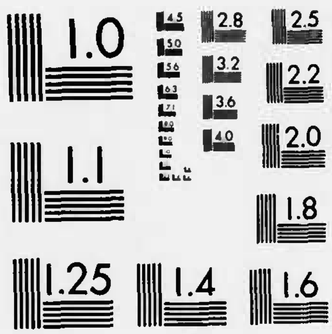
AD-A123 267 A PROGRAM OF ELLIPTIC SOLVER DEVELOPMENT AND IMPLEMENTATION IN SEMI-IMPLI..(U) JAYCOR SAN DIEGO CA D E DIETRICH NOV 81 JAYCOR-J510-81-053/2192 1/1

UNCLASSIFIED N00014-80-C-0298 F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A123267

①

No DTIC
7/82

Report J510-81-053/2192

DTIC "A"

A PROGRAM OF ELLIPTIC SOLVER DEVELOPMENT AND IMPLEMENTATION IN SEMI-IMPLICIT NUMERICAL OCEAN CIRCULATION MODELS

Prepared by
David E. Dietrich
~~JAYCOR~~
11011 Torreyana Road
P. O. Box 85154
San Diego, CA 92138

November 1981

Final Report for the Period 1 March 1980 - 31 July 1981

Prepared for

THE OFFICE OF NAVAL RESEARCH
Office of Procurement
800 N. Quincy Street
Arlington, Virginia 22217

THE NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY
Environmental Models
Numerical Modeling Division
NSTL Station, MS 39529

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

Field	Content
1. REPORT NUMBER	J510-81-053/2192
2. GOVT ACCESSION NO.	AD-A123267
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	A Program of Elliptic Solver Development and Implementation in Semi-Implicit Numerical Ocean Circulation Models
5. TYPE OF REPORT & PERIOD COVERED	Final Report 1 March 1980 - 31 July 1981
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s)	David E. Dietrich
8. CONTRACT OR GRANT NUMBER(s)	N00014-80-C-0298
9. PERFORMING ORGANIZATION NAME AND ADDRESS	JAYCOR 11011 Torreyana Road P. O. Box 85154, San Diego, CA 92138
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS	Office of Naval Research, Department of the Navy 800 N. Quincy Street Arlington, Virginia 22217
12. REPORT DATE	November 1981
13. NUMBER OF PAGES	44
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	Naval Ocean Research and Development Activity Environmental Models, Numerical Modeling Station NSTL Station, MS 39539
15. SECURITY CLASS. (of this report)	Unclassified
15a. DECLASSIFICATION DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)	APPROVED FOR PUBLIC RELEASE DISTRIBUTION UNLIMITED
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)	
18. SUPPLEMENTARY NOTES	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)	elliptic partial difference equation solver, stabilized error vector propagation (SEVP) method, irregular grids with islands, non-iterative

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Several elliptic solver programs have been developed using the highly efficient and versatile "error vector propagation"(EVP) method as the fundamental algorithm component. The basic new program is a modified version of Madala's SEVP method that can handle arbitrary geometry, with or without islands. There are several other versions of this new program: one allows irregular boundaries but not islands; one does not allow irregular boundaries or islands; and some versions exist with and without roundoff relaxation in the forward sweep. All versions are capable of handling Dirichlet, Neuman, and mixed boundary conditions.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

The author would like to thank Drs. Alan Wallcraft and Harley Hurlburt for their advice and support of this project. This research was sponsored by the Naval Oceanographic Research and Development Activity (NORDA) under Contract No. N00014-80-C-0298.

CONTENTS

1. INTRODUCTION

Beginning with the simultaneous discovery of the fundamental algorithm by Hirota, et al. (1970), and by Roache (1971), a new class of algorithms for solving elliptic partial difference equations has been developed during the last 10 years. Roache (1971) noted that the fundamental algorithm, which is analogous to "shooting methods" in ordinary differential equations, and described as the "error vector propagation" (EVP) method, is very efficient when applied to low-resolution problems, but cannot be applied to high-resolution problems due to roundoff error. Dietrich (1974) discovered an iterative application of the algorithm that could be applied to high-resolution problems, and this iterative technique was further explored and refined by Dietrich, et al. (1975) and by Dietrich (1977). Madala (1978) discovered a method, called the "stabilized error vector propagation" (SEVP) technique, whereby the basic algorithm can be applied as a direct method for high-resolution problems. Roache (1978) reviewed the development of EVP and analyzed its characteristics in detail.

One important class of applications of EVP-based elliptic solvers is in semi-implicit formulations of the fluid dynamic equations in which dominant

---

I. Hirota, T. Tokioka, and M. Nishiguchi, J. Meteor. Soc. Japan 48 (1970), pp. 161-167.

P. J. Roache, "A New Direct Method for the Discretized Poisson Equation," in Proceedings of the Second International Conference of Numerical Methods in Fluid Dynamics, Berkeley, California, September 16-19, 1970, pp. 48-53, 1971.

D. Dietrich, J. Meteor. Soc. Japan 52 (1974), pp. 337-338.

D. Dietrich, B. E. McDonald, and A. Warn-Varnas, J. Comp. Phys. 18 (1975), pp. 421-439.

D. Dietrich, Final Report, Contract No. N00014-77-C-0208 for the Naval Oceanographic Laboratory of the Naval Oceanographic Research and Development Activity (1977).

R. Madala, Mon. Wea. Rev. 106 (1978), pp. 1735-1741.

P. J. Roache, Num. Heat Trans. 1 (1978), pp. 1-201.

terms involved in gravity (O'Brien and Hurlburt, 1972) and/or compression wave (Dietrich, et al., 1981) propagation are solved implicitly (with the remaining terms explicit). A more powerful application exists in solving "almost fully implicit" formulations of the fluid dynamic equations (Dietrich, 1975); this application is recommended when the shortest fluid dynamic time scale of interest is long, compared to the well-known CFL time step limit, as appears to be the case in most fluid dynamic problems with constant or slowly varying external forcing. (In some problems with waves generated by time-varying external conditions, this might not be the case.)

In Section 2 of this report, we describe an SEVP algorithm, program JAYNOR, for solving general two-dimensional, second-order elliptic equations. This algorithm can be applied to linear problems with variable coefficients, irregular boundaries, islands, and general boundary conditions. Its limitations are mainly associated with roundoff error, as described in Section 3. In Section 4, we describe how to use a computer program based on this general algorithm, as well as more specialized versions of this program. Appendix II contains listings of this general program and its more specialized versions.

J. J. O'Brien and H. E. Hurlburt, J. Phys. Oceanogr. 2 (1972), pp. 14-26.

D. Dietrich, H. Klein, and R. K.-C. Chan, "FLAME: A Computer Model of Time-Dependent, Multidimensional, Multiphase Reactive Flow," paper presented at the 20th National Heat Transfer Conference, Milwaukee, Wisconsin, August 2-5, 1981.

D. Dietrich, J. Meteor. Soc. Japan 53 (1975), pp. 222-225.

2. THE GENERAL SEVP ALGORITHM

The general SEVP algorithm described here is similar to the one described by Madala (1978). The differences are mainly the capability of including islands in the present version, and the elimination of the homogeneous solution calculation during the forward sweep by more fully utilizing influence coefficient arrays generated during a preprocessor.

The new SEVP algorithm, JAYNOR, uses a master flag array, $N_{i,j}$, defined by:

$$N_{i,j} = \begin{cases} 1 \text{ if the } (i,j) \text{ point is "inside the boundary"} \\ 0 \text{ otherwise} \end{cases} .$$

Points on islands are considered "outside the boundary". If $N_{i,j} = 1$, the following equation is satisfied:

$$A_{i,j} \cdot X_{i,j+1} + B_{i,j} \cdot X_{i+2,j+1} + C_{i,j} \cdot X_{i+1,j} + D_{i,j} \cdot X_{i+1,j+2} + E_{i,j} \cdot X_{i+1,j+1} = F_{i,j} , \qquad (1)$$

where $A_{i,j}$, $B_{i,j}$, $C_{i,j}$, $D_{i,j}$, and $E_{i,j}$ are given coefficient arrays of the five-point elliptic operator; $F_{i,j}$ is a given source term array; and $X_{i,j}$ is the solution array determined by JAYNOR. If $N_{i,j} = 0$, the $X_{i,j}$ is left unchanged. In addition to the above-indicated parameters and others normally needed to define a two-dimensional elliptic partial difference equation with second-order accuracy, JAYNOR requires the user to specify the number of subregions in which the aforementioned SEVP algorithm is to applied, and the grid lines separating these subregions. In general, the more subregions used, the smaller the roundoff error will be in the solution (at the expense of more auxiliary storage and calculation). These grid lines are transverse to the EVP march direction, which is the j-direction in JAYNOR.

We now describe the JAYNOR algorithm in detail. Since Neuman and mixed boundary conditions can be easily transformed into Dirichlet-type boundary conditions by suitably redefining the partial difference operator and source term at points adjacent to boundaries (see Appendix I), we describe its application to problems with Dirichlet conditions. (JAYNOR assumes the user has already performed this simple task.)

First, we describe JAYNOR application to a regular region, illustrated in Figure 1. Two classes of auxiliary influence arrays are generated in the JAYNOR preprocessor, subroutine JAY. Both classes are composed of selected subvectors from the total solution vector, forced by a selected isolated unit source term. In a regular problem, each such subvector of the first class is the solution at the second row from the bottom of a subregion, forced by a unit source term assigned to a selected point in the second row from the top of the same subregion, in an associated problem in the region below the top boundary of the subregion — the associated problem has homogeneous boundary conditions, including at the top bounday of the subregion, and has zero source term everywhere, except at the selected unit source point. In JAYNOR, this array is called RINV; specifically, RINV(I1,I2,K) is the solution at the (I2+1) position of the second row from the bottom of subregion K, forced by an isolated unit source at the (I1+1) position in the second row from the top of subregion K. In a regular problem, each subvector of the second class is the solution at the second row from the top of a subregion, forced by a unit source term assigned to a selected point in the second row from the top of the same subregion, again in the associated problem described above. In JAYNOR, this array is called RINV1; specifically, RINV1(I1,I2,K) is the solution at the (I2+1) position of the second row from the top of subregion K, forced by an isolated unit source at the (I1+1) position in the second row from the top of subregion K. The top subregion does not require an array of this second class. (Although the above description indicates the storage sequence in regular problems, arrays RINV and RINV1 are stored as one-dimensional arrays to save storage in irregular problems due to differing storage requirements in each subregion — see below.)

Using the arrays RINV and RINV1, as determined by the JAYNOR pre-processor, which is performed in subroutine JAY, subroutine NOR calculates the solution to the partial difference equation (1) with assigned boundary conditions. This is done using a forward sweep, followed by a backward sweep, similar to the procedure described by Madala (1978), utilizing RINV and RINV1 data along the way. In the present method, the forward "particular solution sweep" and backward "homogeneous solution sweep" each requires slightly fewer

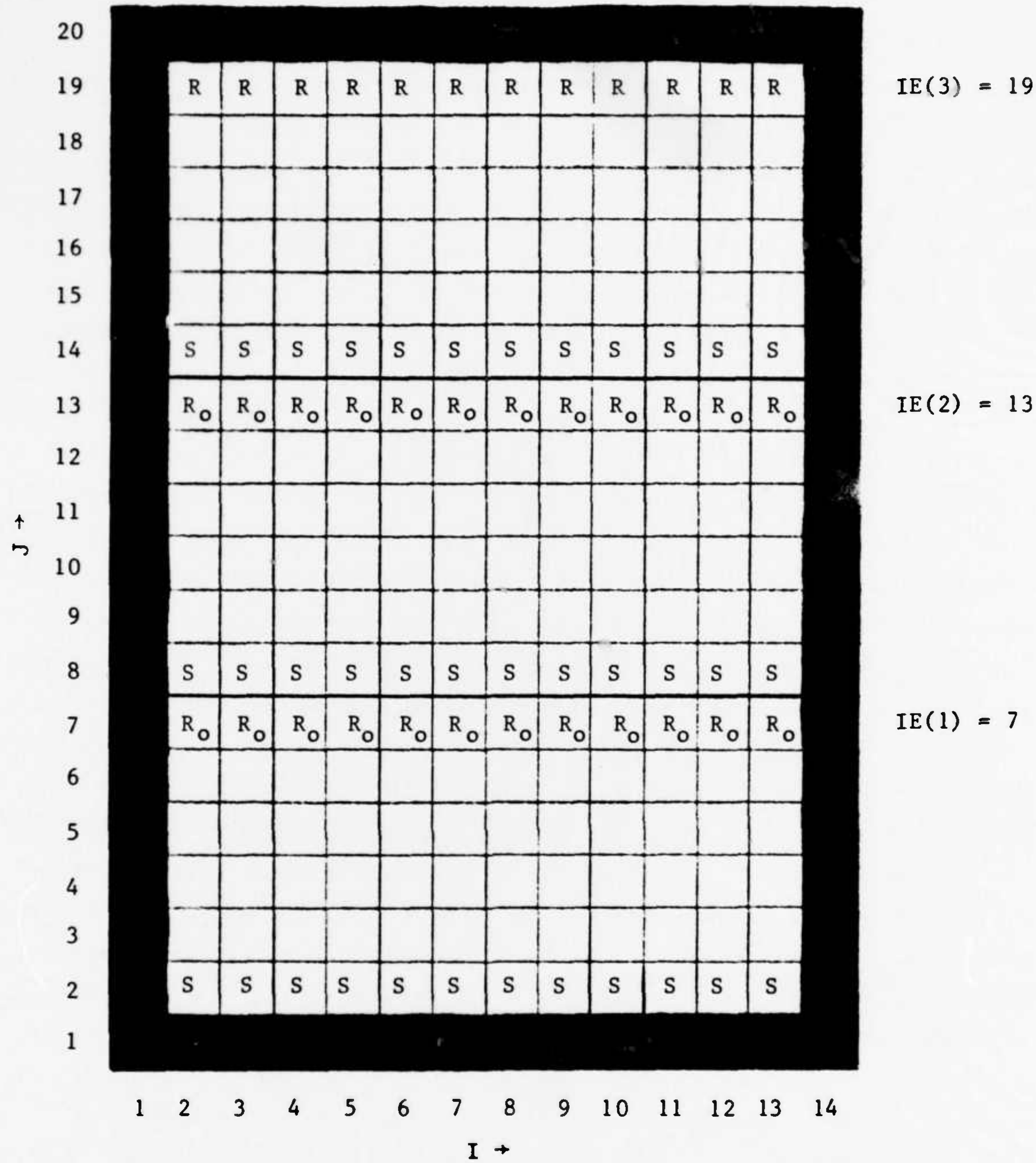


Figure 1. Application of program JAYNOR to a regular mesh. The partial difference equation is satisified at the center of each cell, except those outside the boundary, which are shaded; the source term and solution arrays are cell-centered. Grid lines forming the cell boundaries are drawn. Grid lines between SEVP subregions are darkened. In this example, JAYNOR program topological variables are as follows: M=14, N=20, NSB=3, IE(1)=7, IE(2)=13, IE(3)=19, and the NTYPE(I,J) array has zeroes for I=1, I=14, J=1 and J=20, and ones everywhere else (there are no islands). SEVP starting vector positions are indicated by S's, and SEVP residual vector positions are indicated by R's; most of the R's have zero subscripts to indicate they are "open residual positions" (see main text).

operations than a single ordinary relaxation sweep, except for operations involving RINV and RINV1. Subroutine NOR avoids generating a homogeneous solution on the forward sweep by using RINV1 to adjust the grid point values in the second row from the top of each subregion (except the top subregion) directly during the forward sweep, instead of using RINV, RINV1, and an appropriate homogeneous sweep to create these same adjustments, as described by Madala (1978). Every other step of the present procedure is essentially the same as described by Madala when JAYNOR is applied to a regular region.

Application of JAYNOR to elliptic problems in regions with irregular boundaries and islands is similar to its application to regular problems. The differences are mainly in the definition and use of the preprocessor arrays RINV and RINV1. These differences are associated with the definition "starting vector points" and "error vector points" in each subregion. For regular problems, "starting vector points" are points in the second row from the bottom of each subregion, and "error vector points" are points in the second row from the top of each subregion (see Figure 1). When the problem is irregular, "starting vector points" are all points in the second row from the bottom of each subregion that are inside the boundary (NTYPE=1) plus all points inside the boundary that are *immediately* above points that are outside the boundary (NTYPE=0) (see Figure 2). Similarly, "error vector points" are all points in the second row from the top of each subregion that are inside the boundary plus all points inside the boundary that are immediately below points that are outside the boundary (see Figure 2). The general defintions for RINV and RINV1, applicable for both regular and irregular problems, are then as follows. RINV(L,M,K) is the solution at the M^{th} starting vector position in subregion K, forced by an isolated unit source at the L^{th} error vector position in subregion K, in an associated problem, as described above. RINV1(L,M,K) is the solution at the M^{th} "open" error vector position in subregion K, forced by an isolated unit source at the L^{th} error vector position in subregion K, in the associated problem, where "open" means that the point above the error vector position is inside the boundary (NTYPE=1) (see Figure 2). Since the ranges of L and M can vary from subregion to subregion in irregular problems, arrays RINV and RINV1 are treated as one-dimensional arrays to save storage.

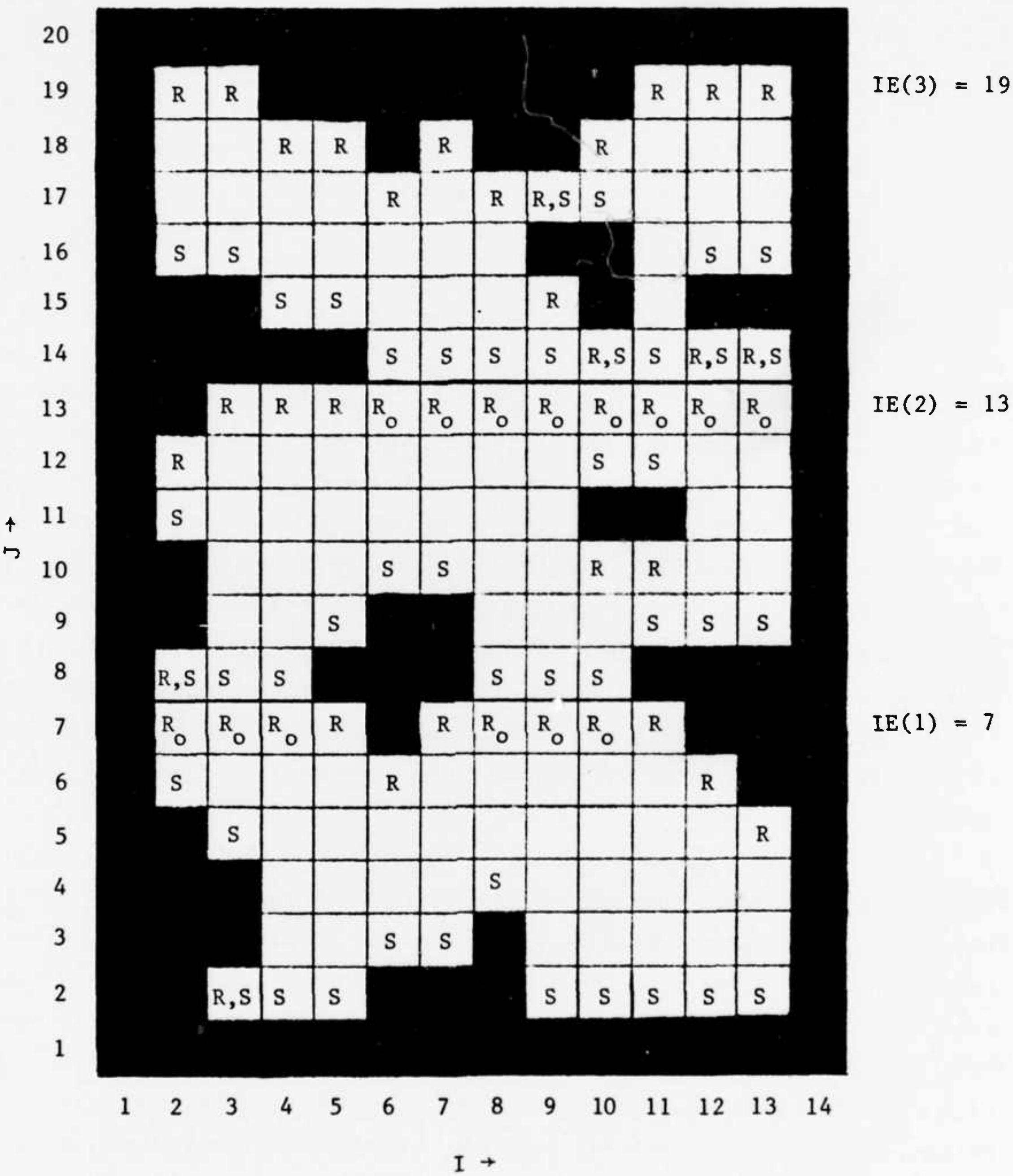


Figure 2. Application of program JAYNOR to an irregular mesh with islands. Refer to Figure 1 caption to interpret this figure. Some of the R's have zero subscripts to indicate these are "open residual positions" (see main text). The user need not specify the R, S, and R_0 positions; they are determined in subroutine TOPOL from the user-specified topological flag array NTYPE (see main text).

3. SELECTION OF MARCHING DIRECTION AND BLOCK SIZE

The choice of marching direction and block size is determined by the accuracy desired in the solution, the computing precision used, and the EVP roundoff error characteristics of the partial difference operator. (In many applications, it is desirable to use double precision on IBM computers and on other computers for which double-precision computation is about as fast as single-precision computation. This can usually be facilitated by using IMPLICIT REAL*8 type statements at the beginning of the EVP subroutines.) Detailed discussions on this are given by Dietrich (1975) and Roache (1978). In general, the choice of march direction and block size depends on a number of factors, and only rough guidelines can be given. Fortunately, this is all that is needed, since even when operating well within roundoff error restrictions, the SEVP method is robust and is usually competitive with the best of other methods, often being superior. However, it should be noted that when the geometry is regular (or nearly regular), and the partial difference operator has constant coefficients, there are transform methods (combined with a capacitance matrix to handle any irregularities) that can be superior to SEVP in terms of storage requirements and competitive in terms of operation count (Wallcraft, 1980).

A general rule-of-thumb guideline for the choice of subregion size when using SEVP is that for an elliptic partial differential equation with low to moderate diagonal dominance (as in a Poisson partial differential equation), the normalized absolute error in the application of the EVP algorithm to a given subregion is of order 5^B times the computing precision used, where B is the ratio of the subregion length in the EVP march direction to the minimum grid interval size in the other direction. For example, for a Poisson equation on a uniform mesh ($\Delta x = \Delta y$), and computing precision of 15 significant figures, the SEVP solution would be accurate to about five significant figures if the maximum subregion length were about 14 grid intervals. However, SEVP can be applied iteratively (as can any direct solver) to reduce roundoff error to any desired level above the computing

---

A. Wallcraft, "Capacity Matrix Techniques," Ph.D. Thesis (1980), Department of Mathematics, Imperial College, London, England S.W.7.

precision available. For example, if the roundoff error is $0(10^{-5})$ after one call to SEVP, the error can be reduced to $0(10^{-10})$ after two SEVP calls. Each succesive SEVP problem is forced by the roundoff error residuals from the previous SEVP problem, and the result is added to the previous solution.

A rule for choosing the SEVP march direction is suggested by Madala (1978). Madala recommends choosing the direction for which "$A > 1$", where A is the ratio of the grid interval transverse to the SEVP march direction divided by the grid interval in the SEVP march direction. However, in some problems there are regions for which $A > 1$ and other regions for which $A < 1$. In such problems the use of SEVP as a direct method may be costly. A good possible alternative in such cases is to use SEVP iteratively (with variable march direction) in the same way EVP was used iteratively by Dietrich, et al. (1975); in some cases, it may be useful to use alternate forms of EVP such as described by Dietrich (1977) and by Roache (1978). Even in problems for which "$A - 1$" has the same sign everywhere, it might not be best to march in the direction for which $A > 1$. Specifically, when $A > 1$ for a march direction with much fewer mesh points than in the other direction, the other march direction may be better even though more SEVP subregions are required, since the influence arrays RINV and RINV1 will be much smaller for a given subregion. This, of course, can be affected by the kind of computer being used (e.g., vector vs. scalar computer).

Finally, when there are islands, it is sometimes advantageous to place the SEVP subregion boundaries in such a manner that the islands straddle the boundaries, especially when this does not increase the number of SEVP subregions required; this reduces the number of starting and error vector points required and the associated auxiliary computation and storage used by the SEVP algorithm. Similarly, if there are no islands, it is desirable to place the SEVP subregion boundaries near the narrowest parts of the SEVP domain, such as the throat of an hourglass-shaped domain.

4. USING THE JAYNOR COMPUTER PROGRAM

Program JAYNOR sets up and solves a general sample problem, using preprocesor subroutines TOPOL and JAY, and repeat solution subroutine NOR; in JAYNOR program variables, the sample problem may be written:

$$AX(I,J)*X(I,J+1)+CX(I,J)*X(I+2,J+1)+AY(I,J)*X(I+1,J)$$

$$+CY(I,J)*X(I+1,J+2)+BB(I,J)*X(I+1,J+1) = F(I,J) \quad . \qquad (2)$$

If NTYPE(I,J) = 1, Eq. (2) is satisfied; if NTYPE(I,J) = 0, X(I,J) is left unchanged and treated as a boundary condition, where appropriate. Thus, the user must specify: the partial difference operator coefficient arrays AX, CX, AY, CY, and BB; the source term array F; the topological array NTYPE; the x-dimension of the solution array M; the y-dimension of the solution array N; the maximum (for all SEVP block subregions) number of starting vector guess positions MXGS; the number of SEVP block subregions NBLK; the dimensions of the influence coefficient arrays (RINV and RINV1) LD1 and LD2; and the grid line number at the top of each subregion, array IE. If any of the user-specified dimensions MXGS, LD1, or LD2 is too small for the specified mesh topology (NTYPE array), appropriate diagnostic messages are printed in subroutine TOPOL and the computation is halted. Most arrays have dimensions selected from the following quantities: M, M-2, N, N-2, MXGS, NBLK, and NBLK-1. Specifically, the appropriate dimensions are as follows:

(M-2,N-2): AX,AY,CX,CY,BB,Q,F
(M,N): RECUR,X,H,ERR,XX,NTYPE
Also: DUMO(MXGS,NBLK-1),DUM1(MXGS),DUM2(MXGS),
ILFT(N),IRGT(N),IGES(MXGS,NBLK),JGES(MXGS,NBLK),
ICHK(MXGS,NBLK),JCHK(MXGS,NBLK),NEV(NBLK),
LINK(M-2,NBLK-1),NLINK(NBLK-1), IE(NBLK),L1(NBLK),
and L2(NBLK-1).

The maximum lengths required by RINV and RINV1 are:

RINV(NBLK*MXGS*MXGS),RINV1((NBLK-1)*MXGS*(M-2)).

A complete alphabetical list of the JAYNOR program significant variables and their definitions follows. User-specified variables are denoted by an asterisk.

TABLE 1: JAYNOR VARIABLES AND DEFINITIONS

Variable Name	Variable Definition
AX(I,J)*	Left coefficient of difference operator at mesh point (I+1,J+1)
AY(I,J)*	Bottom coefficient of difference operator at mesh point (I+1,J+1)
BB(I,J)*	Central coefficient of difference operator at mesh point (I+1,J+1)
CX(I,J)*	Right coefficient of difference operator at mesh point (I+1,J+1)
CY(I,J)*	Top coefficient of difference operator at mesh point (I+1,J+1)
ERR(I,J)	Normalized residual array
ERSUM	Mean absolute residual
F(I,J)*	Source term for difference operator at mesh point (I+1,J+1)
FSUM	Mean absolute source term
H(I,J)	Almost-homogeneous solution component array calculated on backward SEVP sweep
(IC,JC)	Mid-point of island location in SEVP sample problem
ICHK(N,NB)	The N^{th} residual position in the NB^{th} block is on the ICHK(N,NB) vertical mesh line

(continued)

TABLE 1 (continued)

Variable Name	Variable Definition
IE(NB)	The top azimuthal mesh line inside the NB^{th} block (IE(NBLK)=N-1)
IGES(N,NB)	The N^{th} starting vector position in the NB^{th} block is on the IGES(N,NB) vertical mesh line
ILFT(J)	The left-most point inside the boundary on horizontal mesh line J
IRGT(J)	The right-most point inside the boundary on horizontal mesh line J
JCHK(N,NB)	The N^{th} residual position in the NB^{th} block is on the JCHK(N,NB) horizontal mesh line
JGES(N,NB)	The N^{Th} starting vector position in the NB^{th} block is on the JGES(N,NB) horizontal mesh line
LD1*	The dimension of array RINV
LD2*	The dimension of array RINV1
LINK(N,NB)	The N^{th} open residual position in the NB^{th} block is at the LINK(N,NB) residual position (I=ICHK(LINK(N,NB),NB), J=JCHK(LINK(N,NB),NB)=IE(NB))
L1(NB)	The total number of elements in array RINV for blocks below block NB
L2(NB)	The total number of elements in array RINV1 for blocks below block NB
M*	The number of vertical mesh lines spanning the problem to be solved
MXGS*	The maximum number of guess points to be encountered over all blocks (can be larger)
N*	The number of horizontal mesh lines spanning the problem to be solved
NBLK*	The number of blocks (SEVP subregions) to be used
NEV(NB)	The number of starting vector (or residual) positions in block NB

(continued)

TABLE 1 (continued)

Variable Name	Variable Definition
NLINK(NB)	The number of open residual positions in block NB
NTYPE(I,J)*	Topological flag array (NTYPE(I,J)=1 if the partial difference operator is to be satisfied at the grid point located at the intersection of the I^{th} vertical mesh line and J^{th} horizontal mesh line; NTYPE(I,J)=0 otherwise)
Q(I,J)	Same as F(I,J)
RECUR(I,J)	Real array of 1's and 0's related to NTYPE
RINV	Influence coefficient array relating unit source terms at error vector positions to solutions at starting vector positions
RINV1	Influence coefficient array relating unit source terms at error vector positions to solutions at error vector positions
X(I,J)	Solution at mesh point (I,J)
XX(I,J)	Same as X(I,J)

*User-specified variable

5. REFERENCES

1. D. Dietrich, J. Meteor. Soc. Japan 52 (1974), pp. 337-338.

2. D. Dietrich, B. E. McDonald, and A. Warn-Varnas, J. Comp. Phys. 18 (1975), pp. 421-439.

3. D. Dietrich, J. Meteor. Soc. Japan 53 (1975), pp. 222-225.

4. D. Dietrich, Final Report, Contract No. N00014-77-C-0208 for the Naval Oceanographic Laboratory of the Naval Ocean Research and Development Activity (1977).

5. D. Dietrich, H. Klein, and R. K.-C. Chan, "FLAME: A Computer Model of Time-Dependent, Multidimensional, Multiphase Reactive Flow," paper presented at the 20th National Heat Transfer Conference, Milwaukee, Wisconsin, August 2-5, 1981.

6. I. Hirota, T. Tokioka, and M. Nishiguchi, J. Meteor. Soc. Japan 48 (1970), pp. 161-167.

7. R. Madala, Mon. Wea. Rev. 106 (1978), pp. 1735-1741.

8. J. J. O'Brien and H. E. Hurlburt, J. Phys. Oceanogr. 2 (1972), pp. 14-26.

9. P. J. Roache, "A New Direct Method for the Discretized Poisson Equation" in Proceedings of the Second International Conference of Numerical Methods in Fluid Dynamics, Berkeley, California, September 16-19, 1970, pp. 48-53, 1971.

10. P. J. Roache, Num. Heat Trans. 1 (1978), pp. 1-201.

11. A. Wallcraft, "Capacity Matrix Techniques," Ph.D. Thesis (1980), Department of Mathematics, Imperial College, London, England S.W.7.

6. BIBLIOGRAPHY

1. G. J. McAveney and L. M. Leslie, J. Meteor. Soc. Japan 50 (1972), pp. 136-137.

APPENDIX I

TRANSFORMATION TO DIRICHLET BOUNDARY CONDITIONS FROM NEUMAN OR MIXED BOUNDARY CONDITIONS

Using the notation of Eq. (1) in the main text, suppose the point $(i=I+1, j=J+1)$ is inside the boundary, whereby Eq. (1) applies with $i=I$ and $j=J$, and suppose that the point $(i=I, j=J+1)$ lies outside the boundary, whereby Eq. (1) does not apply with $i=I-1$ and $j=J$. Instead, we apply a "boundary condition" of the form

$$X_{I,J+1} = \beta - \alpha \cdot X_{I+1,J+1} \quad , \tag{3}$$

where α and β are specified. If $\alpha = 0$ we have a Dirichlet condition, and no adjustment is necessary before using program JAYNOR. However, if $\alpha \neq 0$ we can use Eq. (3) to eliminate $X_{I,J+1}$ from the equation set. We do this by substituting Eq. (3) into Eq. (1) with $i=I$ and $j=J$ to obtain

$$A'_{I,J} \cdot X_{I,J+1} + B_{I,J} \cdot X_{I+2,J+1} + C_{I,J} \cdot X_{I+1,J} + D_{I,J} \cdot X_{I+1,J+2} + E'_{I,J} \cdot X_{I+1,J+1} = F'_{I,J} \quad , \tag{4}$$

where $A'_{I,J} = 0$, $E'_{I,J} = E_{I,J} - A_{I,J} \cdot \alpha$, and $F'_{I,J} = F_{I,J} - A_{I,J} \cdot \beta$.

The modified partial difference equation (4) can be solved by program JAYNOR; the value of $X_{I,J+1}$ does not matter in this solution procedure, although the user is free to calculate the appropriate value, after JAYNOR gives the corrct $X_{I+1,J+1}$ value, using Eq. (3). Mixed boundary conditions in the other three directions from the point (I+1,J+1) can be treated in an analogous manner.

APPENDIX II

COMPUTER LISTINGS OF FOUR SEVP VERSIONS

The main text describes the most general version of program JAYNOR in detail. This is Version 2. The differences between each successive version and the previous one are indicated by comment cards in the main program JAYNOR.

The listings for four versions follow.

VERSION 2

```
OI          PROGRAM JAYNOR(INPUT,OUTPUT,TAPE5=TNPUT,TAPE6=OUTPUT)
00          COMMON/SEVP/AX(7,9),AY(7,9),BB(7,9),CX(7,9),CY(7,9),Q(7,9),F(7,9),
00         A RECUR(9,I1),RINV(588),RINV1(308),DUM0(I4,2),DUM1(14),
00         B DUM2(I4),X(9,11),H(9,11),ERR(9,1I),XX(9,II),ILFT(II),
00         C IRGT(1I),IGES(I4,3),JGES(14,3),TCHK(I4,3),JCHK(14,3),NEV(3),
00         D LTNK(7,2),NLINK(2),NTYPE(9,1I),TE(3),LI(3),L2(2)
00          DATA M,N,MXGS,NBLK,LDI,LD2/9,II,14,3,588,308/
00          DATA TE/4,7,10/
00    C  DIETRICH'S MODIFIED MADALA SOLVER, VERSION 2.
00    C  THIS VERSION TS IDENTICAL TO VERSION 1, EXCEPT
00    C  FOR MAKING RINV AND RINVI TNTO 1-DIMENSIONAL ARRAYS, AND MODIFYING
00    C  BOOKKEEPING TO REDUCE STORAGE REQUIREMENTS.
00    C EQUATION TO BE SOLVED IS
00    C  NTYPE(I+I,J+I) * (AX(I,J) * X(I,J+I) + CX(I,J) * X(I+2,J+1)
00    C                   + AY(I,J) * X(I+1,J) + CY(I,J) * X(T+1,J+2))
00    C                   + (2 - NTYPE(I+I,J+1)) * BB(I,J) * X(I+I,J+1)
00    C              = 2 * (1 - NTYPE(T+I,J+1)) * BB(I,J) * X(T+I,J+I)
00    C           + NTYPE(T+1,J+I) * F(I,J), ((I=0,I,...,M-I),(J=0,I,...,N-I))
00    C  WHERE TERMS OUTSIDE DIMENSION BOUNDS ARE INTERPRETED AS ZERO'S.
00    C  THUS, TF NTYPE(T,J)=0, X(I,J) IS LEFT UNCHANGED.
00          MI=M-I
00          N1=N-I
00          M2=M-2
00          N2=N-2
00          NBLK1=NBLK-I
00          DO 80 J=1,N
00          DO 80 I=1,M
00    8C    NTYPE(I,J)=0.
00          DO 90 J=2,NI
00          DO 90 I=2,MI
00    9C    NTYPE(T,J)=I
00          DO 2000 JC=2,NI
00          DO 2000 IC=2,M1
00          NTYPE(IC,JC)=0
00          NTYPE(IC-I,JC)=0
00          NTYPE(IC+I,JC)=C
00          NTYPE(IC,JC-1)=0
00          NTYPE(IC,JC+1)=0
00          WRITE(6,92) ((NTYPE(I,J),I=I,M),J=I,N)
00    92    FORMAT(IOX,9II)
01          CALL TOPOL(RECUR,ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,
00         A NTYPE,IE,M,N,MI,N1,M2,NBLK,NBLK1,MXGS,L1,L2,LDI,LD2)
00          DO 100 J=I,N2
00          DO I00 I=I,M2
00          AX(I,J)=1.+.0I*(T-4)**2
00          CX(I,J)=I.-.0I*(I-4)**2
00          AY(I,J)=I.+.0I*(J-5)**2
00          CY(I,J)=1.-.0I*(J-5)**2
00          BB(I,J)=-AX(I,J)-AY(I,J)-CX(I,J)-CY(T,J)-.I
00     100 CONTINUE
00          DENOM=(.25*M*N)**2
00          DO I70 J=I,N
00          DO I70 I=1,M
00          X(I,J)=(I-1.)*(I-M)*(J-I.)*(J-N)/DENOM
00          XX(T,J)=X(I,J)*(1.-NTYPE(T,J))
00     I70 CONTINUE
00          FSUM=0.
```

```
00          DO 171 J=2,N1
00          L=J-1
00          DO 171 I=2,M1
00          K=I-1
00          F(K,L)=AX(K,L)*X(K,J)+AY(K,L)*X(I,L)+CX(K,L)*X(I+1,J)+CY(K,L)*
00         A X(I,J+1)+BB(K,L)*X(I,J)
00    171   FSUM=FSUM+ABS(F(K,L))
00          FSUM=M2*N2/FSUM
00          T0=SECOND(TIM)
01          CALL JAY(AX,AY,BB,CX,CY,RINV,RINV1,H,RECUR,ILFT,IRGT,IGES,JGES,
00         AICHK,JCHK,NEV,LINK,NLINK,NTYPE,IE,M,N,M2,N2,NBLK,NBLK1,MXGS,L1,L2)
00          TIMSUM=0.
00          DO 181 J=2,N1
00          DO 181 I=2,M1
00          D(I-1,J-1)=F(I-1,J-1)
00    181   X(I,J)=0.
00          ALINE=0.
00          T0=SECOND(TIM)
01          CALL NOR(AX,AY,BB,CX,CY,RINV,RINV1,DUMD,DUM1,DUM2,D,H,XX,RECUR,
00         A ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,IE,M,N,M2,N2,NBLK,
00         B NBLK1,MXGS,L1,L2)
00          DO 990 J=2,N1
00          DO 990 I=2,M1
00    990   X(I,J)=X(I,J)+XX(I,J)
00          DO 991 J=1,N
00          DO 991 I=1,M
00    991   XX(I,J)=0.
00          DO 992 J=2,N1
00          L=J-1
00          DO 992 I=2,M1
00          K=I-1
00    992   D(K,L)=F(K,L)-AY(K,L)*X(I,J-1)-AX(K,L)*X(I-1,J)-BB(K,L)*X(I,J)-
00         A CX(K,L)*X(I+1,J)-CY(K,L)*X(I,J+1)
00   1000   CONTINUE
00    905   FORMAT(1X,1P15E8.1)
00          ERSUM=0.
00          DO 740 I=2,M1
00          K=I-1
00          DO 740 J=2,N1
00          L=J-1
00          ERR(I,J)=F(K,L)-AY(K,L)*X(I,J-1)-AX(K,L)*X(I-1,J)-BB(K,L)*
00         A X(I,J)-CX(K,L)*X(I+1,J)-CY(K,L)*X(I,J+1)
00          ERR(I,J)=ERR(I,J)*NTYPE(I,J)
00          ERSUM=ERSUM+ABS(ERR(I,J))
00    740   ERR(I,J)=ERR(I,J)*FSUM
00          ERSUM=ERSUM*FSUM/(M2*N2)
00          WRITE(6,904)
00    904   FORMAT(/40X,20HNORMALIZED RESIDUALS)
00          DO 742 I=2,M1
00          WRITE(6,905)(ERR(I,J),J=2,N1)
00    742   CONTINUE
00          WRITE(6,906) ERSUM
00    906   FORMAT(/37H MEAN ABSOLUTE NORMALIZED RESIDUAL = ,1PE9.2)
00          IF (ERSUM.GT.1.E-4) STOP
00          IF (IC.NE.2) NTYPE(IC-1,JC)=1
00          IF (IC.NE.M1) NTYPE(IC+1,JC)=1
00          IF (JC.NE.2) NTYPE(IC,JC-1)=1
00          IF (JC.NE.N1) NTYPE(IC,JC+1)=1
00   2000   NTYPE(IC,JC)=1
00          END
```

```
02            SUBROUTINE JAY(AX,AY,BB,CX,CY,RINV,RINVI,H,RECUR,ILFT,
00           1IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,NTYPE,IE,MO,NO,M2,N2,
00           A NBLK,NBLKI,MXGS,L1,L2)
00            DIMENSION AX(M2,N2),AY(M2,N2),BB(M2,N2),CX(M2,N2),CY(M2,N2),
00           A RINV(1),RINVI(1),RECUR(MO,NO),H(MO,NO),
00           B ILFT(NO),IRGT(NO),L1(NBLK),L2(NBLKI),IGES(MXGS,NBLK),
00           C JGES(MXGS,NBLK),ICHK(MXGS,NBLK),JCHK(MXGS,NBLK),NEV(NBLK),
00           D LINK(M2,NBLKI),NLINK(NBLKI),NTYPE(MO,NO),IE(NBLK)
00            JL=1
00            LO=L1(NBLK)
00            NB=0
00        100 NB=NB+1
00            NGES=NEV(NB)
00            NGES1=NEV(NB-1)
00            LA=L1(NB)
00            LB=L2(NB-1)
00            LC=L2(NB)
00            JM=IE(NB)
00            JMP=JM+1
00            JMM=JM-2
00            DO 250 NG=1,NGES
00            IG=IGES(NG,NB )
00            JG=JGES(NG,NB)
00            JF=JG-1
00            DO 210 J=JL,JMP
00            DO 210 I=1,MO
00        210 H(I,J)=0.
00            H(IG,JG)=1.
00            IF (JF.GT.JMM) GO TO 228
00            IF (NTYPE(IG,JF).EQ.0) GO TO 220
00            NOP=NLINK(NB-1)
00            DO 214 N=1,NOP
00            M=LINK(N,NB-1)
00            IF (ICHK(M,NB-1).EQ.IG.AND.JCHK(M,NB-1).EQ.JF) GO TO 215
00        214 CONTINUE
00        215 DO 218 N=1,NOP
00            L=LINK(N,NB-1)
00        218 H(ICHK(L,NB-1),JF)=RINVI(LB+(N-1)*NGES1+M)*CY(IG-1,JG-2)
00        220 DO 225 J=JF,JMM
00            IL=ILFT(J+2)-1
00            IR=IRGT(J+2)-1
00        222 DO 225 I=IL,IR
00        225 H(I+1,J+2)=-RECUR(I+1,J+2)*(AX(I,J)*H(I,J+1)+AY(I,J)*H(I+1,J)+
00           A BB(I,J)*H(I+1,J+1)+CX(I,J)*H(I+2,J+1))/CY(I,J)
00        228 DO 230 NC=1,NGES
00            I=ICHK(NC,NB)-1
00            J=JCHK(NC,NB)-1
00        230 RINV(LA+(NC-1)*NGES+NG)=AX(I,J)*H(I,J+1)+AY(I,J)*H(I+1,J)+BB(I,J)*
00           A H(I+1,J+1)+CX(I,J)*H(I+2,J+1)
00            IF (NB.EQ.NBLK) GO TO 250
00            NOP=NLINK(NB)
00            J=IE(NB)
00            DO 240 N=1,NOP
00            M=LINK(N,NB)
00        240 RINV(LO+(N-1)*NGES+NG)=H(ICHK(M,NB),J)
00        250 H(IG,JG)=0.
00            CALL MATINV(RINV(LA+1),NGES,NGES)
00            IF (NB.EQ.NBLK) RETURN
00            DO 260 I=1,NGES
00            DO 260 J=1,NOP
00            RINVI(LC+(J-1)*NGES+I)=0.
00            DO 260 K=1,NGES
00        260 RINVI(LC+(J-1)*NGES+I)=RINVI(LC+(J-1)*NGES+I)-
00           A RINV(LA+(K-1)*NGES+I)*RINV(LO+(J-1)*NGES+K)
00      C  RINVI(I,J,NB) IS THE ''ALMOST HOMOGENEOUS'' SOLUTION AT THE J-TH OPEN
00      C  RESIDUAL POSITION FORCED BY A RESIDUAL VALUE OF 1 AT THE I-TH
00      C  RESIDUAL POSITION. HOMOGENEOUS B.C.'S ARE ASSUMED EVERYWHERE,
00      C  INCLUDING THE TOP OF THE PRESENT SUBREGION NB.
00            JL=JM
00            GO TO 100
00            END
```

VERSION 2
(continued)

```
01         SUBROUTINE NOR(AX,AY,BB,CX,CY,RINV,RINV1,DUMD,DUM1,DUM2,F,H,X,
00        1RECUR,ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,IE,MD,ND,M2,
00        A N2,NBLK,NBLK1,MXGS,L1,L2)
00         DIMENSION AX(M2,N2),AY(M2,N2),BB(M2,N2),CX(M2,N2),CY(M2,N2),
00        A RINV(1),RINV1(1),RECUR(MD,ND),H(MD,ND),
00        B ILFT(ND),IRGT(ND),L1(NBLK),L2(NBLK1),IGES(MXGS,NBLK),
00        C JGES(MXGS,NBLK),ICHK(MXGS,NBLK),JCHK(MXGS,NBLK),NEV(NBLK),
00        D LINK(M2,NBLK1),NLINK(NBLK1),IE(NBLK)
00         DIMENSION DUMD(M2,NBLK1),DUM1(MXGS),DUM2(MXGS),F(M2,N2),X(MD,ND)
00         COMMON/BSM/AIT,ALINE,BIT,BLINE,ITMAX
00         JS=1
00         DO 150 NB=1,NBLK
00         LC=L2(NB)
00         JF=IE(NB)-2
00         DO 105 J=JS,JF
00         IL=ILFT(J+2)-1
00         IR=IRGT(J+2)-1
00         DO 105 I=IL,IR
00     105 X(I+1,J+2)=RECUR(I+1,J+2)*(F(I,J)-AX(I,J)*X(I,J+1)-AY(I,J)*
00        A X(I+1,J)-BB(I,J)*X(I+1,J+1)-CX(I,J)*X(I+2,J+1))/CY(I,J)+(1.-
00        B RECUR(I+1,J+2))*X(I+1,J+2)
00         IF (NB.EQ.NBLK) GO TO 150
00         NGES=NEV(NB)
00         DO 115 N=1,NGES
00         I=ICHK(N,NB)-1
00         J=JCHK(N,NB)-1
00     115 DUM1(N)=F(I,J)-AX(I,J)*X(I,J+1)-AY(I,J)*X(I+1,J)-BB(I,J)*
00        A X(I+1,J+1)-CX(I,J)*X(I+2,J+1)-CY(I,J)*X(I+1,J+2)
00         NOP=NLINK(NB)
00         J=IE(NB)
00         DO 120 N=1,NOP
00         DUM2(N)=0.
00         DO 118 M=1,NGES
00     118 DUM2(N)=DUM2(N)+DUM1(M)*RINV1(LC+(N-1)*NGES+M)
00         M=LINK(N,NB)
00         I=ICHK(M,NB)
00         DUMD(N,NB)=X(I,J)
00     120 X(I,J)=X(I,J)-DUM2(N)
00     150 JS=IE(NB)
00         DO 300 NB1=1,NBLK
00         NB=NBLK-NB1+1
00         JS=1
00         IF (NB.NE.1) JS=IE(NB-1)
00         JF=IE(NB)-2
00         LA=L1(NB)
00         LB=L2(NB-1)
00         NGES=NEV(NB)
00         IF (NB.EQ.NBLK) GO TO 201
00         J=IE(NB)
00         NOP=NLINK(NB)
00         DO 200 N=1,NOP
00         M=LINK(N,NB)
00         I=ICHK(M,NB)
00     200 X(I,J)=DUMD(N,NB)
00     201 CONTINUE
00         N=IE(NB)
00         DO 202 J=JS,N
```

```
      DO 202 I=1,MO
  202 H(I,J)=0.
      DO 210 N=1,NGES
      I=ICHK(N,NB)-1
      J=JCHK(N,NB)-1
  210 DUM1(N)=F(I,J)-AX(I,J)*X(I,J+1)-AY(I,J)*X(I+1,J)-BB(I,J)*
     A X(I+1,J+1)-CX(I,J)*X(I+2,J+1)-CY(I,J)*X(I+1,J+2)
      DO 220 N=1,NGES
      DUM2(N)=0.
      DO 218 M=1,NGES
  218 DUM2(N)=DUM2(N)+DUM1(M)*RINV(LA+(N-1)*NGES+M)
      I=IGES(N,NB)
      J=JGES(N,NB)
      H(I,J)=DUM2(N)
  220 X(I,J)=X(I,J)+DUM2(N)
      IF (NB.EQ.1) GO TO 250
      DO 222 N=1,MXGS
  222 DUM1(N)=0.
      NOP=NLINK(NB-1)
      J=JS
      DO 230 N=1,NOP
      M=LINK(N,NB-1)
      I=ICHK(M,NB-1)
  230 DUM1(M)=H(I,J+1)*CY(I-1,J-I)
C  WHEN BOUNDARY IS IRREGULAR, THERE IS SOME WASTED CALCULATION HERE IN
C  ORDER TO AVLID LOGICAL DECISIONS OR EXTRA STORAGE (NGES COULD BE
C  REPLACED BY NLINK(NB))
      NGES=NEV(NB-1)
      DO 240 N=1,NOP
      DUM2(N)=0.
      DO 238 M=1,NGES
  238 DUM2(N)=DUM2(N)+DUM1(M)*RINV1(LB+(N-1)*NGES+M)
      M=LINK(N,NB-1)
  240 H(ICHK(M,NB-1),JCHK(M,NB-1))=DUM2(N)
  250 DO 300 J=JS,JF
      IL=ILFT(J+2)-1
      IR=IRGT(J+2)-1
      DO 300 I=IL,IR
      DUM1(I)=-RECUR(I+1,J+2)*(AX(I,J)*H(I,J+1)+AY(I,J)*H(I+1,J)+
     A BB(I,J)*H(I+1,J+1)+CX(I,J)*H(I+2,J+1))/CY(I,J)
      H(I+1,J+2)=DUM1(I)+(1.-RECUR(I+1,J+2))*H(I+1,J+2)
  300 X(I+1,J+2)=X(I+1,J+2)+DUM1(I)
      END
```

VERSION 2
(continued)

```
      SUBROUTINE TOPOL(RECUR,ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,
     1NLINK,NTYPE,IE,M,N,M1,N1,M2,NBLK,NBLK1,MXGS,L1,L2,LD1,LD2)
      DIMENSION  RECUR(M,N),     ILFT(N),IRGT(N),IGES(MXGS,NBLK),
     A JGES(MXGS,NBLK),ICHK(MXGS,NBLK),JCHK(MXGS,NBLK),NEV(NBLK),
     BLINK(M2,NBLK1),NLINK(NBLK1),NTYPE(M,N),IE(NBLK),L1(NBLK),L2(NBLK))
      DO 10 J=2,N1
      I=0
    2 I=I+1
      IF (NTYPE(I,J)*NTYPE(I,J-1).EQ.0.AND.I.LT.M1) GO TO 2
      ILFT(J)=I
      I=M
    4 I=I-1
      IF (NTYPE(I,J)*NTYPE(I,J-1).EQ.0.AND.I.GT.1) GO TO 4
   10 IRGT(J)=I
      IRGT(1)=2
      ILFT(1)=2
      ILFT(N)=2
      IRGT(N)=2
      DO 15 J=2,N1
      IF (ILFT(J).LE.IRGT(J)) GO TO 15
      IRGT(J)=2
      ILFT(J)=2
   15 CONTINUE
      JMIN=2
      MXLNK=0
      L1(1)=0
      L2(1)=0
      DO 100 NB=1,NBLK
      NGES=0
      NCHK=0
      JMAX=IE(NB)
      DO 40 I=2,M1
      NM=NTYPE(I,JMIN-1)
      NU=NTYPE(I,JMIN)
      DO 40 J=JMIN,JMAX
      NL=NM
      NM=NU
      RECUR(I,J)=NM
      IF (NL.EQ.0) RECUR(I,J)=0.
      NU=NTYPE(I,J+1)
      IF (NM.EQ.0) GO TO 40
      IF (NL.EQ.1.AND.J.NE.JMIN) GO TO 20
      NGES=NGES+1
      IF (NGES.GT.MXGS) GO TO 200
      IGES(NGES,NB)=I
      JGES(NGES,NB)=J
   20 IF (NU.EQ.1.AND.J.NE.JMAX) GO TO 40
      NCHK=NCHK+1
      ICHK(NCHK,NB)=I
      JCHK(NCHK,NB)=J
   40 CONTINUE
      NEV(NB)=NGES
      IF (NB.EQ.NBLK) GO TO 100
      NLNK=0
      DO 80 K=1,NGES
      I=ICHK(K,NB)
      J=JCHK(K,NB)
      IF (NTYPE(I,J+1).EQ.0) GO TO 80
      NLNK=NLNK+1
      LINK(NLNK,NB)=K
   80 CONTINUE
      NLINK(NB)=NLNK
      MXLNK=MAX0(MXLNK,NLNK*NGES)
      L1(NB+1)=L1(NB)+NGES**2
      IF (NB.NE.NBLK1) L2(NB+1)=L2(NB)+NGES*NLNK
      LRINV1=L2(NB)+NGES*NLNK
  100 JMIN=JMAX+1
C  IF MXLNK.GT.NGES*NGES, THERE IS SOME WASTED STORAGE THAT IS VERY
C  DIFFICULT TO PROGRAM AROUND, BECAUSE OF THE DUAL ROLE OF THE LAST
C  BLOCK OF RINV, WHICH IS USED AS A SCRATCH AREA FOR RINV1 DATA.
      LRINV=L1(NBLK)+MAX0(MXLNK,NGES**2)
      WRITE(6,150) LRINV,LRINV1
  150 FORMAT(55H DIMENSIONS REQUIRED FOR RINV AND RINV1 ARE LD1, LD2 = ,
     AI5,2X,I5,1H./124H IF EITHER INPUT VALUE LD1 OR LD2 IS TOO SMALL, T
     BHE COMPUTATION IS HALTED HERE, AND THE USER MUST INCREASE LD1 AND/
     COR LD2 TO/22H THE INDICATED VALUES.)
      IF ((LD1.LT.LRINV).OR.(LD2.LT.LRINV1)) STOP
      RETURN
  200 WRITE(6,202)
  202 FORMAT(92H DIMENSION PARAMETER MXGS IS TOO SMALL FOR SPECIFIED TOP
     AOGRAPHY AND SUBREGION SPECIFICATION.)
      STOP
      END
```

VERSION 2
(continued)

```
00              SUBROUTINE MATINV(B,N,M)
00              DIMENSION B(N,1),B1(100),B2(100)
00              M1=M-1
00              DO 110 I=1,M1
00              B1(1)=1./B(I,I)
00              B(I,I)=1.0
00              DO 112 J=I,M
00        112   B(I,J)=B(I,J)*B1(1)
00              IP1=I+1
00              DO 120 I1=IP1,M
00        120   B1(I1)=B(I1,I)
00              DO 125 I1=IP1,M
00        125   B(I1,I)=0.
00              DO 127 J=1,M
00        127   B2(J)=B(I,J)
00              DO 135 I1=IP1,M
00              DO 135 J=1,M
00        135   B(I1,J)=B(I1,J)-B1(I1)*B2(J)
00        110   CONTINUE
00              B1(1)=1./B(M,M)
00              B(M,M)=1.
00              DO 140 J=1,M
00        140   B(M,J)=B(M,J)*B1(1)
00              DO 150 I=2,M
00              DO 155 I2=1,I
00        155   B1(I2)=B(I2,I)
00              IM1=I-1
00              DO 156 I2=1,IM1
00        156   B(I2,I)=0.
00              DO 157 J=1,M
00        157   B2(J)=B(I,J)
00              IM1=I-1
00              DO 160 I2=1,IM1
00              DO 160 J=1,M
00        160   B(I2,J)=B(I2,J)-B1(I2)*B2(J)
00        150   CONTINUE
00              END
```

VERSION 3

```
01          PROGRAM JAYNOR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
00          COMMON/SEVP/AX(7,9),AY(7,9),BB(7,9),CX(7,9),Q(7,9),F(7,9),
00         A RECUR(9,11),RINV(588),RINV1(308),DUMD(14,2),DUM1(14),
00         B DUM2(14),X(9,11),H(9,11),ERR(9,11),XX(9,11),ILFT(11),
00         C IRGT(11),IGES(14,3),JGES(14,3),ICHK(14,3),JCHK(14,3),NEV(3),
00         D LINK(7,2),NLINK(2),NTYPE(9,11),IE(3),L1(3),L2(2)
00          DATA M,N,MXGS,NBLK,LD1,LD2/9,11,14,3,588,308/
00          DATA IE/4,7,10/
00    C  DIETRICH'S MODIFIED MADALA SOLVER, VERSION 3.
00    C  THIS VERSION IS IDENTICAL TO VERSION 2, EXCEPT
00    C  CY HAS BEEN ELIMINATED.  (BY DIVIDING THROUGH THE ELLIPTIC EQUATION
00    C  BY THE COEFFICIENT OF X(I+1,J+2), THE USER CAN USUALLY GET THIS FORM
00    C  IN A STRAIGHTFORWARD MANNER.  IF THIS COEFFICIENT VANISHES NEAR UPPER
00    C  J-BOUNDARIES, A MODIFIED FORM OF THIS VERSION, IN WHICH THE RESIDUAL
00    C  CALCULATION AT THE CORRESPONDING ERROR POINTS IS MODIFIED, WOULD BE
00    C  USEFUL.)
00    C EQUATION TO BE SOLVED IS
00    C  NTYPE(I+1,J+1) * (AX(I,J) * X(I,J+1) + CX(I,J) * X(I+2,J+1)
00    C                   + AY(I,J) * X(I+1,J) +          X(I+1,J+2))
00    C                   + (2 - NTYPE(I+1,J+1)) * BB(I,J) * X(I+1,J+1)
00    C               = 2 * (1 - NTYPE(I+1,J+1)) * BB(I,J) * X(I+1,J+1)
00    C           + NTYPE(I+1,J+1) * F(I,J), ((I=0,1,...,M-1),(J=0,1,...,N-1))
00    C  WHERE TERMS OUTSIDE DIMENSION BOUNDS ARE INTERPRETED AS ZERO'S.
00    C  THUS, IF NTYPE(I,J)=0, X(I,J) IS LEFT UNCHANGED.
00          M1=M-1
00          N1=N-1
00          M2=M-2
00          N2=N-2
00          NBLK1=NBLK-1
00          DO 80 J=1,N
00          DO 80 I=1,M
00     80   NTYPE(I,J)=0.
00          DO 90 J=2,N1
00          DO 90 I=2,M1
00     90   NTYPE(I,J)=1
00          DO 2000 JC=2,N1
00          DO 2000 IC=2,M1
00          NTYPE(IC,JC)=0
00          NTYPE(IC-1,JC)=0
00          NTYPE(IC+1,JC)=0
00          NTYPE(IC,JC-1)=0
00          NTYPE(IC,JC+1)=0
00          WRITE(6,92) ((NTYPE(I,J),I=1,M),J=1,N)
00     92   FORMAT(10X,9I1)
01          CALL TOPOL(RECUR,ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,
00         A NTYPE,IE,M,N,M1,N1,M2,NBLK,NBLK1,MXGS,L1,L2,LD1,LD2)
00          DO 100 J=1,N2
00          DO 100 I=1,M2
00          AX(I,J)=1.+.01*(I-4)**2
00          CX(I,J)=1.-.01*(I-4)**2
00          AY(I,J)=1.+.01*(J-5)**2
00          BB(I,J)=(-AX(I,J)-AY(I,J)-CX(I,J)+.01*(J-5)**2-1.1)/(1.-.01*(J-5)
00         A**2)
00      100 CONTINUE
00          DENOM=(.25*M*N)**2
00          DO 170 J=1,N
00          DO 170 I=1,M
```

```
00          X(I,J)=(I-1.)*(I-M)*)J-1.)*(J-N)/DENOM
00          XX(I,J)=X(I,J)*(1.-NTYPE)I,J))
00     170  CONTINUE
00          FSUM=0.
00          DO 171 J=2,N1
00          L=J-1
00          DO 171 I=2,M1
00          K=I-1
00          F)K,L)=AX(K,L)*X)I,J)+AY(K,L)*X)I,L)+CX(K,L)*X(I+1,J)+
00         A X(I,J+1)+BB(K,L)*X(I,J)
00     171  FSUM=FSUM+ABS(F)K,L))
00          FSUM=M2*N2/FSUM
00          TO=SECOND(TIM)
01          CALL JAY)AX,AY,BB,CX,RINV,RINV1,H,RECUR,ILFT,IRGT,IGES,JGES,
00         AICHK,JCHK,NEV,LINK,NLINK,NTYPE,IE,M,N,M2,N2,NBLK,NBLK1,MXGS,L1,L2)
00          TIMSUM=0.
00          DO 181 J=2,N1
00          DO 181 I=2,M1
00          D(I-1,J-1)=F(I-1,J-1)
00     181  X(I,J)=0.
00          ALTNE=0.
00          TO=SECOND(TIM)
01          CALL NOR)AX,AY,BB,CX,RINV,RINV1,DUM0,DUM1,DUM2,D,H,XX,RECUR,
00         A ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,IE,M,N,M2,N2,NBLK,
00         B NBLK1,MXGS,L1,L2)
00          DO 990 J=2,N1
00          DO 990 I=2,M1
00     990  X)I,J)=X)I,J)+XX(I,J)
00          DO 991 J=1,N
00          DO 991 I=1,M
00     991  XX(I,J)=0.
00          DO 992 J=2,N1
00          L=J-1
00          DO 992 I=2,M1
00          K=I-1
00     992  D)K,L)=F(K,L)-AY(K,L)*X(I,J-1)-AX(K,L)*X(I-1,J)-BB)K,L)*X(I,J)-
00         A CX(K,L)*X(I+1,J)-X(I,J+1)
00    1000  CONTINUE
00     905  FORMAT(1X,1P15E8.1)
00          ERSUM=0.
00          DO 740 I=2,M1
00          K=I-1
00          DO 740 J=2,N1
00          L=J-1
00          ERR(I,J)=F)K,L)-AY(K,L)*X(I,J-1)-AX(K,L)*X(I-1,J)-BB(K,L)*
00         A X(I,J)-CX(K,L)*X(I+1,J)-X(I,J+1)
00          ERR)I,J)=ERR(I,J)*NTYPE(I,J)
00          ERSUM=ERSUM+ABS(ERR(I,J))
00     740  ERR(I,J)=ERR(I,J)*FSUM
00          ERSUM=ERSUM*FSUM/(M2*N2)
00          WRITE(6,904)
00     904  FORMAT(/40X,20HNORMALIZED RESIDUALS)
00          DO 742 I=2,M1
00          WRITE(6,905)(ERR(I,J),J=2,N1)
00     742  CONTINUE
00          WRITE(6,906) ERSUM
00     906  FORMAT(/37H MEAN ABSOLUTE NORMALIZED RESIDUAL = ,1PE9.2)
00          IF )ERSUM.GT.1.E-4) STOP
00          IF (IC.NE.2) NTYPE)IC-1,JC)=1
00          IF (IC.NE.M1) NTYPE(IC+1,JC)=1
00          IF (JC.NE.2) NTYPE(IC,JC-1)=1
00          IF (JC.NE.N1) NTYPE(IC,JC+1)=1
00    2000  NTYPE(IC,JC)=1
00          END
```

```
OI          SUBROUTINE JAY(AX,AY,BB,CX,RINV,RINV1,M,RECUR,ILFT,
00         1IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,NTYPE,IE,MD,ND,M2,N2,
00         A NBLK,NBLK1,MXGS,L1,L2)
00          DIMENSION AX(M2,N2),AY)M2,N2),BB(M2,N2),CX(M2,N2),
00         A RINV)1),RINV1(1),RECUP(MO,NO),H(MD,ND),
00         B ILFT(ND),IRGT(ND),L1(NBLK),L2(NBLK1),IGES(MXGS,NBLK),
00         C JGES(MXGS,NBLK),ICHK(MXGS,NBLK),JCHK(MXGS,NBLK),NEV(NBLK),
00         O LINK(M2,NBLK1),NLINK(NBLK1),NTYPE)MD,NO),IE(NBLK)
00          JL=1
00          LD=L1(NBLK)
00          NB=D
00     1CD  NB=NB+1
00          NGES=NEV)NB)
00          NGESI=NEV(NB-1)
00          LA=L1(NB)
00          LB=L2(NB-1)
00          LC=L2(NB)
00          JH=IE(NB)
00          JHP=JH+1
00          JHM=JH-2
00          DO 25D NG=1,NGES
00          IG=IGES(NG,NB )
00          JG=JGES(NG,NB)
00          JF=JG-I
00          DO 2ID J=JL,JHP
00          OO 2ID I=1,MD
00     2ID  H(1,J)=G.
00          H(IG,JG)=1.
00          IF )JF.GT.JHM) GO TO 228
00          IF )NTYPE(IG,JF).EQ.D) GO TO 220
00          NOP=NLINK)NB-1)
00          DO 214 N=1,NOP
00          M=LINK)N,NB-1)
00          IF (ICHK(M,NB-1).EO.IG.ANO.JCHK(M,NB-1).EQ.JF) GO TO 215
00     214  CONTINUE
00     215  OO 218 N=1,NOP
CO          L=LINK(N,NB-1)
00     218  H(ICHK(L,NB-1),JF)=RINV1)LB+)N-1)*NGES1+M)
00     220  DO 225 J=JF,JHM
00          IL=ILFT)J+2)-I
00          IR=IRGT(J+2)-1
GO     222  DO 225 I=IL,IR
00     225  H(I+1,J+2)=-RECUR(I+1,J+2)*(AX)I,J)*H)I,J+I)+AY)I,J)*H(I+1,J)+
00         A BB(I,J)*H(I+1,J+1)+CX)I,J)*H)I+2,J+I))
00     228  DO 230 NC=1,NGES
00          I=ICHK)NC,NB)-I
00          J=JCHK(NC,NB)-1
00     230  RINV(LA+(NC-1)*NGES+NG)=AX)I,J)*H)I,J+I)+AY(I,J)*H(I+1,J)+BB(I,J)+
00         A H(I+1,J+I)+CX(I,J)*H)I+2,J+I)
00          IF (NB.EO.NBLK) GO TO 250
00          NOP=NLINK)NB)
00          J=IE(NB)
00          DO 24C N=1,NOP
00          M=LINK)N,NB)
00     240  RINV(LD+)N-1)*NGES+NG)=H(ICHK(M,NB),J)
00     2ЕD  H(IG,JG)=D.
00          CALL MATINV(RINV(LA+1),NGES,NGES)
00          IF )NB.EO.NBLK) RETURN
00          DO 260 I=1.NGES
00          DO 260 J=1,NOP
00          RINV1(LC+(J-I)*NGES+I)=O.
00          DO 260 K=1,NGES
00     260  RINV1)LC+)J-1)*NGES+I)=RINV1(LC+)J-1)*NGES+1)-
00         A RINV)LA+)K-1)*NGES+I)*RINV(LD+(J-1)*NGES+K)
00     C  RINV1(1,J,NB) IS THE ''ALMOST HOMOGENEOUS'' SOLUIION AT THE J-TH OPEN
00     C  RESIOUAL POSITION FORCED BY A RESIOUAL VALUE OF 1 AT THE I-TH
00     C  RESIDUAL POSITION.  HOMOGENEOUS B.C.'S ARE ASSUMED EVERYWHERE.
00     C  INCLUOING THE TOP OF THE PRESENT SUBREGION NB.
CO          JL=JH
00          GO IO 100
00          END
```

VERSION 3
(continued)

```
01          SUBROUTINE NOR(AX,AY,BB,CX,RINV,RINV1,DUM0,DUM1,DUM2,F,H,X,
00         1RECUR,ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,IE,MD,ND,M2,
00         A N2,NBLK,NBLK1,MXGS,L1,L2)
00          DIMENSION AX(M2,N2),AY(M2,N2),BB(M2,N2),CX(M2,N2),
00         A RINV(1),RINV1(1),RECUR(MD,ND),H(MD,ND),
00         B ILFT(ND),IRGT(ND),L1(NBLK),L2(NBLK1),IGES(MXGS,NBLK),
00         C JGES(MXGS,NBLK),ICHK(MXGS,NBLK),JCHK(MXGS,NBLK),NEV(NBLK),
00         D LINK(M2,NBLK1),NLINK(NBLK1),IE(NBLK)
00          DIMENSION DUM0(M2,NBLK1),DUM1(MXGS),DUM2(MXGS),F(M2,N2),X(MD,ND)
00          JS=1
00          DO 150 NB=1,NBLK
00          LC=L2(NB)
00          JF=IE(NB)-2
00          DO 105 J=JS,JF
00          IL=ILFT(J+2)-1
00          IR=IRGT(J+2)-1
00          DO 105 I=IL,IR
00      105 X(I+1,J+2)=RECUR(I+1,J+2)*(F(I,J)-AX(I,J)*X(I,J+1)-AY(I,J)*
00         A X(I+1,J)-BB(I,J)*X(I+1,J+1)-CX(I,J)*X(I+2,J+1))+(1.-
00         B RECUR(I+1,J+2))*X(I+1,J+2)
00          IF (NB.EQ.NBLK) GO TO 150
00          NGES=NEV(NB)
00          DO 115 N=1,NGES
00          I=ICHK(N,NB)-1
00          J=JCHK(N,NB)-1
00      115 DUM1(N)=F(I,J)-AX(I,J)*X(I,J+1)-AY(I,J)*X(I+1,J)-BB(I,J)*
00         A X(I+1,J+1)-CX(I,J)*X(I+2,J+1)-X(I+1,J+2)
00          NOP=NLINK(NB)
00          J=IE(NB)
00          DO 120 N=1,NOP
00          DUM2(N)=0.
00          DO 118 M=1,NGES
00      118 DUM2(N)=DUM2(N)+DUM1(M)*RINV1(LC+(N-1)*NGES+M)
00          M=LINK(N,NB)
00          I=ICHK(M,NB)
00          DUM0(N,NB)=X(I,J)
00      120 X(I,J)=X(I,J)-DUM2(N)
00      150 JS=IE(NB)
00          DO 300 NB1=1,NBLK
00          NB=NBLK-NB1+1
00          JS=1
00          IF (NB.NE.1) JS=IE(NB-1)
00          JF=IE(NB)-2
00          LA=L1(NB)
00          LB=L2(NB-1)
00          NGES=NEV(NB)
00          IF (NB.EQ.NBLK) GO TO 201
00          J=IE(NB)
00          NOP=NLINK(NB)
00          DO 200 N=1,NOP
00          M=LINK(N,NB)
00          I=ICHK(M,NB)
00      200 X(I,J)=DUM0(N,NB)
00      201 CONTINUE
00          N=IE(NB)
00          DO 202 J=JS,N
00          DO 202 I=1,MD
```

```
00        202  H(I,J(=0.
00             DO 210 N=1,NGES
00             I=ICHK(N,NB)-1
00             J=JCHK(N,NB)-1
00        210  DUM1(N)=F(I,J(-AX(I,J(*X(I,J+1(-AY(I,J(*X(I+I,J(-BB(I,J(*
00            A X(I+1,J+1(-CX(I,J(*X(I+2,J+1(-X(I+1,J+2(
00             DO 220 N=1,NGES
00             DUM2(N)=0.
00             DO 218 M=1,NGES
00        218  DUM2(N)=DUM2(N)+DUM1(M(*RINV(LA+(N-1)*NGES+M)
00             I=IGES(N,NB(
00             J=JGES(N,NB)
00             H(I,J(=DUM2(N)
00        220  X(I,J(=X(I,J)+DUM2(N)
00             IF (NB.EQ.1) GO TO 250
00             DO 222 N=1,MXGS
00        222  DUM1(N)=0.
00             NOP=NLINK(NB-1)
00             J=JS
00             DO 230 N=1,NOP
00             M=LINK(N,NB-1)
00             I=ICHK(M,NB-1)
00        230  DUM1(M)=H(I,J+I(
00       C  WHEN BOUNDARY IS IRREGULAR, THERE IS SOME WASTED CALCULATION HERE IN
00       C  ORDER TO AVLID LOGICAL DECISIONS OR EXTRA STORAGE (NGES COULD BE
00       C  REPLACED BY NLINK(NB()
00             NGES=NEV(NB-1)
00             DO 240 N=1,NOP
00             DUM2(N)=0.
00             DO 238 M=1,NGES
00        238  DUM2(N(=DUM2(N)+DUM1(M(*RINV1(LB+(N-1(*NGES+M)
00             M=LINK(N,NB-1(
00        240  H(ICHK(M,NB-1),JCHK(M,NB-1)(=DUM2(N)
00        250  DO 300 J=JS,JF
00             IL=ILFT(J+2(-1
00             IR=IRGT(J+2)-1
00             DO 300 I=IL,IR
00             DUM1(I)=-RECUR(I+1,J+2)*(AX(I,J)*H(I,J+1(+AY(I,J(*H(I+1,J(+
00            A BB(I,J(*H(I+1,J+1)+CX(I,J)*H(I+2,J+1((
00             H(I+1,J+2)=DUM1(I(+(1.-RECUR(I+1,J+2)(*H(I+1,J+2(
00        300  X(I+1,J+2(=X(I+1,J+2(+DUM1(I)
00             END
```

VERSION 3
(continued)

```
01          SUBROUTINE TOPOL(RECUR,ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,
00         1NLINK,NTYPE,IE,M,N,MI,N1,M2,NBLK,NBLK1,MXGS,L1,L2,LO1,LO2(
00          DIMENSION  RECUR(M,N(,     ILFT(N(,IRGT(N),IGES(MXGS,NBLK),
00         A JGES(MXGS,NBLK(,ICHK(MXGS,NBLK(,JCHK(MXGS,NBLK(,NEV(NBLK(,
00         BLINK(M2,NBLK1),NLINK(NBLK1(,NTYPE(M,N(,IE(NBLK(,LI(NBLK(,L2(NBLK1)
00          DO 10 J=2,N1
00          I=0
00     2    I=I+1
00          IF (NTYPE(I,J(*NTYPE(I,J-1(.EQ.0.AND.I.LT.M1) GO TO 2
00          ILFT(J)=I
00          I=M
00     4    I=I-1
00          IF (NTYPE(I,J)*NTYPE(I,J-1(.EQ.0.AND.I.GT.1) GO TO 4
00    10    IRGT(J(=I
00          IRGT(1(=2
00          ILFT(1(=2
00          ILFT(N(=2
00          IRGT(N(=2
00          DO 15 J=2,N1
00          IF (ILFT(J).LE.IRGT(J)( GO TO 15
00          IRGT(J(=2
00          ILFT(J(=2
00    15    CONTINUE
00          JMIN=2
00          MXLNK=0
00          L1(1(=0
00          L2(1(=0
00          DO 100 NB=1,NBLK
00          NGES=0
00          NCHK=0
00          JMAX=IE(NB)
00          DO 40 I=2,M1
00          NM=NTYPE(I,JMIN-1)
00          NU=NTYPE(I,JMIN(
00          DO 40 J=JMIN,JMAX
00          NL=NM
00          NM=NU
00          RECUR(I,J(=NM
00          IF (NL.EQ.0( RECUR(I,J(=0.
00          NU=NTYPE(I,J+1(
00          IF (NM.EQ.0) GO TO 40
00          IF (NL.EQ.1.AND.J.NE.JMIN) GO TO 20
00          NGES=NGES+1
00          IF (NGES.GT.MXGS( GO TO 200
00          IGES(NGES,NB)=I
00          JGES(NGES,NB)=J
00    20    IF (NU.EQ.1.AND.J.NE.JMAX) GO TO 40
00          NCHK=NCHK+1
00          ICHK(NCHK,NB)=I
00          JCHK(NCHK,NB)=J
00    40    CONTINUE
00          NEV(NB)=NGES
00          IF (NB.EQ.NBLK) GO TO 100
00          NLNK=0
00          DO 80 K=1,NGES
00          I=ICHK(K,NB)
00          J=JCHK(K,NB)
00          IF (NTYPE(I,J+1).EQ.0) GO TO 80
00          NLNK=NLNK+1
00          LINK(NLNK,NB)=K
00    80    CONTINUE
00          NLINK(NB)=NLNK
00          MXLNK=MAX0(MXLNK,NLNK*NGES)
00          L1(NB+1)=L1(NB(+NGES**2
00          IF (NB.NE.NBLK1) L2(NB+1(=L2(NB)+NGES*NLNK
00          LRINVI=L2(NB)+NGES*NLNK
00   100    JMIN=JMAX+1
00    C  IF MXLNK.GT.NGES*NGES, THERE IS SOME WASTED STORAGE THAT IS VERY
00    C  DIFFICULT TO PROGRAM AROUND, BECAUSE OF THE DUAL ROLE OF THE LAST
00    C  BLOCK OF RINV, WHICH IS USED AS A SCRATCH AREA FOR RINVI DATA.
00          LRINV=L1(NBLK)+MAX0(MXLNK,NGES**2)
00          WRITE(6,150( LRINV,LRINVI
00   150    FORMAT(55H DIMENSIONS REQUIRED FOR RINV AND RINVI ARE LO1, LO2 = ,
00         AI5,2X,I5,1H./124H IF EITHER INPUT VALUE LO1 OR LO2 IS TOO SMALL, T
00         BHE COMPUTATION IS HALTED HERE, AND THE USER MUST INCREASE LO1 AND/
00         COR LO2 TO/22H THE INDICATED VALUES.)
00          IF ((LO1.LT.LRINV).OR.(LO2.LT.LRINVI)) STOP
00          RETURN
00   200    WRITE(6,202)
00   202    FORMAT(92H DIMENSION PARAMETER MXGS IS TOO SMALL FOR SPECIFIED TOP
00         AOGRAPHY AND SUBREGION SPECIFICATION.(
00          STOP
00          END
```

VERSION 3
(continued)

```
00          SUBROUTINE MATINV(B,N,M)
00          DIMENSION B(N,1),B1(100),B2(100)
00          M1=M-1
00          DO 110 I=1,M1
00          B1(I)=1./B(I,I)
00          B(I,I)=1.0
00          DO 112 J=1,M
00     112  B(I,J)=B(I,J)*B1(I)
00          IP1=I+1
00          DO 120 I1=IP1,M
00     120  B1(I1)=B(I1,I)
00          DO 125 I1=IP1,M
00     125  B(I1,I)=0.
00          DO 127 J=1,M
00     127  B2(J)=B(I,J)
00          DO 135 I1=IP1,M
00          DO 135 J=1,M
00     135  B(I1,J)=B(I1,J)-B1(I1)*B2(J)
00     110  CONTINUE
00          B1(I)=1./B(M,M)
00          B(M,M)=1.
00          DO 140 J=1,M
00     140  B(M,J)=B(M,J)*B1(I)
00          DO 150 I=2,M
00          DO 155 I2=1,I
00     155  B1(I2)=B(I2,I)
00          IM1=I-1
00          DO 156 I2=1,IM1
00     156  B(I2,I)=0.
00          DO 157 J=1,M
00     157  B2(J)=B(I,J)
00          IM1=I-1
00          DO 160 I2=1,IM1
00          DO 160 J=1,M
00     160  B(I2,J)=B(I2,J)-B1(I2)*B2(J)
00     150  CONTINUE
00          END
```

VERSION 4

```
00        PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
00        COMMON/SEVP/AX(7,9),AY(7,9(,BB(7,9(,CX(7,9(,O(7,9),F(7,9(,
00       A RINV(S88),RINVI(308),OUMO(14,2(,OUM1(14(,
00       B DUM2(14(,X(9,1I),H(9,11(,ERR(9,I1(,XX(9,I1(,ILFT(1I(,
00       C IRGT(1I),IGES(I4,3),JGES(14,3(,ICHK(I4,3(,JCHK(14,3(,NEV(3(,
00       O LINK(7,2(,NLINK(2(,NTYPE(9,1I(,IE(3(,LI(3(,L2(2)
00        OATA M,N,MXGS,NBLK,LOI,LO2/9,I1,14.3,588,308/
00        OATA IE/4,7,1G/
00    C  OIETRICH'S MOOIFIEO MAOALA SOLVER, VERSION 4.
00    C  THIS VERSION IS IOENIICAL IO VERSION 3, EXCEPT
00    C  NO ISLANOS ARE ALLOWEO ANO IRREGULAR BOUNDARIES MUST BE SUCH THAT ALL
00    C  NONZERO NTYPE VALUES IN A GIVEN ROW MUST BE CONTIGUOUS, THEREBY
00    C  REQUIRING THAT THE RIGHT ANO LEFI BOUNDARY CURVES BE SINGLE VALUEO
00    C  FUNCTIONS OF Y.  THIS ELIMINATES THE NEEO FOR ARRAY RECUR AND
00    C  ASSOCIATEO COMPUTATIONS.
00    C EQUATION TO BE SOLVEO IS
00    C  NTYPE(1+1,J+I) * (AX(I,J( * X(I,J+I( + CX(I,J) * X(I+2,J+1(
00    C                    + AY(I,J) * X(I+I,J( +            X(I+1,J+2)(
00    C                    + (2 - NTYPE(I+1,J+1(( * BB(I,J( * X(I+1,J+1)
00    C               = 2 * (1 - NTYPE(I+I,J+1() * BB(I,J( * X(I+1,J+1)
00    C            + NTYPE(I+1,J+1( * F(I,J(, ((I=0,I,....M-I(,(J=0,1,....,N-1)(
00    C  WHERE IERMS OUTSIOE OIMENSION BOUNOS ARE INTERPRETEO AS ZERO'S.
00    C  THUS, IF NTYPE(I,J(=0, X(I,J( IS LEFT UNCHANGEO.
00        MI=M-1
00        N1=N-1
00        M2=M-2
00        N2=N-2
00        NBLK1=NBLK-I
00        DO 80 J=1,N
00        DO 80 I=1,M
00     8C NTYPE(I,J(=0.
00        DO 90 J=2,N1
00        DO 90 I=2,M1
00     9C NTYPE(I,J(=1
01        NTYPE(2,3(,NTYPE(2,4),NTYPE(3,4(,NTYPE(2,5(,NTYPE(3,5(,NTYPE(2,6)
00       A =0
00        WRITE(6,92) ((NTYPE(I,J),I=I,M(,J=1,N(
00     92 FORMAI(IOX,9I1(
02        CALL TOPOL(ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,
00       A NTYPE,IE,M,N,M1,N1,M2,NBLK,NBLK1,MXGS,L1,L2,LOI,LO2(
00        DO IOO J=I,N2
00        DO 100 I=I,M2
00        AX(I,J)=1.+.0I*(I-4(**2
00        CX(1,J(=1.-.0I*(I-4)**2
00        AY(1,J(=1.+.01*(J-S(**2
00        BB(1,J)=(-AX(I,J)-AY(I,J(-CX(I,J(+.01*(J-S(**2-I.1)/(1.-.01*(J-5(
00       A**2(
00    100 CONIINUE
00        DENOM=(.2S*M*N)**2
00        DO 170 J=1,N
00        DO 170 I=I,M
00        X(I,J)=(I-1.)*(I-M(*(J-I.)*(J-N)/DENOM
00        XX(I,J)=X(I,J(*(1.-NTYPE(1,J)(
00    170 CONTINUE
00        FSUM=0.
00        DO 171 J=2,N1
00        L=J-1
```

VERSION 4
(continued)

```
00            DO I7I I=2,MI
00            K=I-I
00            F(K,L(=AX(K,L)*X(I,J)+AY(K,L(*X(I,L(+CX(K,L(*X(I+I,J(+
00           A X(I,J+1)+BB(K,L(*X(I,J(
00      I7I   FSUM=FSUM+ABS(F(K,L)(
00            FSUM=M2*N2/FSUM
00            TO=SECOND(TIM(
02            CALL JAY(AX,AY,BB,CX,RINV,RINVI,M,ILFI,IRGT,IGES,JGES,
00           AICHK,JCMK,NEV,LINK,NLINK,NTYPE,IE,M,N,M2,N2,NBLK,NBLKI,MXGS,LI,L2(
00            TIMSUM=0.
00            DO I8I J=2,NI
00            DO I8I I=2,MI
00            O(I-I,J-I(=F(I-I,J-I(
00      18I   X(I,J(=0.
00            ALINE=0.
00            TO=SECOND(TIM(
02            CALL NOR(AX,AY,BB,CX,RINV,RINV1,DUMO,DUMI,DUM2,O,M,XX,
00           A ILFT,IRGI,IGES,JGES,ICHK,JCMK,NEV,LINK,NLINK,IE,M,N,M2,N2,NBLK,
00           B NBLK1,MXGS,LI,L2(
00            DO 990 J=2,NI
00            DO 990 I=2,MI
00      990   X(I,J)=X(I,J)+XX(I,J(
00            DO 991 J=I,N
00            DO 99I I=I,M
00      991   XX(I,J(=0.
00            DO 992 J=2,NI
00            L=J-I
00            DO 992 1=2,MI
00            K=I-1
00      992   O(K,L(=F(K,L)-AY(K,L)*X(I,J-I)-AX(K,L(*X(I-1,J)-BB(K,L)*X(I,J)-
00           A CX(K,L(*X(I+I,J(-X(I,J+I(
00      1C00 CONTINUE
00      9CS   FORMAI(IX,IPISE8.I(
00            ERSUM=0.
00            DO 740 I=2,MI
00            K=I-1
00            DO 740 J=2,NI
00            L=J-1
00            ERR(I,J(=F(K,L(-AY(K,L(*X(I,J-1)-AX(K,L)*X(I-I,J)-BB(K,L)*
00           A X(I,J(-CX(K,L(*X(I+I,J(-X(I,J+I(
00            ERR(I,J(=ERR(1,J)*NTYPE(I,J(
00            ERSUM=ERSUM+ABS(ERR(I,J()
00      740   ERR(I,J(=ERR(I,J(*FSUM
00            ERSUM=ERSUM*FSUM/(M2*N2)
00            WRITE(6,904)
00      9C4   FORMAI(/40X,20HNORMALIZED RESIDUALS)
00            DO 742 I=2,MI
00            WRIIE(6,905((ERR(I,J(,J=2,N1)
00       742  CONTINUE
00            WRITE(6,906) ERSUM
00      9C6   FORMAT(/37H MEAN ABSOLUTE NORMALIZED RESIDUAL = ,IPE9.2(
00            IF (FRSUM.GT.1.E-4) SIOP
00            END
```

VERSION 4
(continued)

```
01        SUBROUTINE JAY(AX,AY,BB,CX,RINV,RINV1,H,ILFT,
00       1IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,NTYPE,IE,MD,ND,M2,N2,
00       A NBLK,NBLK1,MXGS,L1,L2)
00        DIMENSION AX(M2,N2),AY(M2,N2),BB(M2,N2),CX(M2,N2),
00       A RINV(1),RINV1(1),H(MD,ND),
00       B ILFT(ND),IRGT(ND),L1(NBLK),L2(NBLK1),IGES(MXGS,NBLK),
00       C JGES(MXGS,NBLK),ICHK(MXGS,NBLK),JCHK(MXGS,NBLK),NEV(NBLK),
00       D LINK(M2,NBLK1),NLINK(NBLK1),NTYPE(MD,ND),IE(NBLK)
00        JL=1
00        LD=L1(NBLK)
00        NB=0
00    100 NB=NB+1
00        NGES=NEV(NB)
00        NGES1=NEV(NB-1)
00        LA=L1(NB)
00        LB=L2(NB-1)
00        LC=L2(NB)
00        JH=IE(NB)
00        JHP=JH+1
00        JHM=JH-2
00        DO 250 NG=1,NGES
00        IG=IGES(NG,NB )
00        JG=JGES(NG,NB)
00        JF=JG-1
00        DO 210 J=JL,JHP
00        DO 210 I=1,MD
00    210 H(I,J)=0.
00        H(IG,JG)=1.
00        IF (JF.GT.JHM) GO TO 228
00        IF (NTYPE(IG,JF).EQ.0) GO TO 220
00        NOP=NLINK(NB-1)
00        DO 214 N=1,NOP
00        M=LINK(N,NB-1)
00        IF (ICHK(M,NB-1).EQ.IG.AND.JCHK(M,NB-1).EQ.JF) GO TO 215
00    214 CONTINUE
00    215 DO 218 N=1,NOP
00        L=LINK(N,NB-1)
00    218 H(ICHK(L,NB-1),JF)=RINV1(LB+(N-1)*NGES1+M)
00    220 DO 225 J=JF,JHM
00        IL=ILFT(J+2)-1
00        IR=IRGT(J+2)-1
00    222 DO 225 I=IL,IR
00    225 H(I+1,J+2)=-(AX(I,J)*H(I,J+1)+AY(I,J)*H(I+1,J)+
00       A BB(I,J)*H(I+1,J+1)+CX(I,J)*H(I+2,J+1))
00    228 DO 230 NC=1,NGES
00        I=ICHK(NC,NB)-1
00        J=JCHK(NC,NB)-1
00    230 RINV(LA+(NC-1)*NGES+NG)=AX(I,J)*H(I,J+1)+AY(I,J)*H(I+1,J)+BB(I,J)*
00       A H(I+1,J+1)+CX(I,J)*H(I+2,J+1)
00        IF (NB.EQ.NBLK) GO TO 250
00        NOP=NLINK(NB)
00        J=IE(NB)
00        DO 240 N=1,NOP
00        M=LINK(N,NB)
00    240 RINV(LD+(N-1)*NGES+NG)=H(ICHK(M,NB),J)
00    250 H(IG,JG)=0.
00        CALL MATINV(RINV(LA+1),NGES,NGES)
00        IF (NB.EQ.NBLK) RETURN
00        DO 260 I=1,NGES
00        DO 260 J=1,NOP
00        RINV1(LC+(J-1)*NGES+I)=0.
00        DO 260 K=1,NGES
00    260 RINV1(LC+(J-1)*NGES+I)=RINV1(LC+(J-1)*NGES+I)-
00       A RINV(LA+(K-1)*NGES+I)*RINV(LD+(J-1)*NGES+K)
00    C  RINV1(I,J,NB) IS THE ''ALMOST HOMOGENEOUS'' SOLUTION AT THE J-TH OPEN
00    C  RESIDUAL POSITION FORCED BY A RESIDUAL VALUE OF 1 AT THE I-TH
00    C  RESIDUAL POSITION.  HOMOGENEOUS B.C.'S ARE ASSUMED EVERYWHERE,
00    C  INCLUDING THE TOP OF THE PRESENT SUBREGION NB.
00        JL=JH
00        GO TO 100
00        END
```

```
01          SUBROUTINE NOR(AX,AY,BB,CX,RINV,RINV1,DUM0,DUM1,DUM2,F,H,X,
00         1ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,NLINK,IE,MO,NO,M2,
00         A N2,NBLK,NBLK1,MXGS,L1,L2)
00          DIMENSION AX(M2,N2),AY(M2,N2),BB(M2,N2(,CX(M2,N2),
0C         A RINV(1(,RINV1(1),H(MO,NO),
00         B ILFT(NO(,IRGT(NO(,L1(NBLK(,L2(NBLK1),IGES(MXGS,NBLK(,
00         C JGES(MXGS,NBLK),ICHK(MXGS,NBLK),JCHK(MXGS,NBLK),NEV(NBLK),
00         D LINK(M2,NBLK1),NLINK(NBLK1),IE(NBLK)
00          DIMENSION DUM0(M2,NBLK1),DUM1(MXGS),DUM2(MXGS),F(M2,N2),X(MO,NO)
00          JS=1
00          DO 150 NB=1,NBLK
00          LC=L2(NB)
00          JF=IE(NB)-2
00          DO 105 J=JS,JF
00          IL=ILFT(J+2)-1
00          IR=IRGT(J+2)-1
00          DO 105 I=IL,IR
00     1C5  X(I+1,J+2)=F(I,J)-AX(I,J)*X(I,J+1(-AY(I,J(*X(I+1,J(-BB(I,J)*
00         A X(I+1,J+1)-CX(I,J)*X(I+2,J+I)
00          IF (NB.EQ.NBLK) GO TO I50
00          NGES=NEV(NB)
00          DO 115 N=1,NGES
00          I=ICHK(N,NB)-1
G0          J=JCHK(N,NR)-1
00     I15  DUM1(N)=F(I,J)-AX(I,J)*X(I,J+I)-AY(I,J)*X(I+1,J)-BB(I,J)*
00         A X(I+1,J+1(-CX(I,J)*X(I+2,J+1)-X(I+I,J+2)
00          NOP=NLINK(NB)
00          J=IE(NB)
00          DO 120 N=1,NOP
00          DUM2(N)=0.
00          DO 1I8 M=1,NGES
00     I18  DUM2(N)=DUM2(N)+DUM1(M)*RINV1(LC+(N-1)*NGES+M)
C0          M=LINK(N,NB)
00          I=ICHK(M,NB)
00          DUM0(N,NB)=X(I,J)
00     120  X(I,J)=X(I,J)-DUM2(N)
00     I50  JS=IE(NB)
00          DO 300 NB1=1,NBLK
00          NB=NBLK-NB1+1
00          JS=1
00          IF (NB.NE.I) JS=IE(NB-1)
0C          JF=IE(NB)-2
00          LA=L1(NB)
00          LB=L2(NB-1)
00          NGES=NEV(NB)
00          IF (NB.EQ.NBLK) GO TO 201
00          J=IE(NB)
C0          NOP=NLINK(NB)
00          DO 200 N=1,NOP
00          M=LINK(N,NB)
00          I=ICHK(M,NB)
00     2C0  X(I,J)=DUM0(N,NB)
00     2C1  CONTINUE
00          N=IE(NB)
00          DO 202 J=JS,N
00          DO 202 I=1,MO
```

```
00       202  H(I,J)=0.
00            DO 210 N=1,NGES
00            I=ICHK(N,NB)-1
00            J=JCHK(N,NB)-1
00       210  DUM1(N)=F(I,J)-AX(I,J)*X(I,J+1)-AY(I,J)*X(I+1,J)-BB(I,J)*
00           A X(I+1,J+1)-CX(I,J)*X(I+2,J+1)-X(I+1,J+2)
00            DO 220 N=1,NGES
00            DUM2(N)=0.
00            DO 218 M=1,NGES
00       218  DUM2(N)=DUM2(N)+DUM1(M)*RINV(LA+(N-1)*NGES+M)
00            I=IGES(N,NB)
00            J=JGES(N,NB)
00            H(I,J)=DUM2(N)
00       220  X(I,J)=X(I,J)+DUM2(N)
00            IF (NB.EQ.1) GO TO 250
00            DO 222 N=1,MXGS
00       222  DUM1(N)=0.
00            NOP=NLINK(NB-1)
00            J=JS
00            DO 230 N=1,NOP
00            M=LINK(N,NB-1)
00            I=ICHK(M,NB-1)
00       230  DUM1(M)=H(I,J+1)
00      C  WHEN BOUNDARY IS IRREGULAR, THERE IS SOME WASTED CALCULATION HERE IN
00      C  ORDER TO AVLID LOGICAL DECISIONS OR EXTRA STORAGE (NGES COULD BE
00      C  REPLACED BY NLINK(NB))
00            NGES=NEV(NB-1)
00            DO 240 N=1,NOP
00            DUM2(N)=0.
00            DO 238 M=1,NGES
00       238  DUM2(N)=DUM2(N)+DUM1(M)*RINVI(LB+(N-1)*NGES+M)
00            M=LINK(N,NB-1)
00       240  H(ICHK(M,NB-1),JCHK(M,NB-1))=DUM2(N)
00       250  DO 300 J=JS,JF
00            IL=ILFT(J+2)-1
00            IR=IRGT(J+2)-1
00            DO 300 I=IL,IR
00            H(I+1,J+2)=-(AX(I,J)*H(I,J+1)+AY(I,J)*H(I+1,J)+
00           A BB(I,J)*H(I+1,J+1)+CX(I,J)*H(I+2,J+1))
00       300  X(I+1,J+2)=X(I+1,J+2)+H(I+1,J+2)
00            END
```

```
00              SUBROUTINE TOPOL(ILFT,IRGT,IGES,JGES,ICHK,JCHK,NEV,LINK,
00             1NLINK,NTYPE,IE,M,N,M1,N1,M2,NBLK,NBLK1,MXGS,L1,L2,LO1,LO2)
00              DIMENSION ILFT(N),IRGT(N),IGES(MXGS,NBLK),
00             A JGES(MXGS,NBLK),ICHK(MXGS,NBLK),JCHK(MXGS,NBLK),NEV(NBLK),
00             BLINK(M2,NBLK1),NLINK(NBLK1),NTYPE(M,N),IE(NBLK),L1(NBLK),L2(NBLK1)
00              DO 10 J=2,N1
00              I=0
00        2     I=I+1
00              IF (NTYPE(I,J)*NTYPE(I,J-1).EQ.0.AND.I.LT.M1) GO TO 2
00              ILFT(J)=I
00              I=M
00        4     I=I-1
00              IF (NTYPE(I,J)*NTYPE(I,J-1).EQ.0.AND.I.GT.1) GO TO 4
00        10    IRGT(J)=I
00              IRGT(1)=2
00              ILFT(1)=2
00              ILFT(N)=2
00              IRGT(N)=2
00              DO 15 J=2,N1
00              IF (ILFT(J).LE.IRGT(J)) GO TO 15
00              IRGT(J)=2
00              ILFT(J)=2
00        15    CONTINUE
00              JMIN=2
00              MXLNK=0
00              L1(1)=0
00              L2(1)=0
00              DO 100 NB=1,NBLK
00              NGES=0
00              NCHK=0
00              JMAX=IE(NB)
00              DO 40 I=2,M1
00              NM=NTYPE(I,JMIN-1)
00              NU=NTYPE(I,JMIN)
00              DO 40 J=JMIN,JMAX
00              NL=NM
00              NM=NU
00              NU=NTYPE(I,J+1)
00              IF (NM.EQ.0) GO TO 40
00              IF (NL.EQ.1.AND.J.NE.JMIN) GO TO 20
00              NGES=NGES+1
00              IF (NGES.GT.MXGS) GO TO 200
00              IGES(NGES,NB)=I
00              JGES(NGES,NB)=J
00        20    IF (NU.EQ.1.AND.J.NE.JMAX) GO TO 40
00              NCHK=NCHK+1
00              ICHK(NCHK,NB)=I
00              JCHK(NCHK,NB)=J
00        40    CONTINUE
00              NEV(NB)=NGES
00              IF (NB.EQ.NBLK) GO TO 100
00              NLNK=0
00              DO 80 K=1,NGES
00              I=ICHK(K,NB)
00              J=JCHK(K,NB)
00              IF (NTYPE(I,J+1).EQ.0) GO TO 80
00              NLNK=NLNK+1
00              LINK(NLNK,NB)=K
00        80    CONTINUE
00              NLINK(NB)=NLNK
00              MXLNK=MAX0(MXLNK,NLNK*NGES)
00              L1(NB+1)=L1(NB)+NGES**2
00              IF (NB.NE.NBLK1) L2(NB+1)=L2(NB)+NGES*NLNK
00              LRINV1=L2(NB)+NGES*NLNK
00        100   JMIN=JMAX+1
00       C  IF MXLNK.GT.NGES*NGES, THERE IS SOME WASTED STORAGE THAT IS VERY
00       C  DIFFICULT TO PROGRAM AROUND, BECAUSE OF THE DUAL ROLE OF THE LAST
00       C  BLOCK OF RINV, WHICH IS USED AS A SCRATCH AREA FOR RINV1 DATA.
00              LRINV=L1(NBLK)+MAX0(MXLNK,NGES**2)
00              WRITE(6,150) LRINV,LRINV1
00        150   FORMAT(55H DIMENSIONS REQUIRED FOR RINV AND RINV1 ARE LO1, LO2 = ,
00             AI5,2X,I5,1H./124H IF EITHER INPUT VALUE LO1 OR LO2 IS TOO SMALL, T
00             BHE COMPUTATION IS HALTED HERE, AND THE USER MUST INCREASE LO1 AND/
00             COR LO2 TO/22H THE INDICATED VALUES.)
00              IF ((LO1.LT.LRINV).OR.(LO2.LT.LRINV1)) STOP
00              RETURN
00        200   WRITE(6,202)
00        202   FORMAT(92H DIMENSION PARAMETER MXGS IS TOO SMALL FOR SPECIFIED TOP
00             AOGRAPHY AND SUBREGION SPECIFICATION.)
00              STOP
00              END
```

VERSION 4
(continued)

```
00          SUBROUTINE MATINV(B,N,M)
00          DIMENSION B(N,1),B1(100),B2(100)
00          M1=M-1
00          DO 110 I=1,M1
00          B1(1)=1./B(I,I)
00          B(I,I)=1.0
00          DO 112 J=1,M
00    112   B(I,J)=B(I,J)*B1(1)
00          IP1=I+1
00          DO 120 I1=IP1,M
00    120   B1(I1)=B(I1,I)
00          DO 125 I1=IP1,M
00    125   B(I1,I)=0.
00          DO 127 J=1,M
00    127   B2(J)=B(I,J)
00          DO 135 I1=IP1,M
00          DO 135 J=1,M
00    135   B(I1,J)=B(I1,J)-B1(I1)*B2(J)
00    110   CONTINUE
00          B1(1)=1./B(M,M)
00          B(M,M)=1.
00          DO 140 J=1,M
00    140   B(M,J)=B(M,J)*B1(1)
00          DO 150 I=2,M
00          DO 155 I2=1,I
00    155   B1(I2)=B(I2,I)
00          IM1=I-1
00          DO 156 I2=1,IM1
00    156   B(I2,I)=0.
00          DO 157 J=1,M
00    157   B2(J)=B(I,J)
00          IM1=I-1
00          DO 160 I2=1,IM1
00          DO 160 J=1,M
00    160   B(I2,J)=B(I2,J)-B1(I2)*B2(J)
00    150   CONTINUE
00          END
```

VERSION 5

```
02        PROGRAM JAYNOR)INPU1,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
00        DIMENSION AX(7,9),AT)7,9),BB)7,9),CX)7,9),O(7,9),F(7,9),
00       A RTNV)7,7,3),RINV1(7,7,2),DUMO)7,2),DUM1(7),DUM2)7),X)9,11),
00       B H)9,11),ERR(9,11),XX(9,11),IE)3)
00        DATA M,N,NBLK/9,11,3/
00        DATA IE/4,7,10/
00  C  DIETRICH'S MODIFIED MADALA SOLVER, VERSION 5.
00  C  THIS VERSION IS IDENTICAL TO VERSION 4, EXCEPT
00  C  IRREGULAR BOUNDARIES ARE NOT ALLOWED.  IT IS ALSO EQUIVALENT TO
00  C  MADALA'S BASIC SOLVER, ALLOWING NON-SEPARABLE OPERATORS ON A
00  C  NON-UNIFORM MESH WITH REGULAR BOUNDARIES.
00  C  THE EQUATION TO BE SOLVED IS
00  C  AX)I,J) * X(I,J+1) + CX)I,J) * X(I+2,J+1) + BB)I,J) * X)I+1,J+1) +
00  C  AY)I,J) * X(I+1,J) +            X(I+1,J+2) = F(I,J),
00  C                                         ((I=1,M-2),)J=1,N-2))
00  C  WHERE X)1,J), X)M,J), X)I,1), AND X(I,N) ARE LEFT UNCHANGED.
00        M1=M-1
00        N1=N-1
00        M2=M-2
00        N2=N-2
00        NBLK1=NBLK-1
00        DO 100 J=1,N2
00        DO 100 I=1,M2
00        AX(I,J)=1.+.01*)I-4)**2
00        CX(I,J)=1.-.01*)I-4)**2
00        AY)I,J)=1.+.01*)J-5)**2
00        BB)I,J)=(-AX)I,J)-AY)I,J)-CX)I,J)+.01*)J-5)**2-1.1)/
00       A (1.-.01*)J-5)**2)
00    100 CONTINUE
00        DENOM=).25*M*N)**2
00        DO 170 J=1,N
00        DO 170 I=1,M
00        X)I,J)=)I-1.)*)I-M)*(J-1.)*)J-N)/DENOM
00        XX)I,J)=0.
00        IF )I.EQ.1.OR.I.EQ.M.OR.J.EQ.I.OR.J.EQ.N) XX)I,J)=X)I,J)
00    170 CONTINUE
00        FSUM=0.
00        DO 171 J=2,N1
00        L=J-1
00        DO 171 I=2,M1
00        K=I-1
00        F)K,L)=AX(K,L)*X(I,J)+AY)K,L)*X)I,L)+CX)K,L)*X)I+1,J)+
00       A X(I,J+1)+BB(K,L)*X)I,J)
00   171  FSUM=FSUM+ABS(F)K,L))
00        FSUM=M2*N2/FSUM
00        TO=SECOND)TIM)
02        CALL JAY)AX,AT,BB,CX,RINV,RINV1,H,TE,M,N,M2,N2,NBLK,NBLK1)
00        TIMSUM=0.
00        DO 181 J=2,N1
00        DO 181 I=2,M1
00        O)I-1,J-1)=F)I-1,J-1)
00   181  X)T,J)=0.
00        ALINE=0.
00        TO=SECOND)TIM)
02        CALL NOR)AX,AT,BB,CX,RINV,RINV1,DUMO,DUM1,DUM2,O,M,XX,IE,M,N,M2,
00       A N2,NBLK,NBLK1)
00        DO 990 J=2,N1
00        DO 990 I=2,M1
00   990  X)I,J)=X)I,J)+XX)I,J)
00        DO 991 J=1,N
00        DO 991 I=1,M
00   991  XX)I,J)=0.
00        DO 992 J=2,N1
00        L=J-1
00        DO 992 I=2,M1
00        K=I-I
00   992  O)K,L)=F)K,L)-AT)K,L)*X)I,J-1)-AX)K,L)*X(I-1,J)-BB(K,L)*X)I,J)-
00       A CX)K,L)*X)I+1,J)-X)I,J+1)
00   1000 CONTINUE
00   905  FORMAT)1X,1P15E8.1)
00        ERSUM=0.
00        DO 740 I=2,M1
00        K=I-1
00        DO 740 J=2,N1
00        L=J-I
00        ERR)I,J)=F)K,L)-AT)K,L)*X(I,J-1)-AX)K,L)*X)I-1,J)-BB)K,L)*
00       A X)I,J)-CX)K,L)*X)I+1,J)-X)I,J+1)
00        ERSUM=ERSUM+ABS)ERR)I,J))
00   740  ERR)I,J)=ERR)I,J)*FSUM
00        ERSUM=ERSUM*FSUM/(M2*N2)
00        WRITE)6,904)
00   904  FORMAT)/40X,20HNORMALIZED RESIDUALS)
00        DO 742 I=2,M1
00        WRITE)6,905))ERR)I,J),J=2,N1)
00    742 CONTINUE
00        WRITE(6,906) ERSUM
00   906  FORMAT)/37H MEAN ABSOLUTE NORMALIZED RESIDUAL = ,1PE9.2)
00        END
```

```
01              SUBROUTINE JAY(AX,AY,BB,CX,R1NV,R1NV1,H,1E,MD,ND,M2,N2,NBLK,
00             INBLKI)
00              DIMENSION AX(M2,N2),AY(M2,N2),BB(M2,N2),CX(M2,N2),R1NV(M2,M2,NBLK)
00             A,RINV1(M2,M2,NBLKI),H(MD,ND),IE(NBLK)
00              JL=I
00              NB=0
00        1C0   NB=NB+1
00              JH=1E(NB)
00              JHP=JH+1
00              JHM=JH-2
00              JG=JL+I
00              DO 250 NG=1,M2
00              IG=NG+I
00              DO 210 J=JL,JHP
00              DO 210 I=1,MD
00        210   H(1,J)=0.
00              H(IG,JG)=I.
00              IF (NB.EO.I) GO TO 220
00              DO 218 N=1,M2
00        218   H(N+I,JL)=R1NVI(NG,N,NB-1)
GD        220   DO 225 J=JL,JHM
GO              DO 225 I=1,M2
00        225   H(I+I,J+2)=-(AX(I,J)*H(1,J+1)+AY(1,J)*H(1+1,J)+BB(I,J)*H(1+1,J+1)+
00             A CX(I,J)*H(1+2,J+I))
00              J=JH-1
00              DO 230 1=1,M2
00        230   RINV(NG,1,NB)=AX(I,J)*H(I,J+I)+AY(I,J)*H(1+1,J)+BB(1,J)*
0C             A H(I+1,J+1)+CX(I,J)*H(I+2,J+I)
00              IF (NB.EO.NBLK) GO TO 250
00              J=IE(NB)
00              DO 240 N=1,M2
00        240   RINV(NG,N,NBLK)=H(N+I,J)
00        250   H(16,JG)=0.
00              CALL MATINV(RINV(1,I,NB),M2,M2)
00              IF (NB.EO.NBLK) RETURN
00              DO 260 I=1,M2
00              DO 260 J=1,M2
00              RINVI(1,J,NB)=0.
00              DO 260 K=1,M2
00        260   RINVI(1,J,NB)=R1NVI(I,J,NB)-RINV(I,K,NB)*RINV(K,J,NBLK)
00       C  RINV1(I,J,NB) IS THE ''ALMOST HOMOGENEOUS'' SOLUTION AT THE J-TH OPEN
00       C  RESIDUAL POSITION FORCED BY A RESIDUAL VALUE OF 1 AT THE I-TH
00       C  RESIDUAL POSITION.  HOMOGENEOUS B.C.'S ARE ASSUMED EVERYWHERE,
00       C  INCLUDING THE IOP OF THE PRESENT SUBREGION NB.
00              JL=JH
00              GO TO IDD
00              END
```

VERSION 5
(continued)

```
01          SUBROUTINE NOR(AX,AY,BB,CX,RINV,RINV1,DUMO,DUM1,DUM2,F,H,X,IE,
00         1MD,ND,M2,N2,NBLK,NBLK1)
00          DIMENSION AX(M2,N2),AY(M2,N2),BB(M2,N2),CX(M2,N2),RINV(M2,M2,NBLK)
00         A,RINV1(M2,M2,NBLK1),H(MD,ND),IE(NBLK),DUMO(M2,NBLK1),DUM1(M2),
00         B DUM2(M2),F(M2,N2 ),X(MD,ND)
00          JS=1
00          DO 150 NB=1,NBLK
00          JF=IE(NB)-2
00          DO 105 J=JS,JF
00          DO 105 I=1,M2
00      105 X(I+1,J+2)=F(I,J)-AX(I,J)*X(I,J+1)-AY(I,J)*X(I+1,J)-BB(I,J)*
00         A X(I+1,J+1)-CX(I,J)*X(I+2,J+1)
00          IF (NB.EQ.NBLK)GO TO 150
00          J=IE(NB)-1
00          DO 115 I=1,M2
00      115 DUM1(I)=F(I,J)-AX(I,J)*X(I,J+1)-AY(I,J)*X(I+1,J)-BB(I,J)*
00         A X(I+1,J+1)-CX(I,J)*X(I+2,J+1)-X(I+1,J+2)
00          J=IE(NB)
00          DO 120 N=1,M2
00          DUM2(N)=0.
00          DO 118 M=1,M2
00      118 DUM2(N)=DUM2(N)+DUM1(M)*RINV1(M,N,NB)
00          DUMO(N,NB)=X(N+1,J)
00      120 X(N+1,J)=X(N+1,J)-DUM2(N)
00      150 JS=IE(NB)
00          DO 300 NB1=1,NBLK
00          NB=NBLK-NB1+1
00          JS=1
00          IF (NB.NE.1) JS=IE(NB-1)
00          JF=IE(NB)-2
00          IF (NB.EQ.NBLK) GO TO 201
00          J=IE(NB)
00          DO 200 N=1,M2
00      200 X(N+1,J)=DUMO(N,NB)
00      201 N=IE(NB)
00          DO 202 J=JS,N
00          DO 202 I=1,MD
00      202 H(I,J)=0.
00          J=IE(NB)-1
00          DO 210 I=1,M2
00      210 DUM1(I)=F(I,J)-AX(I,J)*X(I,J+1)-AY(I,J)*X(I+1,J)-BB(I,J)*
00         A X(I+1,J+1)-CX(I,J)*X(I+2,J+1)-X(I+1,J+2)
00          DO 220 N=1,M2
00          DUM2(N)=0.
00          DO 218 M=1,M2
00      218 DUM2(N)=DUM2(N)+DUM1(M)*RINV(M,N,NB)
00          H(N+1,JS+1)=DUM2(N)
00      220 X(N+1,JS+1)=X(N+1,JS+1)+DUM2(N)
00          IF (NB.EQ.1) GO TO 250
00          J=IE(NB-1)
00          DO 240 N=1,M2
00          DUM2(N)=0.
00          DO 238 M=1,M2
00      238 DUM2(N)=DUM2(N)+H(M+1,JS+1)*RINV1(M,N,NB-1)
00      240 H(N+1,J)=DUM2(N)
00      250 DO 300 J=JS,JF
00          DO 300 I=1,M2
00          H(I+1,J+2)=-AX(I,J)*H(I,J+1)-AY(I,J)*H(I+1,J)-BB(I,J)*
00         A H(I+1,J+1)-CX(I,J)*H(I+2,J+1)
00      300 X(I+1,J+2)=X(I+1,J+2)+H(I+1,J+2)
00          END
```

VERSION 5
(continued)

```
00          SUBROUTINE MATINV(B,N,M)
00          DIMENSION B(N,1),B1(100),B2(100)
00          M1=M-1
00          DO 110 I=1,M1
00          B1(I)=1./B(I,I)
00          B(I,I)=1.0
00          DO 112 J=1,M
00     112  B(I,J)=B(I,J)*B1(I)
00          IP1=I+1
00          DO 120 II=IP1,M
00     120  B1(II)=B(II,I)
00          DO 125 II=IP1,M
00     125  B(II,I)=0.
00          DO 127 J=1,M
00     127  B2(J)=B(I,J)
00          DO 135 II=IP1,M
00          DO 135 J=1,M
00     135  B(II,J)=B(II,J)-B1(II)*B2(J)
00     110  CONTINUE
00          B1(1)=1./B(M,M)
00          B(M,M)=1.
00          DO 140 J=1,M
00     140  B(M,J)=B(M,J)*B1(1)
00          DO 150 I=2,M
00          DO 155 I2=1,I
00     155  B1(I2)=B(I2,I)
00          IM1=I-1
00          DO 156 I2=1,IM1
00     156  B(I2,I)=0.
00          DO 157 J=1,M
00     157  B2(J)=B(I,J)
00          IM1=I-1
00          DO 160 I2=1,IM1
00          DO 160 J=1,M
00     160  B(I2,J)=B(I2,J)-B1(I2)*B2(J)
00     150  CONTINUE
00          END
```

END

FILMED

2-83

DTIC