

287

DUPLICATE ALSO



# Forecasting Research

Met O 11 Technical Note No. 8

Comparison of Algorithms for the solution of  
Cyclic, Block, Tridiagonal System

by

M.H. Mawson

May 1988

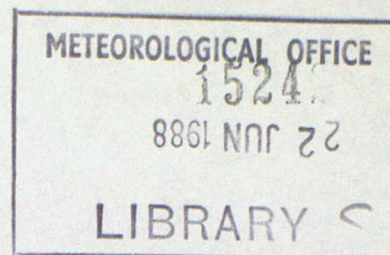
ORGS UKMO M

Me  
London Road, National Meteorological Library (O 11)  
FitzRoy Road, Exeter, Devon. EX1 3PB G12 2SZ, England

FH2A



DUPLICATE ALSO



MET O 11 TECHNICAL NOTE No 8

Comparison of Algorithms for the solution of

Cyclic, Block, Tridiagonal Systems.

M. H. Mawson

LONDON, METEOROLOGICAL OFFICE.  
Met.O.11 Technical Note (New Series) No.8

Comparison of algorithms for the solution of  
cyclic, block, tridiagonal systems.

08170688

FH2A



## Section 1. Introduction.

The object of this note is to describe three schemes for solving cyclic block tridiagonal systems and compare them. A comparison of schemes for cyclic tridiagonal matrices was done by Temperton (1975).

## Section 2. Formulation of the Problem.

If we consider a 2-dimensional grid then we seek to solve equations of the form

$$a_0 \theta_{i,j} + a_1 \theta_{i-1,j} + a_2 \theta_{i+1,j} + a_3 \theta_{i,j-1} + a_4 \theta_{i,j+1} = R_{i,j} \quad (1)$$

where the  $a_i$ 's are variable coefficients,  $\theta_{i,j}$  is the value of  $\theta$  at point  $(i,j)$  of the grid. The first co-ordinate is periodic with values from 1 to  $M$ , the second co-ordinate has values from 1 to  $N$  and indices out of this range are ignored. Writing this in matrix form we have

$$C_1 \underline{\theta}_{i-1} + A_1 \underline{\theta}_i + B_1 \underline{\theta}_{i+1} = \underline{R}_i$$

where

$$\underline{\theta}_i = \begin{bmatrix} \theta_{i,1} \\ \theta_{i,2} \\ \vdots \\ \theta_{i,N} \end{bmatrix}, \quad \underline{R}_i = \begin{bmatrix} R_{i,1} \\ R_{i,2} \\ \vdots \\ R_{i,N} \end{bmatrix}$$

$A$  is a tridiagonal matrix,  $B$  and  $C$  are diagonal matrices.



Thus the whole system can be written

$$E \underline{\theta} = \underline{R} \quad \text{where}$$

$$E = \begin{array}{cccccccccccc} & i=1 & i=2 & i=3 & . & . & . & . & . & . & i=M-1 & i=M \\ \begin{array}{l} E = \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} & A_1 & B_1 & 0 & . & . & . & . & . & . & 0 & C_1 \\ & C_2 & A_2 & B_2 & 0 & . & . & . & . & . & . & 0 \\ & 0 & C_3 & A_3 & B_3 & 0 & . & . & . & . & . & 0 \\ & . & & & & & & & & & & . \\ & . & & & & & & & & & & . \\ & . & & & & & & & & & & . \\ & 0 & . & . & . & . & . & . & 0 & C_{M-1} & A_{M-1} & B_{M-1} \\ & B_M & 0 & . & . & . & . & . & . & 0 & C_M & A_M \end{array}$$

where 0 denotes the zero matrix



$$\begin{array}{ccc} \underline{\theta} = \underline{\theta}_1 & , & \underline{R} = \underline{R}_1 \\ & & \underline{R}_2 \\ & & \cdot \\ & & \cdot \\ & & \cdot \\ \underline{\theta}_M & & \underline{R}_M \end{array}$$

In this paper we consider the case where in general there is no symmetry

$$\text{i.e. } C_{i+1} \neq B_i, \quad A_i^T \neq A_i$$

We also assume a unique solution exists, which is true for diagonally dominant E. The three methods are as follows :

### Section 3. The Three Algorithms.

#### 1) Ahlberg-Nilson-Walsh Algorithm.

Details of this algorithm are given (for a general cyclic tridiagonal matrix) in Ahlberg, Nilson and Walsh (1967). Here it is applied to the block, tridiagonal cyclic case.

If we split the system as follows:

$$\begin{array}{ccc} E = \begin{bmatrix} E_1 & F \\ G^T & E_2 \end{bmatrix} , & \underline{\theta} = \begin{bmatrix} \underline{\theta}_1 \\ \underline{\theta}_M \end{bmatrix} , & \underline{R} = \begin{bmatrix} \underline{R}_1 \\ \underline{R}_M \end{bmatrix} \end{array}$$



where

$$\begin{array}{cccccccc}
 E_1 = & A_1 & B_1 & 0 & . & . & . & 0 \\
 & C_2 & A_2 & B_2 & 0 & . & . & 0 \\
 & . & . & . & . & . & . & . \\
 & . & . & . & . & . & . & . \\
 & . & . & . & . & . & . & . \\
 & 0 & . & . & . & C_{M-2} & A_{M-2} & B_{M-2} \\
 & 0 & . & . & . & 0 & C_{M-1} & A_{M-1}
 \end{array} , \quad F = \begin{array}{c} C_1 \\ 0 \\ . \\ . \\ . \\ 0 \\ B_{M-1} \end{array}$$

$$\begin{array}{ccc}
 \underline{\theta}^1 = \underline{\theta}_1 & , & \underline{R}^1 = \underline{R}_1 \\
 \underline{\theta}_2 & & \underline{R}_2 \\
 . & & . \\
 . & & . \\
 . & & 0 \\
 \underline{\theta}_{M-1} & & \underline{R}_{M-1}
 \end{array} , \quad G = B_M , \quad E_2 = A_M$$

Then we can write the system  $E\theta = R$  as

$$1. E_1 \underline{\theta}^1 + F \underline{\theta}_M = \underline{R}^1$$

$$2. G^T \underline{\theta}^1 + E_2 \underline{\theta}_M = \underline{R}_M$$

Then writing  $\underline{V} = E_1^{-1} \underline{R}^1$  and  $U = E_1^{-1} F$

1. becomes

$$3. \underline{\theta}^1 = \underline{V} - U \underline{\theta}_M$$

substituting in 2 yields

$$4. (E_1 - F^T U) \underline{\theta}_M = \underline{R}_M - F^T \underline{V}$$

Equation 4 can be solved for  $\underline{\theta}_M$  and then  $\underline{\theta}^1$  can be found from 3 by using  $\underline{\theta}_M$ .

The algorithm is therefore ...



1. Reduce  $E_1$  to Upper Diagonal Form (UDF) updating  $\underline{R}^1$  and  $F$ . This is done in the same way as the direct algorithm for reduction of  $E$  but removes the need to eliminate the matrices  $B_M$  and  $B_M^1$ .

2. Back-substitute to find  $\underline{V}$  and  $U$  using the equations

$$E_1 \underline{V} = \underline{R}^1 \quad \text{and} \quad E_1 U = F.$$

3. Form  $E_1 - F^T U$  and  $\underline{R}_M - F^T \underline{V}$

4. Reduce  $E_1 - F^T U$  to UDF updating  $\underline{R}_M - F^T \underline{V}$

5. Back-substitute using equation 4 to find  $\underline{\theta}_M$

6. Then solve equation 3 to find  $\underline{\theta}^1$

#### 11). Block Algorithm

This is a block implementation of the Ahlberg-Nilson-Walsh algorithm which takes advantage of the block tri-diagonal structure but without assuming any particular structure for the matrices  $A_i$ ,  $B_i$  and  $C_i$ . Using the previous notation and with  $\underline{V}$  similar to  $\underline{V}$  and  $D$  similar to  $U$  in the notation of 2(i) the procedure is as follows ...

1. Forward sweep.

$$A_1 = A_1, \quad G_1 = C_1, \quad \underline{R}_1 = \underline{R}_1$$

$$G_i = 0 \quad 2 \leq i \leq M-2$$

$$G_{M-1} = B_{M-1}$$

$$A_{i+1} = A_{i+1} - C_{i+1} A_i^{-1} B_i \quad 1 \leq i \leq M-2$$

$$\underline{R}_{i+1} = \underline{R}_{i+1} - C_{i+1} A_i^{-1} \underline{R}_i \quad 1 \leq i \leq M-2$$

$$G_{i+1} = G_{i+1} - C_{i+1} A_i^{-1} G_i \quad 1 \leq i \leq M-2$$



2. Solution of equations at end point.

$$\underline{V}_{M-1} = \underline{A}_{M-1}^{-1} \underline{R}_{M-1}$$

$$\underline{D}_{M-1} = \underline{A}_{M-1}^{-1} \underline{G}_{M-1}$$

3. Back-substitute

$$\underline{V}_i = \underline{A}_i^{-1} (\underline{R}_i - \underline{B}_i \underline{V}_{i+1}) \quad M-2 \geq i \geq 1$$

$$\underline{D}_i = \underline{A}_i^{-1} (\underline{G}_i - \underline{B}_i \underline{D}_{i+1}) \quad M-2 \geq i \geq 1$$

4. Solve for Mth point.

$$(\underline{A}_M - \underline{B}_M \underline{D}_1 - \underline{C}_M \underline{D}_{M-1}) \underline{\theta}_M = \underline{R}_M - \underline{B}_M \underline{V}_1 - \underline{C}_M \underline{V}_{M-1}$$

5. Correct other points.

$$\underline{\theta}_i = \underline{V}_i - \underline{D}_i \underline{\theta}_M \quad 1 \leq i \leq M-1$$

#### iii) Direct Method.

The idea is simply to reduce E to UDF and then back-substitute to find  $\underline{\theta}$ .

With the same notation as before the algorithm is as follows:-

1. Reduce  $\underline{A}_1$  to UDF and update  $\underline{B}_1$ ,  $\underline{C}_1$  and  $\underline{R}_1$ .

Note: This changes  $\underline{B}_1$  and  $\underline{C}_1$  from diagonal matrices to lower diagonal form matrices.



2. Remove  $C_2$  using  $A_1$  and update  $A_2$ ,  $R_2$  and create new matrix  $C_2'$  in the last column of the second row ie:

$$\begin{array}{cccccccc}
 A_1 & B_1 & 0 & . & . & . & . & 0 & C_1 \\
 0 & A_2 & B_2 & 0 & . & . & . & 0 & C_2' \\
 0 & C_3 & A_3 & B_3 & . & . & . & 0 & 0 \\
 . & . & . & . & . & . & . & . & . \\
 . & . & . & . & . & . & . & . & . \\
 . & . & . & . & . & . & . & . & .
 \end{array}$$

Note :  $C_2'$  is of lower triangular form and  $A_2$  is now of the form

$$A_2 = \begin{array}{ccccccc}
 x & x & & & & & \\
 x & x & x & & & & \\
 x & x & x & x & & & \\
 . & . & . & . & . & . & 0 \\
 . & . & . & . & . & . & \\
 . & . & . & . & . & . & \\
 . & . & . & . & . & . & x & x \\
 x & . & . & . & . & . & x & x
 \end{array}$$

ie: lower triangular with one super-diagonal.

3. Remove  $B_m$  using  $A_1$  and update  $A_m$ ,  $R_m$  and create new matrix  $B_m'$  in last row, second column. Note:  $B_m'$  is lower triangular,  $A_m$  is now of the same form as  $A_2$ . So E after step 3 looks like



$$\begin{array}{cccccccc}
A_1 & B_1 & 0 & . & . & . & . & 0 & C_1 \\
0 & A_2 & B_2 & 0 & . & . & . & 0 & C_2' \\
0 & C_3 & A_3 & B_3 & . & . & . & 0 & 0 \\
. & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . \\
0 & . & . & . & . & . & 0 & C_{M-1} & A_{M-1} & B_{M-1} \\
0 & B_M' & 0 & . & . & . & . & 0 & C_M & A_M
\end{array}$$

For  $i = 2$  until  $i = M-2$  repeat steps 4, 5, 6.

4. Reduce  $A_i$  to UDF and update  $B_i$ ,  $C_i'$ ,  $\underline{R}_i$ .

Note: When  $i=2$  this makes  $B_i$ ,  $C_i'$  full matrices.

5. Remove  $C_{i+1}$  using  $A_i$ , update  $A_{i+1}$ ,  $\underline{R}_{i+1}$  and create  $C_{i+1}'$  in last column, row  $i+1$ .

Note:  $C_{i+1}'$  is a full matrix,  $A_{i+1}$  becomes a full matrix when  $i=2$ .

6. Remove  $B_{i+1}'$  using  $A_i$ , update  $A_{i+1}$ ,  $\underline{R}_{i+1}$  and create new matrix  $B_{i+1}'$  in last row, column  $i+1$ .

Note:  $B_{i+1}'$  is a full matrix and  $A_{i+1}$  becomes a full matrix when  $i=2$ .

When  $i = M-2$  the newly generated matrices  $C_{M-1}'$  and  $B_M'$  have to be added to  $B_{M-1}$  and  $C_M$  respectively which occupy the positions of the newly generated matrices in E. So

$$7. B_M' = B_{M-1}' + C_M$$

$$C_{M-1}' = C_{M-1}' + B_{M-1}$$



ie: E looks like

$$\begin{array}{cccccccc}
 A_1 & B_1 & 0 & . & . & . & . & 0 & C_1 \\
 0 & A_{22} & B_{22} & 0 & . & . & . & 0 & C_{22}' \\
 . & & & & & & & & . \\
 . & & & & & & & & . \\
 . & & & & & & & & . \\
 0 & . & . & . & . & . & . & 0 & A_{M-1} & B_{M-1} + C_{M-1}' \\
 0 & . & . & . & . & . & . & 0 & C_M + B_M' & A_M
 \end{array}$$

8. Reduce  $A_{M-1}$  to UDF, updating  $C_{M-1}'$ ,  $R_{M-1}$ .
9. Remove  $B_M'$  using  $A_{M-1}$  and update  $A_M$ ,  $R_M$ .
10. Reduce  $A_M$  to UDF, update  $R_M$ .
11. Back-substitute to find  $\theta$ .

### Section 3. Results.

The three algorithms were all programmed with equal care in Fortran and run in single precision on an IBM 3081 computer. The values assigned to M and N were 400 and 15 respectively. The cost of each algorithm is expressed in terms of accounting units used which is proportional to the time taken for the program execution.

<u>Algorithm.</u>	<u>Units.</u>
Block Algorithm	43.67
Direct Method	51.09
Ahlberg-Nilson-Walsh	37.53



#### Section 4. Conclusion.

The Direct method, although very simple, is slow due to the creation of full matrices in the last row when reducing  $E$  to UDF. The use of the Ahlberg-Nilson-Walsh algorithm removes this problem but at the expense of more back-substitution. Since back-substitution is comparatively inexpensive this algorithm is considerable cheaper. The Block algorithm is slightly more expensive but this difference is less noticable when a more complicated form is given to the  $B_i$ 's and  $C_i$ 's, for example if they are tridiagonal.

#### References.

H.H., Ahlberg, E.N., Nilson, J.L., Walsh.

"The Theory of Splines and their Applications."

Academic Press, 1967.

C., Temperton.

Algorithms for the Solution of Cyclic Tridaigonal Systems.

J. Comp. Phys. 19 pp317-323. 1975.