

Chapter 2

Preliminaries

The governing differential equations in fluid flow problems contain first and second order partial derivative terms. In order to obtain the solution of these equations, a process called ‘discretization’ must be applied onto these equations. This process involves the approximation of the derivative terms at each point of the solution region, by discrete functional values on and around these points. For this purpose, finite difference approximations of the derivative terms are required. These approximations can be obtained in various ways. One way is to use a quadratic interpolation function. Another way is to utilize Taylor series approximations around the point at which the approximation is made. This requires some knowledge of Taylor series. Utilizing quadratic interpolation is easier for the novice reader, but many are generally accustomed to using Taylor series. Taylor series has an advantage that the truncation error associated with the approximations is readily obtained. Here we will present both methods and leave the choice to the reader.

Moreover, in various stages of the block solutions, certain techniques are required for the solution of linear systems of equations which contain matrices of some special forms.

This chapter is devoted to the presentation of the mathematical preliminaries to be utilized for these purposes.

2.1 Quadratic Interpolation

Suppose we are given three points x_1 , x_2 and x_3 in the x direction, and corresponding discrete functional values for a function $f(x)$ as $f_1 = f(x_1)$, $f_2 = f(x_2)$ and $f_3 = f(x_3)$. If this is the only knowledge about the function $f(x)$, then the Fundamental Theorem of Linear Algebra can be used to pass a parabola of the form

$$f(x) = a + bx + cx^2 \quad (2.1)$$

from all of the three points (x_1, f_1) , (x_2, f_2) and (x_3, f_3) . We can obtain this parabola by substituting these values into Eq. (2.1) as

$$\begin{aligned} f_1 &= a + bx_1 + cx_1^2 \\ f_2 &= a + bx_2 + cx_2^2 \\ f_3 &= a + bx_3 + cx_3^2 \end{aligned} \quad (2.2)$$

These equations are linear in a , b and c , and thus can be easily solved. Solving these equations for a , b and c , replacing in Eq. (2.1) and rearranging terms give

$$f(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}f_1 + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}f_2 + \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}f_3 \quad (2.3)$$

This equation is called the Lagrangian Polynomial. It is of a more suitable form for our purposes in the discretization of the derivatives in the governing differential equations.

Using this polynomial, we will derive expressions for the first and second order derivative terms $\frac{df}{dx}$ and $\frac{d^2f}{dx^2}$ of $f(x)$.

2.2 Approximations Using Lagrangian Polynomial

For notational purposes which will be clear later on in our formulations, we assume a grid arrangement with suitable notations as shown in Fig. 2.1.

The points W , P and E are the points x_1 , x_2 and x_3 at which $f(x)$ is given as $f_1 = f_W$, $f_2 = f_P$ and $f_3 = f_E$, respectively. Points w and e are at midway between the points W and P , and P and E , respectively. Also note that $\delta_w = x_2 - x_1$ and $\delta_e = x_3 - x_2$.

Differentiating $f(x)$ in Eq. (2.3) with respect to x gives

$$\frac{df}{dx} = \frac{(x - x_3) + (x - x_2)}{(x_1 - x_2)(x_1 - x_3)}f_W + \frac{(x - x_3) + (x - x_1)}{(x_2 - x_1)(x_2 - x_3)}f_P + \frac{(x - x_2) + (x - x_1)}{(x_3 - x_1)(x_3 - x_2)}f_E \quad (2.4)$$

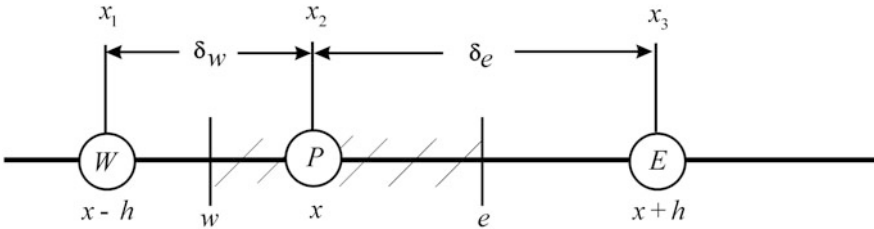


Fig. 2.1 Grid arrangement in the x direction

Now, the value of $\frac{df}{dx}$ at the point $x_2 = P$ can be calculated by replacing x by x_2 in Eq. (2.4). This gives

$$\left. \frac{df}{dx} \right|_P = \frac{(x_2 - x_3) + (x_2 - x_2)}{(x_1 - x_2)(x_1 - x_3)} f_W + \frac{(x_2 - x_3) + (x_2 - x_1)}{(x_2 - x_1)(x_2 - x_3)} f_P + \frac{(x_2 - x_2) + (x_2 - x_1)}{(x_3 - x_1)(x_3 - x_2)} f_E \quad (2.5)$$

Replacing differences in x_i 's with the corresponding δ_i 's, gives

$$\left. \frac{df}{dx} \right|_P = -\frac{\delta_e}{\delta_w(\delta_e + \delta_w)} f_W + \frac{\delta_e - \delta_w}{\delta_e \delta_w} f_P + \frac{\delta_w}{\delta_e(\delta_w + \delta_e)} f_E \quad (2.6)$$

For the second order derivative $\frac{d^2f}{dx^2}$, we differentiate $\frac{df}{dx}$ in Eq. (2.4) with respect to x which gives

$$\begin{aligned} \frac{d^2f}{dx^2} &= \frac{d}{dx} \left(\frac{df}{dx} \right) = \frac{2}{(x_1 - x_2)(x_1 - x_3)} f_W + \frac{2}{(x_2 - x_1)(x_2 - x_3)} f_P \\ &\quad + \frac{2}{(x_3 - x_1)(x_3 - x_2)} f_E \end{aligned} \quad (2.7)$$

$\left. \frac{d^2f}{dx^2} \right|_P$ can now be obtained as

$$\left. \frac{d^2f}{dx^2} \right|_P = \frac{2}{\delta_w(\delta_e + \delta_w)} f_W - \frac{2}{\delta_e \delta_w} f_P + \frac{2}{\delta_e(\delta_w + \delta_e)} f_E \quad (2.8)$$

For a function $f(x)$ which is given continuously over the space x , the expressions given in Eqs. (2.6) and (2.8) for any three consecutive data points W , P and E , are only approximations to the corresponding derivatives. Therefore, their use in the governing differential equations leads only to approximations of the corresponding derivative terms. These approximations are exact only if the function $f(x)$ is a quadratic polynomial function, that is to say, a parabola of the form $f(x) = a + bx + cx^2$. Now, if the function is known to behave in a widely different form, the above approximations will not be appropriate because of the resulting poor approximations. In such a case, a numerical simulation of the unknown function $f(x)$ in which these approximations are used may not only give unreasonable results, but may well lead to unexpected divergence in the numerical solution as well. Therefore, if the behavior of the function is known, or even anticipated, a more suitable form of these approximations should be generated. The approximations will still be a linear combination of the three functional values, but with a slight alteration of the weighting factors. We will address this issue in Chap. 4 for the functions involved in the governing equations.

If we let $\delta_e = \delta_w = h$ in the above formulations, that is to say, if equidistant spacing between the x_i 's is used, then Eqs. (2.6) and (2.8) become

$$\left. \frac{df}{dx} \right|_P = \frac{f_E - f_W}{2h} \quad (2.9)$$

and

$$\left. \frac{d^2f}{dx^2} \right|_P = \frac{f_W - 2f_P + f_E}{h^2} \quad (2.10)$$

respectively. These are the so called 'central difference' approximations for the derivative terms $\frac{df}{dx}$ and $\frac{d^2f}{dx^2}$ at a point P .

In the following chapters, for the sake of generality, we will use the non-equidistant spaced approximations as given in Eqs. (2.6) and (2.8).

2.3 Approximations Using Taylor Series

The Taylor's Theorem states that under certain conditions, the value of a function at a point $x + h$, say $f(x + h)$, can be written in an infinite series form which contains the function and its derivatives at the point x as follows:

$$f(x + h) = f(x) + h \left. \frac{df}{dx} \right|_x + \frac{h^2}{2!} \left. \frac{d^2f}{dx^2} \right|_x + \frac{h^3}{3!} \left. \frac{d^3f}{dx^3} \right|_x + \dots \quad (2.11)$$

If h is small enough and the higher order derivatives are bounded, we can write an approximation to $f(x + h)$ by dropping the terms containing derivatives of order three and higher, as

$$f(x + h) \approx f(x) + h \left. \frac{df}{dx} \right|_x + \frac{h^2}{2!} \left. \frac{d^2f}{dx^2} \right|_x \quad (2.12)$$

If we apply this formula by considering Fig. 2.1, we can write

$$f_E \approx f_P + \delta_e \left. \frac{df}{dx} \right|_P + \frac{\delta_e^2}{2!} \left. \frac{d^2f}{dx^2} \right|_P \quad (2.13)$$

and

$$f_W \approx f_P - \delta_w \left. \frac{df}{dx} \right|_P + \frac{\delta_w^2}{2!} \left. \frac{d^2f}{dx^2} \right|_P \quad (2.14)$$

To obtain an approximation for $\frac{df}{dx}\big|_P$, we can now manipulate Eqs. (2.13) and (2.14) by eliminating the terms $\frac{d^2f}{dx^2}\big|_P$. For this, we multiply Eq. (2.13) by $\frac{2}{\delta_e^2}$ and Eq. (2.14) by $-\frac{2}{\delta_w^2}$ and add the two resulting expressions side by side. We get

$$\frac{2}{\delta_e^2}f_E - \frac{2}{\delta_w^2}f_W \approx \left[\frac{2}{\delta_e^2} - \frac{2}{\delta_w^2}\right]f_P + \left[\frac{2}{\delta_e} + \frac{2}{\delta_w}\right]\frac{df}{dx}\bigg|_P \quad (2.15)$$

Rearranging Eq. (2.15), we can write

$$\frac{\delta_e + \delta_w}{\delta_e \delta_w} \frac{df}{dx}\bigg|_P \approx -\frac{1}{\delta_w^2}f_W + \frac{\delta_e^2 - \delta_w^2}{\delta_e^2 \delta_w^2}f_P + \frac{1}{\delta_e^2}f_E \quad (2.16)$$

from which,

$$\frac{df}{dx}\bigg|_P \approx -\frac{\delta_e}{\delta_w(\delta_e + \delta_w)}f_W + \frac{\delta_e - \delta_w}{\delta_e \delta_w}f_P + \frac{\delta_w}{\delta_e(\delta_w + \delta_e)}f_E \quad (2.17)$$

This is the same as the expression derived in Eq. (2.6).

To obtain the approximation for $\frac{d^2f}{dx^2}\big|_P$, this time we must eliminate the terms $\frac{df}{dx}\big|_P$. For this, we multiply Eq. (2.13) by $\frac{1}{\delta_e}$ and Eq. (2.14) by $\frac{1}{\delta_w}$ and add the two expressions side by side. We get

$$\frac{1}{\delta_e}f_E + \frac{1}{\delta_w}f_W \approx \left[\frac{1}{\delta_e} + \frac{1}{\delta_w}\right]f_P + \left[\frac{\delta_e}{2} + \frac{\delta_w}{2}\right]\frac{d^2f}{dx^2}\bigg|_P \quad (2.18)$$

Rearranging Eq. (2.18) gives

$$\left[\frac{\delta_e + \delta_w}{2}\right]\frac{d^2f}{dx^2}\bigg|_P \approx \frac{1}{\delta_w}f_W - \left[\frac{\delta_e + \delta_w}{\delta_e \delta_w}\right]f_P + \frac{1}{\delta_e}f_E \quad (2.19)$$

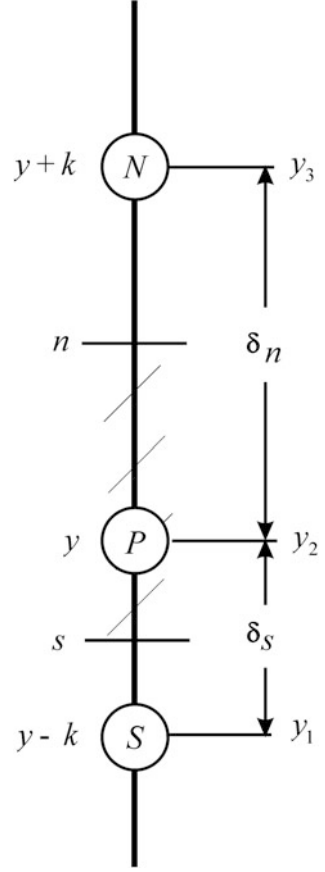
from which

$$\frac{d^2f}{dx^2}\bigg|_P \approx \frac{2}{\delta_w(\delta_e + \delta_w)}f_W - \frac{2}{\delta_e \delta_w}f_P + \frac{2}{\delta_e(\delta_w + \delta_e)}f_E \quad (2.20)$$

which is the same as the expression derived in Eq. (2.8).

For the purposes that will be clear later on, it is beneficial to consider approximations for the above derivatives in the y direction as well. This will aid in our

Fig. 2.2 Grid arrangement in the y direction



notational convention to be used for the partial derivatives in the governing differential equations of our interest.

This time we consider a grid arrangement in the y direction as shown in Fig. 2.2.

Suppose that we are given a function $f(y)$. Now, following similar derivations as shown above, we can write approximations for the derivatives $\frac{df}{dy}$ and $\frac{d^2f}{dy^2}$ at the point P as follows:

$$\left. \frac{df}{dy} \right|_P \approx -\frac{\delta_n}{\delta_s(\delta_n + \delta_s)} f_S + \frac{\delta_n - \delta_s}{\delta_n \delta_s} f_P + \frac{\delta_s}{\delta_n(\delta_s + \delta_n)} f_N \quad (2.21)$$

$$\left. \frac{d^2f}{dy^2} \right|_P \approx \frac{2}{\delta_s(\delta_n + \delta_s)} f_S - \frac{2}{\delta_n \delta_s} f_P + \frac{2}{\delta_n(\delta_s + \delta_n)} f_N \quad (2.22)$$

2.4 General Elimination Technique for Linear Systems

A linear system of equations with N equations and N unknowns can be written as

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N &= b_1 \\
 a_{21}x_1 + a_{22}x_2 + \cdots + a_{2N}x_N &= b_2 \\
 &\vdots \\
 a_{N1}x_1 + a_{N2}x_2 + \cdots + a_{NN}x_N &= b_N
 \end{aligned} \tag{2.23}$$

in which a_{ij} , $i = 1, N$ and $j = 1, N$ are constant coefficients and x_i , $i = 1, N$ are the unknown variables. A solution to this system is the set of x_i values which satisfy all of the equations simultaneously. This system is written in a compact form called a matrix equation as follows:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \tag{2.24}$$

The matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}$$

is called the **coefficient matrix** and

$$A_{aug} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2N} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} & b_N \end{bmatrix}$$

is called the **augmented matrix** of the system.

One of the basic strategies for solving the above system directly is to apply a series of operations, called **elementary row operations** onto the rows of the augmented matrix and update it to an equivalent form that is easier to solve. The

following operations are used for this purpose, each of which does not have any effect on the solution set of the equations:

1. Multiply all entries in a row by a nonzero constant
2. Replace one row by adding to it another row, coefficient by coefficient
3. Interchange the places of two rows

The aim in the first step of the solution process is to eliminate all of the nonzero coefficients in the lower triangular part of the augmented matrix which corresponds to the matrix of the original system. To eliminate the coefficient a_{21} of the augmented matrix, i.e., to make it zero, we multiply row 1 by $-\frac{a_{21}}{a_{11}}$

$$A_{aug'} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} & b_1 \\ 0 & a'_{22} & \cdots & a'_{2N} & b'_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} & b_N \end{bmatrix}$$

and add it to row 2. We get

We note that all the coefficients in row 2 have changed now. Then we proceed to eliminate a_{31} , a_{41}

until a_{N1} , in the same way. At the end of this process, the augmented matrix becomes

$$A_{aug'} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} & b_1 \\ 0 & a'_{22} & \cdots & a'_{2N} & b'_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a'_{N2} & \cdots & a'_{NN} & b'_N \end{bmatrix}$$

We repeat the elimination process for the coefficients a_{32} , a_{42} until a_{N2} . The whole process is continued until the coefficient $a_{N,N-1}$ is eliminated.

The complete process is named as the **forward elimination**, after which we have an **upper triangular matrix**, in which all of the coefficients in the lower triangular part are zeros.

$$A_{aug'} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} & b_1 \\ 0 & a'_{22} & \cdots & a'_{2N} & b'_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a'_{NN} & b'_N \end{bmatrix}$$

Now the second step in the solution process is to obtain the solution to the original system by applying a **back substitution process**. For this, we solve x_N from the N th equation, x_{N-1} from equation $(N - 1)$ etc., until x_1 .

An algorithm for the complete solution process can now be written as follows:
Forward elimination:

```

for k=1,N-1
  for i=k+1,N
    t=a(i,k)/a(k,k)
    a(i,k)=0
    for j=k+1,N
      a(i,j)=a(i,j)-a(k,j)*t
    end j
    b(i)=b(i)-b(k)*t
  end i
end k

```

Backward substitution:

```

x(N)=b(N)/a(N,N)
for i=N-1,1,-1
  sum=0
  for j=i+1,N
    sum=sum+a(i,j)*x(j)
  end j
  x(i)=(b(i)-sum)/a(i,i)
end i

```

The process described is named as the **Gauss Elimination method** in the literature.

We note that the third elementary row operation has not been utilized in this process. If, however, any diagonal coefficient a_{ii} becomes zero at any stage of the forward elimination process, row i must be interchanged by another row having a nonzero coefficient on the main diagonal. Otherwise, a division by zero will occur while the back substitution process is performed. Luckily enough, as we will see later on, the matrix systems that are involved in incompressible fluid flow solutions do not pose this problem, if of course, the systems are formed properly. Hence this issue will not be elaborated any further.

2.5 Solution Techniques for Special Linear Systems

Within the block solution procedures described later, various linear sub-systems whose matrices are of a certain special form need to be solved. Therefore, a brief review of the techniques used to solve such systems is appropriate.

Type 1: Lower triangular systems with two off-diagonal vectors

Suppose we have a system of linear equations given as

$$\begin{bmatrix} c_1 & & & & & & \\ b_2 & c_2 & & & & & \\ & b_3 & c_3 & & & & \\ & & \cdot & \cdot & & & \\ a_{m+1} & & & \cdot & \cdot & & \\ & \cdot & & & \cdot & \cdot & \\ & & \cdot & & & \cdot & \cdot \\ & & & a_N & & b_3 & c_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ r_N \end{bmatrix}$$

This matrix contains only two vectors in the lower triangular part and the upper triangular part is completely zero. One of the vectors is just below the main diagonal vector, and the other is far below by m columns. All other coefficients in the lower diagonal part are zeros. Here it is possible to store the complete matrix by utilizing only three vectors in computer's memory.

The system can be solved easily by a forward substitution sweep using the following algorithm:

```

x(1)=r(1)/c(1)
for i=2,m
    x(i)=(r(i)-b(i)*x(i-1))/c(i)
end i
for i=m+1,N
    x(i)=(r(i)-a(i)*x(i-m)-b(i)*x(i-1))/c(i)
end i

```

Type 2: Upper triangular systems with two off-diagonal vectors

In this case we consider the following linear system of equations:

$$\begin{bmatrix} c_1 & d_1 & & e_1 & & & \\ & c_2 & d_2 & & \cdot & & \\ & & c_3 & d_3 & & \cdot & \\ & & & \cdot & \cdot & & e_{N-m} \\ & & & & \cdot & \cdot & \\ & & & & & \cdot & \cdot \\ & & & & & & d_{N-1} \\ & & & & & & c_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ r_N \end{bmatrix}$$

Here we have only three vectors in the upper triangular part. All other coefficients, including those at the lower triangular part are zeros. For this system, a backward substitution sweep is applied, using the following algorithm:

```

x(N)=r(N)/c(N)
for i=N-1,N-m+1,-1
    x(i)=(r(i)-d(i)*x(i+1))/c(i)
end i
for i=N-m,1,-1
    x(i)=(r(i)-d(i)*x(i+1)-e(i)*x(i+m))/c(i)
end i

```

Type 3: A tri-diagonal system

Suppose we are given the following system which contains a diagonal vector and two off-diagonal vectors just below and above this diagonal vector as follows:

$$\begin{bmatrix} b_1 & c_1 & & & & & & & \\ a_2 & b_2 & c_2 & & & & & & \\ & a_3 & b_3 & c_3 & & & & & \\ & & \cdot & \cdot & \cdot & & & & \\ & & & \cdot & \cdot & \cdot & & & \\ & & & & \cdot & \cdot & \cdot & & \\ & & & & & \cdot & \cdot & c_N & \\ & & & & & a_N & b_N & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ r_N \end{bmatrix}$$

This system can be solved by a forward sweep to eliminate the coefficients a_i , followed by a backward sweep to obtain the solution. An algorithm similar to the one used in the Gaussian Elimination method can be used here. Application of the usual algorithm as given above however, would change the values of the coefficients of the vectors. But for reasons that will be clear later on, we do not want to spoil the original values of the coefficients of the three vectors. We can achieve this by not actually updating the a_i 's and utilizing an auxiliary vector t for keeping the updated versions of the b_i 's. The c_i 's do not change, so they are kept untouched. The following algorithm performs the forward elimination and backward substitution processes.

```

t(1)=b(1)
for i=2,N
    q=a(i)/t(i-1)
    t(i)=b(i)-c(i-1)*q
    r(i)=r(i)-r(i-1)*q
end i
x(N)=r(N)/t(N)
for i=N-1,1,-1
    x(i)=(r(i)-c(i)*x(i+1))/t(i)
end i

```

Type 4: Lower triangular systems with three off-diagonal vectors

Suppose we are given the system

$$\begin{bmatrix} c_1 & & & & & & & & & & \\ b_2 & c_2 & & & & & & & & & \\ & b_3 & c_3 & & & & & & & & \\ e_m & & & \cdot & \cdot & & & & & & \\ a_{m+1} & e_{m+1} & & & \cdot & \cdot & & & & & \\ & \cdot & & & \cdot & \cdot & \cdot & & & & \\ & & & \cdot & \cdot & \cdot & \cdot & & & & \\ & & & & a_N & e_N & & b_3 & c_N & & \\ & & & & & & & & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ r_N \end{bmatrix}$$

In this case the matrix is similar to that given in Type 1, but it contains an additional vector just above the lower far-diagonal vector. This system can be solved by a forward sweep using the following algorithm:

```

x(1)=r(1)/c(1)
for i=2,m-1
    x(i)=(r(i)-b(i)*x(i-1))/c(i)
end i
x(m)=(r(m)-e(m)*x(i-m+1)-b(i)*x(i-1))/c(i)
for i=m+1,N
    x(i)=(r(i)-a(i)*x(i-m)-e(i)*x(i-m+1)-b(i)*x(i-1))/c(i)
end i

```

Type 5: Upper triangular systems with three off-diagonal vectors

The system to be solved is given by

$$\begin{bmatrix} c_1 & d_1 & y_1 & e_1 & & & & & & & \\ & c_2 & d_2 & & \cdot & & & & & & \\ & & c_3 & d_3 & & \cdot & \cdot & & & & \\ & & & \cdot & \cdot & & \cdot & e_{N-m} & & & \\ & & & & \cdot & \cdot & & y_{N-m+1} & & & \\ & & & & & \cdot & \cdot & & & & \\ & & & & & & \cdot & & & & \\ & & & & & & & d_{N-1} & & & \\ & & & & & & & c_N & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ r_N \end{bmatrix}$$

In this case, the coefficient matrix is similar to that in Type 2, but it contains an additional vector just below the far-diagonal vector. The algorithm given below uses a backward substitution sweep to solve this system:

```

x(N)=r(N)/c(N)
for i=N-1,N-m+2,-1
    x(i)=(r(i)-d(i)*x(i+1)-y(i)*x(i+m-1))/c(i)
end i
i=N-m+1
x(i)=(r(i)-d(i)*x(i+1)-y(i)*x(i+m-1))/c(i)
for i=N-m,I,-1
    x(i)=(r(i)-d(i)*x(i+1)-y(i)*x(i+m-1)-e(i)*x(i+m))/c(i)
end i

```

Fully Implicit, Coupled Procedures in Computational
Fluid Dynamics

An Engineer's Resource Book

Mazhar, Z.

2016, XVI, 173 p. 50 illus., Hardcover

ISBN: 978-3-319-29894-8