TRI-DIAGONAL MATRIX ALGORITHM (TDMA)

In general, either ordinary (ODE) or partial differential equations (PDE) are used to model heat transfer and fluid flow problems. These model equations are coupled and/or non-linear thus, precluding the possibility of using a closed form solution technique. This is the motivation for resorting to a numerical solution procedure. A numerical procedure has two main steps. The first step is to convert a given ODE/PDE into a system of linear algebraic equations and the second step is to solve these sets of linear algebraic equations. These algebraic equations are often referred to as the discretization equations. The focus of this note is the second step involving the solution to a set of algebraic equations.

In the area of heat transfer and fluid flow, the discretization equations for one-dimensional problems display the tri-diagonal pattern. Two and three-dimensional problems display penta and hepta diagonal patterns, respectively. In a system of tri-diagonal equations, only the main diagonal elements and its two neighbors are non-zero. The objective is to exploit this pattern by storing only non-zero elements which results in significant reduction in memory requirement. The discretization equations for two and three dimensional problems are solved by sweeping each plane in one direction at a time. Thus, both penta and hepta diagonal systems of equations can be split into several tri-diagonal systems and therefore, it is important to learn how to solve a tri-diagonal system effectively.

DEVELOPMENT OF TRI-DIAGONAL MATRIX/THOMAS ALGORITHM

The tri-diagonal system of equations can be solved by the standard Gauss-Elimination method. Because of the simple and special pattern of the equations, the elimination process turns into a convenient and efficient algorithm. Such an algorithm is referred to as the Tri-Diagonal matrix Algorithm (TDMA) or the Thomas algorithm. Consider a one-dimensional discretization domain as shown below:

X	X	X	X	X	(N=5)
1	2	3	4	5	

The discretization equation for a field variable (φ) at a node I for a tri-diagonal system is:

$$a_I \varphi_I = B_I \varphi_{I+1} + C_I \varphi_{I-1} + d_I \qquad \mathbf{0}$$

here I =1 and I = N are the boundary points. If the boundary condition is of the Dirichlet type, φ_1 and φ_N are known. On the other hand, if the boundary conditions are of the second or third kind, then φ_1 and φ_N have to be determined as a part of the solution. From Equation 1 for I = 1, φ_o has no meaningful role to play. Similarly, for I = N, φ_{N+1} has no meaningful role and these situations can be taken care of by setting $c_1 = b_N = 0$. If φ_1 is given $a_1 = 1$, $b_1 = c_1 = 0$, and $d_1 = \varphi_1$ (given the value of φ_1).

The development of TDMA will be illustrated for the situation where φ_1 and φ_N are not known a priori.

I = 1, yields:

$$a_1\ \varphi_1 = b_1\ \varphi_2 + d_1 \quad (Note\ c_1 = 0)$$

$$\vee \quad \varphi_1 = f_1\ (\varphi_2\ )$$

I = 2, yields:

$$a_2\ \varphi_2 = b_2\ \varphi_3 + c_2\ \varphi_1 + d_2$$

From the expression for I = 1, $\varphi_1 = f_1\ (\varphi_2\ )$0 therefore, $\varphi_2 = f_2\ (\varphi_3\ )$0. Similarly, for I = 3, 4 and 5 leads to:

$$\varphi_3 = f_3\ (\varphi_4\ )$$

$$\varphi_4 = f_4\ (\varphi_5\ )$$

$$\varphi_5 = f_5\ (\varphi_6\ )$$

In general, if there are N points in the domain, then:

$$\varphi_N = f_N\ (\varphi_{N+1}\ )$$

This completes the process of forward elimination. Recall that $\ddot{o}_{N+1}$ has no meaningful existence hence, the value of $\ddot{o}_N$ can be obtained from f_N. Once $\ddot{o}_N$ is obtained, $\ddot{o}_{N-1}$, $\ddot{o}_{N-2}$, ...$\ddot{o}_1$ can be obtained by back substitution using f_{N-1}, f_{N-2}, ...f_1.

The algorithm will be developed following the discussion of Patankar (1980). The forward substitution expression can be written as:

$$\varphi_I = P_I\ \varphi_{I+1} + Q_I \qquad (2)$$

replacing I with I - 1

$$\varphi_{I\text{-}1} = P_{I\text{-}1}\ \varphi_I + Q_{I\text{-}1} \qquad (3)$$

and substituting Equation 3 into Equation 1 we get:

$$\varphi_I = \frac{b_I}{(a_I - P_{I\text{-}1}\, c_I\,)}\, \varphi_{I+1} + \frac{Q_{I\text{-}1}\, c_I + d_I}{(a_I - P_{I\text{-}1}\, c_I\,)} \tag{4}$$

comparing Equation 4 and Equation 2 we get:

$$P_I = \frac{b_I}{a_I - c_I\, P_{I\text{-}1}}$$

$$Q_I = \frac{d_I + c_I\, Q_{I\text{-}1}}{a_I - c_I\, P_{I\text{-}1}} \tag{5}$$

Note that the expressions for P_I and Q_I are recursive in nature.

For I = 1:

$$P_I = \frac{b_1}{a_1} \div Q_1 = \frac{d_1}{a_1}$$

On the other hand, for I = N, and $P_N = 0$ from Equation 5, one gets:

$$\varphi_N = Q_N \tag{6}$$

Thus, once $\ddot{o}_N$ is established, the back substitution can be started.

COMPUTATIONAL SEQUENCE

1. Calculate P_1 and Q_1
2. Calculate P_I and Q_I, I = 2, 3, 4,....N
3. Set $\ddot{o}_N = Q_N$
4. Using Equation 2, calculate $\ddot{o}_{N-1}$, $\ddot{o}_{N-2}$, ... for I = N-1, N-2 ... 1.

SUBROUTINE FOR TDMA

In order to minimize the memory requirement, Equation 1 will be written as

$$A(I,1)\varphi_{I-1} + A(I,2)\varphi_I + A(I,3)\varphi_{I+1} = B(I) \qquad (7)$$

Now, comparing Equations 7 and 1, we have:

$$c_I = -A(I,1)$$

$$a_I = A(I,2)$$

$$b_I = -A(I,3)$$

$$d_I = B(I)$$

Using the arrays, A and B, the following TDMA subroutine is written.

```
      SUBROUTINE THOMAS (N, A, B)
cc    Algorithm to solve a tri-diagonal system
cc    N:  Number of Equations
cc    B:  Solution Vector
cc    Written by:  N.K. Anand
      DIMENSION A(101,3),B(101)
      A(1,3) = -A(1,3) / A(1,2)
      B(1) = B(1) / A(1,2)
      DO I = 2,N
      A(I,3) = -A(I,3) / (A(I,2) + A(I,1) * A(I-1,3))
      B(I) = (B(I)-A(I,1)*B(I-1)) / (A(I,2) + A(I,1)*A(I-1,3))
      END DO
      DO I = N-1, 1, -1
      B(I) = A(I,3)*B(I+1)+B(I)
      END DO
      RETURN
      END
```

REFERENCE

Patankar, S.V., (1980), "Numerical Heat Transfer and Fluid Flow," McGraw Hill Book Co.

N.K. Anand