

**CIVIL & ENVIRONMENTAL ENGINEERING 253**

**Mathematical Models for Water Quality**

**CLASS NOTES**

**Michael K. Stenstrom**

**Civil & Environmental Engineering Department**

**October, 2007**

## PREFACE

These notes cover the majority of the topics included in Civil & Environmental Engineering 253, Mathematical Models for Water Quality. The course stresses practical ways of solving partial differential equations (PDEs) that arise in environmental engineering. The course and the notes do not address the development or applications models, and the derivation of a PDE for a particular problem is left to the student, and is covered in other courses in our curriculum.

The course stresses finite difference methods (FDMs), as opposed to finite element methods (FEMs). I prefer this approach because it is conceptually easier. Furthermore, most environmental engineering problems have simple, regular geometries, with the frequent exception of groundwater simulations. For these geometrically simple problems, finite difference methods are superior.

Finite element methods have traditionally been used for irregular geometries. Recently, computer-based grid generation techniques have been developed, which are extending the utility of the finite difference methods to irregular geometrics. With this extension finite difference methods possess the geometric flexibility of finite element methods. Finite difference methods are also easily extended to 2 and 3 dimensions.

The course also discusses issues with numeric solutions that allow the use of parallel computers.

# 1. DISCRETE VARIABLES AND FINITE DIFFERENCES

Consider the following function:

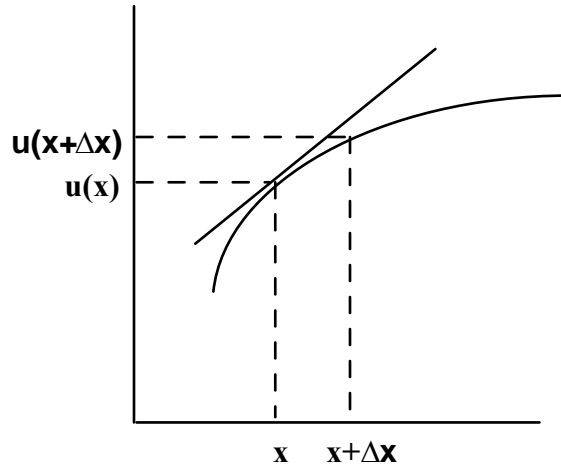


Figure 1.1 Continuous Differentiable Function

If  $\Delta x$  approaches zero we have

$$u(x + dx) = u(x) + \frac{du(x)}{dx} \cdot dx \tag{1.1}$$

or

$$dx \frac{du}{dx} = du \tag{1.2}$$

The relationship is true for the case of  $u(x)$  being a differentiable function.

In two dimensions we have

$$dy \frac{\partial u}{\partial y} + dx \frac{\partial u}{\partial x} = du \tag{1.3}$$

When we work with finite differences we cannot let  $\Delta x \rightarrow 0$ , because our ability to compute is finite. Therefore we have a finite difference approximation for Equation (1.1), as follows:

$$u(x + \Delta x) = u(x) + \Delta x \frac{du(x)}{dx} + \frac{\Delta^2 x}{2!} \frac{d^2 u(x)}{dx^2} + \frac{\Delta^3 x}{3!} \frac{d^3 u(x)}{dx^3} + \dots + \text{HOTs} \tag{1.4}$$

where HOTs refers to the higher order terms.

This is a Taylor Series.

For most of our work we will use constant  $\Delta x$ 's or  $\Delta y$ 's, which means we can write our finite difference variables as follows:

$$x_0 = 0, \quad x_i = i(\Delta x) \quad (1.5)$$

$i$  takes on values of 0 to  $iR$ , where

$$iR \cdot \Delta x = x_{\max} \quad (1.6)$$

and

$$x_{i+1} = x_i + \Delta x \quad (1.7)$$

$$x_{i+1} = \Delta x(i + 1) \quad (1.8)$$

$$x_{i-1} = x_i - \Delta x \quad (1.9)$$

$$x_{i-1} = (i - 1)\Delta x \quad (1.10)$$

Using this notation  $x_0$  is equal to 0. This can be confusing when writing computer codes, especially in FORTRAN. Pre-FORTRAN-77 compilers do not allow zero array subscripts, which requires one to code  $x(1)$  equal to  $x_0$ , or  $x = 0$ . Most FORTRAN codes still avoid using array subscripts equal to zero. Mathematically it is more convenient to use  $x_0$  to denote  $x = 0$ . Programmers must often use  $x(1)$  to denote  $x = 0$ .

We can write our Taylor Series using partials instead of total derivatives, as follows:

$$u_{i+1} = u_i + \Delta x \left. \frac{\partial u}{\partial x} \right|_i + \frac{\Delta^2 x}{2!} \left. \frac{\partial^2 u}{\partial x^2} \right|_i + \frac{\Delta^3 x}{3!} \left. \frac{\partial^3 u}{\partial x^3} \right|_i + \dots + \frac{\Delta^n x}{n!} \left. \frac{\partial^n u}{\partial x^n} \right|_i + \text{HOTs} \quad (1.11)$$

This equation estimates  $u_{i+1}$  from  $u_i$ , and is called a *forward* Taylor Series, since it estimates  $u_{i+1}$  in the direction of increasing  $x$ .

In the direction of decreasing  $x$  we can write a *backwards* Taylor Series, as follows:

$$u_{i-1} = u_i - \Delta x \left. \frac{\partial u}{\partial x} \right|_i + \frac{\Delta^2 x}{2!} \left. \frac{\partial^2 u}{\partial x^2} \right|_i - \frac{\Delta^3 x}{3!} \left. \frac{\partial^3 u}{\partial x^3} \right|_i + \dots + (-1)^n \frac{\Delta^n x}{n!} \left. \frac{\partial^n u}{\partial x^n} \right|_i + \text{HOTs} \quad (1.12)$$

We can solve Equation (1.11) or (1.12) for the first partial as follows:

$$\left. \frac{\partial u}{\partial x} \right|_i = \frac{u_i - u_{i-1}}{\Delta x} + \frac{\Delta x}{2!} \left. \frac{\partial^2 u}{\partial x^2} \right|_i - \frac{\Delta^2 x}{3!} \left. \frac{\partial^3 u}{\partial x^3} \right|_i + \dots + \text{HOTs} \quad (1.13)$$

We truncate the series to obtain:

$$\left. \frac{\partial u}{\partial x} \right|_i = \frac{u_i - u_{i-1}}{\Delta x} \quad (1.14)$$

Equation (1.14) is called an analog, because it is analogous to the first derivative.

The largest truncated term is  $\left. \frac{\Delta x}{2!} \frac{\partial^2 u}{\partial x^2} \right|_i$ . It is of the magnitude  $\Delta x$ , or the first power of  $\Delta x$ .

We refer to this analog as *first-order correct*. We could have truncated before we divided by  $\Delta x$ , but the order of correctness would be the same.

We can also solve Equation (1.11) for  $\left. \frac{\partial u}{\partial x} \right|_i$  and obtain similar results. Both the forward Taylor series (Equation (1.11)) and the backward Taylor series (Equation (1.12)) give first-order correct analogs.

To obtain an approximation for the second derivative we add Equations (1.11) and (1.12) to obtain

$$u_{i+1} + u_{i-1} = 2u_i + \Delta^2 x \left. \frac{\partial^2 u}{\partial x^2} \right|_i + 2 \frac{\Delta^4 x}{4!} \left. \frac{\partial^4 u}{\partial x^4} \right|_i + \dots + \text{HOTs} \quad (1.15)$$

After solving for the second partial derivative we obtain:

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta^2 x} - \frac{\Delta^2 x}{12} \left. \frac{\partial^4 u}{\partial x^4} \right|_i - \dots - \text{HOTs} \quad (1.16)$$

We can truncate to obtain the second-order correct analog for the second derivative:

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta^2 x} \quad (1.17)$$

Finally we can truncate an improved first derivative by subtracting Equations (1.11) and (1.12) to obtain

$$\left. \frac{\partial u}{\partial x} \right|_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} - \frac{\Delta^2 x}{6} \left. \frac{\partial^3 u}{\partial x^3} \right|_i - \dots - \text{HOTs} \quad (1.18)$$

We truncate to obtain

$$\left. \frac{\partial u}{\partial x} \right|_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} \quad (1.19)$$

This is a second-order correct analog for the first partial derivative.

To summarize we have three finite difference analogs.

$$\left. \frac{\partial u}{\partial x} \right|_i = \frac{u_{i+1} - u_i}{\Delta x} \quad (\text{first-order correct}) \quad (1.20)$$

$$\left. \frac{\partial u}{\partial x} \right|_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} \quad (\text{second-order correct}) \quad (1.21)$$

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta^2 x} \quad (\text{second-order correct}) \quad (1.22)$$

From these analog we can construct finite difference equations for most partial differential equations. Occasionally we develop additional analogs for special purposes. Analog of any desired order of correctness can be developed, but usually second-order correct analogs are used for partial differential equations using finite differences.

The analogs we chose will depend upon the nature of the equations, parabolic, hyperbolic, or elliptic, and our needs for accuracy and efficiency.

## 2. CLASSIFICATIONS OF PARTIAL DIFFERENTIAL EQUATIONS

It is useful to classify partial differential equations into three types: parabolic, hyperbolic, and elliptic. The names arise due to the similarity of the classification mechanism to analytic geometry. Solution techniques exist for each type of equation. Generally an appropriate technique for one type of equation is not appropriate for another type.

Consider a system of two 1st order PDE's as follows:

$$a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + c \frac{\partial v}{\partial x} + d \frac{\partial v}{\partial y} = f_1 \quad (2.1)$$

$$e \frac{\partial u}{\partial x} + g \frac{\partial u}{\partial y} + h \frac{\partial v}{\partial x} + i \frac{\partial v}{\partial y} = f_2 \quad (2.2)$$

We define the system as *quasi-linear* if coefficients a- to -i can be functions (linear or non-linear) of x, y, or u or v. The problem is quasi-linear as long as none of these coefficients contain terms involving  $\frac{\partial u}{\partial x}$  or  $\frac{\partial u}{\partial y}$ .

If the coefficients a to i and the functions  $f_1$  and  $f_2$  are constant or functions only of x or y, then the problem is linear. If the coefficients involve functions of  $\frac{\partial u}{\partial x}$  or  $\frac{\partial u}{\partial y}$ , the problem is nonlinear.

We can conveniently classify linear and quasi-linear sets of the previous equations. Nonlinear equations do not frequently arise in environmental problems.

Now, assuming u or v are differentiable, then we can represent the increment of u and v by *partial derivatives* along the x and y directions as follows

$$\Delta x \frac{\partial u}{\partial x} + \Delta y \frac{\partial u}{\partial y} \cong \Delta u \quad (2.3)$$

$$\Delta x \frac{\partial v}{\partial x} + \Delta y \frac{\partial v}{\partial y} \cong \Delta v \quad (2.4)$$

or as  $\Delta x, \Delta y \rightarrow 0$

$$dx \frac{\partial u}{\partial x} + dy \frac{\partial u}{\partial y} = du \quad (2.5)$$

$$dx \frac{\partial v}{\partial x} + dy \frac{\partial v}{\partial y} = dv \quad (2.6)$$

The existence of the above derivatives are assured because we are working with continuously differentiable function.

If we combine the directional derivatives with Equations (2.1) and (2.2) we have four equations and four unknowns, and we can write them as follows:

$$\begin{bmatrix} a & b & c & d \\ e & g & h & i \\ dx & dy & 0 & 0 \\ 0 & 0 & dx & dy \end{bmatrix} \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ du \\ dv \end{bmatrix} \quad (2.7)$$

Suppose the above system of equations is written at a particular point P in the domain x and y, and that u and v are known at this point. Since a through i, and f<sub>1</sub> and f<sub>2</sub> are functions of x, y, v or u they are also known. Suppose that u and v are known at another point in the domain Γ. Therefore, du and dv will be known. In this case all the coefficients in Equation (2.7) will be known. Equation (2.7) will have a unique solution if the determinant is non-zero. This implies that the derivative as we approach point P from either direction will be equal.

The case where the determinant is zero is more interesting. A zero determinant implies no unique solution for the derivatives in Equation (2.7). Consequently, discontinuities in the derivatives may occur as one moves along different approaches to point P.

To determine the values of Δx or Δy we take the determinate as follows:



$$dx \begin{vmatrix} b & c & d \\ g & h & i \\ 0 & dx & dy \end{vmatrix} - dy \begin{vmatrix} a & c & d \\ e & h & i \\ 0 & dx & dy \end{vmatrix} = 0$$

$$dx \left[ -dx \begin{vmatrix} b & d \\ g & i \end{vmatrix} + dy \begin{vmatrix} b & c \\ g & h \end{vmatrix} \right] - dy \left[ -dx \begin{vmatrix} a & d \\ e & i \end{vmatrix} + dy \begin{vmatrix} a & c \\ e & h \end{vmatrix} \right] = 0 \quad (2.8)$$

$$-d^2x(bi - gd) + dx dy(bh - cg) + dy dx(ai - de) - d^2y(ah - ec) = 0 \quad (2.9)$$

collecting terms we obtain

$$d^2y(ce - ah) + dy dx(ai - de + bh - cg) + d^2x(gd - bi) = 0 \quad (2.10)$$

Now we want to see what properties of  $dx$  and  $dy$  will cause the determinate to be zero.

We see that we have a quadratic equation in  $\frac{dy}{dx}$

$$\left(\frac{dy}{dx}\right)^2 (ce - ah) + \frac{dy}{dx} (ai - de + bh - cg) + dg - bi = 0 \quad (2.11)$$

Recall the quadratic formula

$$AX^2 + BX + C = 0 \quad (2.12)$$

$$X = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (2.13)$$

We let

$$\begin{aligned} A &= ce - ah \\ B &= (ai - de + bh - cg) \\ C &= dg - bi \end{aligned}$$

We have real or complex roots depending on  $B^2 - 4AC$ . The following cases exist:

$$B^2 - 4AC > 0 \quad 2 \text{ real roots - hyperbolic}$$

$$B^2 - 4AC = 0 \quad 1 \text{ real root - parabolic}$$

$$B^2 - 4AC < 0 \quad \text{complex roots - elliptic}$$

The value of the roots,  $\frac{dx}{dy}$  or  $\frac{dy}{dx}$  are called characteristic directions. The solution propagates along characteristic directions or lines. When numerically solving PDEs, we

must not select values of  $\Delta x$  or  $\Delta y$  which cross the characteristic directions; otherwise we have non-unique solutions.

Now consider a second example.

$$a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} = f \quad (2.14)$$

This is a second-order problem in one equation. Note that we could reduce this to a 2 first-order equation.

The directional derivatives become

$$d\left(\frac{\partial u}{\partial x}\right) = \frac{\partial^2 u}{\partial x^2} dx + \frac{\partial^2 u}{\partial x \partial y} dy \quad (2.15)$$

$$d\left(\frac{\partial u}{\partial y}\right) = \frac{\partial^2 u}{\partial x \partial y} dx + \frac{\partial^2 u}{\partial y^2} dy \quad (2.16)$$

our matrix form is

$$\begin{bmatrix} a & b & c \\ dx & dy & 0 \\ 0 & dx & dy \end{bmatrix} \begin{bmatrix} \frac{\partial^2 u}{\partial x^2} \\ \frac{\partial^2 u}{\partial x \partial y} \\ \frac{\partial^2 u}{\partial y^2} \end{bmatrix} = \begin{bmatrix} f \\ d\left(\frac{\partial u}{\partial x}\right) \\ d\left(\frac{\partial u}{\partial y}\right) \end{bmatrix} \quad (2.17)$$

The determinant is

$$-dx \begin{vmatrix} a & c \\ dx & 0 \end{vmatrix} + dy \begin{vmatrix} a & b \\ dx & dy \end{vmatrix} = 0$$

$$-dx(0 - cdx) + dy(ady - bdx) = 0 \quad (2.18)$$

We take

$$c(dx)^2 + a(dy)^2 - bdx dy = 0 \quad (2.19)$$

or

$$a\left(\frac{dy}{dx}\right)^2 - b\frac{dy}{dx} + c = 0 \quad (2.20)$$

if we let  
 $A = a$   
 $B = -b$   
 $C = c$

So we have

|                 |                             |
|-----------------|-----------------------------|
| $B^2 - 4AC > 0$ | 2 real roots for hyperbolic |
| $B^2 - 4AC = 0$ | 1 real root for parabolic   |
| $B^2 - 4AC < 0$ | complex roots for elliptic  |

Consider a third example - the linear wave problem:

$$\frac{\partial^2 \mathbf{u}}{\partial x^2} - \alpha \frac{\partial^2 \mathbf{u}}{\partial y^2} = 0$$

|             |   |               |
|-------------|---|---------------|
| A           | = | 1             |
| B           | = | 0             |
| C           | = | $-\alpha$     |
| $B^2 - 4AC$ | = | $0 + 4\alpha$ |

The parameter  $\alpha$  is always positive so this PDE is always hyperbolic.

Note that the forcing functions,  $f_1$  and  $f_2$  in Equation (2.7) do not enter into the classification mathematics. Furthermore, note that the coefficients a- to i- in Equation (2.7) are not necessarily time invariant. Therefore, a transient problem could change its properties during its solution, which might require two entirely different solution techniques.

In later sections we will develop numerical techniques to solve hyperbolic PDEs using mathematics very similar to the classification procedures we used here. The technique involves finding the characteristic directions and selecting values of  $\Delta x$  and  $\Delta y$  (or  $\Delta x$ ,  $\Delta y$  and  $\Delta z$ ) in order to place grid points along the characteristic directions. In this way we avoid crossing a characteristic direction.

### 3. NON-DIMENSIONAL FORMS

It is customary to scale partial differential equations by introducing dimensionless variables. This was quite popular before the advent of computers. Many handbooks have tabulated solutions of commonly used PDE's. By solving the equation one can often use a tabulated solution. Such techniques are no longer considered "state-of-the-art", and many tabulated solutions contain serious errors or restrictions on the domain which are not apparent to the unsuspecting user. Nevertheless, it is still desirable to scale PDE's prior to their solution, and it is important to know the procedure.

Consider the dispersive-convective transport equation:

$$D \frac{\partial^2 C}{\partial x^2} - V \frac{\partial C}{\partial x} = \frac{\partial C}{\partial t} \quad (3.1)$$

where C is a concentration, V is velocity, D is a diffusivity or dispersion coefficient, and x and t are space and time. We check for dimensional consistency

$$\frac{l^2}{t} \cdot \frac{m}{l^3} \cdot \frac{1}{l^2} - \frac{l}{t} \cdot \frac{m}{l^3 \cdot l} = \frac{m}{l^3 t} \quad (3.2)$$

all terms have units of  $\frac{m}{l^3 t}$ . Note that differentiating with respect to a variable adds the units of the variable to the denominator of the term.

Let's make the problem dimensionless by letting

$$\begin{aligned} Z &= \frac{x}{x_{\max}} = \frac{x}{L} & \text{or } x &= LZ \\ u &= \frac{C}{C_{\max}} & \text{or } C &= uC_{\max} \\ \theta &= t \frac{D}{L^2} & \text{or } t &= \frac{\theta L^2}{D} \end{aligned}$$

By substituting for x, C, and t we obtain

$$D \left[ \frac{\partial}{\partial(LZ)} \frac{\partial(uC_{\max})}{\partial(LZ)} \right] - V \frac{\partial(uC_{\max})}{\partial(LZ)} = \frac{\partial(uC_{\max})}{\partial(\theta L^2 / D)} \quad (3.3)$$

$$\frac{C_{\max} D}{L^2} \frac{\partial^2 u}{\partial Z^2} - \frac{C_{\max} V}{L} \frac{\partial u}{\partial Z} = \frac{C_{\max}}{L^2 / D} \frac{\partial u}{\partial \theta} \quad (3.4)$$

$$\frac{\partial^2 u}{\partial Z^2} - \frac{LV}{D} \frac{\partial u}{\partial Z} = \frac{\partial u}{\partial \theta} \quad (3.5)$$

or

$$\frac{\partial^2 u}{\partial Z^2} - \text{Pe} \frac{\partial u}{\partial Z} = \frac{\partial u}{\partial \theta} \quad (3.6)$$

Therefore, we can describe all cases of this equation with one parameter,  $\frac{LV}{D}$ , which is called the Peclet Number (Pe). Some authors define the Peclet Number as  $\frac{D}{LV}$ , which is the reciprocal of its more common definition.

Consider this equation over ranges of the Peclet.

$$\begin{aligned} \text{as Pe} &\rightarrow \infty && \text{the problem becomes plug flow or hyperbolic} \\ \text{or Pe} &\rightarrow 0 && \text{the problem becomes pure diffusion or parabolic} \end{aligned}$$

The equation is parabolic for all finite values of Pe. For cases where Pe becomes very large we have the “near hyperbolic” problem. Many finite difference and finite element techniques becomes oscillatory for large values of Pe. We have special techniques to solve near hyperbolic problems. When  $D = 0$ , we scale the problem differently, and Pe does not exist.

Consider a second example of scaling an equation.

$$D \frac{\partial^2 C}{\partial x^2} - V \frac{\partial C}{\partial x} = \frac{\partial C}{\partial t} \quad (3.7)$$

where  $0 \leq x \leq x_{\max}$

Let

$$Z = \frac{x}{x_{\max}} = \frac{x}{L} \quad (3.8)$$

$$\theta = t \frac{V}{L} \quad (3.9)$$

$$u = \frac{C}{C_{\max}} \quad (3.10)$$

So we obtain

$$D \left[ \frac{\partial}{\partial(LZ)} \frac{\partial(uC_{\max})}{\partial(LZ)} \right] - V \frac{\partial(uC_{\max})}{\partial(LZ)} = \frac{\partial(uC_{\max})}{\partial(\frac{\theta L}{V})} \quad (3.11)$$

or

$$\frac{D}{LV} \frac{\partial^2 u}{\partial Z^2} - \frac{\partial u}{\partial Z} = \frac{\partial u}{\partial \theta} \quad (3.12)$$

or

$$\frac{1}{Pe} \frac{\partial^2 u}{\partial Z^2} - \frac{\partial u}{\partial Z} = \frac{\partial u}{\partial \theta} \quad (3.13)$$

It is easy to show that this is the same as our earlier sample. We still have one parameter, Pe, to describe our equation. We note that there is not a unique scaling procedure.

Consider a third and final example: heat conduction in a cylinder. We have a one dimensional problem in polar coordinates.

$$\alpha \left[ \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} \right] = \frac{\partial T}{\partial t} \quad (3.14)$$

This equation is the same as our previous examples, except in polar coordinates, with  $V = 0$ , and  $D = \alpha$ .

To scale we let

$$x = \frac{r}{r_{\max}} = \frac{r}{R} \quad (3.15)$$

$$\tau = t \frac{\alpha}{R^2} \quad (3.16)$$

$$u = \frac{T - T_i}{\alpha T_i} \quad (3.17)$$

when  $T_i$  is a reference temperature.

Substitute

$$\begin{aligned} T &= \alpha T_i u + T_i \\ t &= \frac{R^2 \tau}{\alpha} \\ r &= R x \end{aligned}$$

$$\alpha \left[ \frac{\partial^2 (\alpha T_i u + T_i)}{\partial (R x)^2} + \frac{1}{R x} \frac{\partial (\alpha T_i u + T_i)}{\partial (R x)} \right] = \frac{\partial (\alpha T_i u + T_i)}{\alpha \left( \frac{R^2 \tau}{\alpha} \right)} \quad (3.18)$$

which becomes

$$\frac{\alpha^2 T_i}{R^2} \frac{\partial^2 u}{\partial x^2} + \frac{\alpha^2 T_i}{R x} \frac{\partial u}{\partial x} = \frac{\alpha^2 T_i}{R^2} \frac{\partial u}{\partial \tau} \quad (3.19)$$

which simplifies to

$$\frac{\partial^2 u}{\partial x^2} + \frac{1}{x} \frac{\partial u}{\partial x} = \frac{\partial u}{\partial \tau} \quad (3.20)$$

One can see the value of scaling from Equation (3.20). A single numerical solution could be applied to many different problems using scaling factors, as defined by Equations 3.15 - 3.17.

## 4. PARABOLIC EQUATIONS IN ONE DIMENSION

The simplest and most often encountered partial differential equations are parabolic. They arise from heat and mass transfer problems. Let's consider our non-steady state diffusion-advection or diffusion-advection-reaction transport equation, as follows:

$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2} - V \frac{\partial C}{\partial x} - KC \quad (4.1)$$

with

$$\begin{aligned} C = 1, \quad x = 0, \quad t \geq 0 & \quad (\text{entrance boundary condition}), \\ \frac{\partial C}{\partial x} = 0, \quad x = 1, \quad t \geq 0 & \quad (\text{exit boundary condition}) \\ C = 0, \quad x > 0, \quad t = 0 & \quad (\text{initial condition}) \end{aligned}$$

### 4.1 Forward Difference

Let us apply the analogs developed previously. We will not scale the problem in order to see the effects of  $D$  and  $V$ . We will use second-order correct analogs for the spatial derivatives, and a first-order correct analog for the time derivative.

$$\frac{C_{i,n+1} - C_{i,n}}{\Delta t} = D \left[ \frac{C_{i+1,n} - 2C_{i,n} + C_{i-1,n}}{\Delta x^2} \right] - V \left[ \frac{C_{i+1,n} - C_{i-1,n}}{2\Delta x} \right] - KC_{i,n} \quad (4.2)$$

Note that we have two subscripts,  $i$  and  $n$ , for space and time, respectively.

Our shorthand for this equation becomes:

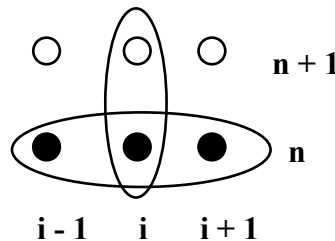


Figure 4.1 Forward Difference Analog - Interior Points

We note in Equation (4.1) that we have initial conditions and two boundary conditions. Therefore the problem is referred to as a “split boundary, initial value” problem. We need initial conditions to insure that we have a unique solution. The boundary conditions exist because we know something from the physics and chemistry of the problem at the



boundaries. For this problem we have a *fixed* inlet boundary and a *zero-flux* exit boundary. We will define a third boundary condition later.

We note from our analog picture above that the problem is explicit, since  $u_{i,n}$  is known from the initial conditions. We have one equation in one unknown, which we solve as follows;

$$C_{i,n+1} = C_{i,n} + \frac{D\Delta t}{\Delta^2 x} (C_{i+1,n} - 2C_{i,n} + C_{i-1,n}) - \frac{V\Delta t}{2\Delta x} (C_{i+1,n} - C_{i-1,n}) - \Delta t K C_{i,n} \quad (4.3)$$

Equation (4.3) is used to solve for all interior points. For the inlet boundary we place our analog as follows:

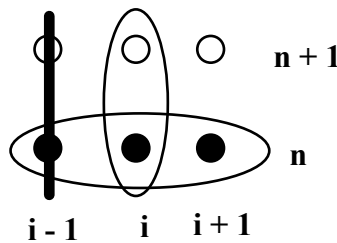


Figure 4.2 Forward Difference - Inlet Boundary

The value of  $u_{i-1,n}$  in Equation (4.3) becomes the boundary value. Since  $u_{i-1,n}$  is known at the boundary, we simply use it in our solution technique. When we are able to reduce the analogue to a single unknown equated to several known, we have an explicit equations. They are usually the easiest problems to solve.

For the exit boundary, we place the analog as follows:

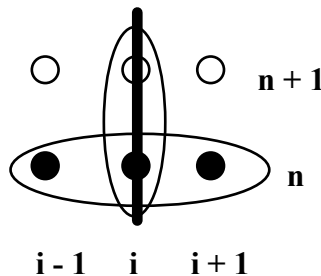


Figure 4.3 Forward Difference - Exit Boundary (zero flux)

The point at  $i = iR + 1$  does not physically exist and is called a false point. It is placed there for convenience. We now apply our exit boundary condition, as follows:

$$\left. \frac{\partial C}{\partial x} \right|_{iR} = 0 \quad (4.4)$$

$$0 = \frac{C_{i+1,n} - C_{i-1,n}}{2\Delta x} \quad (4.5)$$

$$C_{i+1,n} = C_{i-1,n} \quad (4.6)$$

$u_{i+1,n}$  is the false point, and we remove it by substituting Equation (4.6) into (4.3) as follows:

$$C_{i,n+1} = C_{i,n} + \frac{D\Delta t}{\Delta^2 x} (2C_{i-1,n} - 2C_{i,n}) - \frac{V\Delta t}{2\Delta x} (C_{i-1,n} - C_{i-1,n}) - K\Delta t C_{i,n} \quad (4.7)$$

The advective term involving the velocity vanishes because the partial derivative at the exit boundary is equal to zero.

This method is explicit since it only has one unknown term in each analog. It is called *forward difference* since it uses a forward Taylor series for the time derivative. One should not refer to this method as the “explicit method” because there are many explicit methods to solve this equation.

Forward difference is a popular method, especially with those who wish to use the simplest computer code. The method is only first-order correct in the time derivative and is *conditionally* stable. The method lends itself better to “pipe-lining” and parallel processing better than many other methods.

Conditional stability or instability results when the higher order terms that were truncated (Equations (1.11) or (1.13)) grow and become infinitely large. The error introduced by assuming the higher order terms are insignificant is called *truncation error*. This type of error is different than *round-off error*, which results because of a computer’s finite precision.

Problems are stable if the truncation error at each time step decreases with increasing time. When the error increases with each time step, the problem is unstable, and an infinitesimally small error will grow to extremely large error in only a few time steps. Also note that stable techniques do not insure accurate solutions, only finite solutions. One frequently hears that the solution is “unstable” when it appears to be inaccurate or oscillatory. This is an error. Solutions tend to infinity (“blow-up”) when they are unstable. Solutions are inaccurate when they are stable but have excessive round-off or truncation error.

We shall later see that forward difference is stable when

$$\frac{D\Delta t}{(\Delta^2 x)} \leq \frac{1}{2} \quad (4.8)$$

This constraint becomes very severe for small values of  $\Delta x$ , or large values of  $D$ . Many time steps (small  $\Delta t$ ) are required.

## 4.2 Backward Difference

We can improve our techniques for solving Equation (4.1) by using a backward Taylor series for the time derivative. The analog looks like this.

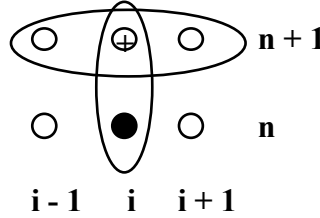


Figure 4.4 Backward Difference - Interior Points

Our substituted equation becomes

$$\frac{C_{i,n+1} - C_{i,n}}{\Delta t} = D \left[ \frac{C_{i+1,n+1} - 2C_{i,n+1} + C_{i-1,n+1}}{\Delta^2 x} \right] - V \left[ \frac{C_{i+1,n+1} - C_{i-1,n+1}}{2\Delta x} \right] - KC_{i,n+1} \quad (4.9)$$

We note that we have three unknowns ( $u_{i+1,n+1}$ ,  $u_{i,n+1}$ ,  $u_{i-1,n+1}$ ) in this equation. Collecting terms we have

$$\begin{aligned} & \left( \frac{-D}{\Delta^2 x} - \frac{V}{2\Delta x} \right) C_{i-1,n+1} \\ & \left( \frac{1}{\Delta t} + \frac{2D}{\Delta^2 x} + K \right) C_{i,n+1} = \frac{1}{\Delta t} C_{i,n} \\ & \left( \frac{-D}{\Delta^2 x} + \frac{V}{2\Delta x} \right) C_{i+1,n+1} \end{aligned} \quad (4.10)$$

We employ a canonical form as follows:

$$a_i C_{i-1,n+1} + b_i C_{i,n+1} + c_i C_{i+1,n+1} = d_i \quad (4.11)$$

where  $a_i$ ,  $b_i$ , and  $c_i$  are the coefficients of the unknowns in Equation (4.10), and  $d_i$  is the sum of all terms on the right hand side of Equation (4.10). In general  $d_i$  will be comprised of terms involving  $C_{i+1,n}$ ,  $C_{i-1,n}$ ,  $C_{i,n}$  and source/sink terms. We will use this canonical form through the rest of this course.

A solvable matrix form of these equations for  $iR = 5$  is:

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 \\ 0 & 0 & a_4 & b_4 & c_4 \\ 0 & 0 & 0 & a_5 & b_5 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix}_{n+1} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{bmatrix}_n \quad (4.12)$$

Note that there are five equations and five unknowns. The terms  $a_1$  and  $c_5$  are missing. It is always necessary to remove these two coefficients using the two boundary equations; otherwise no unique solution to the equations will exist (e.g., five equations with seven unknowns).

To remove these two terms from Equation (4.10) we substitute our boundary conditions as follows:

**Inlet**

$$\begin{aligned} \left( \frac{1}{\Delta t} + \frac{2D}{\Delta^2 x} + K \right) C_{i,n+1} &= \frac{1}{\Delta t} C_{i,n} + \left( \frac{D}{\Delta^2 x} + \frac{V}{2\Delta x} \right) C_{i-1,n+1} \\ \left( \frac{-D}{\Delta^2 x} + \frac{V}{2\Delta x} \right) C_{i+1,n+1} & \end{aligned} \quad (4.13)$$

where  $C_{i-1,n+1}$  is a known boundary point.

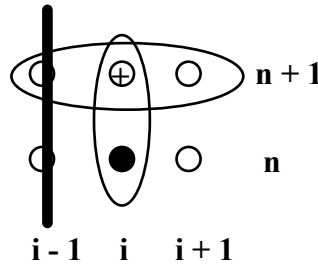


Figure 4.5 Backward Difference - Inlet Boundary

**Exit**

$$\begin{aligned} \left( \frac{-2D}{\Delta^2 x} \right) C_{i-1,n+1} \\ \left( \frac{1}{\Delta t} + \frac{2D}{\Delta^2 x} + K \right) C_{i,n+1} &= \frac{1}{\Delta t} C_{i,n} \end{aligned} \quad (4.14)$$

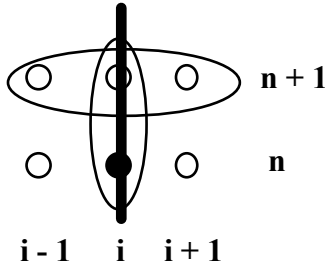


Figure 4.6 Backward Difference - Exit Boundary (zero flux)

Note that  $C_{i-1,n+1} = C_{i+1,n+1}$  and the terms resulting from the first spatial derivative vanish as they did in Equation (4.7).

Equation (4.12) is a tridiagonal matrix. A system of equations described by a tridiagonal matrix is very easy to solve if one uses the specialized Gaussian Elimination technique called the Thomas Algorithm. Appendix A shows the Thomas Algorithm.

### 4.3 Crank-Nicolson

The backward difference approach can be improved by using a centered, second-order correct analog for the time derivative. The method is frequently called Crank-Nicolson, after its developers.

The analog appears as follows:

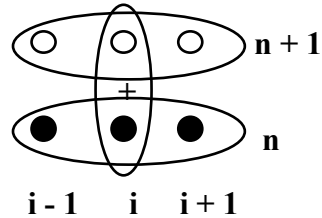


Figure 4.7 Crank-Nicolson -Interior Points

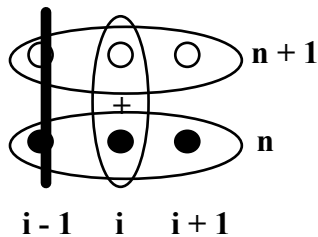


Figure 4.8 Crank-Nicolson - Entrance Boundary

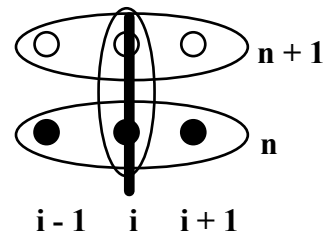


Figure 4.9 Crank-Nicolson - Exit Boundary

The “+” sign represents the center of all analogs. The spatial analogs are averaged over the two time levels. After substituting the analogs into Equation (4.1) we obtain:

$$\begin{aligned}
 \left( \frac{-D}{2\Delta^2x} - \frac{V}{4\Delta x} \right) C_{i-1,n+1} &= \left( \frac{D}{2\Delta^2x} + \frac{V}{4\Delta x} \right) C_{i-1,n} \\
 \left( \frac{1}{\Delta t} + \frac{D}{\Delta^2x} + \frac{K}{2} \right) C_{i,n+1} &= \left( \frac{1}{\Delta t} - \frac{D}{\Delta^2x} - \frac{K}{2} \right) C_{i,n} \\
 \left( \frac{-D}{2\Delta^2x} + \frac{V}{4\Delta x} \right) C_{i+1,n+1} &= \left( \frac{D}{2\Delta^2x} - \frac{V}{4\Delta x} \right) C_{i+1,n}
 \end{aligned} \tag{4.15}$$

This set of equation forms a tridiagonal matrix just as the backward different set of equations formed a tridiagonal matrix. The boundary conditions are also applied in a similar fashion, except that terms on the right hand side (e.g.,  $C_{i-1,n}$  for the inlet boundary) must also be eliminated or resolved by substitution. Crank-Nicolson, like backward difference, is unconditionally stable.

If we define the parameter  $f$  and  $f'$  (which equals  $1 - f$ ) we can generalize a computer code to solve forward difference, backward difference, Crank-Nicolson, and intermediate values. The equation becomes:

$$\begin{aligned}
 \left( \frac{-fD}{\Delta^2x} - \frac{fV}{2\Delta x} \right) C_{i-1,n+1} &= \left( \frac{f'D}{\Delta^2x} + \frac{f'V}{2\Delta x} \right) C_{i-1,n} \\
 \left( \frac{1}{\Delta t} + \frac{2fD}{\Delta^2x} + fK \right) C_{i,n+1} &= \left( \frac{1}{\Delta t} - \frac{2f'D}{\Delta^2x} - f'K \right) C_{i,n} \\
 \left( \frac{-fD}{\Delta^2x} + \frac{fV}{2\Delta x} \right) C_{i+1,n+1} &= \left( \frac{f'D}{\Delta^2x} - \frac{f'V}{2\Delta x} \right) C_{i+1,n}
 \end{aligned} \tag{4.16}$$

When  $f = 0.5$  Equation (4.16) becomes Crank-Nicolson. When  $f = 0$  it becomes forward difference, and with  $f = 1.0$  it is backward difference.

We reviewed these three methods to get you started on solving PDEs. While you developing your computer codes we will cover several other techniques, which have specialized purposes. Figure 4.10 shows a block diagram of a program that could be either Crank-Nicolson or backward difference.

#### 4.4 Richardson’s Method

Richardson’s method is the “intuitively obvious” method but has unfortunate results; however, it is important to be aware of it and know when it can be used.

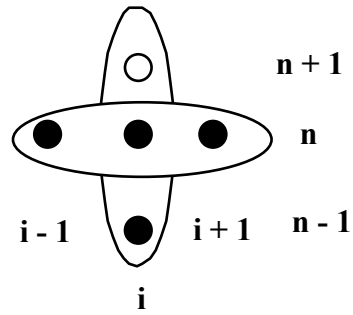


Figure 4.11 Richardson's Method

Consider the equation

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} \quad (4.17)$$

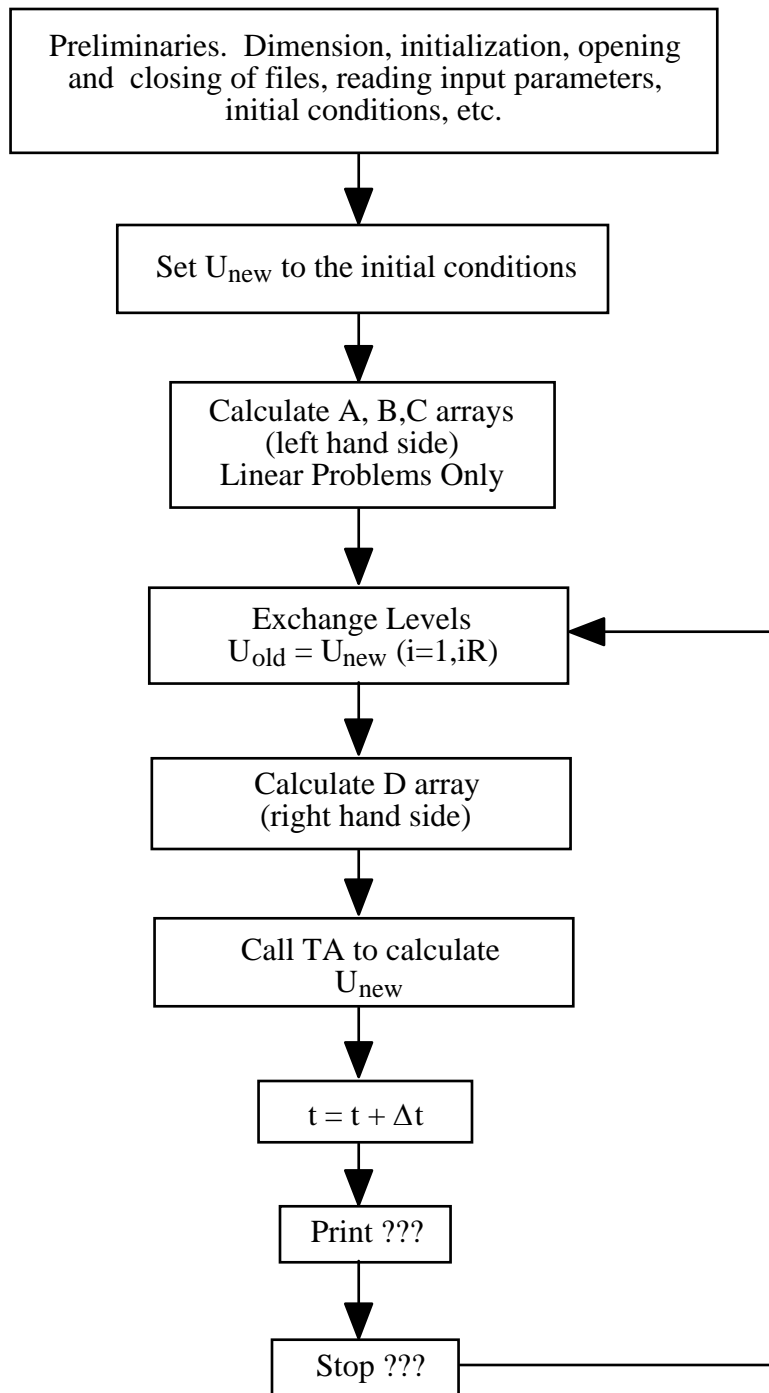


Figure 4.10 Block Diagram of a Crank-Nicolson or Backward Difference Program.

The obvious method is to use the following analog.



$$\frac{u_{i,n+1} - u_{i,n-1}}{2\Delta t} = \frac{D}{\Delta x^2} (u_{i+1,n} - 2u_{i,n} + u_{i-1,n}) \quad (4.18)$$

This method is second-order correct but is unconditionally unstable. Richardson realized the instability only after using it for several years.

The method has very little application, though it can be used for short term problems which require only a few time steps. Weather prediction is one such problem.

The method is *non-self-starting*. It requires two values of  $u_i$  at old time level. We obtain one level from the initial condition. We must use a self-starting method, such as forward difference to obtain the second. Normally we cannot assume values for both initial time levels. If we assume that  $u_{i-1,n} = u_{i,n}$ , we are assuming that  $\frac{\partial u}{\partial t} = 0$  at  $t = n + 1/2$ , which is usually not true.

#### 4.5 Dufort-Frankel

A modification to Richardson's method was proposed by Duford and Frankel. We can obtain a stable condition by averaging to remove the central point from the space derivative in Richardson's method, as follows:

$$u_{i,n} = \frac{1}{2} (u_{i,n+1} + u_{i,n-1}) \quad (4.19)$$

After substituting this in Equation (4.18) our new analog becomes

$$u_{i,n+1} - u_{i,n-1} = \frac{2D\Delta t}{\Delta x^2} (u_{i+1,n} - (u_{i,n+1} + u_{i,n-1}) + u_{i-1,n}) \quad (4.20)$$

We can solve for  $u_{i,n+1}$  as follows:

$$u_{i,n+1} \left( 1 + \frac{2D\Delta t}{\Delta x^2} \right) = \left( 1 - \frac{2D\Delta t}{\Delta x^2} \right) u_{i,n-1} + \frac{2D\Delta t}{\Delta x^2} u_{i-1,n} + \frac{2D\Delta t}{\Delta x^2} u_{i+1,n} \quad (4.21)$$

This method is second-order correct and unconditionally stable. Its properties are inferior to Crank-Nicolson and it is seldom used, unless simplified computational requirements are needed. The problem with this method, like all explicit methods for parabolic PDEs, is that the solution crosses characteristic lines.

#### 4.6 Parabolic equations with small D compared to V and L

A frequently encountered modeling need is to solve parabolic equations with small values of  $D$ . This occurs, for example, in a long, tubular reactor that is almost plug flow but has a significant mixing. The concept of axial mixing, originally described by Taylor, is one of the phenomenon that creates dispersion along the traveling velocity front.

It can be shown that Crank-Nicolson produces oscillatory solutions when the following when the following condition:

$$\Delta x > \frac{2D}{VL} \quad (4.22)$$

or 
$$\Delta x > \frac{2}{Pe} \quad (4.23)$$

This can be a severe constraint and can control execution time. Rather than simply reducing  $\Delta x$ , alternative methods can be used. These methods have sometimes been called “upwind” methods.

The following sections describe several methods to solve these problems without suffering the time penalty associated with equation 4.22.

#### 4.7 Characteristics Averaging (CA)

Crank-Nicolson can be modified to solve the near-hyperbolic problem by diagonally averaging the terms involving the first spatial derivatives. We use a second-order analog written over  $\frac{\Delta x}{2}$ , as follows:

$$\left. \frac{\partial u}{\partial x} \right|_{i+1/2} = \frac{u_{i+1} - u_i}{\Delta x} \quad (4.24)$$

We substitute this analog into Equation (4.1), with the other second-order spatial and time derivatives as follows:

$$\begin{aligned} u_{i+1,n+1} \left( \frac{-D}{2\Delta^2 x} \right) &= u_{i,n} \left( \frac{1}{2\Delta t} - \frac{V}{2\Delta x} + \frac{D}{\Delta^2 x} - \frac{K}{4} \right) \\ u_{i,n+1} \left( \frac{1}{2\Delta t} + \frac{V}{2\Delta x} + \frac{D}{\Delta^2 x} + \frac{K}{4} \right) &= u_{i-1,n} \left( \frac{1}{2\Delta t} + \frac{V}{2\Delta x} - \frac{D}{\Delta^2 x} - \frac{K}{4} \right) \\ u_{i-1,n+1} \left( \frac{1}{2\Delta t} - \frac{V}{2\Delta x} - \frac{D}{\Delta^2 x} + \frac{K}{4} \right) &= u_{i-2,n} \left( \frac{+D}{2\Delta^2 x} \right) \end{aligned} \quad (4.25)$$

To include the reaction term we used a four point averaging scheme. This technique can solve the range of parabolic to hyperbolic conditions if the constraint  $V = \frac{\Delta x}{\Delta t}$  is followed. By requiring  $\frac{\Delta x}{\Delta t} = V$ , many of the terms in Equation (4.25) vanish, and a two point diagonal average for the reaction term should be used. Equation (4.25) becomes

$$\begin{aligned}
 u_{i+1,n+1} \left( \frac{-D}{2\Delta^2 x} \right) &= u_{i,n} \left( \frac{D}{2\Delta^2 x} \right) \\
 u_{i,n+1} \left( \frac{1}{\Delta t} + \frac{D}{\Delta^2 x} + \frac{K}{2} \right) &= u_{i-1,n} \left( \frac{1}{\Delta t} - \frac{D}{\Delta^2 x} - \frac{K}{2} \right) \\
 u_{i-1,n+1} \left( \frac{-D}{2\Delta^2 x} \right) &= u_{i-2,n} \left( \frac{D}{2\Delta^2 x} \right)
 \end{aligned}
 \tag{4.26}$$

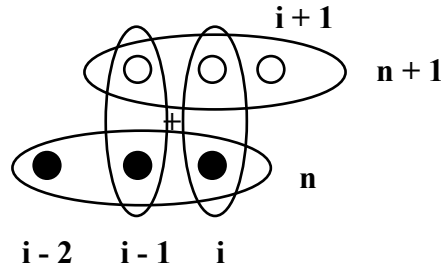


Figure 4.12 Characteristics Averaging - Interior Points

This solution technique is quite useful and can be generalized like Crank-Nicolson. It can be used for a wide variety of Peclet numbers. We shall later see that this solution technique, with  $V = \frac{\Delta x}{\Delta t}$  is always parallel to the characteristic lines. It is similar to a technique called Keller-Box.

The exit boundary condition can be conveniently handled similarly to Crank-Nicolson. The entrance boundary condition is more difficult. If the boundary is located at  $i - 1$ , then both  $C_{i-1,n}$  and  $C_{i-2,n}$  must be known. This condition is similar to non self-starting analogs with respect to time, such as Richardson's method.

If one assumes  $C_{i-1} = C_{i-2}$ , a large error results. It is the same as assuming that  $\frac{\partial C}{\partial x}$  is zero at  $i - 1/2$ , which is certainly not true. No entirely acceptable procedure exists for the entrance boundary condition. One can write a Crank-Nicolson or Backward Difference analog for the first analog and obtain acceptable results if the spatial and time increments are carefully selected.

The CA technique works very well when  $V$  is constant. It is a simple matter to set  $V = \frac{\Delta x}{\Delta t}$  and a non-oscillatory solution is obtained. Problems arise when  $V$  is not constant. In this case, the locations of the grid points must be calculated before the solution is calculated. In cases where  $V$  must be estimated or is a function of the solution, the grid points cannot be exactly located along the trajectories defined by  $V$  and numerical dispersion occurs. The Method of Characteristics is a better way of handling the changing locations of the grid points, but is only useful for truly hyperbolic equations ( $D=0$ ).

#### 4.8 State Variable Formulation

Weaver et al (19xx) introduced a technique which is actually a clever use of the state variable concept from process control theory. The procedure is to reduce the second-order PDE to two first-order PDEs. If we start with equation 4.1 and substitute as follows:

$$\frac{\partial u}{\partial X} = Z \tag{4.27}$$

$$\frac{1}{Pe} \frac{\partial Z}{\partial X} - Z = \frac{\partial u}{\partial t} \tag{4.28}$$

For this formulation, the Centered Difference technique (introduced in the chapter on hyperbolic equations) can be used. After the terms are collected a penta-diagonal matrix is created, which can be solved using the pentadiagonal algorithm or the bi-tri algorithm. (see the Appendix). Weaver et al report that the method is broadly applicable, spanning a wide range of  $Pe$ . The method has an important condition related to the boundary conditions, which can be implemented exactly, unlike CA or in the following cases.

#### 4.9 Non-central analogs

A simple way of overcoming the oscillatory behavior is to use non central analogs. For example, if Centered Difference analogs are combined with the second derivative analogs as shown in Figure 4.13

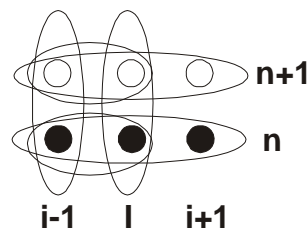


Figure 4.13 Non-central analogs

This formulation is not second-order correct but can work. Handling the boundary conditions requires some creativity.

Another alternative, proposed by Price, Varga and Warrren (19xx) uses a second-order correct analog for the first spatial derivative as follows:

$$\left. \frac{\partial u}{\partial x} \right|_{i,n+1/2} = \frac{3u_i - 4u_{i-1} + u_{i-2}}{2\Delta x} \quad (4.29)$$

This analog can be written with the second-order correct second spatial derivative analog, centered at  $i$  and  $n+1/2$  to create penta diagonal form (the new time level includes for values of  $u$ , from  $i-2$  to  $i+1$ ).

#### 4.10 Summary

Crank-Nicolson and Characteristics-Averaging are the methods of choice for solving one dimensional parabolic PDE's with split boundary conditions. They are computationally very fast. Some investigators prefer finite element techniques, but their preferences are made on the basis of ease of use, or super computer compatibility. For the simple equations described herein with regular geometries, the finite difference techniques are superior.

## 5. PARABOLIC EQUATIONS IN TWO AND THREE DIMENSIONS

Our one dimensional transport equation in one dimension frequently occurs in two dimensions. We can extend some of our one dimensional concepts to two dimensions, although there are a number of complications. Consider the following equations:

$$\frac{\partial u}{\partial t} = D_x \frac{\partial^2 u}{\partial x^2} + D_y \frac{\partial^2 u}{\partial y^2} - V_x \frac{\partial u}{\partial x} - V_y \frac{\partial u}{\partial y} - Ku \quad (5.1)$$

### 5.1 Forward Difference in Two Dimensions

First we extend Forward Difference to two dimensions, as follows:

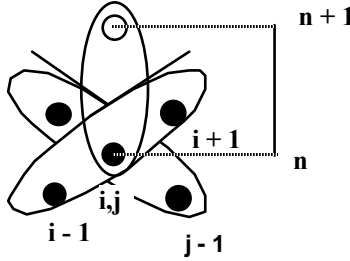


Figure 5.1. Forward Difference Extended to Two Dimensions

$$\begin{aligned} \frac{u_{i,j,n+1} - u_{i,j,n}}{\Delta t} = & D_x \left[ \frac{u_{i+1,j,n} - 2u_{i,j,n} + u_{i-1,j,n}}{\Delta^2 x} \right] + D_y \left[ \frac{u_{i,j+1,n} - 2u_{i,j,n} + u_{i,j-1,n}}{\Delta^2 y} \right] \\ & - V_x \left[ \frac{u_{i+1,j,n} - u_{i-1,j,n}}{2\Delta x} \right] - V_y \left[ \frac{u_{i,j+1,n} - u_{i,j-1,n}}{2\Delta y} \right] - Ku_{i,j,n} \end{aligned} \quad (5.2)$$

Solving for  $u_{i,j,n+1}$  and collecting terms we obtain

$$u_{i,j,n+1} = + u_{i,j,n} \left[ 1 - \frac{2D_x \Delta t}{\Delta^2 x} - \frac{2D_y \Delta t}{2\Delta y} - K\Delta t \right] + u_{i-1,j,n} \left[ \frac{D_x \Delta t}{\Delta^2 x} + \frac{V_x \Delta t}{2\Delta x} \right]$$

$$\begin{aligned}
& + u_{i+1,j,n} \left[ \frac{D_x \Delta t}{\Delta^2 x} - \frac{V_x \Delta t}{2\Delta x} \right] \\
& + u_{i,j-1,n} \left[ \frac{D_y \Delta t}{\Delta^2 y} + \frac{V_y \Delta t}{2\Delta y} \right] \\
& + u_{i,j+1,n} \left[ \frac{D_y \Delta t}{\Delta^2 y} - \frac{V_y \Delta t}{2\Delta y} \right]
\end{aligned} \tag{5.3}$$

for

$$1 \leq i \leq iR \qquad 1 \leq j \leq jR$$

We see our equation is explicit. The equation is much more forward than backward, and has severe stability constraints. Boundary conditions are included as before with Forward Difference. However, we now have four sets of boundary equations.

Consider the case when  $\frac{\partial u}{\partial y} = 0$  at  $j = jR$ .

This boundary condition exists for all  $i$ , as shown below.

The analogs become

$$\begin{aligned}
& u_{i-1,jR,n} \left[ \frac{D_x \Delta t}{\Delta^2 x} + \frac{V_x \Delta t}{2\Delta x} \right] \\
& u_{i,jR,n} \left[ 1 - \frac{2D_x \Delta t}{\Delta^2 x} - \frac{2D_y \Delta t}{\Delta^2 x} - K\Delta t \right] \\
u_{i,jR,n+1} & = u_{i+1,jR,n} \left[ \frac{D_x \Delta t}{\Delta^2 x} - \frac{V_x \Delta t}{2\Delta x} \right] \\
& u_{i,jR-1,n} \left[ \frac{2D_y \Delta t}{\Delta^2 y} \right]
\end{aligned} \tag{5.4}$$

For the special case at the four corners, we have two boundary conditions to implement, as when

$$\begin{aligned}
i = 1, j = jR & \quad \text{and} \quad u_{i,jR,n} = u_o(t) \\
u_{1,jR,n+1} & = u_o(t + \Delta t)
\end{aligned}$$

or in the case when  $\frac{\partial u}{\partial x} = 0$ , and  $\frac{\partial u}{\partial y} = 0$

$$\begin{aligned}
\text{at } i = iR, \quad j = jR \\
u_{iR,jR,n+1} & = u_{iR-1,jR,n} \left[ \frac{2D_x \Delta t}{\Delta^2 x} \right]
\end{aligned}$$

$$\begin{aligned}
& + u_{iR,jR,n} \left[ 1 - \frac{2D_x \Delta t}{\Delta^2_x} - \frac{2D_y \Delta t}{\Delta^2_y} - K \Delta t \right] \\
& + u_{iR,jR-1,n} \left[ \frac{2D_y \Delta t}{\Delta^2_y} \right]
\end{aligned} \tag{5.5}$$

Conceptually the boundary conditions are the same as for one dimension, but must be implemented along all four sides. The number of boundary analogs becomes  $2jR + 2iR - 2$ .

The stability constraints on the forward difference make it a poor choice for most problems. It is conditionally stable subject to the following constraint.

$$D_x \frac{\Delta t}{\Delta^2_x} + D_y \frac{\Delta t}{\Delta^2_y} \leq 1/2 \tag{5.6}$$

## 5.2 Backward Difference and Crank-Nicolson in Two Dimensions

Now let's extend our one dimensional backward difference technique to two dimensions. We use the following analogs:

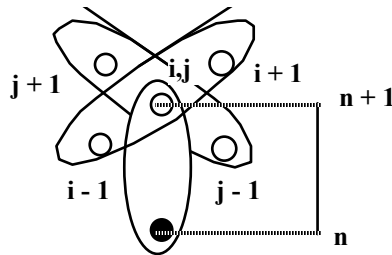


Figure 5.2 Backward Difference Extended to Two Dimensions

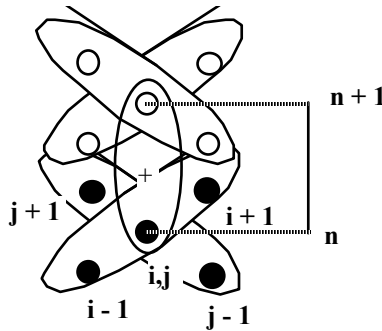


Figure 5.3 Crank-Nicolson Extended to Two Spatial Dimensions



We obtain

$$\begin{aligned}
 & u_{i-1,j,n+1} \left[ -\frac{D_x \Delta t}{\Delta^2 x} - \frac{V_x \Delta t}{2 \Delta x} \right] \\
 & u_{i,j,n+1} \left[ 1 + \frac{2D_x \Delta t}{\Delta^2 x} + \frac{2D_y \Delta t}{\Delta^2 y} + K \Delta t \right] = u_{i,j,n} \\
 & u_{i+1,j,n+1} \left[ -\frac{D_x \Delta t}{\Delta^2 x} + \frac{V_x \Delta t}{2 \Delta x} \right] \\
 & u_{i,j-1,n+1} \left[ -\frac{D_y \Delta t}{\Delta^2 y} - \frac{V_y \Delta t}{2 \Delta y} \right] \\
 & u_{i,j+1,n+1} \left[ -\frac{D_y \Delta t}{\Delta^2 y} + \frac{V_y \Delta t}{2 \Delta y} \right]
 \end{aligned} \tag{5.7}$$

We see that we have an implicit solution with five unknowns. At first glance it seems that we can use the band algorithm.

If we use a canonical form, with the  $i-1, i, i+1, j-1, j+1$  terms corresponding to  $a, b, c, e,$  and  $f$  coefficients, we will obtain the following matrix. Only one equation is shown from an arbitrary size matrix. The dots represent other terms or zeros.

$$\begin{bmatrix}
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & a_i & b_i & c_i & e_i & f_i & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
 \end{bmatrix}
 \begin{bmatrix}
 \cdot \\
 \cdot \\
 u_{i-1,j} \\
 u_{i,j} \\
 u_{i+1,j} \\
 u_{i,j-1} \\
 u_{i,j+1} \\
 \cdot
 \end{bmatrix}
 =
 \begin{bmatrix}
 \cdot \\
 \cdot \\
 d_i \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot
 \end{bmatrix} \tag{5.8}$$

Note that the  $u_{i-1,j}, u_{i,j}$  and  $u_{i+1,j}$  terms are arranged in the  $u$  vector in ascending order of  $i$ . If the next equation is written (i.e., incrementing  $i$  to the next grid point) in the same way as the equation shown, an error will result. The  $a_i, b_i$  and  $c_i$  terms will be multiplied by  $u_{i,j}, u_{i+1,j}$  and  $u_{i,j+1}$ .

It is not possible to order the terms in Equation (5.8) to produce a pentadiagonal matrix. Correct ordering results in either a sparse matrix or a matrix with 2 diagonals separated by zeros from 3 main diagonals. In either case, matrix inversion is quite time consuming and not possible with existing computers for large problems.

The later case, with 2 diagonals separated from the 3 main diagonals, is easier to solve, and there are proprietary super computer routine that can invert this matrix, for well behaved situation.

We can extend Crank-Nicolson to two dimensions as we tried with backward difference, but we would still have a sparse matrix.

### 5.3 Alternating Direction Implicit in Two Dimensions

A technique which combines the properties of backward difference and forward difference exists, which is unconditionally stable, and procedures an easily solved tridiagonal matrix. The method is called Alternating Direction Implicit (ADI) and was developed by Peaceman and Rachford (1955). The name arises because we write hybrid analogs that are implicit in one direction and explicit in the opposite direction. We alternate the explicit and implicit directions at each new time level. Usually we divide each time step into two parts, and integrate over each direction at  $\Delta t/2$ . The analogs appear as follows:

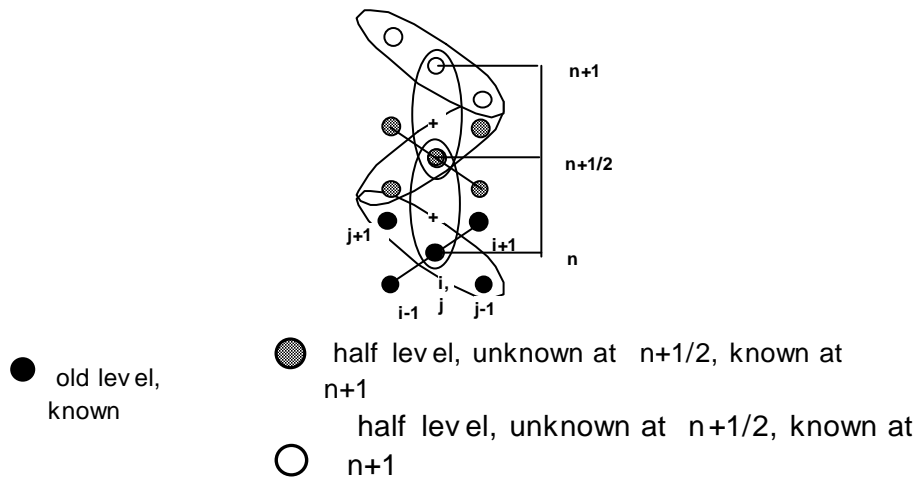


Figure 5.4 ADI Analogs [Implicit X(i), then Implicit Y(j)]

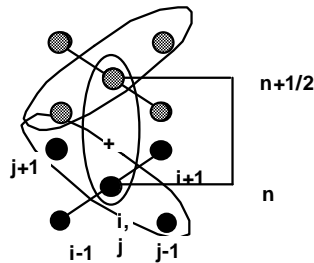


Figure 5.5 ADI Analogs [Explicit Y(j), Implicit X(i)]

The analogs are developed as follows, using  $\Delta t/2$  as the basic time step.

**implicit x, explicit y**

$$\begin{aligned}
 u_{i-1,j,n+1/2} \left[ -\frac{D_x \Delta t}{2\Delta^2_x} - \frac{V_x \Delta t}{4\Delta x} \right] &= u_{i,j-1,n} \left[ \frac{D_y \Delta t}{2\Delta^2_y} + \frac{V_y \Delta t}{4\Delta y} \right] \\
 u_{i,j,n+1/2} \left[ 1 + \frac{D_x \Delta t}{\Delta^2_x} + \frac{K\Delta t}{4} \right] &= u_{i,j,n} \left[ 1 - \frac{D_y \Delta t}{\Delta^2_y} - \frac{K\Delta t}{4} \right] \\
 u_{i+1,j,n+1/2} \left[ -\frac{D_x \Delta t}{2\Delta^2_x} + \frac{V_x \Delta t}{4\Delta x} \right] &= u_{i,j+1,n} \left[ \frac{D_y \Delta t}{2\Delta^2_y} - \frac{V_y \Delta t}{4\Delta y} \right]
 \end{aligned} \tag{5.10}$$

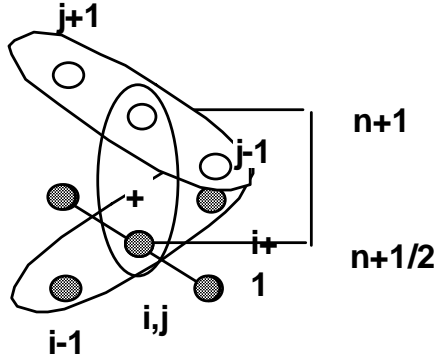


Figure 5.6 ADI Analog [Implicit Y (j), Explicit X (i)]

**implicit y, explicit x**

$$\begin{aligned}
 u_{i,j-1,n+1} \left[ -\frac{D_y \Delta t}{2\Delta^2_y} - \frac{V_y \Delta t}{4\Delta y} \right] &= u_{i-1,j,n+1/2} \left[ \frac{D_x \Delta t}{2\Delta^2_x} + \frac{V_x \Delta t}{4\Delta x} \right] \\
 u_{i,j,n+1} \left[ 1 + \frac{D_y \Delta t}{\Delta^2_y} + \frac{K\Delta t}{4} \right] &= u_{i,j,n+1/2} \left[ 1 - \frac{D_x \Delta t}{\Delta^2_x} - \frac{K\Delta t}{4} \right] \\
 u_{i,j+1,n+1} \left[ -\frac{D_y \Delta t}{2\Delta^2_y} + \frac{V_y \Delta t}{4\Delta y} \right] &= u_{i+1,j,n+1/2} \left[ \frac{D_x \Delta t}{2\Delta^2_x} - \frac{V_x \Delta t}{4\Delta x} \right]
 \end{aligned} \tag{5.10}$$

This procedure works well and is very widely used to solve 2-dimensional parabolic and elliptic PDE's with regular boundaries. The procedure requires that we divide our approach into two types of analogs. We write our program as follows:

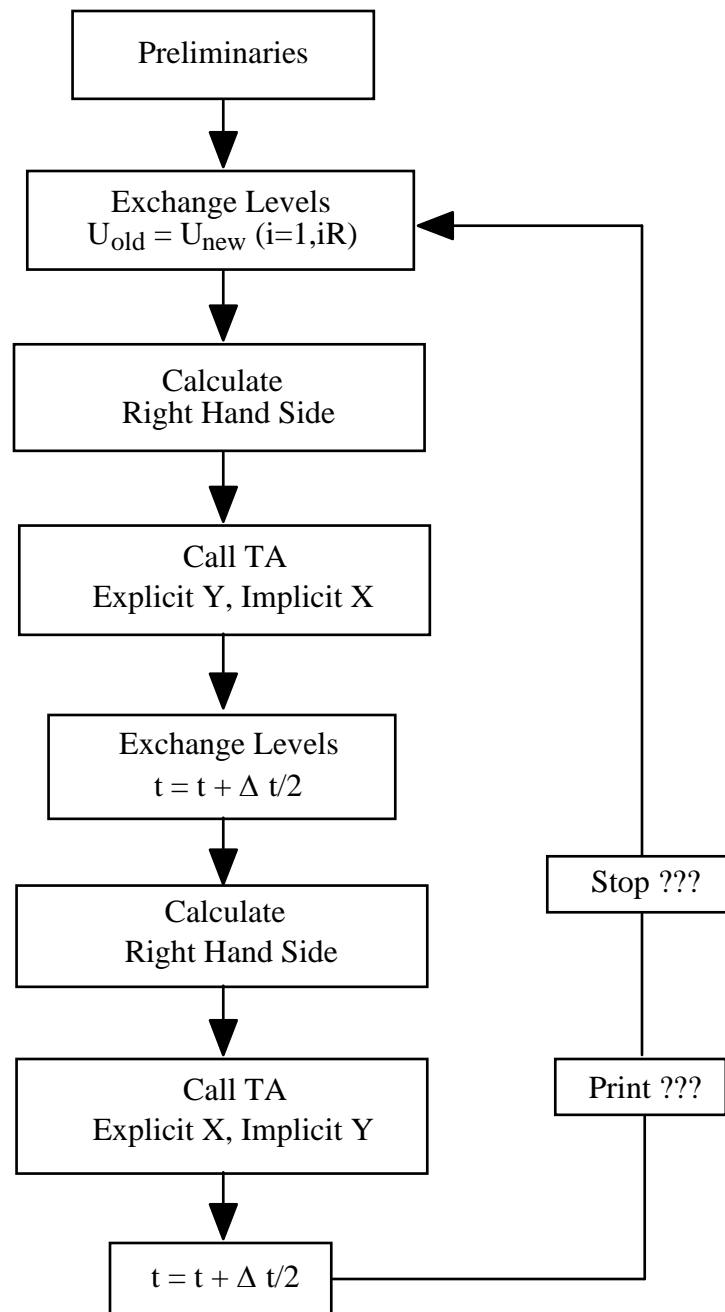


Figure 5.7

We can improve our implicit analogs in ADI by averaging them at the old time level. The analogs for implicit x, explicit y would be written as follows:

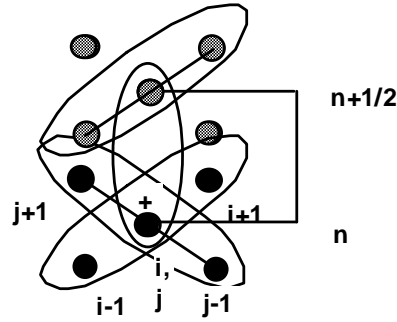


Figure 5.8 Modified ADI Analogs [Explicit Y(j), Implicit X(i)]

$$\begin{aligned}
 u_{i-1,j,n+1/2} \left[ \frac{-D_x \Delta t}{4\Delta^2_x} - \frac{V_x \Delta t}{8\Delta x} \right] &= u_{i,j-1,n} \left[ \frac{D_y \Delta t}{2\Delta^2_y} + \frac{V_y \Delta t}{4\Delta y} \right] \\
 u_{i,j,n+1/2} \left[ 1 + \frac{D_x \Delta t}{2\Delta^2_x} + \frac{K \Delta t}{8} \right] &= u_{i,j+1,n} \left[ \frac{D_y \Delta t}{2\Delta^2_y} - \frac{V_y \Delta t}{4\Delta y} \right] \\
 u_{i+1,j,n+1/2} \left[ \frac{-D_x \Delta t}{4\Delta^2_x} + \frac{V_x \Delta t}{8\Delta x} \right] &= u_{i,j,n} \left[ 1 - \frac{D_y \Delta t}{\Delta^2_y} - \frac{D_x \Delta t}{2\Delta^2_x} - \frac{K \Delta t}{4} \right] \\
 &= u_{i-1,j,n} \left[ \frac{D_x \Delta t}{4\Delta^2_x} + \frac{V_x \Delta t}{8\Delta x} \right] \\
 &= u_{i+1,j,n} \left[ \frac{D_x \Delta t}{4\Delta^2_x} - \frac{V_x \Delta t}{8\Delta x} \right]
 \end{aligned} \tag{5.11}$$

We would develop similar analogs for the opposite direction.

At first this appears to be an improvement over ADI, since the implicit analogs are second-order correct. Unfortunately the procedure is conditionally stable, and is generally inferior to ADI. One might choose to use it for a stiff problem (very large  $K$ ) where the required time step is so small that stability is assured.

#### 5.4 Improved Notation

To see that this method ADI is more forward than backward, we can develop a shorthand. We use the one dimensional diffusion equation for starters.

Let

$$\delta_x u_{i,n} = \frac{u_{i+1,n} - u_{i-1,n}}{2\Delta x} \tag{5.12}$$

Similarly

$$\delta_t u_{i,n} = \frac{u_{i,n+1} - u_{i,n}}{\Delta t} \quad (5.13)$$

and finally

$$\delta_x^2 u_{i,n} = \frac{u_{i+1,n} - 2u_{i,n} + u_{i-1,n}}{\Delta x^2} \quad (5.14)$$

We can use this notation to “shorthand” our previous work. For example, we can describe solution techniques for Equation (4.1) as follows:

*Forward Difference*

$$\delta_t u_{i,n} = D \delta_x^2 u_{i,n} - V \delta_x u_{i,n} - K u_{i,n} \quad (5.15)$$

*Backward Difference*

$$\delta_t u_{i,n} = D \delta_x^2 u_{i,n+1} - V \delta_x u_{i,n+1} - K u_{i,n+1} \quad (5.16)$$

*Crank-Nicolson*

$$\delta_t u_{i,n} = \frac{D}{2} \delta_x^2 u_{i,n} + \frac{D}{2} \delta_x^2 u_{i,n+1} - \frac{V}{2} \delta_x u_{i,n} - \frac{V}{2} \delta_x u_{i,n+1} - \frac{K}{2} u_{i,n} - \frac{K}{2} u_{i,n+1} \quad (5.17)$$

or

$$\delta_t u_{i,n} = D \delta_x^2 u_{i,n+1/2} - V \delta_x u_{i,n+1/2} - K u_{i,n+1/2} \quad (5.18)$$

We can also write Crank-Nicolson as a two step procedure. We drop the advective and reaction terms for simplicity.

$$\delta_t u_{i,n} = \delta_x^2 u_{i,n} \quad (5.19)$$

$$\delta_t u_{i,n} = \delta_x^2 u_{i,n+1} \quad (5.20)$$

Summing

$$2 \delta_t u_{i,n} = \delta_x^2 u_{i,n} + \delta_x^2 u_{i,n+1} \quad (5.21)$$

or

$$\delta_t u_{i,n} = 1/2 \delta_x^2 u_{i,n} + 1/2 \delta_x^2 u_{i,n+1} \quad (5.22)$$

It is easy to see that Crank-Nicolson is the alternating application of backward and forward differences.

Now lets apply our shorthand to ADI. We first need to define derivatives in two spatial directions

$$\delta_t u_{i,j,n} = \frac{u_{i,j,n+1} - u_{i,j,n}}{\Delta t} \quad (5.23)$$

$$\delta_x^2 u_{i,j,n} = \frac{u_{i+1,j,n} - 2u_{i,j,n} + u_{i-1,j,n}}{\Delta^2_x} \quad (5.24)$$

$$\delta_y^2 u_{i,j,n} = \frac{u_{i,j+1,n} - 2u_{i,j,n} + u_{i,j-1,n}}{\Delta^2_y} \quad (5.25)$$

Next we can write the two step ADI procedure as follows:

$$\delta_t u_{i,j,n} = \delta_x^2 u_{i,j,n+1/2} + \delta_y^2 u_{i,j,n} \quad (5.26)$$

$$\delta_t u_{i,j,n+1/2} = \delta_x^2 u_{i,j,n+1/2} + \delta_y^2 u_{i,j,n+1} \quad (5.27)$$

The location of the time derivatives can be determined by averaging the time indices of the spatial derivatives on the right hand side, e.g.,  $(n)$  and  $(n + 1/2)$  averages to  $n + 1/4$  for the  $x$  direction, and  $(n + 1/2)$  and  $(n + 1)$  averages to  $n + 3/4$  for the  $y$  direction. Averaging the two analogs  $(1/4$  and  $3/4)$  gives  $1/2$ . We see that the time derivatives are located at exactly the half-way point between the old and new time levels. ADI is analogous to Crank-Nicolson in this case.

Now let us examine the modified ADI which we noted was conditionally stable.

$$\delta_t u_{i,j,n} = \delta_x^2 u_{i,j,n+1/4} + \delta_y^2 u_{i,j,n} \quad (5.28)$$

$$\delta_t u_{i,j,n+1/2} = \delta_x^2 u_{i,j,n+1/2} + \delta_y^2 u_{i,j,n+3/4} \quad (5.29)$$

We average to find the  $x$  derivative at  $3/8$ , and the  $y$  derivative at  $3/8$ . This indicates that the procedure is more forward than backward, and this causes conditional stability.

## 5.5 Three Dimensional Problems

We now apply our techniques to three dimensional problems. Consider three dimensional diffusion equations. Extrapolation of ADI to three dimensions gives the following:

$$\delta_t u_{i,j,k,n}^* = \delta_x^2 u_{i,j,k,n+1/3}^* + \delta_y^2 u_{i,j,k,n}^* + \delta_z^2 u_{i,j,k,n}^* \quad (5.30)$$

$$\delta_t u_{i,j,k,n+1/3}^{**} = \delta_x^2 u_{i,j,k,n+1/3}^{**} + \delta_y^2 u_{i,j,k,n+2/3}^{**} + \delta_z^2 u_{i,j,k,n}^{**} \quad (5.31)$$

$$\delta_t u_{i,j,k,n+2/3}^{***} = \delta_x^2 u_{i,j,k,n+1/3}^{***} + \delta_y^2 u_{i,j,k,n+2/3}^{***} + \delta_z^2 u_{i,j,k,n+1}^{***} \quad (5.32)$$

The increasing number of asterisks denote implicitly calculated analogs.

Now we average our spatial derivatives and discover that  $\delta_x^2$  is centered at  $1/3$ ,  $\delta_y^2$  is centered at  $4/9$  and  $\delta_z^2$  is centered at  $1/3$ . Therefore our extrapolation of ADI to three

dimensions is more forward than backward, and we should expect conditional stability. In fact this procedure is conditionally stable. We must look for other methods.

Rachford proposed a modified ADI for three dimensions which is unconditionally stable by only first-order correct. We can describe it as follows:

$$\delta_t u_{i,j,k,n+1}^* = \delta_x^2 u_{i,j,k,n+1}^* + \delta_y^2 u_{i,j,k,n} + \delta_z^2 u_{i,j,k,n} \quad (5.33)$$

$$\delta_t u_{i,j,k,n+1}^{**} = \delta_x^2 u_{i,j,k,n+1}^* + \delta_y^2 u_{i,j,k,n+1}^{**} + \delta_z^2 u_{i,j,k,n} \quad (5.34)$$

$$\delta_t u_{i,j,k,n+1}^{***} = \delta_x^2 u_{i,j,k,n+1}^* + \delta_y^2 u_{i,j,k,n+1}^{**} + \delta_z^2 u_{i,j,k,n+1}^{***} \quad (5.35)$$

The technique requires that the new time level,  $n+1$ , be calculated three times. It is calculated implicitly, using the x-direction derivatives in Equation (5.33). It is again calculated implicitly in Equation (5.34), using y-direction derivatives. In Equation (5.34), the results for  $u$  calculated in (5.33) are used as “knowns” in the x-direction derivatives, while the z-direction derivatives are still represented using the known values of  $u$  at the  $n$  time level. In Equation (5.35), the values of  $u$  are calculated a third time and final time, using the z-direction derivatives. The results for  $u$  from Equations (5.33) and (5.34) are used to represent the known values in the x and y directions, respectively. In all three equations the time derivative contains a  $u_{n+1}$  unknown, which is calculated implicitly the unknowns in one spatial derivative.

Only the last values of  $n+1$  are remembered for the next time step. That is to say, only the results of Equation (5.35) are stored and they become the old time level at the next time step. The time derivative is represented by the shift from left to right in the paper, since we cannot draw in four dimensions.

To further improve upon the Peaceman and Rachford three dimensional ADI, Brian (1961) developed a modified set of analogs which are unconditionally stable and 2nd order correct. They are written as follows:

$$\delta_t u_{i,j,k,n+1/2}^* = \delta_x^2 u_{i,j,k,n+1/2}^* + \delta_y^2 u_{i,j,k,n} + \delta_z^2 u_{i,j,k,n} \quad (5.36)$$

$$\delta_t u_{i,j,k,n+1/2}^{**} = \delta_x^2 u_{i,j,k,n+1/2}^* + \delta_y^2 u_{i,j,k,n+1/2}^{**} + \delta_z^2 u_{i,j,k,n} \quad (5.37)$$

$$\delta_t u_{i,j,k,n+1/2}^{***} = \delta_x^2 u_{i,j,k,n+1/2}^* + \delta_y^2 u_{i,j,k,n+1/2}^{**} + \delta_z^2 u_{i,j,k,n+1/2}^{***} \quad (5.38)$$

$$\delta_t u_{i,j,k,n+1} = \delta_x^2 u_{i,j,k,n+1/2}^* + \delta_y^2 u_{i,j,k,n+1/2}^{**} + \delta_z^2 u_{i,j,k,n+1/2}^{***} \quad (5.39)$$

The values of  $u$  are calculated three times just as in the Rachford ADI. Equations (5.36 through 5.38) implicitly calculate  $u$  using a different set of spatial derivatives. The final value of  $u$  is calculated in Equation (5.39). which is completely explicit. In practice Equation (5.39) can be combined into Equation (5.38) to save computer time, and is shown here for clarity.



The computational requirements for multi-dimensional problems increase exponentially. It is easy to see that a one-dimensional problem required one use the Thomas Algorithm to solve for the new time level. For ADI with an even number of grid points in each direction, the Thomas algorithm is called  $iR$  times for each new half level, or  $2jR$  times to advance one time level. For three-dimensional problems with an even number of grid points, the Thomas algorithm is called  $3iR^2$  times for each new time level. To see this impact on computer time, a 100 grid point problem, which is a frequently used number of grid points for a one-dimensional problem, would require 1, 200, and 30,000 uses of the Thomas algorithm for one, two, and three-dimensional problems, respectively. We can see that the computational requirements for three-dimensional transient problems are quite severe, which explains the dearth of three dimensional problem solutions, especially with large numbers of grid points.

## 6. ELLIPTIC EQUATIONS

The most common occurrence of elliptic PDE's for us is steady state diffusion problems. Consider non-steady state diffusion in two dimensions.

$$D_x \frac{\partial^2 u}{\partial x^2} + D_y \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t} \quad (6.1)$$

at steady state  $\frac{\partial u}{\partial t} = 0$

$$D_x \frac{\partial^2 u}{\partial x^2} + D_y \frac{\partial^2 u}{\partial y^2} = 0 \quad (6.2)$$

recalling our classifications

$$a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} = 0 \quad (6.3)$$

so  $A = D_x$ , and  $C = D_y$ , and  $B = 0$

Since  $D_x > 0$  and  $D_y > 0$

$$B^2 - 4AC = -4D_x D_y < 0 \quad (6.5)$$

Therefore this equation is always elliptic for all values of  $D_x$  and  $D_y$ . The problem is parabolic when  $\frac{\partial u}{\partial t} \neq 0$ .

There are a number of possible solution techniques. One method is to use the parabolic method, integrating to steady state. ADI is popular to use. A second method, altogether different from parabolic methods are a class of methods, called relaxation or iteration techniques.

Consider 2nd order correct analogs for  $\frac{\partial^2 u}{\partial y^2}$  and  $\frac{\partial^2 u}{\partial x^2}$  for Equation (6.2) with a source or sink term, b.

$$D_x \left[ \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \right] + D_y \left[ \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} \right] = b \quad (6.5)$$

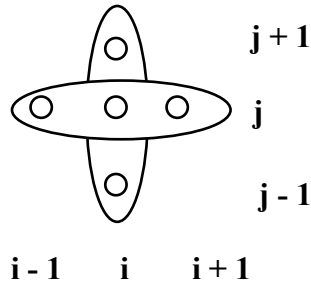


Figure 6.1 Elliptic Analog

for the convenient case of

$$D_x = D_y = 1.0, \text{ and } \Delta x = \Delta y, \text{ we get}$$

$$u_{i,j} - \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) = b_{i,j} \frac{\Delta^2 x}{4} \quad (6.6)$$

Consider the following grid

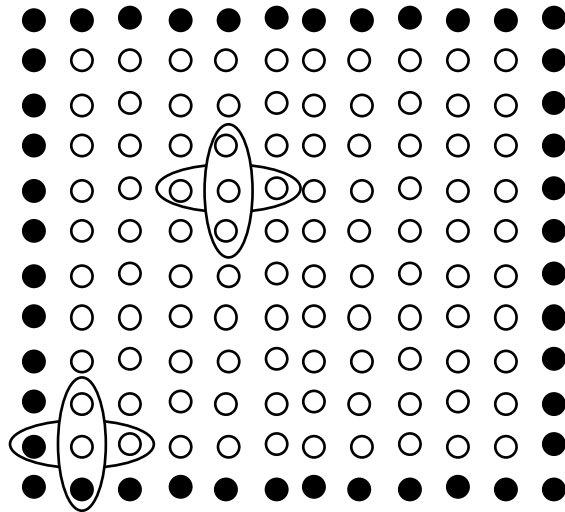


Figure 6.2 Grid of Points Showing Analog Placement (filled circles are known boundary points)

Note that there are no initial values!

We have a problems with five unknowns. As with the direct extrapolation of Backward Difference to two dimensions, we have a set of equations that are not convenient to solve conventional tools for solving simultaneous equations. The analogs lead to a sparse matrix. Consequently we develop iterative methods, whereby we guess for some of the unknowns and use these guesses to solve for other unknowns. By comparing solutions to guesses to

create new guesses we obtain better and better solutions, until we eventually converge. There are five popular methods, often called *relaxation techniques* or *iterative methods*.

1. Jacobi
2. Gauss-Seidel
3. Line Relaxation
4. Successive Over Relaxation (SOR)
5. Successive Line Over Relaxation (SLOR)

We develop techniques for each.

### 6.1 Jacobi and Gauss-Seidel

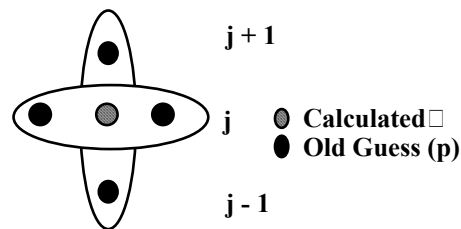


Figure 6.3 Jacobi Analog

Write the equation as

$$u_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) + \frac{\Delta^2 x}{4} b_{i,j} \quad (6.7)$$

We superscript the variables to differentiate between the guess and calculated values, as follows:

$$u_{i,j}^{p+1} = \frac{1}{4}(u_{i+1,j}^p + u_{i-1,j}^p + u_{i,j+1}^p + u_{i,j-1}^p) + \frac{\Delta^2 x}{4} b_{i,j}^{p+1} \quad (6.8)$$

where

- $p$  denotes initial guess or estimate for  $u$  or  $b$
- $p+1$  denotes calculated value for  $u$  or  $b$ .

By iteratively applying the above equation we can eventually produce solutions. We solve for all values of  $u^{p+1}$  at  $i,j$ , and then substitute  $u_{i,j}^{p+1}$  for  $u_{i,j}^p$ .

The problem is that Jacobi is very, very slow to converge. We can improve the Jacobi technique by using a coarse grid early in the problem, then refining it. In general, the finer

the grid, the longer it takes to converge. Now if we pay attention to our grid, we get a little smarter.

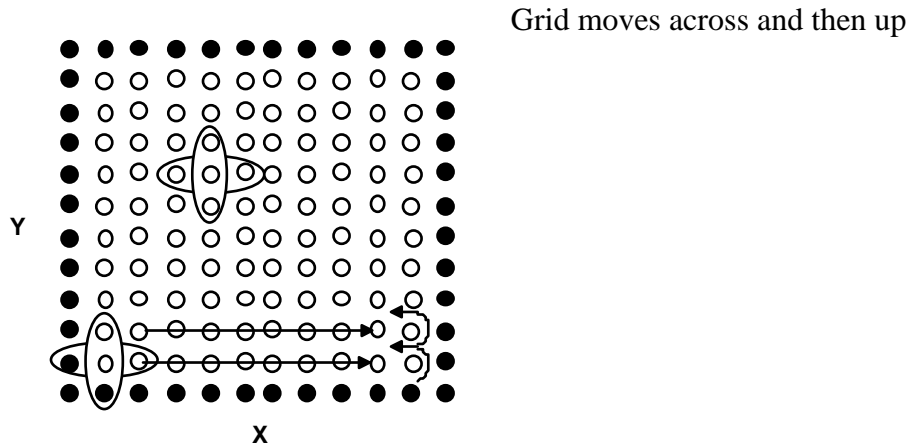


Figure 6.4 Grid of Points Showing Analog Placement (filled circles are known boundary points)

We note that we have  $p+1$  values for other analogs.

We can improve the Jacobi method as follows:

$$u_{i,j}^{p+1} = \frac{1}{4} (u_{i+1,j}^p + u_{i-1,j}^{p+1} + u_{i,j+1}^p + u_{i,j-1}^{p+1}) + \frac{\Delta^2 x}{4} b_{i,j}^p \quad (6.9)$$

Note that we have already calculated  $u_{i-1,j}^{p+1}$  and  $u_{i,j-1}^{p+1}$  when we wish to calculate  $u_{i,j}^{p+1}$ . We will always know two of the terms no matter which way we progress through our grid points. This modification of the Jacobi method is called Gauss-Seidel or successive relaxation. Gauss-Seidel converges twice as fast as Jacobi.

We always choose to use Gauss-Seidel over Jacobi if we are able to do so (usually you can). If Jacobi converges, so will Gauss-Seidel. These two methods are often called Relaxation methods because they compute the data in a successive manner, and “relax” to a solution.

## 6.2 Line Relaxation

We note that for Jacobi we were only using one point at the new iteration level,  $p+1$ . With Gauss-Seidel we used two more new points, for a total of three. What about using four new points?

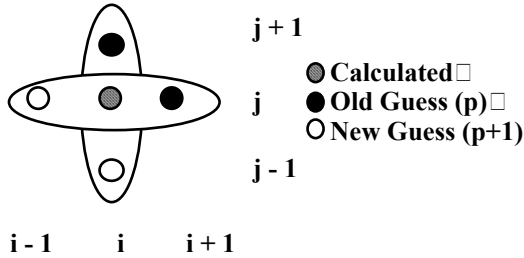


Figure 6.5 Gauss-Seidel Analog

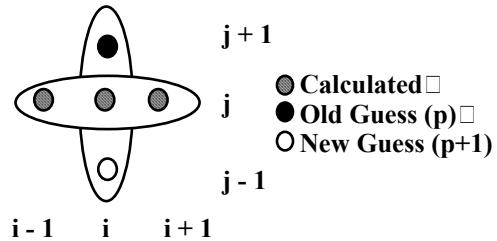


Figure 6.6 Line Relaxation

We see that we no longer have an explicit problem. (For the sake of simplicity we assume that we have no source or sink terms). We no longer can calculate new guess, i.e.,

$$-\frac{1}{4}u_{i+1,j}^{p+1} + u_{i,j}^{p+1} - \frac{1}{4}u_{i-1,j}^{p+1} = \frac{1}{4}u_{i,j-1}^{p+1} + \frac{1}{4}u_{i,j+1}^p \quad (6.10)$$

We have three unknowns. We can obtain  $u_{i-1,j}^{p+1}$  or  $u_{i,j-1}^{p+1}$  (not both!) from previous steps.

Recall that with Jacobi or Gauss-seidel we had two unknowns at the new level, and we eliminated one using the boundary conditions. Now that at the boundary we have at least two unknowns.

$$u_{i,j}^{p+1} \quad u_{i,j+1}^{p+1} \quad \text{and/or} \quad u_{i,j-1}^{p+1} \quad (6.11)$$

So we can never get started!

We need to solve this problem by using an implicit procedure. We arrange the equation as follows:

$$-\frac{1}{4}u_{i-1,j}^{p+1} + u_{i,j}^{p+1} - \frac{1}{4}u_{i+1,j}^{p+1} = \frac{1}{4}(u_{i,j-1}^{p+1} + u_{i,j+1}^p) \quad (6.12)$$

By writing our problem in matrix form we see our problem better.

$$[\text{coefficients}] \begin{bmatrix} u_{i,j}^{p+1} \end{bmatrix} = \begin{bmatrix} u_{i,j-1}^{p+1} + u_{i,j}^p \end{bmatrix} \quad (6.13)$$

$$\begin{bmatrix} 1 & -1/4 & 0 & 0 & 0 \\ -1/4 & 1 & -1/4 & 0 & 0 \\ 0 & -1/4 & 1 & -1/4 & 0 \\ 0 & 0 & -1/4 & 1 & -1/4 \\ 0 & 0 & 0 & -1/4 & 1 \end{bmatrix} \begin{bmatrix} u_{2,j}^{p+1} \\ u_{3,j}^{p+1} \\ u_{4,j}^{p+1} \\ u_{5,j}^{p+1} \\ u_{n-1,j}^{p+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} u_{2,j+1}^p + \frac{1}{4} u_{2,j-1}^{p+1} + \frac{1}{4} b_I \\ \frac{1}{4} u_{3,j+1}^p + \frac{1}{4} u_{3,j-1}^{p+1} \\ \frac{1}{4} u_{4,j+1}^p + \frac{1}{4} u_{4,j-1}^{p+1} \\ \frac{1}{4} u_{5,j+1}^p + \frac{1}{4} u_{5,j-1}^{p+1} \\ \frac{1}{4} u_{n-1,j+1}^p + \frac{1}{4} u_{n-1,j-1}^{p+1} + \frac{1}{4} b_E \end{bmatrix} \quad (6.14)$$

where  $b_I$  and  $b_E$  are inlet and exit boundary values, and are equal to the points at  $u_{1,j}$  and  $u_{n,j}$ .

We see we have a linear algebra problem with  $n-2$  unknowns. The matrix is tridiagonal, and we can use the Thomas Algorithm.

### 6.3 Over-relaxation

We observe after solving many examples using any of the three previously described relaxation techniques that the solutions monotonically progress from the initial guess to the final solutions. It is natural to take advantage of this information, and we do this by making guesses of the unknown values as opposed to simply using the last calculated value. The procedure is an extrapolation or acceleration, and is most commonly called overrelaxation. The procedure can be applied to both Gauss-Seidel and Line Relaxation. Consider the previous equation for Gauss-Seidel:

$$u_{i,j}^{p+1} = \frac{1}{4}(u_{i+1,j}^p + u_{i-1,j}^{p+1} + u_{i,j+1}^p + u_{i,j-1}^{p+1}) - \frac{1}{4} b_{i,j}^p \quad (6.15)$$

At the time we perform this calculation, we also know the value of  $u_{i,j}^p$ . To extrapolate we can use the difference in  $u_{i,j}^p$  and  $u_{i,j}^{p+1}$ , as follows:

$$\bar{u}_{i,j}^{p+1} = u_{i,j}^{p+1} + \alpha(u_{i,j}^{p+1} - u_{i,j}^p) \quad (6.16)$$

where  $u_{i,j}^{p+1}$  is the guess we would have made without extrapolation, and  $\bar{u}_{i,j}^{p+1}$  is an improved guess.  $\alpha$  is an extrapolation factor if it is greater than zero. Solving for  $u_{i,j}^{p+1}$  we obtain

$$\bar{u}_{i,j}^{p+1} = (1 + \alpha)u_{i,j}^{p+1} - \alpha u_{i,j}^p \quad (6.17)$$

letting  $\omega = 1 + \alpha$

$$\bar{u}_{i,j}^{p+1} = \omega u_{i,j}^{p+1} + (1 - \omega) u_{i,j}^p \quad (6.18)$$

and finally

$$u_{i,j}^{p+1} = \frac{1}{\omega} \bar{u}_{i,j}^{p+1} - \frac{(1 - \omega)}{\omega} u_{i,j}^p \quad (6.19)$$

Substituting Equation (6.17) in (6.8), we obtain

$$\bar{u}_{i,j}^{p+1} = \frac{\omega}{4} (u_{i+1,j}^p + u_{i-1,j}^{p+1} + u_{i,j+1}^p + u_{i,j-1}^{p+1}) - \frac{\omega}{4} b_{i,j}^p + (1 - \omega) u_{i,j}^p \quad (6.20)$$

We can drop the bar above  $u_{i,j}^{p+1}$ . Equation (6.18) will converge much faster than Jacobi, Gauss-Seidel or Line Relaxation. The same technique can be applied to Line Relaxation to obtain an even faster converging routine. Equation (6.20) shows the LSOR form.

The value of  $\omega$  must be greater than 1 to obtain increased convergence. If it is too large the solution will oscillate and will converge more slowly. Optional values of  $\omega$  can be obtained by determining the eigenvalues of the coefficient matrix.

Typical convergence terms to obtain solutions with  $10^{-6}$  of the final solutions, for Equation (6.2), for 5 grid points in Table 6.1. and the trend shown is typical for most problems.

Table 6.1 Typical Number of Interaction to Obtain  $10^{-6}$  Error.

|                            |    |
|----------------------------|----|
| Jacobi                     | 70 |
| Gauss-Seidel               | 38 |
| Line Relaxation            | 21 |
| Successive Over-relaxation | 15 |
| Line Over-relaxation       | 12 |



## 7. HYPERBOLIC METHODS

Hyperbolic equations are among the most difficult to accurately solve. This results because of the rigid requirement to place grid points in specific locations. The information in a hyperbolic equation flows along lines, called characteristic lines. We shall see how to manipulate the location of the grid points to place them on the characteristic lines.

### 7.1 First-Order Equations

The single approach to solving a hyperbolic equation would be to try the parabolic methods. Consider this equation:

$$\frac{\partial u}{\partial t} = -V \frac{\partial u}{\partial x} - Ku \tag{7.1}$$

We would use an analog like this for a forward different or explicit approach:

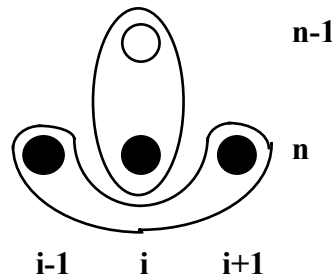


Figure 7.1 Forward Difference Technique Applied to a Hyperbolic Equation

We see that the spatial and time derivatives have no common points. This method fails to provide a solution. Backward difference and Crank-Nicolson fail as well. Alternative approaches might be constructed as shown in in Figure 7.2.

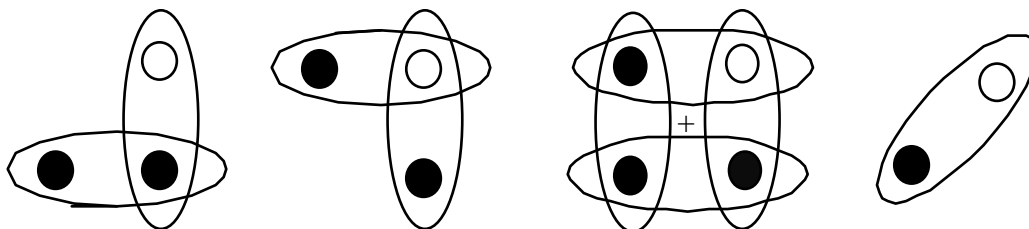


Figure 7.2 Alternative Analogs

The approach that works and works well is shown in Figure 7.3.

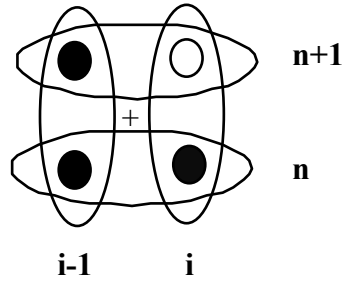


Figure 7.3 Centered Difference Analogs

This method is called centered difference and is a convenient and practical method. If we substitute the analogs into Equation (7.1) we obtain the following:

$$\frac{1}{2} \left[ \frac{u_{i,n+1} - u_{i,n}}{\Delta t} + \frac{u_{i-1,n+1} - u_{i-1,n}}{\Delta t} \right] = -\frac{V}{2} \left[ \frac{u_{i,n+1} - u_{i-1,n+1}}{\Delta x} + \frac{u_{i,n} - u_{i-1,n}}{\Delta x} \right] - \frac{K}{4} [u_{i,n+1} + u_{i,n} + u_{i-1,n+1} + u_{i-1,n}] \quad (7.2)$$

The reaction term can be represented in two ways. Equation (7.2) shows a *four point* average. A *two point* average is also possible, using the terms  $u_{i,n+1}$  and  $u_{i-1,n}$ .

The method is second-order correct in both time and space, and is unconditionally stable. The second-order correctness results because the two time derivatives and two spatial derivatives average to a common point, shown by the "+" in Figure 7.3. Figure 7.4 shows how the analog would be placed at the entrance boundary.

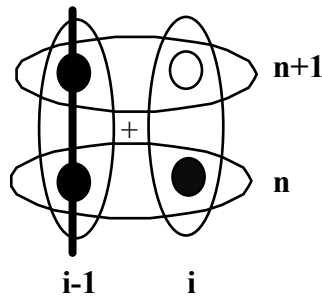


Figure 7.4 Inlet Boundary Condition

If we know the boundary condition the method is explicit. The exit boundary condition is interesting, as shown in Figure 7.5.

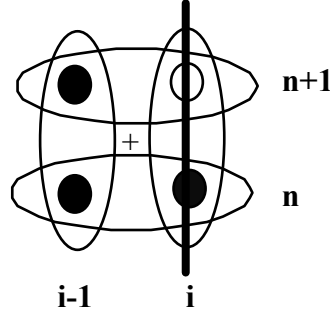


Figure 7.5 Exit Boundary Condition

We see that we can calculate the exit point, and we do not need an exit boundary condition. A first-order hyperbolic equation only has an entrance boundary condition. A second-order hyperbolic equation will have two boundary conditions.

To solve Equation (7.2) we average it to solve for  $u_{i,n+1}$ , as follows:

$$u_{i,n+1} \left[ \frac{1}{\Delta t} + \frac{V}{\Delta x} + \frac{K}{2} \right] = u_{i-1,n} \left[ \frac{1}{\Delta t} + \frac{V}{\Delta x} - \frac{K}{2} \right] + u_{i-1,n+1} \left[ -\frac{1}{\Delta t} + \frac{V}{\Delta x} - \frac{K}{2} \right] \quad (7.3)$$

Note that the "2" in each term canceled. Equation (7.3) can be readily programmed and the final result is no more complicated than a forward difference program for parabolic equations.

It is interesting to consider Equation (7.3) when  $V = \Delta x / \Delta t$  and for  $K = 0$ . We can substitute  $\Delta x = V \Delta t$  as follows:

$$u_{i,n+1} \left[ \frac{1}{\Delta t} + \frac{V}{V\Delta t} \right] = u_{i-1,n} \left[ \frac{1}{\Delta t} - \frac{V}{V\Delta t} \right] + u_{i-1,n+1} \left[ -\frac{1}{\Delta t} + \frac{V}{V\Delta t} \right] \quad (7.4)$$

The equation simplifies to

$$u_{i,n+1} \left[ \frac{1}{2\Delta t} \right] = u_{i-1,n} \left[ \frac{1}{2\Delta t} \right] \quad (7.5)$$

The  $u_{i-1,n+1}$  and  $u_{i,n}$  terms vanish, and we can conclude that

$$u_{i,n+1} = u_{i-1,n} \quad (7.6)$$

Equation (7.6) is the solution to Equation (7.1) when  $K = 0$  and for the condition  $V = \Delta x / \Delta t$ . In fact this is the only correct solution to Equation (7.1). Solutions when  $V \neq \Delta x / \Delta t$  are incorrect. This incorrectness is sometimes call numerical dispersion. Now we should investigate the case when  $K \neq 0$ .

We first observe that when we use a four-point average that the  $u_{i-1,n+1}$  and  $u_{i,n}$  terms do not vanish. Therefore we shall try a two-point average. Equation (7.4) for this case becomes:

$$u_{i,n+1} \left[ \frac{2}{\Delta t} + K \right] = u_{i-1,n} \left[ \frac{2}{\Delta t} - K \right] \quad (7.7)$$

If we rearrange as follows:

$$u_{i,n+1} = u_{i-1,n} - \frac{K}{2} \Delta t [u_{i,n+1} + u_{i-1,n}] \quad (7.8)$$

or

$$u_{i,n+1} = u_{i-1,n} - K \Delta t u_{i-1/2,n+1/2} \quad (7.9)$$

This is a second-order correct ordinary differential equation. We can write ordinary differential equations along the characteristic lines. Each initial grid point is connected to additional grid points - as many as you wish to write, along lines with slope  $1/V = \Delta t / \Delta x$ . Figure 7.6 shows the characteristic lines.

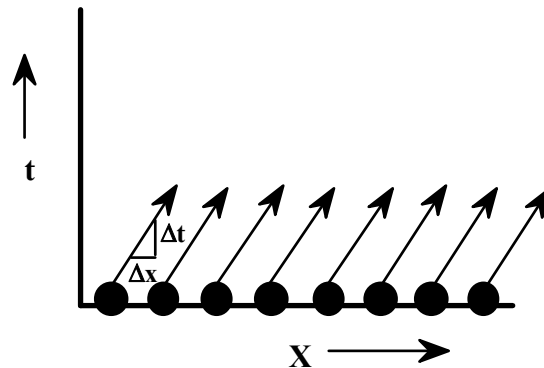


Figure 7.6 Characteristic Lines and Slope

## 7.2 Method of Characteristics for First-Order Equations

The method of characteristics (MOC) is a powerful but sometimes difficult to use procedure to solve hyperbolic equations. Centered difference, with  $V = \Delta x/\Delta t$ , can be considered a special case of MOC. We begin MOC by recalling the procedure for classifying equations. Consider Equation (7.1) with directional derivatives:

$$\begin{bmatrix} 1 & V \\ dt & dx \end{bmatrix} \begin{bmatrix} \frac{\partial u}{\partial t} \\ \frac{\partial u}{\partial x} \end{bmatrix} = \begin{bmatrix} -Ku \\ du \end{bmatrix} \quad (7.10)$$

We calculate the determinant of the coefficient matrix and set it to zero to obtain:

$$dx - Vdt = 0 \quad (7.11)$$

or

$$V = \frac{dx}{dt} \quad (7.12)$$

We now substitute the forcing vector into the coefficient matrix as follows:

$$\begin{bmatrix} 1 & -Ku \\ dt & du \end{bmatrix} \quad (7.13)$$

The determinant becomes:

$$du + Ku dt = 0 \quad (7.14)$$

or

$$\frac{du}{u} = -Kdt \quad (7.15)$$

Equation (7.15) is easy to integral analytically, to obtain

$$\frac{u}{u_B} = e^{-Kt} \quad (7.16)$$

where  $u_B$  is the initial condition [ $u_B = u(x, 0)$ ]. In the analytical solution it is introduced as an integration constant. This is also the analytical solution to Equation (7.8). Note that Equation (7.16) is valid only when  $\Delta x$  and  $\Delta t$  are selected so that  $V = \Delta x/\Delta t$ . If we do not follow this constraint, we obtain an erroneous solution. We call Equation (7.12) the *characteristic equation*, and we call Equation (7.15) the *integration equation*. You can also think of Equation (7.12) as telling you where you can calculate a solution, while Equation (7.15) tells you the value of the solution.

Both centered difference and MOC provide solutions to Equation (7.1). For  $K = 0$  and  $V = \Delta x/\Delta t$ , both techniques are *exact*, which means the solution they provide has zero truncation error. The two point average should for the reaction terms be used when  $V = \Delta x/\Delta t$ . In some cases it may not be possible to use  $V = \Delta x/\Delta t$ , and under these circumstances numerical dispersion will occur, and it may be better to use a four point average.

We now define the *Courant* number. For hyperbolic systems, a stable solution is possible when the Courant number,  $V\Delta t/\Delta x$  is less than 1.0.

### 7.3 Method of Characteristics for Second-Order Equations

For second-order hyperbolic equations, MOC can still be used and provide a mechanism for keeping track of grid points. We return to Equation (2.7), which is two simultaneous first-order equations which could represent the second-order hyperbolic PDE,

$$\frac{\partial^2 u}{\partial t^2} - \alpha \frac{\partial^2 u}{\partial x^2} = f \quad (7.17)$$

$$\begin{bmatrix} a & b & c & d \\ e & g & h & i \\ dx & dy & 0 & 0 \\ 0 & 0 & dx & dy \end{bmatrix} \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ du \\ dv \end{bmatrix} \quad (2.7)$$

We set the determinant to zero and obtain the following equation

$$\left(\frac{dy}{dx}\right)^2 (ce - ah) + \frac{dy}{dx} (ai - de + bh - cg) + dg - bi = 0 \quad (7.18)$$

This equation is hyperbolic if Equation (7.18) has two real roots for  $\frac{dy}{dx}$ . We make the following assumptions and we can use the quadratic formula, as follows:

$$\begin{aligned} A &= ce - ah \\ B &= ai - de + bh - cg \\ C &= dg - bi \end{aligned}$$

$$\frac{dy}{dx}\bigg|_1 = \alpha = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (7.19)$$

$$\left. \frac{dy}{dx} \right|_2 = \beta = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \quad (7.20)$$

Note that if  $B^2 - 4AC \leq 0$ , then the Equation (7.17) is not hyperbolic and we must seek a different solution technique. The results of Equations (7.19) and (7.20) are direction equations. We can pick a  $dx$  and calculate  $dy$ , or vice versa.

Next we define the integration equation, by substituting the forcing vector into one of the columns of the coefficient matrix, and then setting it to zero, as follows:

$$\begin{vmatrix} a & b & c & f_1 \\ e & g & h & f_2 \\ dx & dy & 0 & du \\ 0 & 0 & dx & dv \end{vmatrix} = 0 \quad (7.21)$$

We expand along the bottom row to take advantage of the two zeros, as follows:

$$dx \begin{vmatrix} a & b & f_1 \\ e & g & f_2 \\ dx & dy & du \end{vmatrix} - dv \begin{vmatrix} a & b & c \\ e & g & h \\ dx & dy & 0 \end{vmatrix} = 0 \quad (7.22)$$

We expand to obtain

$$\begin{aligned} & dx[a(gdu - f_2dy) - b(edu - f_2dx) + f_1(edy - gdx)] \\ & - dv[dx(bh - cg) - dy(ah - ce)] = 0 \end{aligned} \quad (7.23)$$

expanding

$$\begin{aligned} & dxduag - dxadyf_2 - bdxedu + d^2xbf_2 + dx f_1edy - f_1gd^2x \\ & - dvdxbh + dvdxcg + dvdyah - dvdyce = 0 \end{aligned} \quad (7.24)$$

Now collecting terms by  $dv$  and  $du$

$$\begin{aligned} & du(dxag - bdx e) + dv(-dxbh + dxcg + dyah - dyce) \\ & - dxadyf_2 + d^2xbf_2 + dx f_1edy - f_1gd^2x = 0 \end{aligned} \quad (7.25)$$

We divide by  $dx$  to obtain:

$$\begin{aligned} & du(ag - be) + dv(-bh + cg + \frac{dy}{dx}ah - \frac{dy}{dx}ce) \\ & - adyf_2 + dxbf_2 + f_1edy - f_1gd^2x = 0 \end{aligned} \quad (7.26)$$

We substitute  $\alpha$  and  $\beta$  into Equation (7.26) to obtain two new equations, as follows:

Let

$$p_1 = ag - be$$

$$p_2 = -bh - cg$$

$$p_3 = adyf_2 - dxbf_2 - f_1edy + f_1gdx \quad (\text{along } \alpha \text{ direction})$$

$$p_4 = adyf_2 - dxbf_2 - f_1edy + f_1gdx \quad (\text{along } \beta \text{ direction})$$

$$p_1 du + dv(p_2 + ah\alpha - ce\alpha) = p_3 \quad (7.27)$$

$$p_1 du + dv(p_2 + ah\beta - ce\beta) = p_4 \quad (7.28)$$

Note that  $p_3$  and  $p_4$  are different because the value of either  $dx$  or  $dy$  will be different along the different characteristic lines.



## 8. NONLINEAR METHODS

The purpose of using numerical methods for solving PDE's is to solve nonlinear equations, for which there are no analytical solutions. When we introduce reaction terms, such as biological reactions or adsorption terms, we often make the equations quasi-linear. Consider this equation:

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} - Ku^2 \quad (8.1)$$

This is a quasi-linear equation. The introduction of the second-order reaction term creates the nonlinearity. If we think about solving this equation using forward difference, we envision these analogs.

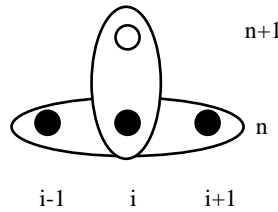


Figure 8.1 Forward Difference Analogs

If we write the nonlinear term at the old time level,  $n$ , it can be calculated from  $u_{i,n}$ , no matter what sort of nonlinear term it is. For forward difference, with the nonlinear term is written at the old time level, the numerical solution is about the same complexity as a linear solution except that the time and distance steps will need to be much smaller. For this reason, "canned" or "packaged" programs use the forward difference technique.

When we use a technique that requires the nonlinear term to be represented at the new time level, we introduce difficulties in solving the resulting system of algebraic equations. For example, with backward difference, the second-order reaction term will appear on the left-hand side of the algebraic equations, in the "B" term of our canonical form, as follows:

$$\begin{aligned} & \left[ \frac{-D}{\Delta x^2} \right] u_{i-1,n+1} \\ & \left[ \frac{1}{\Delta t} + \frac{2D}{\Delta x^2} + Ku_{i,n+1} \right] u_{i,n+1} = \frac{1}{\Delta t} u_{i,n} \\ & \left[ \frac{-D}{\Delta x^2} \right] u_{i+1,n+1} \end{aligned} \quad (8.2)$$

We see that we have a  $u^2$  term in the unknowns and our linear algebra solutions fail. Therefore, we have no way of solving the equations for all our implicit methods, such as Crank-Nicolson and ADI. For explicit methods we have a nonlinear equation for each analog, which may or may not be directly solvable, depending upon the nature of the nonlinearity. It may be required to use an iterative method at each point.

To solve the nonlinear PDE's we introduce two techniques to retain the linear algebraic equations. The first is to *project* the nonlinear term, so that it can be replaced by a known value. The second is *iteration*, which is a technique to determine how well we projected the nonlinear terms, and to improve the projection.

There are four common ways of projecting nonlinear terms: using the old time level, forward projection, backward projection, and central projection. All involve "factoring" the nonlinear term into a linear portion and a projected portion. The term in Equation (8.1),  $u^2$ , would be factored into  $u \cdot u^*$ , where  $u^*$  is a projected value. The "B" term in the canonical form, shown in Equation (8.2) would become

$$\begin{array}{c} \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \\ \left[ \frac{1}{\Delta t} + \frac{2D}{\Delta x^2} + K u_{i,n+1}^* \right] u_{i,n+1} = \frac{1}{\Delta t} u_{i,n} \\ \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \end{array}$$

This equation is linear and the Thomas Algorithm can still be used to solve the resulting system of equations. Any nonlinear term that we can envision can be factored in this way. Here are a few examples:

| Nonlinear term      | Linearized term       |
|---------------------|-----------------------|
| $u^2$               | $u \cdot u^*$         |
| $\frac{u}{K_s + u}$ | $\frac{u}{K_s + u^*}$ |
| $e^{-u}$            | $e^{-u^*}$            |

In the case of the last term,  $e^{-u}$ , the linearized result would be transposed to the right hand side of the canonical form (placed in the "knowns").

The easiest form of projection is to use the old time level. In our example case we would use  $u_{i,n} \cong u_{i,n+1}^*$ . If the solution is not rapidly changing, then the change with each  $\Delta t$  is small, and  $u_{i,n} \cong u_{i,n+1}^*$  is a relatively good assumption. Note that we make the project for each point at the new time level. For more rapidly changing solutions, we could use smaller  $\Delta t$ . Such a procedure could work for almost any quasi-linear equation, but may not be the

fastest. Also we have no way, other than a sensitivity analysis, to determine how accurate we are.

The second method is forward projection. This method uses a forward difference program within the main program (e.g., Crank-Nicolson, ADI, etc.) to calculate the value of  $u_{i,n+1}^*$ . The forward difference program uses the old time level to approximate the nonlinear terms. The improvement of forward projection over using the old time level are mixed. The  $\Delta t$  and  $\Delta x$  for the forward difference program must be selected so that the solution is stable. It is difficult to selected a  $\Delta x$  that is different than the primary technique. Therefore, one must pick smaller  $\Delta t$  's, or accept the instability. The forward difference program is only use for one time step, since the  $n$  time level for the forward difference program is always the  $n$  time level in the primary program. The projected values ( $u_{i,n+1}^*$ ) are always discarded after they are used for linearization; therefore, the error with each time step does not build up. Nevertheless, the error associated with forward projection with  $D\Delta t/\Delta^2x > 1/2$  can be significant.

A better alternative is to use backward projection. In this technique the nonlinear terms are written at the  $n$  time level. An implicit solution is required and the Thomas Algorithm works for this purpose. Stability is not a problem. Central projection is similar to backward projection, except that Crank-Nicolson analogs are used. Backward difference is a large improvement over forward projection or using the old time level. Central projection takes almost as much computer time as the primary technique, and usually does not provide enough improvement to warrant its use. Figure 8.2 shows how a linear Crank-Nicolson program would be modified to include projection.

The next question that arises is "what if the projected value, even with the best projection, is inaccurate?" An obvious remedy is to decrease the time step which reduces truncation error and also reduces the length of the projection (e.g., short  $\Delta t$  as opposed to long  $\Delta t$ ). This technique works but may produce excessive computer time, especially if there is a requirement to maintain  $V = \Delta x/\Delta t$ , as in the near hyperbolic case. A better approach is to use iteration.

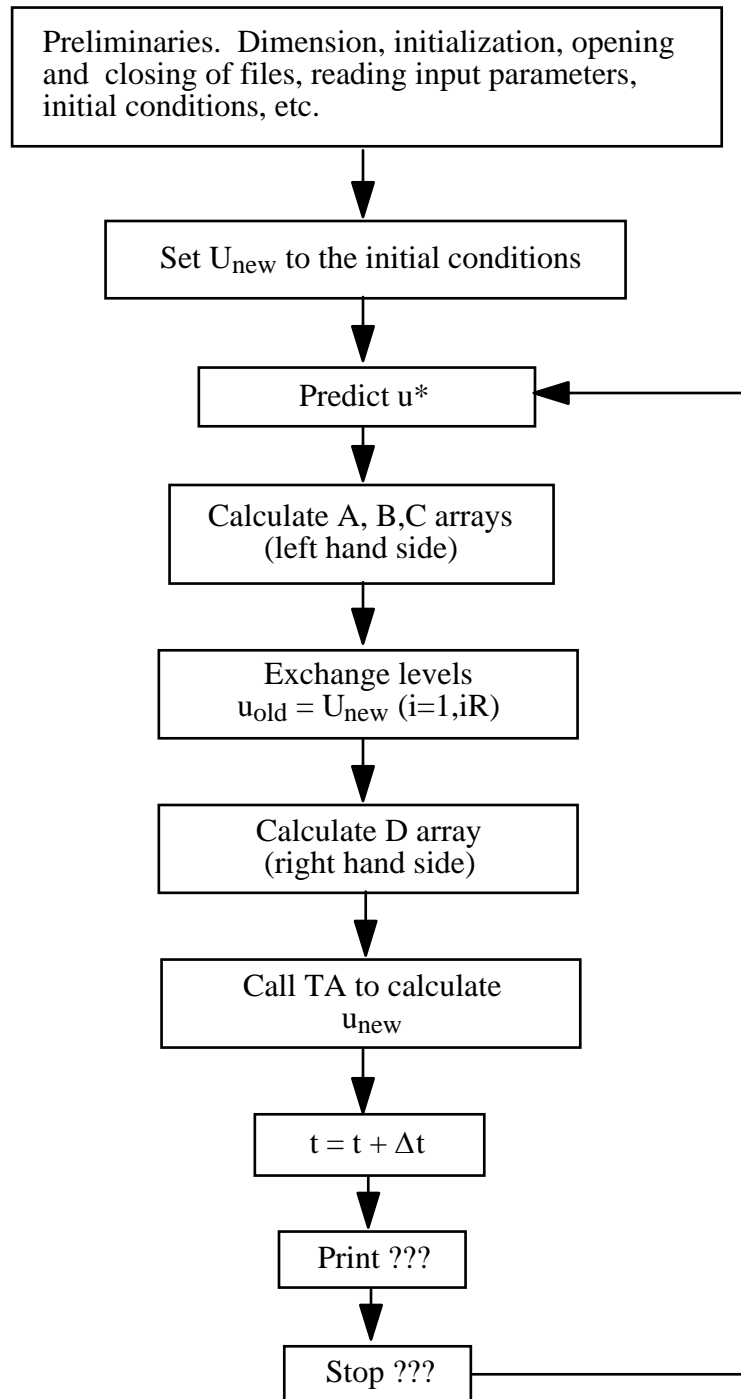


Figure 8.2 Block diagram of a program that uses prediction to solve a quasi-linear equation.

There are three basic types of iteration: direct, modified and using a convergence technique such as regula-falsi, the secant method, or Newton-Raphson. The first two are often called Picard and modified Picard, respectively.

In direct iteration the first projection is the same as discussed previously. The old time level or other methods of projection can be used. The primary method is next used to solve for  $u_{i,n+1}$ . After solution we have three set of u's:  $u_{i,n+1}$ ,  $u_{i,n+1}^*$  and  $Tu_{i,n+1}$ . The  $Tu_{i,n+1}$  is called a *trial* value of  $u_{i,n+1}$ . We have not yet decided to accept it. Presumably, if it is accurate, we should accept it and increment time. If it is inaccurate then we should estimate new and more accurate values of  $u_{i,n+1}$ , then solve for new values of  $Tu_{i,n+1}$ . We should check to see if the difference between  $u_{i,n+1}^*$  and  $Tu_{i,n+1}$ , as follows:

$$\varepsilon \geq \sum_{i=1}^{IR} |u_{i,n+1}^* - Tu_{i,n+1}| \quad (8.3)$$

We use the absolute value in Equation (8.3) because it is usually faster than summing the squares. We select an  $\varepsilon$  that is sufficiently small to insure an accurate solution. A sensitivity analysis can be performed to determine the impact of  $\varepsilon$  on the overall accuracy.

If the difference between  $u_{i,n+1}^*$  and  $Tu_{i,n+1}$  is too large (e.g.,  $> \varepsilon$ ) then we select a new value of  $u_{i,n+1}^*$ . The different methods of selecting  $u_{i,n+1}^*$  give rise to the different iteration techniques. With direct iteration, we substitute  $u_{i,n+1} = Tu_{i,n+1}$ . The trial values become the projected values. It may be necessary to iterate many times, so we need to number our trial values, such as  $Tu_{i,n+1}^1$ ,  $Tu_{i,n+1}^2$ , etc., and our projected values,  $u^{*1}$ ,  $u^{*2}$ , etc.

If the summation is less than  $\varepsilon$ , then the trial u values are accepted and they become  $u_{i,n+1}$ . Execution proceeds, time is incremented, and the program can stop or continue depending upon the value of t, or other criteria.

If the summation is greater than  $\varepsilon$ , then a better estimate of  $u_{i,n+1}^*$  is required. With direct iteration, the new estimate of  $u_{i,n+1}^*$  is set equal to the trial values,  $Tu_{i,n+1}$ , for all values of i.

$$u_{i,n+1}^* = Tu_{i,n+1} \quad (8.4)$$

The trial values are discarded. Execution proceeds until convergence is obtained. For some problems the solution will not converge - it diverges. A counter should be placed in the iteration loop to insure that you do not have an infinite loop.

For modified iteration the value of  $u_{i,n+1}^*$  is calculated from the new trial value and the old projected value, as follows:

$$u_{i,n+1}^{*2} = wfTu_{i,n+1}^1 + (1 - wf)u_{i,n+1}^{*1} \quad (8.5)$$

where  $f$  is a weighing factor that ranges between zero and 1.0, and the  $u_{i,n+1}^*$  on the right hand side are the projected values used to produce the trial values. We begin to number the  $u^*$ 's since we have more than one set. Figure 8.3 show how iteration can be incorporated into a Crank-Nicolson program.

For problems that do not converge with direct iteration, modified iteration is more likely to converge. Problems that converge with direct iteration will also converge with modified iteration, but more slowly. Modified iteration should not be used unless you are relatively confident that direct iteration will not converge.

In cases where modified iteration does not converge, or is very slow to converge, a third option is likely. You can use a convergence technique. Using such a technique, a better guess for  $u_{i,n+1}^*$  is made using old projection for  $u_{i,n+1}^*$  and the trial values they produced. Figure 8.4 shows how such a technique could work.

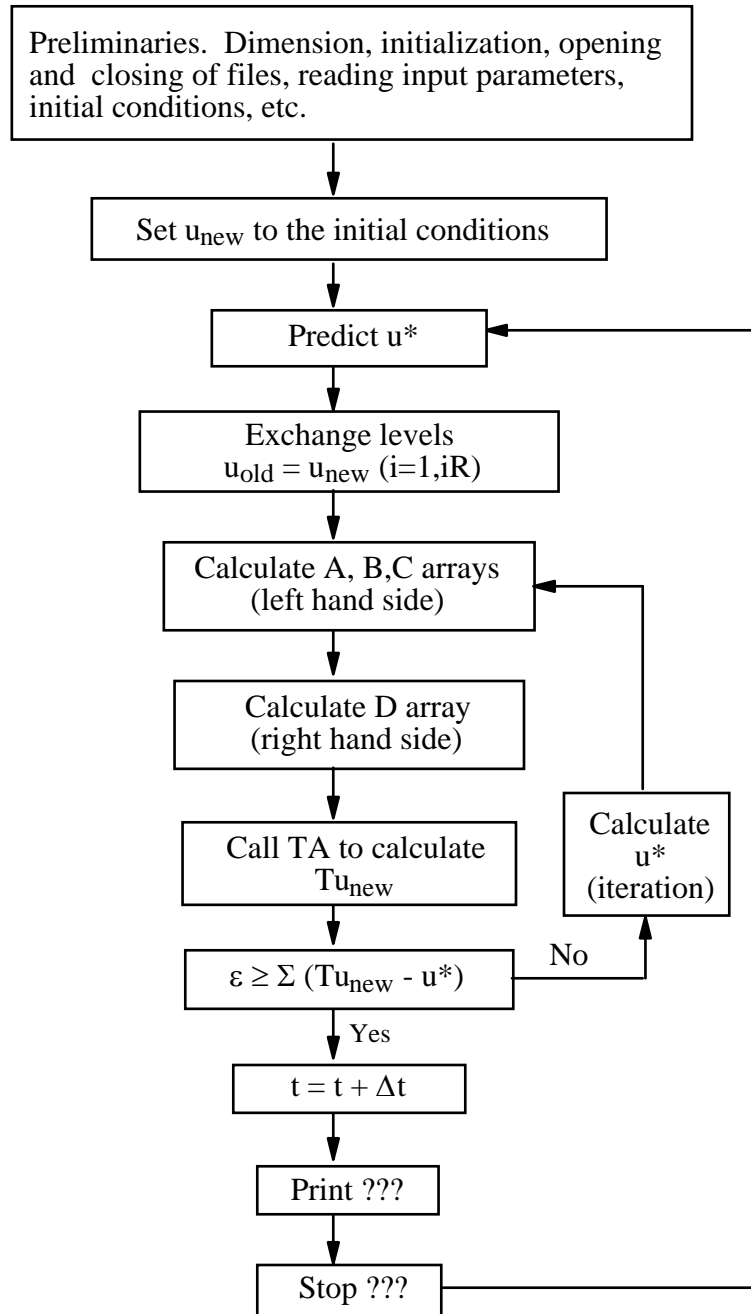


Figure 8.3 Block diagram of a program that uses prediction and iteration to solve a quasi-linear equation.

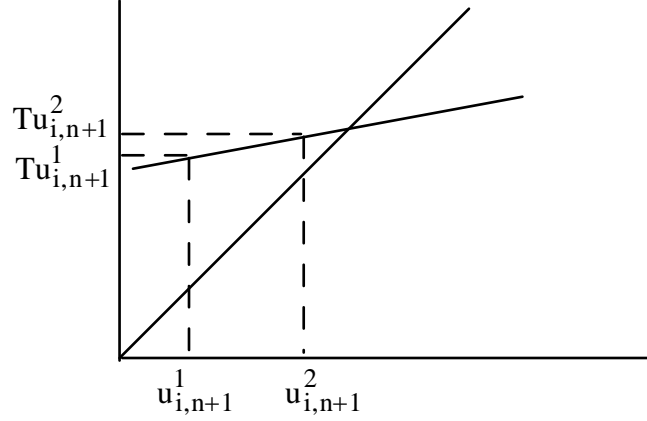


Figure 8.4

There are two lines on Figure 8.4. The first is an identity: we wish  $Tu = u^*$ , and the line with a slope of 1, crossing the origin describes this equation. The second line is defined by the results of the first two iterations. On the first solution, we projected a value of  $u^*$ , either by using the old time level, backward, forward or central projection. We then calculated the first trial value,  $Tu_{i,n+1}^1$ , and we then make a new projection for  $u_{i,n+1}^*$  using either direct or weighted iteration. Finally, we calculate a second trial value  $Tu_{i,n+1}^2$ . At this point we have four variables: two estimates for  $u_{i,n+1}^*$  and two estimates for  $Tu_{i,n+1}$ . These points are plotted on Figure 8.4. Note that we can think of the PDE solution as a function relating  $u_{i,n+1}^*$  and  $Tu_{i,n+1}$ . We provide a projected value, and we calculate a trial value. We can represent this as follows:

$$Tu_{i,n+1} = f(u_{i,n+1}^*) \quad (8.6)$$

We can represent this function as a straight line on Figure 8.4. The slope of the line is defined as

$$m = \text{slope} = \frac{Tu_{i,n+1}^2 - Tu_{i,n+1}^1}{u_{i,n+1}^2 - u_{i,n+1}^1} \quad (8.7)$$

The intercept is calculated from one pair of the trial and projected values, as follows:

$$b = Tu_{i,n+1}^2 - mu_{i,n+1}^{*2} \quad (8.8)$$

We now have two equations relating projected values to trial values, as follows:

$$Tu_{i,n+1}^p = Tu_{i,n+1}^{*p} \quad (8.9)$$

$$Tu_{i,n+1}^p = mu_{i,n+1}^{*p} + b \quad (8.10)$$

where  $m$  and  $b$  are the slope and intercept, as calculated in Equations (8.7) and (8.8), and  $p$  is the iteration (trial) number. Equations (8.9) and (8.10) can be solved simultaneously to



obtain a new projection,  $u_{i,n+1}^{*p}$ . technique can be applied as many times as necessary to obtain convergence, The p counter is increased with each iteration.

There are other methods which can be used to quicken convergence. The secant method can be used, as well as a method called *Regula Falsi*. The disadvantage of the methods is that additional values must be stored. With direct iteration in a nonlinear problem, we stored only two levels of u ( $u_{i,n}$  and  $Tu_{i,n+1}$ ) and one projected value,  $u_{i,n+1}^*$ . With the described convergence, five levels of u must be stored ( $u_{i,n}$ ,  $Tu_{i,n+1}$ ,  $Tu_{i,n+1}^{p-1}$ ,  $u_{i,n+1}^{*p}$  and  $u_{i,n+1}^{*p-1}$ ). This storage requirement could be significant with many grid points as in two-dimensional problems.

## 9. STABILITY ANALYSIS

A solution is stable if the truncation error remains finite with increasing time. To determine if the solution is stable we look at the difference between the real solution,  $u_{i,n}$ , and the truncated solution,  $w_{i,n}$ , as  $n$  goes to infinity. This will be demonstrated using the forward difference for the heat equation.

$$\frac{\partial u}{\partial t} = \alpha \left( \frac{\partial^2 u}{\partial x^2} \right) \quad (9.1)$$

The exact solution uses the forward difference analogs before the higher order terms have been truncated. Substituting these into (9.1) gives

$$\alpha \left[ \frac{u_{i+1,n} - 2u_{i,n} + u_{i-1,n}}{\Delta^2 x} - \frac{\partial^4 u}{\partial x^4} \Big|_{i,n} \frac{2\Delta^2 x}{4!} + \text{HOTS} \right] = \left[ \frac{u_{i,n+1} - u_{i,n}}{\Delta t} - \frac{\partial^2 u}{\partial t^2} \Big|_{i,n} \frac{\Delta t}{2!} + \text{HOTS} \right] \quad (9.2)$$

The approximate solution that is second order correct in the spatial derivative and first order correct in the time derivative yields the equation

$$\alpha \left[ \frac{w_{i+1,n} - 2w_{i,n} + w_{i-1,n}}{\Delta^2 x} \right] = \frac{w_{i,n+1} - w_{i,n}}{\Delta t} \quad (9.3)$$

If  $z$  is the truncation error, then  $z_{i,n} = u_{i,n} - w_{i,n}$ . Subtracting (9.3) from (9.2) and substituting  $z_{i,n}$  for  $u_{i,n} - w_{i,n}$  an equation for the truncation error is obtained.

$$\alpha \left[ \frac{z_{i+1,n} - 2z_{i,n} + z_{i-1,n}}{\Delta^2 x} + \text{HOTS} \right] = \left[ \frac{z_{i,n+1} - z_{i,n}}{\Delta t} + \text{HOTS} \right] \quad (9.4)$$

Neglecting the higher order terms of the error we can now ask if  $z$  becomes infinite or remains finite as  $n \rightarrow \infty$ . Suppose that there are two fixed boundaries

$$\begin{aligned} u_{x=0} &= a \\ u_{x=1} &= b \end{aligned} \quad (9.5)$$

because there is no truncation error at fixed boundaries

$$\begin{aligned} z_{x=0} &= 0 \\ z_{x=1} &= 0 \end{aligned} \quad (9.6)$$

Using the shorthand notation, the equation for  $z$  is

$$\alpha \delta_x^2 z_{i,n} = \delta_t z_{i,n} \quad (9.7)$$

To solve this equation, we will use the separation of variables technique and assume a solution of the form

$$z = \rho_n \phi_i \quad (9.8)$$

where

$\rho_n$  is a function of time only and  
 $\phi_i$  is a function of distance only

Substituting (9.8) into (9.7) we obtain

$$\alpha \delta_x^2 (\rho_n \phi_i) = \delta_t (\rho_n \phi_i) \quad (9.9)$$

Because  $\rho_n$  is not a function of  $x$ , it can be taken out of the parentheses on the left hand side of the equation. Similarly,  $\phi_i$  is not a function of  $t$ , so it is constant with respect to the time derivative. Removing the shorthand notation for the time derivative (9.9) becomes

$$\alpha \rho_n \delta_x^2 \phi_i = \phi_i \left[ \frac{\rho_{n+1} - \rho_n}{\Delta t} \right] \quad (9.10)$$

or

$$\alpha \frac{\rho_n}{\phi_i} \delta_x^2 \phi_i = \frac{\rho_{n+1} - \rho_n}{\Delta t} \quad (9.11)$$

$N = \frac{\rho_{n+1}}{\rho_n}$  is called the time amplification factor. If  $|N|$  is less than 1, then the time portion of the error decreases as  $n \rightarrow \infty$ . This is called the von Neumann condition. Substituting  $N$  in (9.10) and rearranging we obtain

$$\delta_x^2 \phi_i + \frac{1-N}{\alpha \Delta t} \phi_i = 0 \quad (9.11)$$

with boundary conditions

$$\phi_i = 0 \Big|_{x=0}$$

$$\phi_i = 0 \Big|_{x=1}$$

Now assume a solution for  $\phi_i$

$$\phi_i = A \sin(\pi x_i) + B \cos(\pi x_i) \quad (9.12)$$

Applying the boundary conditions

$$\text{at } x = 0 \quad \phi_i = B = 0$$

$$\text{at } x = 1 \quad \phi_i = A \sin(\pi p) = 0$$

From the first condition there are only sine terms in the solution and from the second condition  $p$  must be an integer. Substituting the trial solution,  $\phi_i = A \sin(\pi p x_i)$ , (9.11) becomes

$$\delta_x^2 A \sin(\pi p x_i) + \frac{1-N}{\alpha \Delta t} A \sin(\pi p x_i) = 0 \quad (9.13)$$

$$p = 0, \pm 1, \pm 2, \dots$$

Substituting

$$\delta_x^2 A \sin(\pi p x_i) = \frac{-4}{(\Delta x)^2} \sin^2\left(\pi p \frac{\Delta x}{2}\right) A \sin(\pi p x_i)$$

into (9.13), rearranging and dividing by  $A \sin(\pi p x_i)$  we obtain

$$\sin\left(\pi p \frac{\Delta x}{2}\right) \left[ \frac{-4}{(\Delta x)^2} \sin^2\left(\pi p \frac{\Delta x}{2}\right) + \frac{1-N}{\alpha \Delta t} \right] = 0 \quad (9.14)$$

or

$$N = 1 - \frac{4\alpha \Delta t}{(\Delta x)^2} \sin^2\left(\pi p \frac{\Delta x}{2}\right) \quad p = 0, \pm 1, \pm 2, \dots$$

We can cancel the  $\sin\left(\pi p \frac{\Delta x}{2}\right)$  since it will generally be non-zero.

Recalling that the criterion for stability is  $|N| \leq 1$

$$\left| 1 - \frac{4\alpha \Delta t}{(\Delta x)^2} \sin^2\left(\pi p \frac{\Delta x}{2}\right) \right| \leq 1 \quad (9.15)$$

The absolute value signs can be removed by converting (9.15) to the joint constraint

$$1 - \frac{4\alpha \Delta t}{(\Delta x)^2} \sin^2\left(\pi p \frac{\Delta x}{2}\right) \leq 1 \quad (9.16)$$

and

$$1 - \frac{4\alpha \Delta t}{(\Delta x)^2} \sin^2\left(\pi p \frac{\Delta x}{2}\right) \geq -1 \quad (9.17)$$

Since  $\sin^2$  is always positive and  $\frac{4\alpha \Delta t}{(\Delta x)^2}$  is positive for  $\alpha > 0$ , (9.16) is met for all  $\Delta x$  and  $\Delta t$ .

The maximum value for  $\sin^2\left(\pi p \frac{\Delta x}{2}\right)$  is 1, so (9.17) becomes

$$1 - \frac{4\alpha\Delta t}{(\Delta x)^2} \geq -1 \quad (9.18)$$

or

$$\frac{\alpha \Delta t}{(\Delta x)^2} \leq \frac{1}{2} \quad (9.19)$$

This is a necessary condition for stability of the forward difference method. The conditions for stability of other methods can be determined from the same procedure. Recall that the differential equation was first written using the complete Taylor series expansion. Second, the equation was written using the analog approximations. An equation for the truncation error was then obtained by subtracting the approximate equation from the complete Taylor series equation. A solution consisting of a time dependent component and a spatially dependent component was assumed and substituted into the error equation. For stability we required that the magnitude of the ratio of the time component at one time step to the time component at the previous time step be less than 1. This is necessary for the truncation error to remain finite. Note that stability does not guarantee an accurate solution because the truncation error may be very large even though it is finite.

The stability of the backward difference method will now be demonstrated using (9.1). The differential equation is first written using Taylor expansion around the point  $u_{i,n+1}$ . The backward difference analog equivalent of the equation is then subtracted. After truncating higher order terms this leaves the following equation

$$\alpha \left[ \frac{z_{i+1,n+1} - 2z_{i,n+1} + z_{i-1,n+1}}{\Delta^2 x} \right] = \frac{z_{i,n+1} - z_{i,n}}{\Delta t} \quad (9.20)$$

Again assuming a solution of the form  $z = \rho_n \phi_i$  (9.20) becomes

$$\alpha \delta_x^2(\rho_{n+1} \phi_i) = \phi_i \frac{(\rho_{n+1} - \rho_n)}{\Delta t} \quad (9.21)$$

Taking  $\rho_{n+1}$  outside of the spatial derivative and substituting in  $N = \frac{\rho_{n+1}}{\rho_n}$  yields

$$\delta_x^2 \phi_i = \frac{\phi_i}{\alpha \Delta t} \frac{N-1}{N} \quad (9.22)$$

Since the boundary conditions are the same as in the derivation of (9.19), we assume a solution of the form  $\phi_i = A \sin(\pi p x_i)$ . After substitution and rearrangement, (9.22) becomes

$$N = \frac{1}{1 + \frac{4\alpha\Delta t}{\Delta^2 x} \sin^2 \frac{\pi p \Delta x}{2}} \quad (9.23)$$

For  $\alpha > 0$  and  $\frac{\Delta t}{(\Delta x)^2}$  finite,  $0 < N \leq 1$ . Therefore, the von Neumann condition is met for any choice of  $\Delta x$  and  $\Delta t$ , and the method is unconditionally stable.

If this procedure is repeated using Crank-Nicolson analogs, the following equation for  $N$  is obtained

$$N = \frac{1 - 2 \left[ \frac{\alpha \Delta t}{\Delta^2 x} \sin^2 \frac{\pi p \Delta x}{2} \right]}{1 + 2 \left[ \frac{\alpha \Delta t}{\Delta^2 x} \sin^2 \frac{\pi p \Delta x}{2} \right]} \quad (9.24)$$

In this case  $-1 < N \leq 1$  for  $\alpha > 0$  and any choice of  $\Delta x$  and  $\Delta t$ . This method, then, is also unconditionally stable.

For the final example for stability analysis the stability condition for the two dimensional advection-dispersion equation will be derived for the explicit analogs. The governing equation is

$$\frac{\partial u}{\partial t} = D_x \frac{\partial^2 u}{\partial x^2} + D_y \frac{\partial^2 u}{\partial y^2} - V_x \frac{\partial u}{\partial x} - V_y \frac{\partial u}{\partial y} \quad (9.25)$$

Expressed using the Taylor series expansion using  $j$ ,  $k$  and  $n$  as the indices for  $x$ ,  $y$ , and  $t$ , this becomes

$$\begin{aligned} &= D_x \left[ \frac{u_{j+1,k,n} - 2u_{j,k,n} + u_{j-1,k,n}}{\Delta^2 x} - 2 \frac{\Delta^2 x}{4!} \frac{\partial^4 u}{\partial x^4} \Bigg|_{j,k,n} + \text{HOTs} \right] \\ &+ D_y \left[ \frac{u_{j,k+1,n} - 2u_{j,k,n} + u_{j,k-1,n}}{\Delta^2 y} - 2 \frac{\Delta^2 y}{4!} \frac{\partial^4 u}{\partial y^4} \Bigg|_{j,k,n} + \text{HOTs} \right] \\ &- V_x \left[ \frac{u_{j+1,k,n} - u_{j-1,k,n}}{2 \Delta x} - \frac{\Delta^2 x}{3!} \frac{\partial^3 u}{\partial x^3} \Bigg|_{j,k,n} - \text{HOTs} \right] \\ &- V_y \left[ \frac{u_{j,k+1,n} - u_{j,k-1,n}}{2 \Delta y} - \frac{\Delta^2 y}{3!} \frac{\partial^3 u}{\partial y^3} \Bigg|_{j,k,n} - \text{HOTs} \right] \end{aligned} \quad (9.26)$$

Subtracting the forward difference analog (5.3), substituting  $z = u - w$  and truncating the higher order error terms we obtain

$$D_x \delta_x^2(z_{j,k,n}) + D_y \delta_y^2(z_{j,k,n}) - V_x \delta_x(z_{j,k,n}) - V_y \delta_y(z_{j,k,n}) = \delta_t(z_{j,k,n}) \quad (9.27)$$

We will assume a solution of the form

$$z_{j,k,n} = \rho_n e^{i p j \Delta x} e^{i q k \Delta y} \quad (9.28)$$

where  $i = \sqrt{-1}$  and  $\rho_n$  is again a function of  $t$  only. In general, the constants  $p$  and  $q$  can be complex. Substituting (9.28) into (9.27) the governing equation and taking terms that are constant outside the derivatives we obtain

$$\begin{aligned} & D_x \rho_n e^{i k q \Delta y} \delta_x^2(e^{i j p \Delta x}) + D_y \rho_n e^{i j p \Delta x} \delta_y^2(\rho_n e^{i k q \Delta y}) \\ & - V_x \rho_n e^{i k q \Delta y} \delta_x(e^{i j p \Delta x}) + V_y \rho_n e^{i j p \Delta x} \delta_y(\rho_n e^{i k q \Delta y}) \\ & = e^{i j p \Delta x} e^{i k q \Delta y} \left( \frac{\rho_{n+1} - \rho_n}{\Delta t} \right) \end{aligned} \quad (9.29)$$

Taking the derivatives and dividing by  $e^{i j p \Delta x} e^{i k q \Delta y} \rho_n$ , (9.29) becomes

$$\begin{aligned} & \frac{-4 \Delta t D_x}{\Delta^2 x} \sin^2\left(\frac{p \Delta x}{2}\right) - 4 \frac{\Delta t D_y}{\Delta^2 y} \sin^2\left(\frac{q \Delta y}{2}\right) \\ & - i \frac{\Delta t}{\Delta x} V_x \sin p \Delta x - i \frac{\Delta t}{\Delta y} V_y \sin q \Delta y + 1 = N \end{aligned} \quad (9.30)$$

Thus,  $N$  is a complex number so its magnitude can be expressed as

$$\begin{aligned} |N| = & \left[ \left( 1 - 4 \frac{\Delta t D_x}{\Delta^2 x} \sin^2 \frac{p \Delta x}{2} - 4 \frac{\Delta t D_y}{\Delta^2 y} \sin^2 \frac{q \Delta y}{2} \right)^2 \right. \\ & \left. + \left( - \frac{\Delta t}{\Delta x} V_x \sin p \Delta x - \frac{\Delta t}{\Delta y} V_y \sin q \Delta y \right)^2 \right]^{1/2} \end{aligned} \quad (9.31)$$

While it is tedious to do, it can be shown that  $|N| \leq 1$  reduces to the stability condition

$$D_x \frac{\Delta t}{\Delta^2 x} + D_y \frac{\Delta t}{\Delta^2 y} \leq 1/2 \quad (9.32)$$

In general, the lower derivatives, such as  $\frac{\partial u}{\partial x}$ , do not effect the stability condition.

## 10. REFERENCES

Brian, P.L.T. (1961) Dimensional Transient Heat Conduction Problems, *AIChE J.* **7** (3), 367-370.

Peaceman, D.W. and Rachford, Jr. H.H.(1955) The numerical solution of parabolic and elliptic differential equations, *J Society for industrial and applied mathematics*, **3**(1) 28-41.

Price, H.S., Varga, R.S. and Warren, J.E. (1966) Application of oscillation matrices to diffusion-convection equations, *J. of Mathematical Physics*, **45** (3), 301-311.

von Rosenberg, D.U.(1969) *Methods for the numerical solution of partial differential equations*, American Elsevier Publishing Co, New York, NY.



## Appendix A. Tridiagonal Matrix Solution

The tridiagonal matrix is the “heart” of finite difference problems. It occurs in implicit parabolic problems, elliptic line relaxation, and certain types of hyperbolic problems.

The problem is how to solve for the unknowns without performing calculations on the off diagonal zero's. Gaussian elimination generally takes  $\frac{(ST)^3}{3}$  calculations, where S and T are the array dimensions. The tridiagonal matrix solution, or Thomas Algorithm, which is a special case of Gaussian elimination, take much less.

Here's how we start.

At the boundaries we must eliminate two unknowns. We seen unknowns and n-2 equations less than we need. We adopt the notation using  $a_i$  for the  $u_{i-1, n+1}$  terms,  $b_i$  for the  $u_{i, n+1}$  terms,  $c_i$  for the  $u_{i+1, n+1}$  terms, and  $d_i$  for all other terms. We eliminate the inlet and exit boundary terms, which insures that  $a_1$  and  $c_{iR} = 0$ .

For the case of five interior grid points we have

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 \\ 0 & 0 & a_4 & b_4 & c_4 \\ 0 & 0 & 0 & a_5 & b_5 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{bmatrix} \quad (\text{A.1})$$

Now we need to solve for u.

The following Gaussian elimination procedure is suggested.

First perform the forward substitution, calculating intermediate variables b and g for  $i = 2, iR$

$$\beta_i = b_i - \frac{a_i c_{i-1}}{\beta_{i-1}} \quad (\text{A.2})$$

$$\gamma_i = \frac{d_i - a_i \gamma_{i-1}}{\beta_i} \quad (\text{A.3})$$

for  $i = 1$

$$\beta_1 = b_1 \quad (\text{A.4})$$

$$\gamma_1 = \frac{d_1}{b_1} \quad (\text{A.5})$$

Next calculate  $u$  as follows:

$$u_i = \gamma_i - \frac{c_i u_{i+1}}{\beta_i} \quad (\text{A.6})$$

$$u_{iR} = \gamma_{iR} \quad (\text{A.7})$$

The following code solves the tridiagonal matrix resulting from Crank-Nicolson or Backwards Difference. The code is written assuming that the inlet boundary,  $u(1)$  is known and need not be calculated by the routine. The exit boundary point  $u(iR)$  is calculated by the routine. This implementation is most convenient when using a false point for a zero flux exit boundary. Be very careful when using this code or others to solve the tridiagonal or band matrices. The vast majority of programming errors are associated with incorrect handling of the boundary conditions.

The following techniques are available for other frequently occurring band matrices. They are copied from the out of print book, *Numerical Methods for the Solution of Partial Differential Equations*, by Dale U. Von Rosenberg.