The Block TriDiagonal Matrix Algorithm (BTDMA)

The BTDMA is a matrix version of the TDMA, and therefore many of the details of its algorithm are very similar to the TDMA. On the other hand, it is more complicated to encode and therefore I shall be providing the appropriate subroutines for any assignments in which it is to be used.

Let us write the Finite Difference equations in the form:

$$\begin{pmatrix} B_1 & C_1 & & & & \\ A_2 & B_2 & C_2 & & & \\ & A_3 & B_3 & C_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & A_{M-1} & B_{M-1} & C_{M-1} \\ & & & & A_M & B_M \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_{M-1} \\ Y_M \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_{M-1} \\ R_M \end{pmatrix}.$$

where A, B, and C now represent square matrices (— 3×3 matrices for the problem we are considering), and both Y and R are column vectors (— 3×1 matrices).

1. Reduce to bidiagonal form

The first step is to reduce the matrix to block bidiagonal form by eliminating the subdiagonal blocks.

The A_2 block is removed in two stages: first premultiply the row 1 by B_1^{-1} and then subtract $A_2 \times$ row 1 from the row 2. The first stage yields the system,

$$\begin{pmatrix} I & C_1^* & & & & \\ A_2 & B_2 & C_2 & & & \\ & A_3 & B_3 & C_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & A_{M-1} & B_{M-1} & C_{M-1} \\ & & & & A_M & B_M \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_{M-1} \\ Y_M \end{pmatrix} = \begin{pmatrix} R_1^* \\ R_2 \\ R_3 \\ \vdots \\ R_{M-1} \\ R_M \end{pmatrix},$$

where $B_1 C_1^* = C_1$ and $B_1 R_1^* = R_1$, or, in augmented matrix notation,

$$B_1 \times (C_1^* | R_1^*) = (C_1 | R_1);$$

here the matrices given in brackets are 3×4 (3 rows and 4 columns) for our example problem. The matrix C_1^* and vector R_1^* are now obtained using a standard Gaussian elimination subroutine — this is generally more efficient than computing B_1^{-1} explicitly and then premultiplying C_1 and R_1.

Now we can remove the A_2 block by subtracting $A_2 \times$ row 1 from row 2. Hence we obtain

$$\begin{pmatrix} I & C_1^* & & & & \\ 0 & B_2^* & C_2 & & & \\ & A_3 & B_3 & C_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & A_{M-1} & B_{M-1} & C_{M-1} \\ & & & & A_M & B_M \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_{M-1} \\ Y_M \end{pmatrix} = \begin{pmatrix} R_1^* \\ R_2^* \\ R_3 \\ \vdots \\ R_{M-1} \\ R_M \end{pmatrix},$$

where $B_2^* = B_2 - A_2^* C_1^*$ and $R_2^* = R_2 - A_2^* R_1^*$. Again, this form can be expressed using augmented matrix notation:

$$(B_2^* | R_2^*) = (B_2 | R_2) - A_2^* (C_1^* | R_1^*)$$

.

The two–stage elimination process is now used to remove the A_3 block. We get

$$\begin{pmatrix} I & C_1^* & & & & & \\ 0 & I & C_2^* & & & & \\ & 0 & B_3^* & C_3 & & & \\ & & A_4 & B_4 & C_4 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & A_{M-1} & B_{M-1} & C_{M-1} \\ & & & & & A_M & B_M \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ \vdots \\ Y_{M-1} \\ Y_M \end{pmatrix} = \begin{pmatrix} R_1^* \\ R_2^{**} \\ R_3^* \\ R_4 \\ \vdots \\ R_{M-1} \\ R_M \end{pmatrix},$$

where $B_2^* C_2^* = C_2$, $B_2^* R_2^{**} = R_2^*$, $B_3^* = B_3 - A_3^* C_2^*$ and $R_3^* = R_3 - A_3^* R_2^{**}$. In augmented matrix notation this is expressed more simply as

$$B_2^* \times (C_2^* | R_2^{**}) = (C_2 | R_2^*) \qquad \text{and} \qquad (B_3^* | R_3^*) = (B_3 | R_3) - A_3^* (C_2^* | R_2^{**}),$$

where our objective is to obtain C_2^*, R_2^{**}, B_3^* and R_3^*, in that order.

The process of block subdiagonal elimination is continued until we obtain the block bidiagonal set of equations,

$$\begin{pmatrix} I & C_1 & & & & \\ & I & C_2 & & & \\ & & I & C_3 & & \\ & & & \ddots & \ddots & \\ & & & & I & C_{M-1} \\ & & & & & B_M \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_{M-1} \\ Y_M \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_{M-1} \\ R_M \end{pmatrix},$$

where the superscript asterisks have been omitted.

2. Solution by back–substitution.

The values of Y_i $(i = 1, M)$ are now computed by back–substitution. The last matrix row is equivalent to the equation,

$$B_M Y_M = R_M,$$

which is solved using the Gaussian Elimination subroutine.

The penultimate row gives the equation

$$B_{M-1} Y_{M-1} = R_{M-1} - C_{M-1} Y_M,$$

where the right hand side is known. Again we use Gaussian Elimination to find Y_{M-1}.

The $(M-2)^{\text{nd}}$ row gives

$$B_{M-2} Y_{M-2} = R_{M-2} - C_{M-2} Y_{M-1},$$

to get Y_{M-2}, and so on, until we eventually get to the equation,

$$B_1 Y_1 = R_1 - C_1 Y_2,$$

for Y_1.

3. Writing a Fortran subroutine.

As for the TDMA development, we shall write the above algorithm in as compact a form as is possible. Hence we obtain the following algorithm:

```
For i=2,M:
        solve $B_i \times (C_i|R_i)^{\text{new}} = (C_i|R_i)^{\text{old}}$
        compute $(B_i|R_i)^{\text{new}} = (B_i|R_i)^{\text{old}} - A_i^*(C_{i-1}|R_{i-1})$
Solve: $B_M Y_M = R_M$ to obtain $Y_M$
For i=M-1,1 solve:
        $B_i Y_i = (R_i - C_i Y_{i+1})$ to obtain $Y_i$
```