delta t} = \alpha \frac{u_{i+1}^n - 2 \frac{u_i^{n+1} + u_i^{n-1}}{2} + u_{i-1}^n}{(\Delta x)^2} \quad (4-6)$$

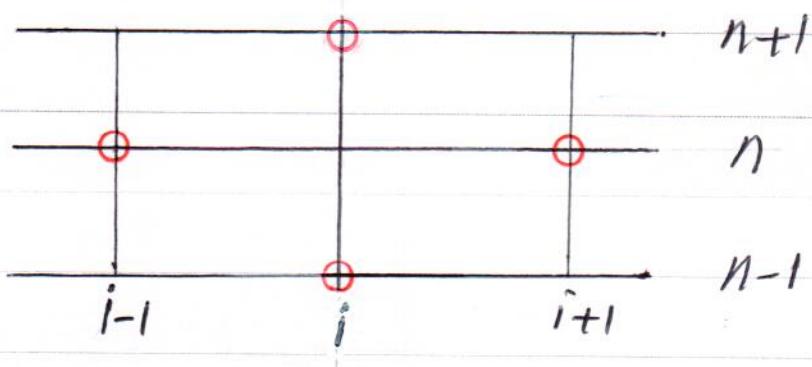
$$u_i^{n+1} = u_i^{n-1} + \frac{2 \alpha (\Delta t)}{(\Delta x)^2} [u_{i+1}^n - u_i^{n+1} - u_i^{n-1} + u_{i-1}^n] \quad (4-7)$$

The above equation can be solved explicitly for the unknown  $u_i$  at the time level  $n+1$ . Thus,

$$\left[ 1 + \frac{2 \alpha (\Delta t)}{(\Delta x)^2} \right] u_i^{n+1} = \left[ 1 - 2 \frac{\alpha (\Delta t)}{(\Delta x)^2} \right] u_i^{n-1} + \frac{2 \alpha (\Delta t)}{(\Delta x)^2} [u_{i+1}^n + u_{i-1}^n] \quad (4-8)$$

This method is unconditionally stable. The values of  $u_i$  at time levels  $n$  and  $n-1$  are required to start the computation. Therefore, either two sets of

data must be specified, or a one-step method can be used as starter. Of course, for the one-step ( $\Delta t$ ) starter solution, only one set of initial data, say at  $n-1$ , is required to generate the solution at  $n$ . With the values of  $u_i$  at  $n-1$  and  $n$  specified, the DuFort-Frankel method can be used. Since the solution at the unknown station requires data from two previous stations, computer storage requirements will increase. The grid points involved in Eq(4-8) are shown in figure(4-2).



Fig(4-2) Grid points for the DuFort-Frankel method.

## 4-2-2 Implicit Methods:

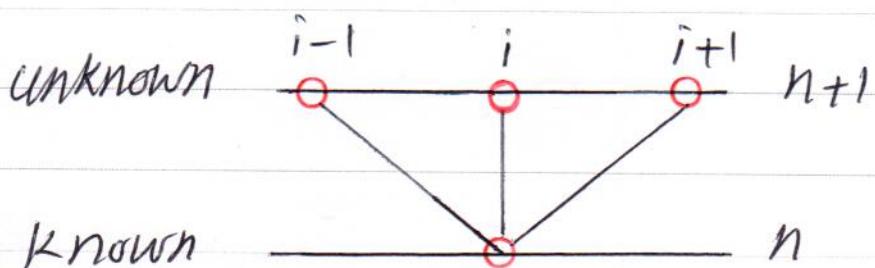
In this section some commonly used implicit formulations are described.

### 4-2-2-1 The Laasonen Method:

Eq(4-1) can be discretized as

$$\frac{U_i^{n+1} - U_i^n}{(\Delta t)} = \alpha \frac{U_{i+1}^{n+1} - 2U_i^{n+1} + U_{i-1}^{n+1}}{(\Delta x)^2} \quad (4-9)$$

The grid points are given in figure (4-3).



Fig(4-3) The grid points for the Laasonen method.

Applying the formulation to all grid points would lead to a set of linear algebraic equations.

Rearranging Eq(4-9), gives

$$\frac{\alpha \Delta t}{\Delta x^2} U_{i-1}^{n+1} - \left[ 1 + 2 \frac{\alpha \Delta t}{\Delta x^2} \right] U_i^{n+1} + \frac{\alpha \Delta t}{\Delta x^2} U_{i+1}^{n+1} = - U_i^n \quad (4-10)$$

Eq(4-10) can be written in the following form

$$a_i^n U_{i-1}^{n+1} + b_i^n U_i^{n+1} + c_i^n U_{i+1}^{n+1} = D_i^n \quad (4-11)$$

Where

$$a_i^n = \frac{\alpha \Delta t}{\Delta x^2}, \quad b_i^n = -\left[1 + 2 \frac{\alpha \Delta t}{\Delta x^2}\right]$$

$$c_i^n = \frac{\alpha \Delta t}{\Delta x^2}, \quad D_i^n = -U_i^n \quad (4-12)$$

Now, applying Eq(4-11) to all the grid points will result in the following set of linear algebraic equations:

$$i=2 \quad a_2 U_1 + b_2 U_2 + c_2 U_3 = D_2 \quad (4-13a)$$

$$i=3 \quad a_3 U_2 + b_3 U_3 + c_3 U_4 = D_3$$

$$i=4 \quad a_4 U_3 + b_4 U_4 + c_4 U_5 = D_4$$

:

$$i=m-2 \quad a_{m-2} U_{m-3} + b_{m-2} U_{m-2} + c_{m-2} U_{m-1} = D_{m-2}$$

$$i=m-1 \quad a_{m-1} U_{m-2} + b_{m-1} U_{m-1} + c_{m-1} U_m = D_{m-1} \quad (4-13b)$$

When the above equations are represented in a matrix form, the coefficient matrix is tridiagonal.

Assume that Dirichlet boundary conditions are imposed. Therefore, the values of the dependent variable  $U$  at  $i=1$ ,  $U_1$ , and at  $i=m$ ,  $U_m$ , are given. Then Eq(4-13a) and Eq(4-13b) can be written as

$$b_2 U_2 + c_2 U_3 = D_2 - a_2 U_1$$

$$a_{m-1} U_{m-2} + b_{m-1} U_{m-1} = D_{m-1} - c_{m-1} U_m$$

Now, the set of equations in matrix formulation is

$$\begin{bmatrix} b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \\ & \vdots & \\ a_{m-2} & b_{m-2} & c_{m-2} \\ a_{m-1} & b_{m-1} \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \\ \vdots \\ U_{m-2} \\ U_{m-1} \end{bmatrix} = \begin{bmatrix} D_2 - a_2 U_1 \\ D_3 \\ D_4 \\ \vdots \\ D_{m-2} \\ D_{m-1} - c_{m-1} U_m \end{bmatrix} \quad (4-14)$$

This is called as the tridiagonal matrix algorithm (TDMA), also known as the Thomas algorithm. However, Gaussian elimination can be used to solve tridiagonal systems of equations. A first sweep eliminates the  $a_i$  coefficients and then a backward substitution produce the solution.

For the description of Gaussian elimination we do not need the column of unknowns, therefore, the matrix can be written in the following form:

$$\left[ \begin{array}{ccc|c} b_2 & c_2 & & D_2 - a_2 u_1 \\ a_3 & b_3 & c_3 & D_3 \\ a_4 & b_4 & c_4 & D_4 \\ \vdots & & & \vdots \\ a_{m-2} & b_{m-2} & c_{m-2} & D_{m-2} \\ a_{m-1} & b_{m-1} & & D_{m-1} - c_{m-1} u_m \end{array} \right] \quad (4-15)$$

Gaussian elimination method for the tridiagonal system (4-15) has the following basic structure:

$$\left\{ \begin{array}{l} R_1: \text{unchanged} \\ R_2 = R_2 - \frac{a_3}{b_2} R_1 \\ R_3 = R_3 - \frac{a_4}{b_3} R_2 \end{array} \right. \quad (4-16)$$

Where  $R_1$ ,  $R_2$  and  $R_3$  are the rows of the extended matrix (4-15). Therefore, the coefficients  $b_i$  and  $D_i$  can be determined as: