

# ALGORITHM 508

## Matrix Bandwidth and Profile Reduction [FI]

H. L. CRANE JR.

Comptek Research, Inc.

NORMAN E. GIBBS, WILLIAM G. POOLE JR., and PAUL K. STOCKMEYER

College of William and Mary

**Key Words and Phrases:** bandwidth reduction, diameter of a graph, profile reduction, sparse matrices

**CR Categories:** 5.14, 5.32

**Language:** Fortran

---

### DESCRIPTION

#### Introduction

This program, REDUCE, reduces the bandwidth and profile of sparse symmetric matrices, using row and corresponding column permutations. It is a realization of the algorithm described by the authors in [4]. It was extensively tested and compared with several other programs [5] and was found to be considerably faster than the others, generally superior for bandwidth reduction and as satisfactory as any other for profile reduction.

#### Outline of the Method

Only an outline of the algorithm is given here; a detailed description can be found in [4]. The algorithm can best be described in terms of the *adjacency graph*  $G$ , which has the characteristic that there is an edge in  $G$  between vertices  $v_i$  and  $v_j$  if and only if  $a_{ij} \neq 0$  and  $i \neq j$ .

**Step 1.** Find the endpoints of a *pseudodiameter*, i.e. a pair of vertices that are at nearly maximal distance apart. This is done by a finite, iterative process of determining a vertex that is a maximum distance away from a given vertex.

**Step 2.** Given pseudodiameter endpoints  $u$  and  $v$  of distance  $k$  apart, partition the set of vertices into *levels*  $L_1, L_2, \dots, L_k$  such that adjacent vertices in  $G$  are in the same or adjacent levels and such that  $\max_i |L_i|$  is nearly minimized.

---

Received 17 May 1975 and 3 March 1976.

Copyright © 1976, Association for Computing Machinery, Inc. General permission to republish, but not for profit, all or part of this material is granted provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery.

This research was supported in part by the Office of Naval Research under Contract N00014-73 A 0374-0001, NR044-459, and in part by the National Aeronautics and Space Administration under Grant NGR 47-102-001.

Authors' addresses: H.L. Crane Jr., Comptek Research, Inc., Hyattsville, MD 20782; N.E. Gibbs, W.G. Poole Jr., P.K. Stockmeyer, Department of Mathematics, College of William and Mary, Williamsburg, VA 23185.

Step 3. Number the vertices of  $G$ , level by level, beginning at an endpoint of the pseudodiameter.

#### Matrix Data Structure

Sparse matrices are typically stored in some compact form which takes advantage of the sparsity. The data structure assumed here is one which is commonly used in bandwidth and profile schemes, e.g. [1], [2], [3], and [6]. Subroutine REDUCE accepts as input a *connection table*,  $C$ , representing the indices of the nonzero elements of the  $n \times n$  matrix  $A$ . The connection table has  $n$  rows and  $m$  columns where  $m$  is the number of off-diagonal nonzero elements in the row which has a maximum number of off-diagonal nonzero elements (i.e. the maximum degree of the graph  $G$ ). The entries in row  $i$  of  $C$  are the column indices of the nonzero elements in row  $i$  of the matrix  $A$ . For example, if

$$A = \begin{bmatrix} X & X & 0 & 0 & X \\ X & X & 0 & X & X \\ 0 & 0 & X & X & 0 \\ 0 & X & X & X & 0 \\ X & X & 0 & 0 & X \end{bmatrix}$$

where  $X$  represents a nonzero element, then

$$C = \begin{bmatrix} 2 & 5 & 0 \\ 1 & 4 & 5 \\ 4 & 0 & 0 \\ 2 & 3 & 0 \\ 1 & 2 & 0 \end{bmatrix}.$$

The order of indices in a row of  $C$  is immaterial. The nonzero elements of the matrix  $A$  are never needed, only their indices.

#### Test Results

REDUCE was tested on an IBM System/360 (model 50) computer using the Fortran IV G and H compilers and on a CDC 6600 computer using the Fortran (RUN) and Fortran extended (FTN) compilers.

In another paper [5], the authors compared the execution times, bandwidths, and profiles produced by REDUCE with those of five other programs on a wide range of problems. REDUCE typically produced the smallest bandwidths; it produced profiles which were on the average as small as those for any other program; and REDUCE was faster than all the others by at least an order of magnitude.

#### REFERENCES

1. COLLINS, R.J. Bandwidth reduction by automatic renumbering. *Int. J. Numer. Meth. Engrg.* 6 (1973), 345-356.
2. CUTHILL, E.H., AND MCKEE, J. Reducing the bandwidth of sparse symmetric matrices. *Proc. ACM 24th Nat. Conf.*, ACM, New York, 1969, pp. 157-172.
3. EVERSTINE, G.C. The BANDIT computer program for the reduction of matrix bandwidth for NASTRAN. Rep. 3827, Naval Ship Res. and Develop. Center, Bethesda, Md., 1972.
4. GIBBS, N.E., POOLE, W.G. JR., AND STOCKMEYER, P.K. An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM J. Numer. Anal.* 13, 2 (April 1976), 235-251.
5. GIBBS, N.E., POOLE, W.G. JR., AND STOCKMEYER, P.K. A comparison of several bandwidth and profile reduction algorithms. *ACM Trans. Math. Software* 2, 4 (Dec. 1976), 322-330. (this issue)

6. WANG, P.T.R. Bandwidth minimization, reducibility, decomposition, and triangularization of sparse matrices. Ph.D. Diss., Dep. Comptr. and Information Sci., Ohio State U., Columbus, Ohio, 1973.

## ALGORITHM

Only that portion of the listing which gives the introductory comments explaining the algorithm is printed here. The complete listing is available from the ACM Distribution Service (see inside back cover for order form), or may be found in "Collected Algorithms from ACM."

```

SUBROUTINE REDUCE(NDSTK, NR, IOLD, RENUM, NDEG, LVL, LVLS1, RED 10
* LVLS2, CCSTOR, IBW2, IPF2) RED 20
C SUBROUTINE REDUCE DETERMINES A ROW AND COLUMN PERMUTATION WHICH, RED 30
C WHEN APPLIED TO A GIVEN SPARSE MATRIX, PRODUCES A PERMUTED RED 40
C MATRIX WITH A SMALLER BANDWIDTH AND PROFILE. RED 50
C THE INPUT ARRAY IS A CONNECTION TABLE WHICH REPRESENTS THE RED 60
C INDICES OF THE NONZERO ELEMENTS OF THE MATRIX, A. THE ALGO- RED 70
C RITHM IS DESCRIBED IN TERMS OF THE ADJACENCY GRAPH WHICH RED 80
C HAS THE CHARACTERISTIC THAT THERE IS AN EDGE (CONNECTION) RED 90
C BETWEEN NODES I AND J IF A(I,J) .NE. 0 AND I .NE. J. RED 100
C DIMENSIONING INFORMATION--THE FOLLOWING INTEGER ARRAYS MUST BE RED 110
C DIMENSIONED IN THE CALLING ROUTINE. RED 120
C NDSTK(NR,D1) D1 IS .GE. MAXIMUM DEGREE OF ALL NODES. RED 130
C IOLD(D2) D2 AND NR ARE .GE. THE TOTAL NUMBER OF RED 140
C RENUM(D2+1) NODES IN THE GRAPH. RED 150
C NDEG(D2) STORAGE REQUIREMENTS CAN BE SIGNIFICANTLY RED 160
C LVL(D2) DECREASED FOR IBM 360 AND 370 COMPUTERS RED 170
C LVLS1(D2) BY REPLACING INTEGER NDSTK BY RED 180
C LVLS2(D2) INTEGER*2 NDSTK IN SUBROUTINES REDUCE, RED 190
C CCSTOR(D2) DGREE, FNDIAM, TREE AND NUMBER. RED 200
C COMMON INFORMATION--THE FOLLOWING COMMON BLOCK MUST BE IN THE RED 210
C CALLING ROUTINE. RED 220
C COMMON/GRA/N, IDPTH, IDEG RED 230
C EXPLANATION OF INPUT VARIABLES-- RED 240
C NDSTK- CONNECTION TABLE REPRESENTING GRAPH. RED 250
C NDSTK(I,J)=NODE NUMBER OF JTH CONNECTION TO NODE RED 260
C NUMBER I. A CONNECTION OF A NODE TO ITSELF IS NOT RED 270
C LISTED. EXTRA POSITIONS MUST HAVE ZERO FILL. RED 280
C NR- ROW DIMENSION ASSIGNED NDSTK IN CALLING PROGRAM. RED 290
C IOLD(I)- NUMBERING OF ITH NODE UPON INPUT. RED 300
C IF NO NUMBERING EXISTS THEN IOLD(I)=I. RED 310
C N- NUMBER OF NODES IN GRAPH (EQUAL TO ORDER OF MATRIX). RED 320
C IDEG- MAXIMUM DEGREE OF ANY NODE IN THE GRAPH. RED 330
C EXPLANATION OF OUTPUT VARIABLES-- RED 340
C RENUM(I)- THE NEW NUMBER FOR THE ITH NODE. RED 350
C NDEG(I)- THE DEGREE OF THE ITH NODE. RED 360
C IBW2- THE BANDWIDTH AFTER RENUMBERING. RED 370
C IPF2- THE PROFILE AFTER RENUMBERING. RED 380
C IDPTH- NUMBER OF LEVELS IN REDUCE LEVEL STRUCTURE. RED 390
C THE FOLLOWING ONLY HAVE MEANING IF THE GRAPH WAS CONNECTED-- RED 400
C LVL(I)- INDEX INTO LVLS1 TO THE FIRST NODE IN LEVEL I. RED 410
C LVL(I+1)-LVL(I)= NUMBER OF NODES IN ITH LEVEL RED 420
C LVLS1- NODE NUMBERS LISTED BY LEVEL. RED 430
C LVLS2(I)- THE LEVEL ASSIGNED TO NODE I BY REDUCE. RED 440
C WORKING STORAGE VARIABLE-- RED 450
C CCSTOR RED 460
C LOCAL STORAGE-- RED 470
C COMMON/CC/-SUBROUTINES REDUCE, SORTZ AND P1KLVL ASSUME THAT RED 480
C THE GRAPH HAS AT MOST 50 CONNECTED COMPONENTS. RED 490
C SUBROUTINE FNDIAM ASSUMES THAT THERE ARE AT MOST RED 500
C 100 NODES IN THE LAST LEVEL. RED 510
C COMMON/LVLW/-SUBROUTINES SETUP AND P1K1VI. ASSUME THAT THERE RED 520
C ARE AT MOST 100 LEVELS. RED 530

```