MASTER

C00-2200-4

23,843

PERIODIC BAND MATRIX SOLUTION

Technical Report

W. P. Gibbons

Northwestern University

Evanston, Illinois

September 30, 1973

Prepared for the U. S. Atomic Energy Commission

Under Contract No. AT(11-1)-2200

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

89

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

PERIODIC BAND MATRIX SOLUTION

by

William P. Gibbons

Computer Sciences Department
Northwestern University
Evanston, Illinois 60201

Description

The purpose of the routine BCMAT presented here is to solve a system of linear algebraic equations with a coefficient matrix of the form shown in Fig. 1a, defined here as a periodic band matrix. Specifically, this is a sparse $n \times n$ matrix with all non-zero elements located in three groups: 1) a narrow band of width m centered about the main diagonal; 2) a triangular group in the upper right, described by $A_{ij} : j \geq i + n - (m-1)/2$; and 3) a triangular group in the lower left described by $A_{ij} : j \leq i - n + (m-1)/2$. The corner elements are used to continue the band in problems involving periodic boundary conditions. The routine BCMAT, which takes advantage of this form to minimize storage and computation, was adapted from BANDSOLVE⁽¹⁾.

Method

The solution is carried out as follows: the leftmost $(m-1)/2$ columns of the coefficient matrix are moved to the right of the matrix as shown in Fig. 2. Gaussian elimination with row interchange is then used to eliminate elements below the main diagonal, and back substitution is used to obtain the solutions. Because of the column move, the last $(m-1)/2$ solutions thus found are actually the first $(m-1)/2$ solutions and must be moved back.

Data Management

In order to conserve storage, the coefficient matrix is not stored in standard form; but only the band and the corner elements are stored in an

($n \times 2m$) matrix. This working matrix is formed by "unwrapping" the corner elements of the original ($n \times n$) matrix to form a single band as shown in Fig. 1b, and then storing only the band itself by shifting the rows until the band is vertical as shown in Fig. 1c. The band thus fits into n rows and the first m columns of the working matrix. The right-hand-side vector is stored in the last column and columns $m+1$ to $2m-1$ are used for temporary storage. The calling routine must provide the coefficient matrix in this form.

Before elimination begins, the first and last $(m-1)/2$ columns of the original ($n \times n$) matrix are copied into columns $m+1$ thru $2m-1$ of the working matrix, and returned to unshifted form as in section B of Fig. 2b. Elimination now proceeds down the band in the left m columns of the working matrix corresponding to section A of Fig. 2b. Since the data in the working matrix has the rows shifted with respect to each other, the code maintains a current section of m rows which have been shifted back so as to be aligned as in the original matrix. Of course, this process shifts off elements, but such elements either have already been eliminated or were initially moved to columns $m+1$ to $2m-1$.

At the first elimination step, rows 1 to m are aligned so that the pivot column is effectively column 1 of section A. This column is then eliminated, so the new rows can be shifted left another position, bringing them into alignment with row $m+1$. Now rows 2 to $m+1$ are aligned, and position 2,2 is used as a pivot. The same elimination operations used on section A are of course performed on section B, which is in standard format. Row interchange is performed among the m aligned rows to always bring the largest element to the pivot position.

The last column so eliminated is effectively the last column of section A; and standard elimination is carried out below the diagonal in section B.

The back substitution is also carried out in two parts because of the data format. The last $m-1$ rows are back substituted in a normal manner to obtain the last $m-1$ solutions. In the first $n-m+1$ rows, only the band and section B may be non-zero, and the routine back substitutes only in these two areas.

Call Parameters

The routine has six call parameters. The input matrix A should be dimensioned at least $n \times 2m$ and should contain the coefficient matrix in column 1 to m , organized as in Fig. 1c, and described above. The right-hand-side vector should be stored in column $2m$; the intervening $m-1$ columns are used for temporary storage and will be destroyed, as will the original coefficients. The solution is returned in column $2m$. The parameters N and M are as used above; NN is the actual row dimension of A, used for variable dimensioning. Singularity is assumed if at some point no pivot larger than ZERO is found; if this condition occurs, the routine exits with IERR = 1. Upon normal completion the routine exits with IERR = 0.

Storage

BCMAT requires an array of $2nm$ floating point cells to solve a system of n equations with m diagonals, while a full matrix solution would require $n^2 + n$ cells. Thus given a constant m , storage required goes up linearly with n instead of quadratically. Of course, m must be less than $(n+1)/2$ for there to be any advantage.

Testing

BCMAT was tested against a routine using the same method but without special data format, and gave the same results as expected. It was also tested against a routine which did not perform the column move,

i.e., a standard solution using Gaussian elimination; the errors were comparable in all cases tested.

Timing was also done for BCMAT and the latter routine above. The tests were done on a CDC 6400 using FTN version 3, optimization level 2. Results are given in Table I.

Acknowledgment

The author would like to thank Professor J. Denavit for his helpful suggestions. This work was supported jointly by the U. S. Atomic Energy Commission under contract AT(11-1)-2200 and the Office of Naval Research under contract N00014-67-A-0356-0026.

Reference

- (1) Donald H. Thurnau, Algorithm 195, CACM 8-63 (441), 12-72 (1074)

TABLE I

N	USING FULL MATRIX	USING BCMAT FOR M =			
		3	5	7	11
11	16	8	15	19	28
20	59	15	26	41	67
50	711	36	72	114	214
75	2225	55	108	172	326
100	5119	74	144	227	460

Execution times for BCMAT and the full matrix version.

Times are in milliseconds on a CDC 6400 using Fortran

Extended.


```

SUBROUTINE RCMAT (A,N,M,NN,ZERO,IERR)
DIMENSION A(NN,1)
INTEGER TWOMM1,TWOM,TWOMP1
NM1=N-1
NP1=N+1
MM1=M-1
MM2=M-2
MP1=M+1
MM=MM1/2
TWOM=M+M
TWOMM1=TWOM-1
TWOMP1=TWOM+1
N1=N-M+1
N2=N-M+2

```

RCM 1
RCM 2
RCM 3
RCM 4
RCM 5
RCM 6
RCM 7
RCM 8
RCM 9
RCM 10
RCM 11
RCM 12
RCM 13
RCM 14
RCM 15
RCM 16
RCM 17
RCM 18
RCM 19
RCM 20
RCM 21
RCM 22
RCM 23
RCM 24
RCM 25
RCM 26
RCM 27
RCM 28
RCM 29
RCM 30
RCM 31
RCM 32
RCM 33
RCM 34
RCM 35
RCM 36
RCM 37
RCM 38
RCM 39
RCM 40
RCM 41
RCM 42
RCM 43
RCM 44
RCM 45
RCM 46
RCM 47
RCM 48
RCM 49
RCM 50
RCM 51
RCM 52
RCM 53
RCM 54
RCM 55
RCM 56
RCM 57

..... COPY FIRST, LAST MM COLUMNS

```

DO 1 I=1,N
  DO 1 J=MP1,TWOMM1
1    A(I,J)=0.
  DO 2 I=1,MM1
    NC=M-I
    NX=NP1-I
    DO 2 J=1,NC
      A(I,MM1+I+J)=A(I,J)
      A(NX,TWOMP1-I-J)=A(NX,MP1-J)
2    A(NX,MP1-J)=0.

```

C
C..... ROW SHIFTING AND ZERO PLACEMENT
C

```

DO 4 I=1,MM1
  NS=M-I
  DO 3 J=1,I
3    A(I,J)=A(I,J+NS)
  IP1=I+1
  DO 4 J=IP1,M
4    A(I,J)=0.

```

C
C..... MAIN ELIMINATION LOOP
C

```

DO 9 I=1,N1
  I2=I+1
  I3=I+MM1
  IMAX=I
  DO 5 J=I2,I3
5    IF (ABS(A(J,1)).GT.ABS(A(IMAX,1))) IMAX=J
    IF (ABS(A(IMAX,1)).LE.ZERO) GO TO 20
    IF (IMAX.EQ.I) GO TO 7
    DO 6 K=1,TWOM
      T=A(I,K)
      A(I,K)=A(IMAX,K)
6    A(IMAX,K)=T
7    DO 9 J=I2,I3
      R=A(J,1)/A(I,1)
      DO 8 K=2,M
8    A(J,K-1)=A(J,K)-R*A(I,K)
      A(J,M)=0.

```

DO 9 K=MP1,TWOM	RCM 58
9 A(J,K)=A(J,K)-R*A(I,K)	RCM 59
C... SECONDARY ELIMINATION LOOP	RCM 60
C	RCM 61
ICP1=M+1	RCM 62
DO 13 IR=N2,NM1	RCM 63
IC=ICP1	RCM 64
ICP1=IC+1	RCM 65
IRP1=IR+1	RCM 66
IMAX=IR	RCM 67
DO 10 IR2=IRP1,N	RCM 68
10 IF (ABS(A(IR2,IC)).GT.ABS(A(IMAX,IC))) IMAX=IR2	RCM 69
IF (ABS(A(IMAX,IC)).LE.ZERO) GO TO 20	RCM 70
IF (IMAX.EQ.IR) GO TO 12	RCM 71
DO 11 IC2=IC,TWOM	RCM 72
T=A(IR,IC2)	RCM 73
A(IR,IC2)=A(IMAX,IC2)	RCM 74
11 A(IMAX,IC2)=T	RCM 75
12 DO 13 IR2=IRP1,N	RCM 76
R=A(IR2,IC)/A(IR,IC)	RCM 77
DO 13 IC2=ICP1,TWOM	RCM 78
13 A(IR2,IC2)=A(IR2,IC2)-R*A(IR,IC2)	RCM 79
IF (ABS(A(N,TWOMM1)).LE.ZERO) GO TO 20	RCM 80
C	RCM 81
C.... BACK SUBSTITUTE SECONDARY SYSTEM	RCM 82
C	RCM 83
A(N,TWOM)=A(N,TWOM)/A(N,TWOMM1)	RCM 84
DO 15 I=1,MM2	RCM 85
-IR=N-I	RCM 86
DO 14 J=1,I	RCM 87
14 A(IR,TWOM)=A(IR,TWOM)-A(IR,TWOM-J)*A(NP1-J,TWOM)	RCM 88
15 A(IR,TWOM)=A(IR,TWOM)/A(IR,TWOMM1-I)	RCM 89
C	RCM 90
C.... BACK SUBSTITUTE PRIMARY SYSTEM	RCM 91
C	RCM 92
DO 17 I=1,N1	RCM 93
IR=N2-I	RCM 94
DO 16 J=1,MM1	RCM 95
16 A(IR,TWOM)=A(IR,TWOM)-A(IR,J+1)*A(IR+J,TWOM)	RCM 96
1 -A(IR,M+J)*A(N1+J,TWOM)	RCM 97
17 A(IR,TWOM)=A(IR,TWOM)/A(IR,1)	RCM 98
C	RCM 99
C.... ROTATE SOLUTIONS TO PROPER POSITIONS	RCM 100
C	RCM 101
DO 18 I=1,N	RCM 102
18 A(I,1)=A(I,TWOM)	RCM 103
DO 19 I=1,N	RCM 104
J=I+MM	RCM 105
IF (J.GT.N) J=J-N	RCM 106
19 A(J,TWOM)=A(I,1)	RCM 107
IEPR=0	RCM 108
RETURN	RCM 109
20 IERR=1	RCM 110
RETURN	RCM 111
END	RCM 112
	RCM 113

(a)

(b)

(c)

Figure 1

(a)

(b)

Figure 2