MASTER

DEC 7 1966

WANL-TME-1526
OCTOBER 1966

Westinghouse Astronuclear Laboratory

VARI-QUIR III: Inclusion of the R-θ Geometry

UNCLASSIFIED NERVA RESEARCH
AND DEVELOPMENT REPORT.

RELEASED FOR ANNOUNCEMENT
IN NUCLEAR SCIENCE ABSTRACTS

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.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

RELEASED FOR ANNOUNCEMENT
IN NUCLEAR SCIENCE ABSTRACTS

Astronuclear
Laboratory
WANL-TME-1526
OCTOBER 1966

SUBMITTED BY:

Westinghouse Electric Corporation
Astronuclear Laboratory
Pittsburgh, Pennsylvania 15236

CPSM 11-10-66

H.C. # 3.00; 10/11/65

VARI-QUIR III: Inclusion of the R-θ Geometry

PREPARED BY:

APPROVED BY:

Main Text -

George Collier
and
J. W. Riese

D. W. Drawbaugh, Manager
Reactor Physics & Mathematics

Appendix A -

W. H. Gray

UNCLASSIFIED NERVA RESEARCH
AND DEVELOPMENT REPORT.

INFORMATION CATEGORY

unclassified
10/3/66
Authorized Classifier - Date

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

TABLE OF CONTENTS

	<u>Page</u>
Abstract	i
Chapter I - ADVANTAGES OF THE NEW CODE	1
A. The R- θ Equations	1
B. Variable Dimensions	2
C. Integration Scheme for VARI-QUIR	3
D. New Input Instructions	5
Chapter II - DESCRIPTION OF CODE	13
A. Present Limitations	13
B. Typical Problem	14
References	20
Appendix A - POTLUK Program	21
Appendix B - FORTRAN Listing	24

Abstract

This report is an extension of WANL-TNR-189. The new version of VARI-QUIR III permits the calculation of R- θ geometries with periodic boundary conditions. Also, the new code has complete variable dimensions at execution time.

I. ADVANTAGES OF THE NEW CODE

A. The R- θ Equations

In WANL-TNR-189 it was stated that the VARI-QUIR III code can solve steady state reactor problems with X-Y and R-Z geometries. The R- θ equations were offered in the report, but they were not incorporated in the program.

The new code can now make R- θ calculations with periodic boundary conditions. The user may make calculations from 2π radians down to any desired segment of the circle. Periodicity is handled by having a false region, and this false region has the same geometrical dimensions and material specifications as the first region.

e.g.

Sketch 1

In sketch 1 there are four physical regions which add up to α radians. α is the segment of the reactor to be calculated, but region A' will be added to the grid automatically. The user does not have to be concerned with the A' region.

B. Variable Dimensions

In the old version of VARI-QUIR III the user was restricted to making reactor calculations of 18 or less energy groups, 36 or less regions, 10 or less regions in either direction, and 23 or less different material regions. In the present version the user has no restrictions (within reason) on any one of these parameters, but only on the total amount of storage. This offers the user greater exchange of X-region, Y-region, materials, etc. If one should try a problem that would exceed the available storage, the program would print out an error message.

Essentially the code has 19,000 locations for problem-variable storage (fluxes, adjoints, sources, etc.) and the rest of the 32,768* locations are used by the program and the available input-output routines. The program takes each variable and calculates the required number of locations which it subtracts from the 19,000. In this way the 19,000 locations are fully packed and there are not any wasted locations. Thus, the largest problem that VARI-QUIR will handle is one that uses 19,000 locations. The user is limited to the above number of locations, but he is free to increase any one or two variables at the expense of any other parameters.

-

*Storage available on IBM-7094 digital computer.

C. Integration Scheme for VARI-QUIR

Difficulty in obtaining converged results from R- θ VARI-QUIR problems involving angularly narrow regions has been found to be caused by the order of integration. The radial variation of fast group fluxes should resemble a Bessel function when an annular ring of thermal absorber is included in the reactor, but heretofore, these fluxes were not even monotonically decreasing functions of radial position.

The integration scheme formerly used in VARI-QUIR III was derived from the successful schemes of the slab and R-Z versions of this program and integrates first radially and then tangentially. Since the EXTERMINATOR⁽¹⁾ program is known to require integration in the direction of minimum buckling and since the order of integration in EXTERMINATOR is under input control, it was decided to run identical R- θ VARI-QUIR and EXTERMINATOR problem to partial convergence. It was observed that the eigenvalue change was the same for 15°, 30°, 60°, and 360° periodic problems when integrating in the θ direction, but was quite different for the same problems when integrating in the R direction. After 16 iterations all of the θ -direction problems were approaching the converged eigenvalue (0.90133), but in the R-direction problems only the 360° problem was approaching the converged value. The other R-direction problems (15°, 30°, and 60°) were oscillating around 0.8750. Figure 1 below is a graph of eigenvalue vs. iteration.

VARI-QUIR III has now been modified to integrate first tangentially and then radially, and the convergence rate improved by a factor of fifteen.

Possibly because the coupling of tangentially adjacent points is weak for angularly narrow regions, the improvement in VARI-QUIR is not as great as for EXTERMINATOR.

Fig. 1 EXTERMINATOR RUNS

D. New Input Instructions

Card No. 1

This card may contain any alphanumeric information that will adequately describe the problem. It is printed at the beginning of the output, following the problem number.

Card No. 2

This card provides information which defines the problem in terms of overall structure.

Punch in Columns --

1-4	NPROB	Problem number
7-8	NG	Number of groups
11-12	NM	Number of materials
15-16	NX	Number of X, Z, or θ regions
19-20	NY	Number of Y or R regions
23-24	NGEOM	= -2 XY double symmetry -1 XY single symmetry 0 XY no symmetry 1 RZ single symmetry 2 RZ double symmetry 3 R θ calculation with integration in the R direction 4 R θ calculation with integration in the θ direction
27-28	NGES	Initial flux option NGES = -1 fluxes and adjoints are computed from scratch 0 flux guess is read into the program, but the adjoint guess is computed from scratch

- 1 flux and adjoint guesses are both read into the program
- 2 the flux and adjoints from the previous problem are used as guesses

32	INTPX	This is the number of interpolated points given in the printout in addition to the three points per region in the X, Z, or θ direction, e.g. if INTPX is 1, the program will print out five points per region in the X, Z, or θ direction. One interpolated point between each regular point. Number of points per region = $2*INTPX+3$.
36	INTPY	The same as above but in the Y or R direction.
39-40	NPCH	Flux and adjoint punch or write tape option. $NPCH < 0$ The program will not punch, or write on tape, the fluxes and adjoints at the end of the problem. ≥ 0 The program will punch, or write on tape, the fluxes and adjoints at the end of the problem. This option is available so that one may restart a problem that is only partially converged.
42-44	ITOUT	Maximum number of outer iterations on the flux computation.
46-48	ININN	Maximum number of inner iterations on the flux computation. This should be 9 or less in most calculations.
51-52	ITPRNT	Number of outer iterations between flux and adjoint printouts.
56	ISSW	Sense switch options $ISSW = -1$ This option permits the program to run as many outer iterations as are required until the relative eigenvalue difference (from one eigenvalue compared with the previous value) is less than $E1$. $\left \frac{\lambda_1 - \lambda_2}{\lambda_1} \right < E1$

When the above relationship is met, the program will terminate the computation, but in any event it will only do the maximum number of outer iterations (ITOUT).

- = 0 The program will run until the number of outer iterations is equal to ITOUT, the maximum permitted.
- = 1 Sense switch 1 is available and the eigenvalue will be printed out on-line after each outer iteration. Sense Switch 1 may be put down at any time during a run to terminate the problem.

60	IPD	<p>This input variable is used when NGES = 1 (flux and adjoint guesses are both read into the program).*</p> <p>IPD < 0 The input flux and adjoints guesses are read off magnetic tape. This tape is mounted on logical tape 9 before execution. When this option is used the output fluxes and adjoints will be written on logical tape 10 in the same format as the input tape. This permits one to restart a problem and continue.</p> <p>≥ 0 The input flux and adjoints guesses are read off IBM cards.</p>
61-64	NPIE	<p>NPIE is set to the integer that when multiplied by the segment of the circle being calculated will equal 360°. (i.e. if one desires to calculate a 30° slice he would set NPIE = 12)</p> <p style="text-align: center;">$(12)(30^\circ) = 360^\circ$</p> <p>This is only used for R-θ calculation and it may be left blank for other geometries.</p>
67-68	NSC	<p>The value of NSC determines whether the program will scale the fluxes and adjoints after each outer iteration. This is done to accelerate convergence.</p> <p>NSC < 0 Scaling is <u>not</u> done.</p> <p>≥ 0 Scaling is done.</p>

*See Appendix A for tape or card preparation.

- 71-72 NORM The value of NORM determines whether the program will normalize the fluxes and adjoints after every outer iteration. In most cases normalization should not be done after every outer iteration.
 $NORM < 0$ Normalization is not done.
 ≥ 0 Normalization is done.
- 75-76 NEXIT Since VARI-QUIR is an overlay program and there are three major links the author thought it might be desirable, at times, to terminate execution in the computational link rather than going back to the input link and attempting to read in the next case. This permits one to have a running dump of the computational link.
 $NEXIT < 0$ Execution will be terminated in the computational link. One should not stack problems using this option.
 ≥ 0 At the end of the problem the program will go on to the next case. This option should be used for the majority of problems.
- 80 NST NST is an input variable that permits one to use the nuclear constant and the flux from the previous problem rather than supplying cards for every stacked problem. The chart below can be used to determine what value should be used for NST.

NST	Nuclear Constants (X-sections)	Flux	NGES
0	new x-section	new flux	< 2
1	new x-section	old flux	$= 2$
2	old x-section	new flux	< 2
3	old x-section	old flux	$= 2$

Card No. 3

Punch in Columns --

1-10	RELAX	is the successive overrelaxation constant. This is the value that is multiplied by the flux change after each iteration. When RELAX is greater than 1.0 the flux change will be increased. If RELAX is set to 1.0, the method of solution reduces to successive iteration method or Seidel's method. Relax should always be equal to or greater than 1.0.
11-20	DRELAX	is the value that will be added to RELAX to permit RELAX to vary as convergence comes about. (i.e., if we make RELAX 1.2 and DRELAX .5 the actual value the program will use for RELAX will be 1.7 on the first iteration. RELAX will vary from 1.7 down to 1.2. The sum of RELAX and DRELAX should never be equal to or greater than 2.0. In all cases though when the relative eigenvalue difference gets smaller than 10^{-5} the program makes the effective RELAX 1.0.)
21-30	E1	is used only when ISSW = -1. This criterion is the input value described on the second input card (ISSW).
31-40	TIM	is the maximum amount of time, this problem can run, in minutes. Execution will be terminated after TIM minutes whether the total number of outer iterations are used up or whether the E1 criterion is met.
41-50	SIZE	is the value the fluxes and adjoints will be normalized to during execution. If this field is left blank, the program will automatically normalize the fluxes and adjoints to 10^6 .
51-60	E3	is a criterion that is used in the <u>inner</u> iteration calculation. The maximum number of inner iterations is input to the code, but if the total flux change for the i th iteration is less than E3 times the total flux change for the first iteration, the program will not do the rest of the iterations up to the maximum.

(e.g., if ITINN = 9 and the

$$\frac{\left(\sum_{j=1}^J |\Delta \phi_g| \right)_4}{\left(\sum_{j=1}^J |\Delta \phi_g| \right)_1} < E3$$

the fifth to the ninth iteration will not be done.)

The program keeps track of the number of inner iterations for each outer iteration for each energy group. This count is called ITCOUNT and it is printed out after every outer iteration along with the eigenvalue. The value of the first digit represents the number of inner iteration required for the first group, and the second digit represent the number of inner iterations for the second group, etc.

Card Group No. 4

The X, Z, or θ values of each region using successive 10-column fields. Starting from the left (where X = 0). The units of X and Z are centimeters and unit of θ is degress.

Card Group No. 5

The Y or R length of each region using successive 10-column fields. Starting from the bottom (where Y = 0). The units here are centimeters, too.

Card Group No. 6

These cards list the material number for each region, using one Y-row per card and as many cards as there are Y regions. On each card, the material numbers for the X regions, starting from the left, are listed in successive 4-column field.

Sample Problem

X regions 3 NX = 3

Y regions 2 NY = 2 The two Y regions require 2 cards.

Y ↑	Material 1	Material 2	Material 4
	Material 2	Material 3	Material 1
X →			

Card Input for Sample Problem

Card 6.1 M(1,1) = 2
 M(2,1) = 3
 M(3,1) = 1

Column	1	4 5	8 9	12
		2	3	1

Card 6.2 M(1,2) = 1
 M(2,2) = 2
 M(3,2) = 4

Column	1	4 5	8 9	12
		1	2	4

Card Group No. 7

These cards contain the nuclear constants for each material.

SABS1(IG) is $\sum_a$ for group IG of material IM.

SFIL(IG,IM) is $\sum_f$ for group IG of material IM.

XNUL(IG) is ν for group IG of material IM.

CHIL(IG,IM) is χ for group IG of material IM.

STR1(IG,IGP) is $\sum^{TR}$ into group IG from group IGP of material IM.

These nuclear constants may be generated with a computer code, called GAM-TNS⁽²⁾, which punches them on cards with the proper format for VARI-QUIR. If one chooses to have these nuclear constant punched by hand, they should see cards 74 to 94, of subroutine PART 1, of the FORTRAN listing (Appendix B).

Card Group No. 8 (omitted if NGES = -1, 2, or if NGES = 1 and IPD = -1)

The type and the amount of cards used here depends upon the value of NGES.

NGES = -1 omit this card group.

= 0 These cards should contain the flux guess that is to be read into the program, but not the adjoints.

= 1 and IPD = -1, omit this card group, because the flux and adjoint guess will be read off magnetic tape.

= 1 and IPD 0, these cards should contain the flux and adjoints guesses. The flux cards first and the adjoint cards second. In all of the cards that are read in here (NGES = 0 or 1) the FORMAT is (5E15.8). Every flux group and every adjoint group will start a new card.

= 2 omit this card group.

II. DESCRIPTION OF CODE

A. Present Limitations

Geometry - XY, RZ, or R θ

Boundary Conditions - $\phi = 0$, $\nabla_n \phi = 0$, or $\phi(r, \theta) = \phi(r, \theta + n\alpha)$

where $n = 0, 1, 2, \dots \left[\frac{360^\circ}{\alpha} - 1 \right]$

and α is the segment of the circle being calculated.

Energy groups, total regions, region in either direct, and different material regions	}	These are free to vary as long as the total number of locations does not exceed 19,000.
---	---	---

Starting options -

- (1) Fluxes and adjoints computed from scratch.
- (2) Flux guess is read into the program, but the adjoint guess is computed from scratch.
- (3)*Flux and adjoint guesses are both read into the program.
- (4) The fluxes and adjoints from the previous problem are used as guesses.

Successive problems - any number may be stacked. If successive problems are run NEXIT ≥ 0 .

Relative Region Size - VARI-QUIR has difficulty in convergence when regions next to each other are quite different in size. It is recommended that no region be more than twice its neighbor. This is especially true for R- θ problems near the center of the reactor (i.e. the center regions, increasing in R, should be about the same size. The worst thing one can do, using VARI-QUIR, is to make the first radial region very small and then make the second radial relatively larger. This procedure will reduce the convergence rate considerably.)

*When using this option one may use card or tape input. This is the option one would use to continue a problem, with successive runs, until the degree of convergence is desired.

B. Typical Problem

The problem chosen was an R- θ geometry, four energy groups, with a core, a reflector and an absorber to demonstrate θ dependence. Sketch 2 below describes the problem.

Sketch 2

This problem was run using VARI-QUIR and EXTERMINATOR⁽¹⁾ with the same grid⁺ size in both codes. This rough grid size was probably an unfair use of EXTERMINATOR since it uses a linear-type approximation, but the author wanted to demonstrate the value of VARI-QUIR's quadratic-in-region approximation.

The results of the two runs are shown in Figures 2-5. The ray* that is plotted goes right through two absorber regions. As the graph shows, the agreement is quite good in the first three groups, but not real good in the fourth group. Figure 6 shows that from 50 to 55 cms. (absorber region) the ⁺7 x 17 grid points per group.

*This is the dashed line in sketch 2.

VARI-QUIR dips lower than the point near the dip would permit. This is legitimate because, as stated in VARI-QUIR III⁽³⁾, the three points in a region only give definition to the quadratic equation in the region (i.e. the flux has definition between the given points). The dip in the curve, from 50 to 55 cms., was calculated from the quadratic in that region. For plotting purposes VARI-QUIR can be made to print out as many in between points as desired.

Fig. 2 GROUP 1 FLUX

Fig. 3 GROUP 2 FLUX

Fig. 4 GROUP 3 FLUX

Fig. 5 GROUP 4 FLUX

REFERENCES

- (1) Fowler, T. B., Tobias, M. L., and Vondy, D. R., "EXTERMINATOR - A Multigroup Code for Solving Neutron Diffusion Equations in One and Two Dimensions", ORNL-TM-842, (1965).
- (2) Gibson, G., Melnick, M. M., and Gray, W. H., "Changes in the GAM-TNS Program and Cross Section Library, WANL-TME-1228, (August 1965).
- (3) Collier, G. and Riese, J. W., "VARI-QUIR III: A Two-Dimensional Steady State Multi-Group Diffusion Code", WANL-TNR-189, (December 1964).
- (4) Killinger, A. H. and Killinger, G. B., "Double Interpolation Subroutine", Westinghouse Analytical Department Report 62-1002, (January 1962).

Appendix A - POTLUK: A FORTRAN IV PROGRAM TO GENERATE A VARI-QUIR FLUX AND ADJOINT GUESS

This code will generate a flux and adjoint guess for a VARI-QUIR problem whose internal region boundaries are different from those of a previous problem using the flux and adjoint from that problem. This enables users to vary the internal boundaries at will and still obtain a flux and adjoint guess that will significantly decrease VARI-QUIR running time (see Table I).

TABLE I

Comparison of Two VARI-QUIR Problems; the original problem uses a "bare reactor" flux guess; the derived problem used the converged flux of the original, with twice as fine a grid.

	Mesh Size	Running Time to the Same Order of Convergence
Original VARI-QUIR Problem	11 x 4	120.12 min.
Derived Problem Using POTLUK Flux and Adjoint Guess	20 x 8	10.56 min.

The method used is double parabolic interpolation using the DI⁽⁴⁾ function subprogram.

The (card or tape) VARI-QUIR fluxes and adjoints may be entered directly into the program, similarly, the program output may be entered directly into VARI-QUIR. The input output options within any one problem (IGO = 1 or 0, see input instructions) are as follows:

<u>INPUT</u>	/	<u>OUTPUT</u>
Card		Tape
Card		Card
Tape		Card
Tape		Tape

Tape input (only tapes generated by VARI-QUIR should be used) should be hung on A-5; tape output is generated on A-6.

Input Instructions

<u>Card</u>	<u>Format</u>	<u>Card Columns</u>	<u>Variable Name</u>	<u>Explanation/Remarks</u>
1	A	1-80	T	80 columns of arbitrary title information.
2	18I4	1-4, 5-8, etc., R.A.*	NOX	Number of old X-regions (horizontal axis).
			NOY	Number of old Y-regions (vertical axis).
			NNX	Number of new X-regions.
			NNY	Number of new Y-regions.
3	8E10.4	1-10, 11-20, etc. ...	DOX	Incremental size (cm.) of each old X-region; there must be NOX entries, 8 per card.
4	8E10.4	1-10, 11-20, etc. ...	DOY	Incremental size (cm.) of each old Y-region; there must be NOY entries, 8 per card.
5	8E10.4	1-10, 11-20, etc. ...	DXN	Incremental size (cm.) of each new X-region; there must be NNX entries, 8 per card.
6	8E10.4	1-10, 11-20, etc. ...	DYN	Incremental size (cm.) of each new Y-region; there must be NNY entries, 8 per card.
7	3A6, I2	1-18, 19-20,* 21-22,* 23-24.*	LABEL	18 columns; arbitrary description of the set of fluxes to be read in.
			ITAPE	= 1, read fluxes (adjoints) from tape (A5), = 0, read fluxes (adjoints) from cards.
			IOUT	= 1, write fluxes (adjoints) on to tape (A6), = 0, punch fluxes (adjoints) on to cards.

* Right Ajusted.

<u>Card</u>	<u>Format</u>	<u>Card Columns</u>	<u>Variable Name</u>	<u>Explanation/Remarks</u>
7		cont'd.	ICH	The number of specially entered fluxes, i.e. fluxes in the new array which the user wishes to input. If $ICH = 0$, skip to card 10. $ICH \leq 99$.
8	18I4	1-4, 5-6, etc...	ICHDX	The position in the flux or adjoint array of each of the specially entered fluxes or adjoints. There must be ICH entries, 18 to a card. (N.B. the size of the flux or adjoint array is $(2*NNX+1) * (2*NNY+1)$. Enter only if $ICH \geq 1$.
9	5E15.8	1-15, 16-30, etc...	ENTR	The fluxes or adjoints to be specially entered; they must be in the same order as ICHDX and there must be ICH entries, 5 to a card. Enter only if $ICH \geq 1$.
10	5E15.5	1-15, 16-30, etc...	PHIO	Old flux or adjoint array, may be VARI-QUIR flux output entered directly. There must be $(2NOX+1)(2NOY+1) = NP$ entries, 5 per card. Enter only if $ITAPE = 0$, otherwise omit.
11	I4	1-4,R.A.	IGO	Next case key: = 0, this is the last problem, get off the machine, = 1, the next case will consist of only new fluxes or adjoints, enter cards 7-11 for the next case, = 2, the next case will be an entirely new problem, enter cards 1-11.

RESTRICTIONS

- (1) If entirely different problems (cards 1-11) are stacked, and tapes are used ($ITAPE$ and/or $INOUT = 1$), new tapes are required for each problem.
- (2) $NOX \leq 800$
 $NOY \leq 800$
 $NP \leq 1600$
 $(2NNX+1)(2NNY+1) \leq 4800$
 $NNX \leq 2500$
 $NNY \leq 2500$
 $ICHDX \leq 99$

Appendix B

FORTRAN LISTING

```

$IBSYS
$EXECUTE      IBJOB
$IBJOB        GO,MAP,SOURCE,ALTIO
$IBFTC MAIN   LIST,DECK,M94/2,SDD,XR7
C             GENERAL VARI-QUIR  STEADY STATE  ( G. COLLIER )    X559
C
C             THREE GEOMETRIES   ( X-Y , R-Z AND R-THETA )
C
C             MAIN PROGRAM
C
C             COMMON /COEF/ZZ,RR,TH,P(3,3),Q(3,3),PR(3,3),QR(3,3),AA(3,3)
1             1,BB(3,3),CC(3,3),DD(3,3),EE(3,3),FF(3,3),GG(3,3),HH(3,3),XX,YY
2             2,TEMP(14)
C
C             COMMON NG,NM,NX,NY,NGEOM,NGES,NPCH,ITOUT,ITINN,ITPRNT,ISSW,RELAX
1             1,DRELAX,E1,TIM,DTIM,S,INTPX,INTPY,NIM1,NIM2,NI,NJ,AMB,SIZE,SIZED
2             2,IPD,IGEOM,KX,JX,NPIE,NSC,NORM,NEXIT,NST,E3
C
C             COMMON CELLS(19000)
C
C             EQUIVALENCE (CELLS,ICELLS)
C
C             DIMENSION ICELLS (19000)
C
10            TIME = CLOCK(XMON,DAY,YEAR)
            DTIM = 0.001*TIME
            MON = XMON
            NDAY = DAY
            NYEAR = YEAR
            READ (5,10001) TEMP
10001         FORMAT(13A6,A2)
            READ (5,10002) NPROB,NG,NM,NX,NY,NGEOM,NGES,INTPX,INTPY,NPCH,ITOUT
1             1,ITINN,ITPRNT,ISSW,IPD,NPIE,NSC,NORM,NEXIT,NST
10002         FORMAT(20I4)
            NI = 2*NX +1
            NJ = 2*NY +1
            LG = NG
            LY = NY
            LJ = NJ
            LX = NX
            LI = NI
            LM = NM
            WRITE (6,10003) NPROB,TIME,MON,NDAY,NYEAR
10003         FORMAT(12H1PROBLEM NO.14,45X16HCOMPUTER TIME = -3PF6.3,5X12,1H/12,
1             11H/12 )
            IF(NGEOM-3)20,30,30
20            NADD = 0

```


GO TO 40	44
30 NADD = 1	45
40 ISAVE = 9*(NX+NADD)	46
IQRX = ISAVE + 1	47
IPRR = ISAVE + IQRX	48
ISAVE = 9*NY	49
IQRR = ISAVE + IPRR	50
IPRT = ISAVE + IQRR	51
IMAT = ISAVE + IPRT	52
IDX = (NX+NADD)*NY + IMAT	53
IDY = NX+NADD + IDX	54
IA1 = NY + IDY	55
IB = NG*NG*NM + IA1	56
ISAVE = NG*NM	57
ICHI1 = ISAVE + IB	58
ID1 = ISAVE + ICHI1	59
ISABS1 = ISAVE + ID1	60
IXNU1 = NG + ISABS1	61
ISFI1 = NG + IXNU1	62
ISTR1 = NG*NM + ISFI1	63
IIY = NG*NG + ISTR1	64
ISAVE = NG*(2*(NX+NADD)+1)*(2*NY+1)	65
IIYD = ISAVE + IIY	66
IIYS = ISAVE + IIYD	67
IIYDS = ISAVE + IIYS	68
IIDEN = ISAVE + IIYDS	69
IRAV = ISAVE + IIDEN	70
IF(NADD)90,60,50	71
50 IDENC = NY + IRAV	72
IYCS = NG + IDENC	73
IYDCS = NG + IYCS	74
ITOTAL = NG + IYDCS	75
LX = NX + 1	76
LI = NI + 2	77
GO TO 70	78
60 ITOTAL = NY + IRAV	79
IDENC = IRAV	80
IYCS = IRAV	81
IYDCS = IRAV	82
70 IF(ITOTAL-19000)80,80,90	83
80 CALL CONT(CELLS,CELLS(IQRX),CELLS(IPRR),CELLS(IQRR),CELLS(IPRT),	84
1ICELLS(IMAT),CELLS(IDX),CELLS(IDY),CELLS(IA1),CELLS(IB),	85
2CELLS(ICHI1),CELLS(ID1),CELLS(ISABS1),CELLS(IXNU1),CELLS(ISFI1),	86
3CELLS(ISTR1),CELLS(IIY),CELLS(IIYD),CELLS(IIYS),CELLS(IIYDS),	87
4CELLS(IIDEN),CELLS(IRAV),CELLS(IDENC),CELLS(IYCS),CELLS(IYDCS),	88
5LX,LY,LG,LI,LJ,LM)	89
GO TO 10	90

90	WRITE (6,10004)									91
10004	FORMAT(40H0ERROR	CHECK	NADD, OR	ITOTAL.	)					92
	STOP									93
	END									94
\$IBFTC	CON	LIST,DECK,M94/2,SDD,XR7								1
		SUBROUTINE CONT(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,SABS1,								2
		1XNU1,SFI1,STR1,Y,YD,YS,YDS,DEN,RAV,DENC,YCS,YDCS,LX,LY,LG,LI,LJ,								3
		2LM)								4
C										5
C		CONTROL PROGRAM								6
C										7
		COMMON /COEF/ZZ,RR,TH,P(3,3),Q(3,3),PR(3,3),QR(3,3),AA(3,3)								8
		1,BB(3,3),CC(3,3),DD(3,3),EE(3,3),FF(3,3),GG(3,3),HH(3,3),XX,YY								9
		2,TEMP(14)								10
C										11
		COMMON NG,NM,NX,NY,NGEOM,NGES,NPCH,ITOUT,ITINN,ITPRNT,ISSW,RELAX								12
		1,DRELAX,E1,TIM,DTIM,S,INTPX,INTPY,NIM1,NIM2,NI,NJ,AMB,SIZE,SIZED								13
		2,IPD,IGEOM,KX,JX,NPIE,NSC,NORM,NEXIT,NST,E3								14
C										15
		DIMENSION QRX(3,3,1),PRR(3,3,1),QRR(3,3,1),PRT(3,3,1)								16
		1,M(LX,1),DX(1),DY(1),A1(LG,LG,1),B(LG,1),CHI1(LG,1)								17
		2,D1(LG,1),SABS1(1),XNU1(1),SFI1(LG,1),STR1(LG,1)								18
		3,Y(LG,LI,1),YD(LG,LI,1),YS(LG,LI,1),YDS(LG,LI,1),DEN(LG,LI,1)								19
		4,DENC(1),YCS(1),YDCS(1),RAV(1),PRX(3,3,1)								20
		CALL PART1(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,SABS1,XNU1								21
		1,SFI1,STR1,Y,YD,DEN,DENC,RAV,LX,LG,LI)								22
		CALL PART2(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,Y,YD,YS,YDS								23
		1,DEN,DENC,YCS,YDCS,LX,LG,LI)								24
		LG1 = 2*INTPX*NX + NI								25
		LGJ = 2*INTPY*NY + NJ								26
		CALL PART3(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,SABS1,XNU1								27
		1,SFI1,STR1,Y,YD,YDS,DEN,DENC,RAV,LX,LG,LI,LG1,LGJ,LM)								28
		RETURN								29
		END								-0
\$IBFTC	BLOK	LIST,DECK,M94/2,SDD,XR7								1
		BLOCK DATA								2
C										3
		COMMON /COEF/ZZ,RR,TH,P(3,3),Q(3,3),PR(3,3),QR(3,3),AA(3,3)								4
		1,BB(3,3),CC(3,3),DD(3,3),EE(3,3),FF(3,3),GG(3,3),HH(3,3),XX,YY								5
		2,TEMP(14)								6
C										7
		DATA XX,YY,ZZ,RR,TH/1HX,1HY,1HZ,1HR,5HTHETA/,((P(I,J),Q(I,J),PR(I,J)								8
		1),QR(I,J),AA(I,J),BB(I,J),CC(I,J),DD(I,J),EE(I,J),FF(I,J),GG(I,J),								9
		2HH(I,J),I=1,3),J=1,3)/4.0,70.0,-1.5,-20.0,2*1.0,2.0,1.0,4*0.0,								10
		32.0,-80.0,-1.0,20.0,-2.0,-1.0,2*2.0,4*0.0,-1.0,10.0,2*0.0,1.0,0.0								11
		4,-1.0,5*0.0,2.0,-80.0,-1.0,20.0,-2.0,-1.0,2*2.0,4*0.0,16.0,160.0								12
		5,2*0.0,4.0,0.0,-10.0,0.0,3.0,8.0,128.0,1.0,2.0,-80.0,1.0,-20.0,-2								

1 10X10HMATERIALS=12 //)	42
IF(NGES)60,70,80	43
60 WRITE (6,10003)	44
10003 FORMAT(6X48HFLUXES AND ADJOINTS ARE COMPUTED FROM SCRATCH. //)	45
GO TO 110	46
70 WRITE (6,10004)	47
10004 FORMAT(6X50HINPUT FLUX GUESS, ADJOINTS COMPUTED FROM SCRATCH. //)	48
GO TO 110	49
80 IF(NGES-1)90,90,100	50
90 WRITE (6,10005)	51
10005 FORMAT(6X 66HFLUX AND ADJOINT GUESSES ARE READ INTO PROGRAM IN THE	52
1S PROBLEM. //)	53
GO TO 110	54
100 WRITE (6,10006)	55
10006 FORMAT(6X50HPRECEDING PROBLEM SUPPLIES FLUX AND ADJOINT GUESS. //)	56
110 READ (5,10007) (DX(IX),IX=1,NX)	57
10007 FORMAT(8E10.4)	58
WRITE (6,10008) (TEMP(1),IX,DX(IX) , IX = 1,NX)	59
10008 FORMAT(6X7HDELTA A5, 5H NO.13, 2H =1PE14.7)	60
READ (5,10007) (DY(JY),JY=1,NY)	61
S = DY(1)**2	62
WRITE (6,10009)	63
10009 FORMAT(1H)	64
WRITE (6,10008) (TEMP(2),JY,DY(JY) , JY = 1,NY)	65
WRITE (6,10009)	66
DO 120 JY = 1,NY	67
READ (5,10010) (M(IX,JY),IX=1,NX)	68
10010 FORMAT(20I4)	69
DO 120 IX = 1,NX	70
120 WRITE (6,10011) IX,JY,M(IX,JY)	71
10011 FORMAT(6X15HMATERIAL NO. M(I2,1H,I2,3H) =I3)	72
IF(NST-1)130,130,180	73
130 DO 170 IM = 1,NM	74
WRITE (6,10012) IM	75
10012 FORMAT(16H0 MATERIAL NO. I2 //)	76
DO 140 IG = 1,NG	77
140 READ (5,10013) SABS1(IG),SF11(IG,IM),XNU1(IG),CHI1(IG,IM),D1(IG,IM	78
1), (STR1(IG,IGP),IGP= 1,NG)	79
10013 FORMAT(5E15.8)	80
DO 160 IG = 1,NG	81
DO 150 IGP = 1,NG	82
150 A1(IG,IGP,IM) = -STR1(IG,IGP)	83
A1(IG,IG, IM) = A1(IG,IG,IM) + SABS1(IG)	84
B(IG,IM) = XNU1(IG)*SF11(IG,IM)	85
WRITE (6,10014) IG,SABS1(IG),SF11(IG,IM),XNU1(IG),CHI1(IG,IM),D1(I	86
1G,IM),IG,NG,(STR1(IG,IGP), IGP = 1,NG)	87
10014 FORMAT(16H0 GROUP NO. I2, // 9X 5HSIGA= 1PE14.7 ,	88

1	3X 5HSIGF= 1PE14.7 , 3X3HNU= 1PE14.7 , 3X4HCHI= 1PE14.7 ,	89
2	3X 2HD= 1PE14.7 , // 14X15HTRANSFERS INTO I2,	90
3	2X17HFROM 1, 2, ... , I2, 4H -- //(21X1PE14.7,1P3E20.7))	91
	DO 160 IGP = 1,NG	92
160	A1(IG,IG,IM) = A1(IG,IG,IM) + STR1(IGP,IG)	93
170	CONTINUE	94
	GO TO 190	95
180	READ (12) (((A1(IG,IGP,IM),IG=1,NG),IGP=1,NG),IM=1,NM),((B(IG,IM)	96
	1,CHI1(IG,IM),DI(IG,IM),SFI1(IG,IM),IG=1,NG),IM=1,NM)	97
	WRITE (6,10015)	98
10015	FORMAT(74H0 THE X-SECTIONS FOR THIS PROBLEM ARE SUPPLIED FROM THE	99
	1 PREVIOUS PROBLEM.)	100
C		101
C	AREA WHERE FLUX AND ADJOINT GUESSES ARE READ INTO PROGRAM.	102
C		103
190	IF(NGES)310,270,200	104
200	IF(NGES-1)270,210,320	105
210	IF(IPD)220,270,270	106
220	READ (9) LFU	107
	IF(LFU-JFU)260,230,260	108
230	DO 240 IG=1,NG	109
240	READ (9) ((Y(IG,I,J),I=1,NI),J=1,NJ)	110
	DO 250 IG=1,NG	111
250	READ (9) ((YD(IG,I,J),I=1,NI),J=1,NJ)	112
	REWIND 9	113
	GO TO 320	114
260	REWIND 9	115
	WRITE (6,10016)	116
	PRINT 10016	117
10016	FORMAT(46H0THIS IS THE WRONG FLUX TAPE. CHECK TAPE NO.)	118
	STOP	119
270	DO 280 IG = 1,NG	120
280	READ (5,10013)((Y(IG,I ,J),I =1,NI),J = 1,NJ)	121
	IF(NGES)310,310,290	122
290	DO 300 IG = 1,NG	123
300	READ (5,10013)((YD(IG,I ,J),I =1,NI),J=1,NJ)	124
	GO TO 320	125
C		126
C	AREA WHERE FLUX AND ADJOINT GUESSES ARE SET TO THEIR STARTING VALU	127
C	WHEN THEY ARE COMPUTED FROM SCRATCH.	128
C		129
310	CALL GUESS(Y,YD,DX,DY,SFI1,M,NGES,NGEOM,NX,NY,NG,NI,NJ,LG,LI,LX)	130
C		131
320	IF(NGES-1)380,380,330	132
330	IF(NST-1)340,340,350	133
340	READ (12)	134
350	DO 360 IG=1,NG	135

```

560 READ (12) ((Y(IG,I,J),I=1,NI),J=1,NJ)
      DO 370 IG=1,NG
370 READ (12) ((YD(IG,I,J),I=1,NI),J=1,NJ)
380 REWIND 12
      IF(NGEOM-2)390,390,500
390 DO 410 IG=1,NG
      DO 400 I=1,NI
        Y(IG,I,1) = 0.0
        YD(IG,I,1) = 0.0
        Y(IG,I,NJ) = 0.0
400 YD(IG,I,NJ) = 0.0
        DO 410 J=1,NJ
          Y(IG,I,J) = 0.0
          YD(IG,I,J) = 0.0
          Y(IG,NI,J) = 0.0
410 YD(IG,NI,J) = 0.0
        IF(NGEOM-1)420,500,500
C
C      SETUP OF THE X-Y COEFFICIENTS
C
420 DO 440 I1=1,3
      DO 440 I2=1,3
        DO 430 IX=1,NX
          PRX(I1,I2,IX) = P(I1,I2)*DX(IX)
430 QRX(I1,I2,IX) = Q(I1,I2)/DX(IX)
          DO 440 JY=1,NY
            PRR(I1,I2,JY) = P(I1,I2)*DY(JY)
            QRR(I1,I2,JY) = Q(I1,I2)/DY(JY)
440 PRT(I1,I2,JY) = PRR(I1,I2,JY)
          IF(NGEOM+1)450,470,750
450 DO 460 IG = 1,NG
          DO 460 J = 1,NJ
            Y(IG,1,J) = (4.0*Y(IG,2,J) - Y(IG,3,J))/3.0
460 YD(IG,1,J) = (4.0*YD(IG,2,J) - YD(IG,3,J))/3.0
470 CALL SYMM(PRR,QRR)
          DO 480 I1 = 1,3
            DO 480 I2 = 1,3
              PRT(I1,I2,1) = PRR(I1,I2,1)
              DO 490 IG = 1,NG
                DO 490 I = 1,NI
                  Y(IG,I,1) = (4.0*Y(IG,I,2) - Y(IG,I,3))/3.0
490 YD(IG,I,1) = (4.0*YD(IG,I,2) - YD(IG,I,3))/3.0
                GO TO 750
C
C      SETUP OF THE RAV COEFFICIENTS
C
500 RA = DY(1)*0.5

```

```

136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182

```

	RAV(1) = RA	183
	DO 510 JY=2,NY	184
	RA = RA + 0.5*(DY(JY) + DY(JY-1))	185
510	RAV(JY) = RA	186
	IF(NGEOM-3)520,600,600	187
C		188
C	SETUP OF THE R-Z COEFFICIENTS	189
C		190
520	DO 540 I1=1,3	191
	DO 540 I2=1,3	192
	DO 530 IX=1,NX	193
	PRX(I1,I2,IX) = P(I1,I2)*DX(IX)	194
530	QRX(I1,I2,IX) = Q(I1,I2)/DX(IX)	195
	DO 540 JY=1,NY	196
	PRR(I1,I2,JY) = DY(JY)*(RAV(JY)*P(I1,I2) + DY(JY)*PR(I1,I2))	197
	QRR(I1,I2,JY) = RAV(JY)*Q(I1,I2)/DY(JY) + QR(I1,I2)	198
540	PRT(I1,I2,JY) = PRR(I1,I2,JY)	199
	IF(NGEOM-1)570,570,550	200
550	DO 560 IG = 1,NG	201
	DO 560 J = 1,NJ	202
	Y (IG,1,J) = (4.0*Y (IG,2,J) -Y (IG,3,J))/3.0	203
560	YD(IG,1,J) = (4.0*YD(IG,2,J) -YD(IG,3,J))/3.0	204
570	CALL SYMM(PRR,QRR)	205
	DO 580 I1 = 1,3	206
	DO 580 I2 = 1,3	207
580	PRT(I1,I2,1) = PRR(I1,I2,1)	208
	DO 590 IG = 1,NG	209
	DO 590 I = 1,NI	210
	Y (IG,I,1) = (4.0*Y (IG,I,2) -Y (IG,I,3))/3.0	211
590	YD(IG,I,1) = (4.0*YD(IG,I,2) -YD(IG,I,3))/3.0	212
	GO TO 750	213
C		214
C	SETUP OF THE R-THETA COEFFICIENTS	215
C		216
600	NX = NX + 1	217
	NI = NI + 2	218
	NIM1 = NI-1	219
	NIM2 = NI-2	220
	DX(NX) = DX(1)	221
	DO 610 IX = 1,NX	222
610	DX(IX) = 1.7453292519943D-2*DX(IX)	223
	DO 630 I1=1,3	224
	DO 630 I2=1,3	225
	DO 620 IX=1,NX	226
	PRX(I1,I2,IX) = P(I1,I2)*DX(IX)	227
620	QRX(I1,I2,IX) = Q(I1,I2)/DX(IX)	228
	PRT(I1,I2,1) = PRR(I1,I2,1)	229

```

      DO 630 JY=1,NY
      PRR(I1,I2,JY) = DY(JY)*(RAV(JY)*P(I1,I2) + DY(JY)*PR(I1,I2))
630   QRR(I1,I2,JY) = RAV(JY)*Q(I1,I2)/DY(JY) + QR(I1,I2)
      DO 690 JY=2,NY
      ALPHA = RAV(JY)/DY(JY)
      IF(ALPHA-2.0)660,640,640
640   S1 = 0.0
      JEND = 1.0 -0.5*DLOG(1.25E-19*ALPHA**(-4))/DLOG(2.0*ALPHA)
      DO 650 J=2,JEND
      I = JEND -J +2
      XI = I
650   S1 = S1 +1./((2.0*XI+1.0)*2.0**(2*I)*ALPHA**(2*I+1))
660   DO 690 I1=1,3
      DO 690 I2=1,3
      IF(ALPHA-2.0)680,670,670
670   PRT(I1,I2,JY) =(S1*(12.0*AA(I1,I2)*ALPHA**4 +12.0*BB(I1,I2)*ALPHA
      1**3 +(2.0*CC(I1,I2)-AA(I1,I2))*ALPHA**2 +(DD(I1,I2)-BB(I1,I2))*
      2ALPHA + EE(I1,I2)) + (12.0*EE(I1,I2)+2.0*CC(I1,I2) -AA(I1,I2)) /
      3 (12.0*ALPHA) +(DD(I1,I2)-BB(I1,I2))/(12.0*ALPHA**2) +EE(I1,I2)/
      4(12.0*ALPHA**3))*10.0
      GO TO 690
680   PRT(I1,I2,JY) = 10.0*( 12.0*AA(I1,I2)*ALPHA**3 +12.0*BB(I1,I2)*
      1ALPHA**2 +2.0*CC(I1,I2)*ALPHA +DD(I1,I2) )*( -1.0 +ALPHA*DLOG(
      2(2.0*ALPHA+1.0)/(2.0*ALPHA-1.0)) ) +10.0*(EE(I1,I2) -BB(I1,I2)*
      3ALPHA -AA(I1,I2)*ALPHA**2 )*DLOG((2.0*ALPHA+1.0)/(2.0*ALPHA-1.0))
690   CONTINUE
710   DO 720 JY = 1,NY
720   M(NX,JY) = M(1,JY)
      DO 740 IG =1,NG
      DO 730 J = 1,NJ
      Y (IG,NIM1,J) = Y (IG, 2,J)
      YD(IG,NIM1,J) = YD(IG, 2,J)
      Y (IG, NI,J) = Y (IG, 3,J)
      YD(IG, NI,J) = YD(IG, 3,J)
      Y (IG, 1,J) = Y (IG,NIM2,J)
730   YD(IG, 1,J) = YD(IG,NIM2,J)
      DO 740 I=1,NI
      Y (IG, I,NJ) = 0.0
      YD(IG, I,NJ) = 0.0
      Y (IG,I,1) = Y (IG,1,1)
740   YD(IG,I,1) = YD(IG,1,1)
C
C   DENOMINATOR CALCULATION
C
750   DO 820 IG = 1,NG
      DO 810 IX = 1,NX
      I = 2*IX

```

230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276

DO 800 JY = 1,NY	277
J = 2*JY	278
IM = M(IX,JY)	279
DEN(IG,I,J) = PRX(2,2,IX)*PRR(2,2,JY)*A1(IG,IG,IM) + D1(IG,IM)*	280
1(PRX(2,2,IX)*QRR(2,2,JY) + QRX(2,2,IX)*PRT(2,2,JY))	281
IF(NX-IX)770,770,760	282
760 IM1 = M(IX+1,JY)	283
DEN(IG,I+1,J) = PRR(2,2,JY)*(PRX(3,3,IX)*A1(IG,IG,IM) + PRX(1,1,	284
1IX+1)*A1(IG,IG,IM1)) + D1(IG,IM)*(QRX(3,3,IX)*PRT(2,2,JY) +	285
2PRX(3,3,IX)*QRR(2,2,JY)) + D1(IG,IM1)*(QRX(1,1,IX+1)*PRT(2,2,JY)	286
3+ PRX(1,1,IX+1)*QRR(2,2,JY))	287
770 IF(NY-JY)800,800,780	288
780 IM2 = M(IX,JY+1)	289
DEN(IG,I,J+1) = PRX(2,2,IX)*(PRR(3,3,JY)*A1(IG,IG,IM) + PRR(1,1,	290
1JY+1)*A1(IG,IG,IM2)) + D1(IG,IM)*(QRX(2,2,IX)*PRT(3,3,JY) +	291
2PRX(2,2,IX)*QRR(3,3,JY)) + D1(IG,IM2)*(QRX(2,2,IX)*PRT(1,1,JY+1)	292
3+ PRX(2,2,IX)*QRR(1,1,JY+1))	293
IF(NX-IX)800,800,790	294
790 IM3 = M(IX+1,JY+1)	295
DEN(IG,I+1,J+1) = PRR(3,3,JY)*(A1(IG,IG,IM)*PRX(3,3,IX) + A1(IG,	296
1IG,IM1)*PRX(1,1,IX+1)) + PRR(1,1,JY+1)*(A1(IG,IG,IM2)*PRX(3,3,IX	297
2) + A1(IG,IG,IM3)*PRX(1,1,IX+1)) + PRT(1,1,JY+1)*	298
3(D1(IG,IM2)*QRX(3,3,IX) + D1(IG,IM3)*QRX(1,1,IX+1)) +	299
4 QRR(1,1,JY+1)*(D1(IG,IM2)*PRX(3,3,IX) + D1(IG,IM3)*PRX(1,1,IX+1)	300
5) + PRT(3,3,JY)*(D1(IG,IM)*QRX(3,3,IX) + D1(IG,IM1)*QRX(1,1,IX+1	301
6)) + QRR(3,3,JY)*(D1(IG,IM)*PRX(3,3,IX) + D1(IG,IM1)*	302
7 PRX(1,1,IX+1))	303
800 CONTINUE	304
810 CONTINUE	305
820 CONTINUE	306
IF(NGEOM-3)860,830,830	307
830 DO 850 IG = 1,NG	308
SUM = 0.0	309
DO 840 IX = 1,KX	310
IM = M(IX,1)	311
840 SUM = SUM+(-30.0*D1(IG,IM)/S-A1(IG,IG,IM))*DX(IX)	312
850 DENC(IG) = SUM	313
860 WRITE (12) (((A1(IG,IGP,IM),IG=1,NG),IGP=1,NG),IM=1,NM),((B(IG,IM)	314
1,CHI1(IG,IM),D1(IG,IM),SF11(IG,IM),IG=1,NG),IM=1,NM)	315
IF(SIZE)870,870,880	316
870 SIZE = 1.0E+6	317
880 TIME = CLOCK(XMON,DAY)	318
WRITE (6,10017) ITOUT,ITINN,ITPRNT,ISSW,RELAX	319
10017 FORMAT(/// 7HOITOUT=I3, 5X6HITINN=I3, 5X7HITPRNT=I2,	320
1 5X5HISSW=I2, 10X6HRELAX= 1PE14.7 ///	321
WRITE (6,10018) TIME	322
10018 FORMAT(77HCENTRY TO THE STEADY STATE SECTION.	323

1	COMPUTER TIME = -3PF6.3)	324
	RETURN	325
	END	326
\$IBFTC	SYM LIST,DECK,M94/2,SDD,XR7	
	SUBROUTINE SYMM(PRR,QRR)	1
C		2
	DIMENSION PRR(3,3,1),QRR(3,3,1)	3
C		4
	PRR(2,2,1) = PRR(2,2,1) + 8.0*PRR(1,2,1)/3.0	5
	PRR(2,3,1) = PRR(2,3,1) + (4.0*PRR(1,3,1) - PRR(1,2,1))/3.0	6
	PRR(3,2,1) = PRR(2,3,1)	7
	PRR(3,3,1) = PRR(3,3,1) - 2.0*PRR(1,3,1)/3.0	8
	PRR(1,3,1) = -PRR(1,1,1)/3.0	9
	PRR(1,2,1) = 4.0*PRR(1,1,1)/3.0	10
	PRR(2,1,1) = PRR(1,2,1)	11
	PRR(3,1,1) = PRR(1,3,1)	12
	PRR(1,1,1) = -PRR(1,1,1)	13
	QRR(2,2,1) = QRR(2,2,1) + 8.0*QRR(1,2,1)/3.0	14
	QRR(2,3,1) = QRR(2,3,1) + (4.0*QRR(1,3,1) - QRR(1,2,1))/3.0	15
	QRR(3,2,1) = QRR(2,3,1)	16
	QRR(3,3,1) = QRR(3,3,1) - 2.0*QRR(1,3,1)/3.0	17
	QRR(1,3,1) = -QRR(1,1,1)/3.0	18
	QRR(1,2,1) = 4.0*QRR(1,1,1)/3.0	19
	QRR(2,1,1) = QRR(1,2,1)	20
	QRR(3,1,1) = QRR(1,3,1)	21
	QRR(1,1,1) = -QRR(1,1,1)	22
	RETURN	23
	END	24
\$IBFTC	GUSS LIST,DECK,M94/2,SDD,XR7	-0
	SUBROUTINE GUESS(Y,YD,DX,DY,SFI1,M,NGES,NGEOM,NX,NY,NG,NI,NJ,LG,LI	1
	1,LX)	2
C		3
	DIMENSION Y(LG,LI,1),YD(LG,LI,1),DX(1),DY(1),SFI1(LG,1),M(LX,1),	4
	1XCOSE(200),YCOSE(200)	5
C		6
	IF(NGES)10,220,260	7
10	XE = 0.0	8
	YE = 0.0	9
	PIE = 3.1415927	10
	XCOSE(1) = 1.0	11
	YCOSE(1) = 1.0	12
	DO 20 I = 1,NX	13
20	XE = XE + DX(I)	14
	DO 30 J = 1,NY	15
30	YE = YE + DY(J)	16
	IF(NGEOM)40,40,110	17
C	XY SECTION INITIAL FLUX GUESS CALCULATION (SCRATCH).	18

	40 IF(NGEOM+1)70,60,50	19
C	NO SYMMETRY	20
	50 XO = 0.0	21
	YO = 0.0	22
	GO TO 80	23
C	SINGLE SYMMETRY	24
	60 XO = 0.0	25
	YO = YE	26
	YE = 2.0*YE	27
	GO TO 80	28
C	DOUBLE SYMMETRY	29
	70 XO = XE	30
	XE = 2.0*XE	31
	YO = YE	32
	YE = 2.0*YE	33
	80 XSUM = XO	34
	DO 90 IX=1,NX	35
	I1 = 2*IX	36
	I2 = I1+1	37
	XSUM = XSUM + 0.5*DX(IX)	38
	XCOSE(I1) = COS(PIE*(XSUM-0.5*XE)/XE)	39
	XSUM = XSUM + 0.5*DX(IX)	40
	90 XCOSE(I2) = COS(PIE*(XSUM-0.5*XE)/XE)	41
	YSUM = YO	42
	DO 100 JY=1,NY	43
	I1 = 2*JY	44
	I2 = I1+1	45
	YSUM = YSUM + 0.5*DY(JY)	46
	YCOSE(I1) = COS(PIE*(YSUM-0.5*YE)/YE)	47
	YSUM = YSUM + 0.5*DY(JY)	48
	100 YCOSE(I2) = COS(PIE*(YSUM-0.5*YE)/YE)	49
	GO TO 200	50
	110 IF(NGEOM-2)120,130,160	51
C	RZ-RTHETA SECTION FLUX GUESS CALCULATION (SCRATCH).	52
	120 XO = 0.0	53
	GO TO 140	54
	130 XO = XE	55
	XE = 2.0*XE	56
	140 XSUM = XO	57
	DO 150 IX=1,NX	58
	I1 = 2*IX	59
	I2 = I1+1	60
	XSUM = XSUM + 0.5*DX(IX)	61
	XCOSE(I1) = COS(PIE*(XSUM-0.5*XE)/XE)	62
	XSUM = XSUM + 0.5*DX(IX)	63
	150 XCOSE(I2) = COS(PIE*(XSUM-0.5*XE)/XE)	64
	GO TO 180	65

160 DO 170 I=2,NI	66
170 XCOSE(I) = 1.0	67
180 YSUM = 0.0	68
DO 190 JY=1,NY	69
I1 = 2*JY	70
I2 = I1+1	71
YSUM = YSUM +0.5*DY(JY)	72
YCOSE(I1) = BESLJO(2.405*YSUM/YE)	73
YSUM = YSUM +0.5*DY(JY)	74
190 YCOSE(I2) = BESLJO(2.405*YSUM/YE)	75
200 DO 210 I=1,NI	76
DO 210 J=1,NJ	77
DO 210 IG=1,NG	78
210 Y(IG,I,J) = XCOSE(I)*YCOSE(J)	79
220 DO 250 IG = 1,NG	80
DO 250 I=1,NI	81
DO 250 J=1,NJ	82
250 YD(IG,I,J) = Y(IG,I,J)	83
260 RETURN	84
END	85
\$IBFTC BEJO LIST,DECK,M94/2,SDD,XR7	-0
FUNCTION BESLJO(X)	1
CC	2
C	3
C BESLJO IS A FUNCTION TYPE SUBPROGRAM DESIGNED TO SOLVE JO(X)	4
C BESSEL FUNCTIONS ACCURATE TO THE NEAREST 0.0001.	5
C	6
C D. JOHNS	7
C OCT 1965	8
C	9
CC	10
DATA EPS/0.00000001/	11
PLY = 1.	12
TEMP = 1.	13
T2 = 1.	14
X2 = X/2.	15
DO 40 I = 1,1000	16
AYE = I	17
PLY = PLY*((X2/AYE)**2)	18
T1 = PLY	19
IF (T1-EPS) 50,50,10	20
10 IF (T2) 30,40,20	21
20 T1 = -T1	22
30 T2 = -T2	23
TEMP = TEMP + T1	24
40 CONTINUE	25
50 BESLJO = TEMP	26

```

      RETURN
      END
$ORIGIN      ALPHA
$IBFTC PRT2  LIST,DECK,M94/2,SDD,XR7
      SUBROUTINE PART2(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,Y,YD,YS
      1,YDS,DEN,DENC,YCS,YDCS,LX,LG,LI)
C
      DIMENSION PRX(3,3,1), QRX(3,3,1), PRR(3,3,1), QRR(3,3,1)
      1,PRT(3,3,1), M(LX,1), DX(1), DY(1), A1(LG,LG,1), B(LG,1)
      2,CHI1(LG,1), D1(LG,1), Y(LG,LI,1), YD(LG,LI,1), YS(LG,LI,1)
      3,YDS(LG,LI,1), DEN(LG,LI,1), DENC(1), YCS(1), YDCS(1)
C
      COMMON /COEF/ZZ,RR,TH,P(3,3),Q(3,3),PR(3,3),QR(3,3),AA(3,3)
      1,BB(3,3),CC(3,3),DD(3,3),EE(3,3),FF(3,3),GG(3,3),HH(3,3),XX,YY
      2,TEMP(14)
C
      COMMON NG,NM,NX,NY,NGEOM,NGES,NPCH,ITOUT,ITINN,ITPRNT,ISSW,RELAX
      1,DRELAX,E1,TIM,DTIM,S,INTPX,INTPY,NIM1,NIM2,NI,NJ,AMB,SIZE,SIZED
      2,IPD,IGEOM,KX,JX,NPIE,NSC,NORM,NEXIT,NST,E3
C
      DOUBLE PRECISION XNUIT
C
      EQUIVALENCE (XNUIT,TEMP)
C
      DATA JFU/6HFLUADJ/
C
      REWIND 10
      XNORM = SIZE
      TAMB = 1.0E+20
      RELAXS = RELAX -1.0
      KI = 2*JX + 1
      KTOUT = 0
      KTPRNT = 0
      IP = 1
      XNUIT = 0.0
      SUM = 0.0
      DO 10 IX=1,JX
10  SUM = SUM + DX(IX)
      HH(1,1) = SUM
      GO TO 80
20  IP = 2
C
C      EIGENVALUE CALCULATION
C
30  XNUM1 = 0.0
      XNUM2 = 0.0
      DEN1 = 0.0

```

```

27
28
-0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

DO 60 IX = 1,JX	44
I = 2*IX	45
DO 60 JY = 1,NY	46
J = 2*JY	47
IM = M(IX,JY)	48
DO 60 IG = 1,NG	49
SUM = 0.0	50
DO 40 I1 = 1,3	51
IA = I + I1-2	52
DO 40 J1 = 1,3	53
JA = J + J1-2	54
DO 40 I2 = 1,3	55
IF = I + I2-2	56
DO 40 J2 = 1,3	57
JF = J + J2-2	58
40 SUM = SUM + YD(IG,IA,JA)*Y(IG,IF,JF)*(QRX(I1,I2,IX)*PRT(J1,J2,JY)	59
1+ PRX(I1,I2,IX)*QRR(J1,J2,JY))	60
XNUM2 = XNUM2 + D1(IG,IM)*SUM	61
DO 60 IGP = 1,NG	62
SUM = 0.0	63
DO 50 I1 = 1,3	64
IA = I + I1-2	65
DO 50 J1 = 1,3	66
JA = J + J1-2	67
DO 50 I2 = 1,3	68
IF = I + I2-2	69
DO 50 J2 = 1,3	70
JF = J + J2-2	71
50 SUM = SUM + YD(IG,IA,JA)*Y(IGP,IF,JF)*PRX(I1,I2,IX)*PRR(J1,J2,JY)	72
XNUM1 = XNUM1 + A1(IG,IGP,IM)*SUM	73
60 DEN1 = DEN1 + CH11(IG,IM)*B(IGP,IM)*SUM	74
AMB = (XNUM1+XNUM2)/DEN1	75
E2 = ABS ((AMB-TAMB)/TAMB)	76
TAMB = AMB	77
TME = 0.001*CLOCK (SUM1,SUM2) -DTIM	78
TIME = 60.0*TME	79
RAMB = 1.0/AMB	80
G = 1.0	81
IF(E2.LT.1.0E-5) G = 0.0	82
RELAX = 1.0 + G*DRELAX*(0.999999-EXP(-88.0*E2)) + G*RELAXS	83
KTPRNT = KTPRNT + 1	84
WRITE (6,10001) RAMB,RELAX,TIME,XNUIT	85
10001 FORMAT(13H0EIGENVALUE = 1PE14.7 , 4X 8HRELAX = OPF9.7 ,	86
1 4X 7HTIME = OPF5.2 , 4X 10HITCOUNT = OPD20.16,8H GROUPS //)	87
IF(ITPRNT-KTPRNT)70 ,70 ,120	88
70 KTPRNT = 0	89
IF(NORM)80,90,90	90

C		91
C	NORMALIZATION	92
C		93
	80 CALL NROM (Y,YD,XNORM,NG,NI,NJ,IG,LI)	94
	90 DO 100 IG = 1,NG	95
	WRITE (6,10002) IG	96
10002	FORMAT(19H0 FLUX GROUP NO. 13)	97
	DO 100 I = 1,KI	98
100	WRITE (6,10003) (Y(IG,I,J),J=1,NJ)	99
10003	FORMAT(1P5E20.7)	100
	DO 110 IG = 1,NG	101
	WRITE (6,10004) IG	102
10004	FORMAT(22H0 ADJOINT GROUP NO. 13)	103
	DO 110 I = 1,KI	104
110	WRITE (6,10003) (YD(IG,I,J),J=1,NJ)	105
	GO TO (20,120,200),IP	106
120	KTOUT = KTOUT +1	107
	IF(ISSW)170,150,130	108
130	PRINT 10001, RAMB,RELAX,TIME,XNUIT	109
140	CALL SSWTCH (1,IS1)	110
	GO TO (180,160),IS1	111
150	IF(TME -TIM)160,180,180	112
160	IF(KTOUT-ITOUT)340,340,180	113
170	IF(E2-E1)180,150,150	114
180	IP = 3	115
	IF(NGEOM-2)180,80,190	116
190	NX = JX	117
	NI = NI-2	118
	GO TO 80	119
200	IF(NPCH)280,210,210	120
210	IF(IPD)220,250,250	121
220	WRITE (10) JFU	122
	DO 230 IG=1,NG	123
230	WRITE (10) ((Y(IG,I,J),I=1,NI),J=1,NJ)	124
	DO 240 IG=1,NG	125
240	WRITE (10) ((YD(IG,I,J),I=1,NI),J=1,NJ)	126
	END FILE 10	127
	REWIND 10	128
	GO TO 280	129
250	DO 260 IG=1,NG	130
260	WRITE (7,10005) ((Y(IG,I,J),I=1,NI),J=1,NJ)	131
	DO 270 IG = 1,NG	132
270	WRITE (7,10005)((YD(IG,I,J),I=1,NI),J=1,NJ)	133
10005	FORMAT(5E15.8)	134
280	TIME = CLOCK (SUM1,SUM2)	135
	WRITE (6,10006) TIME	136
10006	MA-- '// --X1C --MP --R 1 = --PF6)	137

290 IF(ISSW)310,310,300	138
300 PRINT 10007	139
10007 FORMAT(42H SET ALL SENSE SWITCHES UP AND HIT START.)	140
PAUSE	141
310 IF(NEXIT)320,330,330	142
320 STOP	143
330 RETURN	144
C	145
C SOURCE CALCULATION	146
C	147
340 CALL SOURC(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,Y,YD,YS,YDS	148
1,DEN,DENC,YCS,YDCS,LX,LG,LI)	149
IF(NSC.LT.0.OR.NGEOM.GT.2) GO TO 560	
C	151
C INNER ITERATION (SCALING SECTION) Y*SIZE/SUM	152
C	153
350 SUM = 0.0	154
SUMS = 0.0	155
DO 540 IX = 1,KX	156
I = 2*IX	157
IF(IX-1)360,360,370	158
360 I21 = 2	159
GO TO 380	160
370 I21 = 1	161
380 IF(IX-KX)400,390,390	162
390 I22 = 2	163
GO TO 410	164
400 I22 = 3	165
410 DO 540 JY = 1,NY	166
J = 2*JY	167
IM = M(IX,JY)	168
IF(JY-1)420,420,430	169
420 J21 = 2	170
GO TO 440	171
430 J21 = 1	172
440 IF(JY-NY)460,450,450	173
450 J22 = 2	174
GO TO 470	175
460 J22 = 3	176
470 DO 500 I1 = 1,3	177
SUM1 = 0.0	178
SUM2 = 0.0	179
DO 480 I2 = I21,I22	180
SUM1 = SUM1 + PRX(I1,I2,IX)	181
480 SUM2 = SUM2 + QRX(I1,I2,IX)	182
TEMP(I1) = SUM1	183
TEMP(I1+3) = SUM2	184

	SUM1 = 0.0	185
	SUM2 = 0.0	186
	SUM7 = 0.0	187
	DO 490 J2 = J21,J22	188
	SUM7 = SUM7 + PRR(I1,I2,JY)	189
	SUM1 = SUM1 + PRT(I1,J2,JY)	190
490	SUM2 = SUM2 + QRR(I1,J2,JY)	191
	GG(I1,1) = SUM7	192
	TEMP(I1+6) = SUM1	193
500	TEMP(I1+9) = SUM2	194
	DO 530 IG = 1,NG	195
	JG = NG - IG +1	196
	SUM5 = 0.0	197
	SUM6 = 0.0	198
	DO 510 IGP = IG,NG	199
	JGP = NG - IGP +1	200
	SUM5 = SUM5 + A1(JG,JGP,IM)	201
510	SUM6 = SUM6 + A1(IGP,IG,IM)	202
	SUM1 = 0.0	203
	SUM2 = 0.0	204
	SUM3 = 0.0	205
	SUM4 = 0.0	206
	DO 520 I1 = 1,3	207
	IA = I + I1-2	208
	DO 520 J1 = 1,3	209
	JA = J + J1-2	210
	SUM1 = SUM1 + TEMP(I1)*GG(J1,1) *Y(IG,IA,JA)	211
	SUM2 = SUM2 + (TEMP(I1)*TEMP(J1+9) + TEMP(I1+3)*TEMP(J1+6))*	212
1	Y(IG,IA,JA)	213
	SUM3 = SUM3 + TEMP(I1)*GG(J1,1) *YD(JG,IA,JA)	214
520	SUM4 = SUM4 + (TEMP(I1)*TEMP(J1+9) + TEMP(I1+3)*TEMP(J1+6))*	215
1	YD(JG,IA,JA)	216
	SUM5 = SUM5 + SUM3*SUM5 + D1(JG,IM)*SUM4	217
530	SUM = SUM + SUM1*SUM6 + D1(IG,IM)*SUM2	218
540	CONTINUE	219
	SUM1 = SIZE/SUM	220
	SUM2 = SIZED/SUMS	221
	DO 550 IG = 1,NG	222
	DO 550 I = 1,NI	223
	DO 550 J = 1,NJ	224
	Y (IG,I,J) = SUM1*Y (IG,I,J)	225
550	YD(IG,I,J) = SUM2*YD(IG,I,J)	226
C		227
C	THE START OF THE ACTUAL INNER ITERATION	228
C		229
560	IF(NGEDM.GT.3) GO TO 565	
	CALL ITERR(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,Y,YD,YS,YDS	230

1,DEN,DENC,YCS,YDCS,LX,LG,LI)	
GO TO 568	
565 CALL ITERT(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,Y,YD,YS,YDS	230
1,DEN,DENC,YCS,YDCS,LX,LG,LI)	231
C	232
568 IF(NORM)30,570,570	233
570 CALL NROM (Y,YD,XNORM,NG,NI,NJ,LG,LI)	234
GO TO 30	235
END	236
\$IBFTC NRO LIST,DECK,M94/2,SDD,XR7	-0
SUBROUTINE NROM (Y,YD,XNORM,NG,NI,NJ,LG,LI)	1
C	2
DIMENSION Y (LG,LI,1),YD(LG,LI,1)	3
C	4
10 YMAX = Y(1,1,1)	5
YDMAX = YD(1,1,1)	6
DO 50 IG = 1,NG	7
DO 50 I = 1,NI	8
DO 50 J = 1,NJ	9
SUM1 = Y(IG,I,J)	10
IF(SUM1 - YMAX)30,30,20	11
20 YMAX = SUM1	12
30 SUM2 = YD(IG,I,J)	13
IF(SUM2 - YDMAX)50,50,40	14
40 YDMAX = SUM2	15
50 CONTINUE	16
YMAX = YMAX/XNORM	17
YDMAX = YDMAX/XNORM	18
DO 60 IG = 1,NG	19
DO 60 I = 1,NI	20
DO 60 J = 1,NJ	21
Y (IG,I,J) = Y (IG,I,J)/YMAX	22
60 YD(IG,I,J) = YD(IG,I,J)/YDMAX	23
RETURN	24
END	25
\$IBFTC TRAN LIST,DECK,M94/2,SDD,XR7	-0
SUBROUTINE TRANS(A1,Y,YD,I1,I2,I3,I4,I5,NG,LG,LI,SUM1,SUM2)	1
C	2
DIMENSION A1(LG,LG,1),Y(LG,LI,1),YD(LG,LI,1)	3
C	4
IG = I1	5
JG = I2	6
IM = I3	7
IF = I4	8
JF = I5	9
SUM1 = 0.0	10
SUM2 = 0.0	11

DO 10 IGP = 1,IG	12
JGP = NG - IGP +1	13
SUM1 = SUM1 + A1(JGP,JG,IM)*YD(JGP,IF,JF)	14
10 SUM2 = SUM2 + A1(IG,IGP,IM)*Y (IGP,IF,JF)	15
RETURN	16
END	17
\$ORIGIN BETA,SYSUT3	
\$IBFTC SOUR LIST,DECK,M94/2,SDD,XR7	-0
SUBROUTINE SOURC(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,Y,YD,YS	1
1,YDS,DEN,DENC,YCS,YDCS,LX,LG,LI)	2
C	3
DIMENSION PRX(3,3,1), QRX(3,3,1), PRR(3,3,1), QRR(3,3,1)	4
1,PRT(3,3,1), M(LX,1), DX(1), DY(1), A1(LG,LG,1), B(LG,1)	5
2,CHI1(LG,1), D1(LG,1), Y(LG,LI,1), YD(LG,LI,1), YS(LG,LI,1)	6
3,YDS(LG,LI,1), DEN(LG,LI,1), DENC(1), YCS(1), YDCS(1)	7
C	8
COMMON /COEF/ZZ,RR,TH,P(3,3),Q(3,3),PR(3,3),QR(3,3),AA(3,3)	9
1,BB(3,3),CC(3,3),DD(3,3),EE(3,3),FF(3,3),GG(3,3),HH(3,3),XX,YY	10
2,TEMP(14)	11
C	12
COMMON NG,NM,NX,NY,NGEOM,NGES,NPCH,ITOUT,ITINN,ITPRNT,ISSW,RELAX	13
1,DRELAX,E1,TIM,DTIM,S,INTPX,INTPY,NIM1,NIM2,NI,NJ,AMB,SIZE,SIZED	14
2,IPD,IGEOM,KX,JX,NPIE,NSC,NORM,NEXIT,NST,E3	15
C	16
SIZE = 0.0	17
SIZED = 0.0	18
DO 250 IG = 1,NG	19
G = 1.0	20
DO 250 IX = 1,NX	21
IF(NGEOM.GT.2.AND.IX.EQ.NX) G = 0.0	22
I = 2*IX	23
DO 250 JY = 1,NY	24
IA = I	25
J = 2*JY	26
JA = J	27
IM = M(IX,JY)	28
SUM = 0.0	29
SUMD = 0.0	30
T1 = 0.0	31
T2 = 1.0	32
DO 50 IGP = 1,NG	33
IF(IG-IGP)20,10,30	34
10 T2 = 0.0	35
GO TO 30	36
20 T1 = 1.0	37
30 SUM1 = 0.0	38
SUM2 = 0.0	39

DO 40 I1 = 1,3	40
IF = IA+I1-2	41
DO 40 J1 = 1,3	42
JF = JA+J1-2	43
SUM1 = SUM1 + PRX(I1,2,IX)*PRR(J1,2,JY)*YD(IGP,IF,JF)	44
40 SUM2 = SUM2 + PRX(I1,2,IX)*PRR(J1,2,JY)*Y (IGP,IF,JF)	45
SUMD = SUMD + SUM1*(CHI1(IGP,IM)*B(IG,IM)*AMB - A1(IGP,IG,IM)*T2)	46
50 SUM = SUM + SUM2*(CHI1(IG, IM)*B(IGP,IM)*AMB- A1(IG,IGP,IM)*T1)	47
YDS(IG,IA,JA) = SUMD	48
YS (IG,IA,JA) = SUM	49
SIZE = SIZE + SUM*G	50
SIZED = SIZED + SUMD*G	51
IF(NX-IX)120,120,60	52
60 SUM = 0.0	53
SUMD = 0.0	54
IA = I + 1	55
DO 110 IS = 1,2	56
I2 = 5 -2*IS	57
I1 = IX +IS-1	58
IM = M(I1,JY)	59
T1 = 0.0	60
T2 = 1.0	61
DO 110 IGP = 1,NG	62
IF(IG-IGP)80,70,90	63
70 T2 = 0.0	64
GO TO 90	65
80 T1 = 1.0	66
90 SUM1 = 0.0	67
SUM2 = 0.0	68
DO 100 I1 = 1,3	69
IF = IA- 5 + 2*IS +I1	70
DO 100 J1 = 1,3	71
JF = JA+J1-2	72
SUM1 = SUM1 + PRX(I1,I2,I1)*PRR(J1,2,JY)*YD(IGP,IF,JF)	73
100 SUM2 = SUM2 + PRX(I1,I2,I1)*PRR(J1,2,JY)*Y (IGP,IF,JF)	74
SUMD = SUMD + SUM1*(CHI1(IGP,IM)*B(IG,IM)*AMB - A1(IGP,IG,IM)*T2)	75
110 SUM = SUM + SUM2*(CHI1(IG, IM)*B(IGP,IM)*AMB- A1(IG,IGP,IM)*T1)	76
YDS(IG,IA,JA) = SUMD	77
YS (IG,IA,JA) = SUM	78
SIZE = SIZE +SUM*G	79
SIZED = SIZED +SUMD*G	80
120 IF(NY-JY)250,250,130	81
130 SUM = 0.0	82
SUMD = 0.0	83
IA = I	84
JA = J + 1	85
DO 180 JS = 1,2	86

J2 = 5 -2*JS	87
JJ = JY +JS-1	88
IM = M(IX,JJ)	89
T1 = 0.0	90
T2 = 1.0	91
DO 180 IGP = 1,NG	92
IF(IG-IGP)150,140,160	93
140 T2 = 0.0	94
GO TO 160	95
150 T1 = 1.0	96
160 SUM1 = 0.0	97
SUM2 = 0.0	98
DO 170 I1 = 1,3	99
IF = IA+I1-2	100
DO 170 J1 = 1,3	101
JF = JA-5 + 2*JS +J1	102
SUM1 = SUM1 + PRX(I1,2,IX)*PRR(J1,J2,JJ)*YD(IGP,IF,JF)	103
170 SUM2 = SUM2 + PRX(I1,2,IX)*PRR(J1,J2,JJ)*Y (IGP,IF,JF)	104
SUMD = SUMD + SUM1*(CHI1(IGP,IM)*B(IG,IM)*AMB - A1(IGP,IG,IM)*T2)	105
180 SUM = SUM + SUM2*(CHI1(IG ,IM)*B(IGP,IM)*AMB- A1(IG,IGP,IM)*T1)	106
YDS(IG,IA,JA) = SUMD	107
YS (IG,IA,JA) = SUM	108
SIZE = SIZE +SUM*G	109
SIZED = SIZED +SUMD*G	110
IF(NX-IX)250,250,190	111
190 SUM = 0.0	112
SUMD = 0.0	113
IA = I + 1	114
DO 240 IS = 1,2	115
I2 = 5 -2*IS	116
II = IX +IS-1	117
DO 240 JS = 1,2	118
J2 = 5 -2*JS	119
JJ = JY +JS-1	120
IM = M(II,JJ)	121
T1 = 0.0	122
T2 = 1.0	123
DO 240 IGP = 1,NG	124
IF(IG-IGP)210,200,220	125
200 T2 = 0.0	126
GO TO 220	127
210 T1 = 1.0	128
220 SUM1 = 0.0	129
SUM2 = 0.0	130
DO 230 I1 = 1,3	131
IF = IA-5 +2*IS +I1	132
DO 230 J1 = 1,3	133

	J1 = JA-5 4 - S + J1	134
	SUM1 = SUM1 + PRX(I1,I2,I1)*PRR(J1,J2,JJ)*YD(IGP,IF,JF)	135
230	SUM2 = SUM2 + PRX(I1,I2,I1)*PRR(J1,J2,JJ)*Y (IGP,IF,JF)	136
	SUMD = SUMD + SUM1*(CHI1(IGP,IM)*B(IG,IM)*AMB - A1(IGP,IG,IM)*T2)	137
240	SUM = SUM + SUM2*(CHI1(IG ,IM)*B(IGP,IM)*AMB- A1(IG,IGP,IM)*T1)	138
	YDS(IG,IA,JA) = SUMD	139
	YS (IG,IA,JA) = SUM	140
	SIZE = SIZE +SUM*G	141
	SIZED = SIZED +SUMD*G	142
250	CONTINUE	143
	IF(NGEOM.LT.3) GO TO 330	
C		145
C	CENTER POINT SOURCE CALCULATION	146
C		147
260	DO 320 IG = 1,NG	148
	SUM = 0.0	149
	SUMD = 0.0	150
	DO 310 IX = 1,JX	151
	I2 = 2*IX	152
	I1 = I2-1	153
	I3 = I2+1	154
	IM = M(IX,1)	155
	T1 = 0.0	156
	T2 = 1.0	157
	SUM1 = 0.0	158
	SUM2 = 0.0	159
	DO 300 IGP = 1,NG	160
	IF(IG-IGP)280,270,290	161
270	T2 = 0.0	162
	GO TO 290	163
280	T1 = 1.0	164
290	SUM1 = SUM1 +(0.66666666*(0.25*YD(IGP,I1,3)+YD(IGP,I2,3)+0.25*	165
	1YD(IGP,I3,3))-YD(IGP,I,1))*(CHI1(IGP,IM)*B(IG,IM)*AMB	166
	2-A1(IGP,IG,IM)*T2)	167
300	SUM2 = SUM2 +(0.66666666*(0.25*Y (IGP,I1,3)+Y (IGP,I2,3)+0.25*	168
	1Y (IGP,I3,3))-Y (IGP,I,1))*(CHI1(IG,IM)*B(IGP,IM)*AMB	169
	2-A1(IG,IGP,IM)*T1)	170
	SUMD = SUMD +DX(IX)*SUM1	171
310	SUM = SUM +DX(IX)*SUM2	172
	YDCS(IG) = SUMD	173
320	YCS (IG) = SUM	174
330	CONTINUE	175
	RETURN	176
	END	177
\$ORIGIN	BETA,SYSUT3,REW	
\$IBFTC ITER	LIST,DECK,M94/2,SDD,XR7	-0
	SUBROUTINE ITERR(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,Y,YD,YS	1

	1,YDS,DEN,DENC,YCS,YDCS,LX,LG,LI)	2
C		3
	DIMENSION PRX(3,3,1), QRX(3,3,1), PRR(3,3,1), QRR(3,3,1)	4
	1,PRT(3,3,1), M(LX,1), DX(1), DY(1), A1(LG,LG,1), B(LG,1)	5
	2,CHI1(LG,1), D1(LG,1), Y(LG,LI,1), YD(LG,LI,1), YS(LG,LI,1)	6
	3,YDS(LG,LI,1), DEN(LG,LI,1), DENC(1), YCS(1), YDCS(1)	7
C		8
	COMMON /COEF/ZZ,RR,TH,P(3,3),Q(3,3),PR(3,3),QR(3,3),AA(3,3)	9
	1,BB(3,3),CC(3,3),DD(3,3),EE(3,3),FF(3,3),GG(3,3),HH(3,3),XX,YY	10
	2,TEMP(14)	11
C		12
	COMMON NG,NM,NX,NY,NGEOM,NGES,NPCH,ITOUT,ITINN,ITPRNT,ISSW,RELAX	13
	1,DRELAX,E1,TIM,DTIM,S,INTPX,INTPY,NIM1,NIM2,NI,NJ,AMB,SIZE,SIZED	14
	2,IPD,IGEOM,KX,JX,NPIE,NSC,NORM,NEXIT,NST,E3	15
C		16
	DOUBLE PRECISION XNUIT	17
C		18
	EQUIVALENCE (XNUIT,TEMP)	19
C		20
	XNUIT = 0.0	21
	DO 220 IG = 1,NG	22
	JG = NG - IG + 1	23
	KTINN = 0	24
10	KTINN = KTINN + 1	25
	TIN2 = 0.0	26
	DO 150 IX = 1,NX	27
	I = 2*IX	28
	DO 110 JY = 1,NY	29
	IA = I	30
	J = 2*JY	31
	JA = J	32
	IM = M(IX,JY)	33
	SUMD = 0.0	34
	SUM = 0.0	35
	DO 20 I1 = 1,3	36
	IF = IA + I1 - 2	37
	DO 20 J1 = 1,3	38
	JF = JA + J1 - 2	39
	SUM1 = QRX(I1,2,IX)*PRT(J1,2,JY) + PRX(I1,2,IX)*QRR(J1,2,JY)	40
	SUMD = SUMD + SUM1*D1(JG,IM)*YD(JG,IF,JF)	41
	SUM = SUM + SUM1*D1(IG,IM)*Y(IG,IF,JF)	42
	CALL TRANS(A1,Y,YD,IG,JG,IM,IF,JF,NG,LG,LI,SUM1,SUM2)	43
	SUMD = SUMD + SUM1*PRX(I1,2,IX)*PRR(J1,2,JY)	44
20	SUM = SUM + SUM2*PRX(I1,2,IX)*PRR(J1,2,JY)	45
	YD(JG,IA,JA) = YD(JG,IA,JA) + RELAX*(YDS(JG,IA,JA)-SUMD)/	46
	1 DEN(JG,IA,JA)	47
	I = LA) IS ,IA) IM) DI G, JA)	

Y (IG,IA,JA) = Y (IG,IA,JA) + SUM	49
TIN2 = TIN2 + ABS (SUM)	50
IF(NX-IX)50,50,30	51
30 IA = I + 1	52
SUMD = 0.0	53
SUM = 0.0	54
DO 40 IS = 1,2	55
I2 = 5-2*IS	56
I1 = IX + IS - 1	57
IM = M(I1,JY)	58
DO 40 I1 = 1,3	59
IF = IA - 5 + 2*IS + I1	60
DO 40 J1 = 1,3	61
JF = JA + J1 - 2	62
SUM1 = QRX(I1,I2,I1)*PRT(J1,2,JY) + PRX(I1,I2,I1)*QRR(J1,2,JY)	63
SUMD = SUMD + SUM1*D1(JG,IM)*YD(JG,IF,JF)	64
SUM = SUM + SUM1*D1(IG,IM)*Y (IG,IF,JF)	65
CALL TRANS(A1,Y,YD,IG,JG,IM,IF,JF,NG,LG,LI,SUM1,SUM2)	66
SUMD = SUMD + SUM1*PRX(I1,I2,I1)*PRR(J1,2,JY)	67
40 SUM = SUM + SUM2*PRX(I1,I2,I1)*PRR(J1,2,JY)	68
YD(JG,IA,JA) = YD(JG,IA,JA) + RELAX*(YDS(JG,IA,JA)-SUMD)/	69
1 DEN(JG,IA,JA)	70
SUM = RELAX*(YS (IG,IA,JA) - SUM) / DEN(IG,IA,JA)	71
Y (IG,IA,JA) = Y (IG,IA,JA) + SUM	72
TIN2 = TIN2 + ABS (SUM)	73
50 IF(NY-JY)100,100,60	74
60 IA = I	75
JA = J + 1	76
SUMD = 0.0	77
SUM = 0.0	78
DO 70 JS = 1,2	79
J2 = 5 - 2*JS	80
JJ = JY + JS - 1	81
IM = M(IX,JJ)	82
DO 70 I1 = 1,3	83
IF = IA + I1 - 2	84
DO 70 J1 = 1,3	85
JF = JA - 5 + 2*JS + J1	86
SUM1 = QRX(I1,2,IX)*PRT(J1,J2,JJ) + PRX(I1,2,IX)*QRR(J1,J2,JJ)	87
SUMD = SUMD + SUM1*D1(JG,IM)*YD(JG,IF,JF)	88
SUM = SUM + SUM1*D1(IG,IM)*Y (IG,IF,JF)	89
CALL TRANS(A1,Y,YD,IG,JG,IM,IF,JF,NG,LG,LI,SUM1,SUM2)	90
SUMD = SUMD + SUM1*PRX(I1,2,IX)*PRR(J1,J2,JJ)	91
70 SUM = SUM + SUM2*PRX(I1,2,IX)*PRR(J1,J2,JJ)	92
YD(JG,IA,JA) = YD(JG,IA,JA) + RELAX*(YDS(JG,IA,JA)-SUMD)/	93
1 DEN(JG,IA,JA)	94
SUM = RELAX*(YS (IG,IA,JA) - SUM) / DEN(IG,IA,JA)	95

Y (IG,IA,JA) = Y (IG,IA,JA) + SUM	96
TIN2 = TIN2 + ABS (SUM)	97
IF(NX-IX)100,100,80	98
80 IA = I + 1	99
SUMD = 0.0	100
SUM = 0.0	101
DO 90 IS = 1,2	102
I2 = 5-2*IS	103
I1 = IX+IS-1	104
DO 90 JS = 1,2	105
J2 = 5-2*JS	106
JJ = JY+JS-1	107
IM = M(I1,JJ)	108
DO 90 I1 = 1,3	109
IF = IA -5 +2*IS +I1	110
DO 90 J1 = 1,3	111
JF = JA -5 +2*JS +J1	112
SUM1 = QRX(I1,I2,I1)*PRT(J1,J2,JJ) + PRX(I1,I2,I1)*QRR(J1,J2,JJ)	113
SUMD = SUMD + SUM1*D1(JG,IM)*YD(JG,IF,JF)	114
SUM = SUM + SUM1*D1(IG,IM)*Y (IG,IF,JF)	115
CALL TRANS(A1,Y,YD,IG,JG,IM,IF,JF,NG,LG,LI,SUM1,SUM2)	116
SUMD = SUMD +SUM1*PRX(I1,I2,I1)*PRR(J1,J2,JJ)	117
90 SUM = SUM +SUM2*PRX(I1,I2,I1)*PRR(J1,J2,JJ)	118
YD(JG,IA,JA) = YD(JG,IA,JA) + RELAX*(YDS(JG,IA,JA)-SUMD)/	119
1 DEN(JG,IA,JA)	120
SUM = RELAX*(YS (IG,IA,JA) - SUM) / DEN(IG,IA,JA)	121
Y (IG,IA,JA) = Y (IG,IA,JA) + SUM	122
TIN2 = TIN2 + ABS (SUM)	123
100 CONTINUE	124
110 CONTINUE	125
C	126
C	127
C	128
120 IF(IGEOM)150,150,130	129
130 IF(IGEOM-2)140,140,150	130
140 K = I-1	131
YD(JG,I,1) = (4.0*YD(JG,I,2) - YD(JG,I,3))/3.0	132
Y (IG,I,1) = (4.0*Y (IG,I,2) - Y (IG,I,3))/3.0	133
YD(JG,K,1) = (4.0*YD(JG,K,2) - YD(JG,K,3))/3.0	134
Y (IG,K,1) = (4.0*Y (IG,K,2) - Y (IG,K,3))/3.0	135
150 CONTINUE	136
160 IF(IGEOM-2)210,170,200	137
170 DO 180 J = 1,NJ	138
YD(JG,1,J) = (4.0*YD(JG,2,J) - YD(JG,3,J))/3.0	139
180 Y (IG,1,J) = (4.0*Y (IG,2,J) - Y (IG,3,J))/3.0	140
GO TO 210	141
	142

C	R-THETA CENTER POINT CALCULATION AND PERIODICITY SETUP	143
C		144
	200 CALL CENTR(M,DX,A1,D1,Y,YD,YDCS,YCS,DENC,IG,JG,LX,LG,LI)	147
	210 IF(KTINN.LT.2) TIN1 = TIN2	148
	IF(TIN2/TIN1.LT.E3) GO TO 220	149
	IF(KTINN.LT.ITINN) GO TO 10	150
	220 XNUIT = 10.0*XNUIT + FLOAT(KTINN)	151
	RETURN	156
	END	157
	\$IBFTC CENR LIST,DECK,M94/2,SDD,XR7	-0
	SUBROUTINE CENTR(M,DX,A1,D1,Y,YD,YDCS,YCS,DENC,IG,JG,LX,LG,LI)	1
C		2
	DIMENSION M(LX,1),DX(1),A1(LG,LG,1),D1(LG,1),Y(LG,LI,1)	3
	1,YD(LG,LI,1),YDCS(1),YCS(1),DENC(1)	4
C		5
	COMMON /COEF/ZZ,RR,TH,P(3,3),Q(3,3),PR(3,3),QR(3,3),AA(3,3)	6
	1,BB(3,3),CC(3,3),DD(3,3),EE(3,3),FF(3,3),GG(3,3),HH(3,3),XX,YY	7
	2,TEMP(14)	8
C		9
	COMMON NG,NM,NX,NY,NGEOM,NGES,NPCH,ITOUT,ITINN,ITPRNT,ISSW,RELAX	10
	1,DRELAX,E1,TIM,DTIM,S,INTPX,INTPY,NIM1,NIM2,NI,NJ,AMB,SIZE,SIZED	11
	2,IPD,IGEOM,KX,JX,NPIE,NSC,NORM,NEXIT,NST,E3	12
C		13
	10 DO 20 J = 1,NJ	14
	YD(JG,NIM1,J) = YD(JG,2,J)	15
	Y (IG,NIM1,J) = Y (IG,2,J)	16
	YD(JG, NI,J) = YD(JG,3,J)	17
	Y (IG, NI,J) = Y (IG,3,J)	18
	YD(JG, 1,J) = YD(JG,NIM2,J)	19
	20 Y (IG, 1,J) = Y (IG,NIM2,J)	20
	SUM = 0.0	21
	SUMD = 0.0	22
	40 DO 60 IX = 1,KX	38
	I2 = 2*IX	39
	I1 = I2-1	40
	I3 = I2+1	41
	IM = M(IX,1)	42
	SUM1 = D1(JG,IM)*(30.0*YD(JG,1,1)+6.6666666*(0.25*YD(JG,I1,3) +	43
	1YD(JG,I2,3)+0.25*YD(JG,I3,3) - (YD(JG,I1,2)+4.0*YD(JG,I2,2) +	44
	2YD(JG,I3,2))))/S	45
	SUM2 = D1(IG,IM)*(30.0*Y (IG,1,1)+6.6666666*(0.25*Y (IG,I1,3) +	46
	1Y (IG,I2,3)+0.25*Y (IG,I3,3) - (Y (IG,I1,2)+4.0*Y (IG,I2,2) +	47
	2Y (IG,I3,2))))/S	48
	SUM3 = 0.0	49
	SUM4 = 0.0	50
	DO 50 IGP = 1,IG	51
	JGP = NG - IGP +1	52

	SUM3 = SUM3 + A1(JGP,JG,IM)*(YD(JGP,1,1)-0.66666666*(0.25*	53
	1YD(JGP,I1,3)+YD(JGP,I2,3)+0.25*YD(JGP,I3,3)))	54
50	SUM4 = SUM4 + A1(IG,IGP,IM)*(Y (IGP,1,1)-0.66666666*(0.25*	55
	1Y (IGP,I1,3)+Y (IGP,I2,3)+0.25*Y (IGP,I3,3)))	56
	SUMD = SUMD + DX(IX)*(-SUM1- SUM3)*HH(3,3)	57
60	SUM = SUM + DX(IX)*(-SUM2- SUM4)*HH(3,3)	58
	SUMD = YD(JG,1,1) + RELAX*(YDCS(JG) - SUMD)/DENC(JG)	59
	SUM = Y (IG,1,1) + RELAX*(YCS (IG) - SUM)/DENC(IG)	60
70	DO 80 I = 1,NI	61
	YD(JG, I,1) = SUMD	62
80	Y (IG, I,1) = SUM	63
90	RETURN	64
	END	65
	\$ORIGIN BETA,SYSUT3,REW	
	\$IBFTC ITET LIST,DECK,M94/2,SDD,XR7	-0
	SUBROUTINE ITERS(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,Y,YD,YS	1
	1,YDS,DEN,DENC,YCS,YDCS,LX,LG,LI)	2
C		3
	DIMENSION PRX(3,3,1), QRX(3,3,1), PRR(3,3,1), QRR(3,3,1)	4
	1,PRT(3,3,1), M(LX,1), DX(1), DY(1), A1(LG,LG,1), B(LG,1)	5
	2,CHI1(LG,1), D1(LG,1), Y(LG,LI,1), YD(LG,LI,1), YS(LG,LI,1)	6
	3,YDS(LG,LI,1), DEN(LG,LI,1), DENC(1), YCS(1), YDCS(1)	7
C		8
	COMMON /COEF/ZZ,RR,TH,P(3,3),Q(3,3),PR(3,3),QR(3,3),AA(3,3)	9
	1,BB(3,3),CC(3,3),DD(3,3),EE(3,3),FF(3,3),GG(3,3),HH(3,3),XX,YY	10
	2,TEMP(14)	11
C		12
	COMMON NG,NM,NX,NY,NGEOM,NGES,NPCH,ITOUT,ITINN,ITPRNT,ISSW,RELAX	13
	1,DRELAX,E1,TIM,OTIM,S,INTPX,INTPY,NIM1,NIM2,NI,NJ,AMB,SIZE,SIZED	14
	2,IPD,IGEOM,KX,JX,NPIE,NSC,NORM,NEXIT,NST,E3	15
C		16
	DOUBLE PRECISION XNUIT	17
C		18
	EQUIVALENCE (XNUIT,TEMP)	19
C		20
	XNUIT = 0.0	21
	DO 220 IG = 1,NG	22
	JG = NG - IG +1	23
	KTINN = 0	24
10	KTINN = KTINN + 1	25
	TIN2 = 0.0	26
	DO 150 JYY= 1,NY	27
	JY = NY - JYY + 1	28
	J = 2*JY	29
	DO 110 IX=1,KX	30
	I = 2*IX	31
	IF(NY-JY)100,100,60	32

60	I	33
	JA = J + 1	34
	SUMD = 0.0	35
	SUM = 0.0	36
	DO 70 JS = 1,2	37
	J2 = 5 - 2*JS	38
	JJ = JY + JS - 1	39
	IM = M(IX, JJ)	40
	DO 70 I1 = 1,3	41
	IF = IA + I1 - 2	42
	DO 70 J1 = 1,3	43
	JF = JA - 5 + 2*JS + J1	44
	SUM1 = QRX(I1, 2, IX)*PRT(J1, J2, JJ) + PRX(I1, 2, IX)*QRR(J1, J2, JJ)	45
	SUMD = SUMD + SUM1*D1(JG, IM)*YD(JG, IF, JF)	46
	SUM = SUM + SUM1*D1(IG, IM)*Y(IG, IF, JF)	47
	CALL TRANS(A1, Y, YD, IG, JG, IM, IF, JF, NG, LG, LI, SUM1, SUM2)	48
	SUMD = SUMD + SUM1*PRX(I1, 2, IX)*PRR(J1, J2, JJ)	49
70	SUM = SUM + SUM2*PRX(I1, 2, IX)*PRR(J1, J2, JJ)	50
	YD(JG, IA, JA) = YD(JG, IA, JA) + RELAX*(YDS(JG, IA, JA) - SUMD) /	51
	1 DEN(JG, IA, JA)	52
	SUM = RELAX*(YS(IG, IA, JA) - SUM) / DEN(IG, IA, JA)	53
	Y(IG, IA, JA) = Y(IG, IA, JA) + SUM	54
	TIN2 = TIN2 + ABS(SUM)	55
	IF(NX - IX) 100, 100, 80	56
80	IA = I + 1	57
	SUMD = 0.0	58
	SUM = 0.0	59
	DO 90 IS = 1,2	60
	I2 = 5 - 2*IS	61
	I1 = IX + IS - 1	62
	DO 90 JS = 1,2	63
	J2 = 5 - 2*JS	64
	JJ = JY + JS - 1	65
	IM = M(I1, JJ)	66
	DO 90 I1 = 1,3	67
	IF = IA - 5 + 2*IS + I1	68
	DO 90 J1 = 1,3	69
	JF = JA - 5 + 2*JS + J1	70
	SUM1 = QRX(I1, I2, I1)*PRT(J1, J2, JJ) + PRX(I1, I2, I1)*QRR(J1, J2, JJ)	71
	SUMD = SUMD + SUM1*D1(JG, IM)*YD(JG, IF, JF)	72
	SUM = SUM + SUM1*D1(IG, IM)*Y(IG, IF, JF)	73
	CALL TRANS(A1, Y, YD, IG, JG, IM, IF, JF, NG, LG, LI, SUM1, SUM2)	74
	SUMD = SUMD + SUM1*PRX(I1, I2, I1)*PRR(J1, J2, JJ)	75
90	SUM = SUM + SUM2*PRX(I1, I2, I1)*PRR(J1, J2, JJ)	76
	YD(JG, IA, JA) = YD(JG, IA, JA) + RELAX*(YDS(JG, IA, JA) - SUMD) /	77
	1 DEN(JG, IA, JA)	78
	SUM = RELAX*(YS(IG, IA, JA) - SUM) / DEN(IG, IA, JA)	79

Y (IG,IA,JA) = Y (IG,IA,JA) + SUM	80
TIN2 = TIN2 + ABS (SUM)	81
100 CONTINUE	82
IF(IX.NE.1) GO TO 110	
Y (IG,NIM1,J+1) = Y (IG,2,J+1)	84
YD(JG,NIM1,J+1) = YD(JG,2,J+1)	85
Y (IG, NI,J+1) = Y (IG,3,J+1)	86
YD(JG, NI,J+1) = YD(JG,3,J+1)	87
110 CONTINUE	88
Y (IG,1,J+1) = Y (IG,NIM2,J+1)	90
YD(JG,1,J+1) = YD(JG,NIM2,J+1)	91
57 DO 55 IX=1,KX	92
I = 2*IX	93
IA = I	94
JA = J	95
IM = M(IX,JY)	96
SUMD = 0.0	97
SUM = 0.0	98
DO 20 I1 = 1,3	99
IF = IA +I1-2	100
DO 20 J1 = 1,3	101
JF = JA +J1-2	102
SUM1 = QRX(I1,2,IX)*PRT(J1,2,JY) + PRX(I1,2,IX)*QRR(J1,2,JY)	103
SUMD = SUMD + SUM1*D1(JG,IM)*YD(JG,IF,JF)	104
SUM = SUM + SUM1*D1(IG,IM)*Y (IG,IF,JF)	105
CALL TRANS(A1,Y,YD,IG,JG,IM,IF,JF,NG,LG,LI,SUM1,SUM2)	106
SUMD = SUMD + SUM1*PRX(I1,2,IX)*PRR(J1,2,JY)	107
20 SUM = SUM + SUM2*PRX(I1,2,IX)*PRR(J1,2,JY)	108
YD(JG,IA,JA) = YD(JG,IA,JA) + RELAX*(YDS(JG,IA,JA)-SUMD)/	109
1 DEN(JG,IA,JA)	110
SUM = RELAX*(YS (IG,IA,JA) - SUM) / DEN(IG,IA,JA)	111
Y (IG,IA,JA) = Y (IG,IA,JA) + SUM	112
TIN2 = TIN2 + ABS (SUM)	113
IF(JY.NE.1) GO TO 35	
Y (IG,IA,1) = Y (IG,IA,2)	115
YD(JG,IA,1) = YD(JG,IA,2)	116
35 IF(NX-IX)50,50,30	117
30 IA = I + 1	118
SUMD = 0.0	119
SUM = 0.0	120
DO 40 IS = 1,2	121
I2 = 5-2*IS	122
II = IX +IS-1	123
IM = M(II,JY)	124
DO 40 I1 = 1,3	125
IF = IA -5 + 2*IS +I1	126
DO 40 J1 = 1,3	127

	55 = 55 + J1	129
	SUM1 = QRX(I1,I2,I1)*PRT(J1,2,JY) + PRX(I1,I2,I1)*QRR(J1,2,JY)	130
	SUMD = SUMD + SUM1*D1(JG,IM)*YD(JG,IF,JF)	131
	SUM = SUM + SUM1*D1(IG,IM)*Y (IG,IF,JF)	132
	CALL TRANS(A1,Y,YD,IG,JG,IM,IF,JF,NG,LG,LI,SUM1,SUM2)	133
	SUMD = SUMD + SUM1*PRX(I1,I2,I1)*PRR(J1,2,JY)	134
40	SUM = SUM + SUM2*PRX(I1,I2,I1)*PRR(J1,2,JY)	135
	YD(JG,IA,JA) = YD(JG,IA,JA) + RELAX*(YDS(JG,IA,JA)-SUMD)/	136
1	DEN(JG,IA,JA)	137
	SUM = RELAX*(YS (IG,IA,JA) - SUM) / DEN(IG,IA,JA)	138
	Y (IG,IA,JA) = Y (IG,IA,JA) + SUM	139
	TIN2 = TIN2 + ABS (SUM)	141
	IF(JY.NE.1) GO TO 50	142
	Y (IG,IA,1) = Y (IG,IA,2)	144
	YD(JG,IA,1) = YD(JG,IA,2)	145
50	IF(IX.NE.1) GO TO 55	146
	Y (IG,NIM1, J) = Y (IG,2, J)	147
	YD(JG,NIM1, J) = YD(JG,2, J)	148
	Y (IG, NI, J) = Y (IG,3, J)	149
	YD(JG, NI, J) = YD(JG,3, J)	150
	IF(JY.NE.1) GO TO 55	151
	Y (IG,NIM1,1) = Y (IG,2,2)	152
	YD(JG,NIM1,1) = YD(JG,2,2)	153
	Y (IG, NI,1) = Y (IG,3,2)	155
	YD(JG, NI,1) = YD(JG,3,2)	156
55	CONTINUE	157
	Y (IG,1, J) = Y (IG,NIM2, J)	158
	YD(JG,1, J) = YD(JG,NIM2, J)	159
	IF(JY.NE.1) GO TO 56	160
	Y (IG,1,1) = Y (IG,1,2)	171
	YD(JG,1,1) = YD(JG,1,2)	177
56	CONTINUE	179
150	CONTINUE	182
	IF(NGEOM.NE.4) GO TO 210	183
C		184
C	R-THETA CENTER POINT CALCULATION	185
C		186
	CALL CENTT(M,DX,A1,D1,Y,YD,YDCS,YCS,DENC,IG,JG,LX,LG,LI)	191
210	IF(KTINN.LT.2) TIN1 = TIN2	192
	IF(TIN2/TIN1.LT.E3) GO TO 220	
	IF(KTINN.LT.ITINN) GO TO 10	
220	XNUIT = 10.0*XNUIT + FLOAT(KTINN)	
	RETURN	
	END	
\$IBFTC	CENT LIST,DECK,M94/2,SDD,XR7	CENT 1
	SUBROUTINE CENTT(M,DX,A1,D1,Y,YD,YDCS,YCS,DENC,IG,JG,LX,LG,LI)	1
C		CENT 3

	DIMENSION M(LX,1),DX(1),A1(LG,LG,1),D1(LG,1),Y(LG,LI,1)	CENT	4
	1,YD(LG,LI,1),YDCS(1),YCS(1),DENC(1)	CENT	5
C		CENT	6
	COMMON /COEF/ZZ,RR,TH,P(3,3),Q(3,3),PR(3,3),QR(3,3),AA(3,3)	CENT	7
	1,BB(3,3),CC(3,3),DD(3,3),EE(3,3),FF(3,3),GG(3,3),HH(3,3),XX,YY	CENT	8
	2,TEMP(14)	CENT	9
C		CENT	10
	COMMON NG,NM,NX,NY,NGEOM,NGES,NPCH,ITOUT,ITINN,ITPRNT,ISSW,RELAX	CENT	11
	1,DRELAX,E1,TIM,DTIM,S,INTPX,INTPY,NIM1,NIM2,NI,NJ,AMB,SIZE,SIZED	CENT	12
	2,IPD,IGEOM,KX,JX,NPIE,NSC,NORM,NEXIT,NST,E3	CENT	13
C		CENT	14
	SUM = 0.0		
	SUMD = 0.0		
40	DO 60 IX = 1,KX	CENT	30
	I2 = 2*IX	CENT	31
	I1 = I2-1	CENT	32
	I3 = I2+1	CENT	33
	IM = M(IX,1)	CENT	34
	SUM1 = D1(JG,IM)*(30.0*YD(JG,1,1)+6.6666666*(0.25*YD(JG,I1,3) +	CENT	35
	1YD(JG,I2,3)+0.25*YD(JG,I3,3) - (YD(JG,I1,2)+4.0*YD(JG,I2,2) +	CENT	36
	2YD(JG,I3,2)))/S	CENT	37
	SUM2 = D1(IG,IM)*(30.0*Y (IG,1,1)+6.6666666*(0.25*Y (IG,I1,3) +	CENT	38
	1Y (IG,I2,3)+0.25*Y (IG,I3,3) - (Y (IG,I1,2)+4.0*Y (IG,I2,2) +	CENT	39
	2Y (IG,I3,2)))/S	CENT	40
	SUM3 = 0.0	CENT	41
	SUM4 = 0.0	CENT	42
	DO 50 IGP = 1,IG	CENT	43
	JGP = NG - IGP +1	CENT	44
	SUM3 = SUM3 + A1(JGP,JG,IM)*(YD(JGP,1,1)-0.66666666*(0.25*	CENT	45
	1YD(JGP,I1,3)+YD(JGP,I2,3)+0.25*YD(JGP,I3,3)))	CENT	46
50	SUM4 = SUM4 + A1(IG,IGP,IM)*(Y (IGP,1,1)-0.66666666*(0.25*	CENT	47
	1Y (IGP,I1,3)+Y (IGP,I2,3)+0.25*Y (IGP,I3,3)))	CENT	48
	SUMD = SUMD + DX(IX)*(-SUM1- SUM3)	CENT	49
60	SUM = SUM + DX(IX)*(-SUM2- SUM4)	CENT	50
	SUMD = YD(JG,1,1) + RELAX*(YDCS(JG) - SUMD)/DENC(JG)	CENT	51
	SUM = Y (IG,1,1) + RELAX*(YCS (IG) - SUM)/DENC(IG)	CENT	52
70	DO 80 I = 1,NI		
	YD(JG, I,1) = SUMD		
80	Y (IG, I,1) = SUM		
90	RETURN		
	END		
	\$ORIGIN ALPHA,REW		
	\$IBFTC PRT3 LIST,DECK,M94/2,SDD,XR7	PRT3	1
	SUBROUTINE PART3(PRX,QRX,PRR,QRR,PRT,M,DX,DY,A1,B,CHI1,D1,SABS1	PRT3	2
	1,XNU1,SF11,STR1,Y,YD,YDS,DEN,DENC,RAV,LX,LG,LI,LGI,LGJ,LM)	PRT3	3
		PRT3	4
		P	-

	DIMENSION PRX(3,3,1), QRX(3,3,1), PRR(3,3,1), QRR(3,3,1)	PRT3	6
	1, PRT(3,3,1), M(LX,1), DX(1), DY(1), A1(LGI, LGJ), B(LG,1)	PRT3	7
	2, CH1(LG,1), D1(LG,1), SABS1(1), XNU1(1), SF1(LG,1), STR1(LG,1)	PRT3	8
	3, Y(LG,LI,1), YD(LG,LI,1), DEN(LG,LI,1), DENC(1), RAV(1)	PRT3	9
	4, YDS(LG,LI,1)	PRT3	10
C		PRT3	11
	COMMON /COEF/ZZ,RR,TH,P(3,3),Q(3,3),PR(3,3),QR(3,3),AA(3,3)	PRT3	12
	1, BB(3,3), DD(3,3), OO(3,3), EE(3,3), FF(3,3), GG(3,3), HH(3,3), XX, YY	PRT3	13
	2, TEMP(14)	PRT3	14
C		PRT3	15
	COMMON NG,NM,NX,NY,NGEOM,NGES,NPCH,ITOUT,ITINN,ITPRNT,ISSW,RELAX	PRT3	16
	1,DRELAX,E1,TIM,DTIM,S,INTPX,INTPY,NIM1,NIM2,NI,NJ,AMB,SIZE,SIZED	PRT3	17
	2,IPD,IGEOM,KX,JX,NPIE,NSC,NORM,NEXIT,NST,E3	PRT3	18
C		PRT3	19
	IF(NGEOM-2)20,20,10	PRT3	20
	10 NADD = 2	PRT3	21
	GO TO 30	PRT3	22
	20 NADD = 0	PRT3	23
	30 LL = LGI*LGJ	PRT3	24
	L = (LM+1)*LG**2 + 2*LG*(2*LM+1) + 3*NG*NJ*(NI+NADD)	PRT3	25
	DO 40 IG=1,NG	PRT3	26
	40 WRITE (12) ((Y (IG,I,J),I=1,NI),J=1,NJ)	PRT3	27
	DO 50 IG=1,NG	PRT3	28
	50 WRITE (12) ((YD(IG,I,J),I=1,NI),J=1,NJ)	PRT3	29
	REWIND 12	PRT3	30
	DO 60 IG=1,NG	PRT3	31
	DO 60 I=1,NI	PRT3	32
	DO 60 J=1,NJ	PRT3	33
	YDS(IG,I,J) = Y(IG,I,J)	PRT3	34
	60 DEN(IG,I,J) = YD(IG,I,J)	PRT3	35
	IF(LL-L)100,100,70	PRT3	36
	70 WRITE (6,10001)	PRT3	37
	10001 FORMAT(95H1 DECREASE INTPX OR INTPY OR BOTH. THE A1(I,J) MATRIX	PRT3	38
	1 IS NOT LARGE ENOUGH TO HANDLE THIS MANY /	PRT3	39
	2 40H POINTS WITHOUT DESTROYING THE FLUXES.)	PRT3	40
	DO 80 IG = 1,NG	PRT3	41
	WRITE (6,10002) IG	PRT3	42
	10002 FORMAT(19H0 FLUX GROUP NO. I3)	PRT3	43
	DO 80 I = 1,NI	PRT3	44
	80 WRITE (6,10003) (Y(IG,I,J),J=1,NJ)	PRT3	45
	10003 FORMAT(1P5E20.7)	PRT3	46
	WRITE (6,10006)	PRT3	47
	DO 90 IG = 1,NG	PRT3	48
	WRITE (6,10004) IG	PRT3	49
	10004 FORMAT(22H0 ADJOINT GROUP NO. I3)	PRT3	50
	DO 90 I = 1,NI	PRT3	51
	90 WRITE (6,10003) (YD(IG,I,J),J=1,NJ)	PRT3	52

GO TO 110	PRT3	53
100 WRITE (6,10005)	PRT3	54
10005 FORMAT(40H1 INTERPOLATED FLUXES AND ADJOINTS //)	PRT3	55
CALL SECT(7H FLUX,YDS,A1,DX,DY,LG,LI,LGI,LGJ,INTPX,INTPY,NG,NI	PRT3	56
1,NY,NX)	PRT3	57
WRITE (6,10006)	PRT3	58
10006 FORMAT(1H0 //)	PRT3	59
CALL SECT(7HADJOINT,DEN,A1,DX,DY,LG,LI,LGI,LGJ,INTPX,INTPY,NG,NI	PRT3	60
1,NY,NX)	PRT3	61
110 TIME = CLOCK (XMON,DAY)	PRT3	62
WRITE (6,10007) TIME	PRT3	63
10007 FORMAT(/// 61X16HCOMPUTER TIME = -3PF6.3)	PRT3	64
RETURN	PRT3	65
END	PRT3	66
\$IBFTC SEC LIST,DECK,M94/2,SDD,XR7	SECT	1
SUBROUTINE SECT(PRNT,Y,A1,DX,DY,LG,LI,LGI,LGJ,INTPX,INTPY,NG,NI,NY	SECT	2
1,NX)	SECT	3
C	SECT	4
DIMENSION PHI(20),DD(3),YDD(3),PRNT(2),Y(LG,LI,1),A1(LGI,LGJ)	SECT	5
1,DX(1),DY(1)	SECT	6
C	SECT	7
DD(1) = 0.0	SECT	8
NTPX = 2*INTPX + 2	SECT	9
NTPY = 2*INTPY + 2	SECT	10
NTPXM1 = NTPX - 1	SECT	11
NTPYM1 = NTPY - 1	SECT	12
INTPX1 = INTPX + 1	SECT	13
DO 30 IG = 1,NG	SECT	14
DO 10 I = 1,NI	SECT	15
II = (I-1)*INTPX + 1	SECT	16
A1(II,1) = Y(IG,I,1)	SECT	17
DO 10 JY = 1,NY	SECT	18
J = (JY-1)*2 + 1	SECT	19
DD(3) = DY(JY)	SECT	20
DD(2) = 0.5*DD(3)	SECT	21
YDD(1) = Y(IG,I,J)	SECT	22
YDD(2) = Y(IG,I,J+1)	SECT	23
YDD(3) = Y(IG,I,J+2)	SECT	24
CALL INTRP(DD,YDD,PHI,INTPY)	SECT	25
K1 = (JY-1)*NTPY + 2	SECT	26
K2 = K1 + NTPYM1	SECT	27
INK = 1	SECT	28
DO 10 K = K1,K2	SECT	29
INK = INK + 1	SECT	30
10 A1(II,K) = PHI(INK)	SECT	31
DO 20 J = 1,LGJ	SECT	32
DO 20 IX = 1,NX	SECT	33

II = (IX-1)*NTPX + 1	SECT 34
DD(3) = DX(IX)	SECT 35
DD(2) = 0.5*DD(3)	SECT 36
II1 = II + INTPX1	SECT 37
II2 = II1 + INTPX1	SECT 38
YDD(1) = A1(II,J)	SECT 39
YDD(2) = A1(II1,J)	SECT 40
YDD(3) = A1(II2,J)	SECT 41
CALL INTRP(DD,YDD,PHI,INTPX)	SECT 42
K1 = (IX-1)*NTPX + 2	SECT 43
K2 = K1 + NTPXM1	SECT 44
INK = 1	SECT 45
DO 20 K = K1,K2	SECT 46
INK = INK + 1	SECT 47
20 A1(K,J) = PHI(INK)	SECT 48
WRITE (6,10001) PRNT,IG	SECT 49
10001 FORMAT(2H0 A6,A1, 10H GROUP NO. I3)	SECT 50
DO 30 I = 1,LGI	SECT 51
30 WRITE (6,10002) (A1(I,J),J=1,LGJ)	SECT 52
10002 FORMAT(1P5E20.7)	SECT 53
RETURN	SECT 54
END	SECT 55
\$IBFTC INTR LIST,DECK,M94/2,SDD,XR7	INTR 1
SUBROUTINE INTRP(X,Y,AN,INT)	INTR 2
C	INTR 3
DIMENSION X(1),Y(1),AN(1)	INTR 4
C	INTR 5
IX = 2*INT + 3	INTR 6
IXM1 = IX - 1	INTR 7
XIXM1 = IXM1	INTR 8
DIS = X(3)/XIXM1	INTR 9
AN(1) = Y(1)	INTR 10
DO 10 I = 2,IXM1	INTR 11
XII = I - 1	INTR 12
XX = DIS*XII	INTR 13
10 AN(I) = SI(X,Y,XX,3,IND,4,4)	INTR 14
AN(IX) = Y(3)	INTR 15
RETURN	INTR 16
END	INTR 17
\$IBFTC DI NOLIST,NOREF,DECK,M94,XR7	DI 0010
CDI DOUBLE INTERPOLATION (DI) SUBROUTINE	DI 0020
FUNCTION DI(XTBL,YTBL,ZTBL,X,Y,NX,NY,INDX,INDY,INDXX,INDYY,INDA)	DI 0030
C	DI 0040
C INDA = INDICATES WHETHER TO USE ONE OR NY OF THE X TABLES	DI 0050
C (INDA=0 INDICATES NY OF THE X TABLES ARE AVAILABLE)	DI 0060
C (INDA=1 INDICATES ONLY ONE X TABLE IS AVAILABLE)	DI 0070
C INDX = INDICATES TYPE-OF-EXTRAPOLATION USED ON X (IF ANY)	

	UU 220 K=1,3	DI 0550
	IF(IND(K)-1)220,230,240	DI 0560
220	CONTINUE	DI 0570
	INDX=0	DI 0580
	GO TO 260	DI 0590
230	INDX=1	DI 0600
	GO TO 260	DI 0610
240	INDX=2	DI 0620
C		DI 0630
260	RETURN	DI 0640
	END	DI 0650
\$IBFTC	SI NOLIST,NOREF,DECK,M94,XR7	SI 0010
CSI	SINGLE INTERPOLATION (SI) SUBROUTINE	SI 0020
	FUNCTION SI(XTBL,YTBL,XX,NN,IND,INDLE,INDUE)	SI 0030
C		SI 0040
	DIMENSION XTBL(2),YTBL(2)	SI 0050
	X=XX	SI 0060
	N=NN	SI 0070
C		SI 0080
C	IND = INDICATES TYPE-OF-EXTRAPOLATION THAT WAS USED (IF ANY)	SI 0090
C	(IND=0 INDICATES NO EXTRAPOLATION WAS NEEDED ON X)	SI 0100
C	(IND=1 INDICATES LOWER EXTRAPOLATION WAS NEEDED ON X)	SI 0110
C	(IND=2 INDICATES UPPER EXTRAPOLATION WAS NEEDED ON X)	SI 0120
C	INDLE = INDICATES TYPE-OF-LOWER-EXTRAPOLATION TO BE USED ON X	SI 0130
C	(INDLE=1 INDICATES LOWER EXTRAP. ON X IS TO BE LINEAR)	SI 0140
C	(INDLE=2 INDICATES LOWER EXTRAP. ON X IS TO BE PARABOLIC)	SI 0150
C	(INDLE=3 INDICATES LOWER-LINEAR EXTRAP. ON X AND ERROR PRSI	SI 0160
C	(INDLE=4 INDICATES LOWER-PARABOLIC EXTR. ON X AND ERROR PSI	SI 0170
C	INDUE = INDICATES TYPE-OF-UPPER-EXTRAPOLATION TO BE USED ON X	SI 0180
C	(INDUE=1 INDICATES UPPER EXTRAP. ON X IS TO BE LINEAR)	SI 0190
C	(INDUE=2 INDICATES UPPER EXTRAP. ON X IS TO BE PARABOLIC)	SI 0200
C	(INDUE=3 INDICATES UPPER-LINEAR EXTRAP. ON X AND ERROR PRSI	SI 0210
C	(INDUE=4 INDICATES UPPER-PARABOLIC EXTR. ON X AND ERROR PSI	SI 0220
C	N = NUMBER-OF-POINTS IN EACH TABLE	SI 0230
C	X = PARTICULAR VALUE OF INDEPENDENT VARIABLE	SI 0240
C	XTBL = NAME OF INDEPENDENT VARIABLE TABLES	SI 0250
C	YTBL = NAME OF DEPENDENT VARIABLE TABLES	SI 0260
C		SI 0270
C	DIMENSION XTBL(2),YTBL(2)	SI 0280
C		SI 0290
110	IND=0	SI 0300
C		SI 0310
C	CHECK TO SEE IF LOWER OUT-OF-RANGE EXTRAPOLATION WILL BE NEEDED	SI 0320
	IF(X-XTBL(1))120,130,150	SI 0330
C		SI 0340
C	LOWER OUT-OF-RANGE EXTRAPOLATION WAS FOUND NECESSARY (SET IND=1)	SI 0350
120	IND=1	

130	II=2	SI	0360
	GO TO 254	SI	0370
C		SI	0380
C	CHECK TO SEE IF UPPER OUT-OF-RANGE EXTRAPOLATION WILL BE NEEDED	SI	0390
150	IF(XTBL(N)-X)160,180,210	SI	0400
C		SI	0410
C	UPPER OUT-OF-RANGE EXTRAPOLATION WAS FOUND NECESSARY (SET IND=2)	SI	0420
160	IND=2	SI	0430
	II=N	SI	0440
	GO TO (254,180,254,180),INDUE	SI	0450
180	II=N-1	SI	0460
	GO TO 254	SI	0470
C		SI	0480
C	(X IS IN-RANGE MAKE A POINT SEARCH ON X TO OBTAIN II)	SI	0490
210	NM1=N-1	SI	0500
	DO 220 IK=2,NM1	SI	0510
	II=IK	SI	0520
	IF(XTBL(IK)-X)220,254,254	SI	0530
220	CONTINUE	SI	0540
C		SI	0550
254	X1=XTBL(II-1)	SI	0560
	X2=XTBL(II)	SI	0570
	Y1=YTBL(II-1)	SI	0580
	Y2=YTBL(II)	SI	0590
C		SI	0600
C	CHECK IF (UPPER OR LOWER) EXTRAPOLATION WAS FOUND TO BE NECESSARY	SI	0610
	IF(IND-1)259,257,258	SI	0620
C		SI	0630
C	LOWER EXTRAPOLATION IS NEEDED - CHECK INDLE FOR TYPE TO BE USED	SI	0640
257	GO TO (270,259,370,359),INDLE	SI	0650
C		SI	0660
C	UPPER EXTRAPOLATION IS NEEDED - CHECK INDUE FOR TYPE TO BE USED	SI	0670
258	GO TO (270,259,370,359),INDUE	SI	0680
C		SI	0690
C	ERROR PRINTOUT	SI	0700
359	WRITE(6,1000)		
C		SI	0720
259	X3=XTBL(II+1)	SI	0730
	Y3=YTBL(II+1)	SI	0740
C		SI	0750
C	PARABOLIC INTERPOLATION OR EXTRAPOLATION	SI	0760
260	SI=Y1+(1.0+(X2-X)/(X3-X1))*(Y2-Y1)*(X-X1)/(X2-X1)-(X2-X)/(X3-X1)*	SI	0770
	1(X-X1)/(X3-X2)*(Y3-Y2)	SI	0780
	GO TO 300	SI	0790
C		SI	0800
C	ERROR PRINTOUT	SI	0810
370	WRITE(6,1001)		

```

      LINEAR EXTRAPOLATION
270  SI=Y1+(Y2-Y1)*(X-X1)/(X2-X1)
1000 FORMAT(33H TABLE EXTRAPOLATED PARABOLICALLY )
1001 FORMAT(28H TABLE EXTRAPOLATED LINEARLY )
300  RETURN
      END

```

SI 0830
SI 0840
SI 0850

SI 0860
SI 0870

\$DATA

TEST	PROBLEM - 60	DEGREE	PROBLEM WITH	SIMULATED	VANE -														
300	4	3	3	8	4	-1	0	0	0	10	9	5	-1	0	6	0	-1	0	0
1.2		0.3		1.0	E-6	15.0						0.2							
20.0		20.0		20.0															
15.0		15.0		15.0		5.0		5.0		5.0		5.0		5.0		5.0			
1	1	1																	
1	1	1																	
1	1	1																	
2	2	2																	
2	3	2																	
2	2	3																	
2	3	2																	
2	2	2																	
0.23380355E-02	0.79013388E-03	0.26494724E 01	0.75356217E 00	0.26827937E 01	100														
0.	0.	0.	0.	-0.	101														
0.22263291E-02	0.11891828E-02	0.23334724E 01	0.24643213E-00	0.12395053E 01	102														
0.18970601E-01	0.	0.	0.	-0.	103														
0.14952191E-01	0.85840424E-02	0.23196717E 01	0.	0.10389321E 01	104														
0.70583003E-06	0.75373828E-02	0.	0.	-0.	105														
0.47350618E-01	0.38559481E-01	0.23157977E 01	0.	0.97209518E 00	106														
0.46313683E-10	0.12080043E-07	0.97972932E-03	0.30161881E-00	-0.	107														
0.16242607E-02	0.	0.	0.75356217E 00	0.28728710E 01	600														
0.	0.	0.	0.	-0.	601														
0.69637023E-03	0.	0.	0.24643213E-00	0.13448635E 01	602														
0.16804069E-01	0.	0.	0.	-0.	603														
0.66380219E-03	0.	0.	0.	0.11898725E 01	604														
0.20469570E-06	0.83589414E-02	0.	0.	-0.	605														
0.13158737E-02	0.	-0.	0.	0.12108865E 01	606														
0.	0.	0.36159710E-02	0.27495432E-00	-0.	607														
-0.58982494E-02	0.	0.	0.75356217E 00	0.15562864E 01	800														
0.	0.	0.	0.	-0.	801														
0.15435895E-02	0.	0.	0.24643213E-00	0.60818470E 00	802														
0.48674025E-01	0.	0.	0.	-0.	803														
0.41028060E-01	0.	0.	0.	0.50542513E 00	804														
0.58024564E-09	0.29379062E-01	0.	0.	-0.	805														
0.50705355E 00	0.	-0.	0.	0.31632586E-00	806														
0.	0.	0.14357819E-01	0.64679459E 00	-0.	807														

\$IBSYS
\$SCRAM

POTLUK PROGRAM


```

CPOTLUK
C A PROGRAM TO GENERATE A NEW VARI-QUIR FLUX GUESS FROM AN OLD ONE.
C
C READ THE PROBLEM SPECS.
C T(1) = TITLE INFORMATION OF PROBLEM ID.
      EQUIVALENCE (DOX,XU), (DOY,YU), (DXN,XN), (DYN,YN)
      DIMENSION T(14), LABEL(2), ICHDX(99), ENTR(99)
      DIMENSION PHIO(1600), PHIN(4800), XO(800), YO(2400), XN(2500), YN(2500)
      DIMENSION DOX(800), DOY(2400), DXN(2500), DYN(2500)
      DATA ITPCK/6HFLUA7J/,A1/6H OLD X/,A2/6H OLD Y/,A3/6H NEW X/,A4/6H NEW Y/
      INEW Y/
      REWIN 7
      REWIN 10
      ITAPT = 0
      10 READ(5,1001) (T(I),I=1,14)
      WRITE(6,1002) (T(I),I=1,14)
10001 FORMAT(13A6,A2)
10002 FORMAT(1H1,13A6,A2)
C
C NX = NO. OF OLD X-REGIONS(HORIZONTAL AXIS).
C NY = NO. OF OLD Y-REGIONS(VERTICAL AXIS).
C NX = NO. OF NEW X-REGIONS.
C NY = NO. OF NEW Y-REGIONS.
      READ(5,10003) NX, NY, NN, NN
10003 FORMAT(18I4)
C
C DX = OLD X-REGION INCREMENTS IN CM.
C DY = OLD Y-REGION INCREMENTS IN CM.
C DX = NEW X-REGION INCREMENTS IN CM.
C DY = NEW Y-REGION INCREMENTS IN CM.
      READ(5,10004) (DX(I),I=1,NX)
      READ(5,10004) (DY(I),I=1,NY)
      READ(5,10004) (DX(I),I=1,NN)
      READ(5,10004) (DY(I),I=1,NN)
10004 FORMAT(3F10.4)
C CONVERT THESE INCREMENTS TO ACTUAL COORDINATES.
      CALL QUIRT(NX,DX)
      CALL QUIRT(NY,DY)
      CALL QUIRT(NN,DX)
      CALL QUIRT(NN,DY)
C
C CALCULATE THE NUMBER OF FLUXES/ADJOINTS TO BE ENTERED

```

```

POTL 2
POTL 3
POTL 4
POTL 5
POTL 6
POTL 7
POTL 8
POTL 9
POTL 10
POTL 11
POTL 12
POTL 13 1
POTL 14 2
POTL 15
POTL 16 3
POTL 17 10
POTL 18
POTL 19
POTL 20
POTL 21
POTL 22
POTL 23
POTL 24
POTL 25 17
POTL 26
POTL 27
POTL 28
POTL 29
POTL 30
POTL 31
POTL 32 22
POTL 33 29
POTL 34 36
POTL 35 43
POTL 36
POTL 37
POTL 38 50
POTL 39 52
POTL 40 54
POTL 41
POTL 42
POTL 43 56

```

02/08/66

PUTL - FEN SOURCE STATEMENT - IFN(S) -

NP=(2*NOX+1)*(2*NOY+1)	POTL	44	
C	POTL	45	
C WRITE THE INPUT	POTL	46	
I=1	POTL	47	
20 CONTINUE	POTL	48	
GO TO (50,40,50,60,80), I	POTL	49	
30 A=A1	POTL	50	
WRITE(6,10005) NOX, A, (XO(J), J=1,NOX)	POTL	51	62
GO TO 70	POTL	52	
40 A=A2	POTL	53	
WRITE(6,10005) NOY, A, (YO(J), J=1,NOY)	POTL	54	71
GO TO 70	POTL	55	
50 A=A3	POTL	56	
WRITE(6,10005) NNX, A, (XN(J), J=1,NNX)	POTL	57	80
GO TO 70	POTL	58	
60 A=A4	POTL	59	
WRITE(6,10005) NNY, A, (YN(J), J=1,NNY)	POTL	60	89
70 I = I+1	POTL	61	
GO TO 20	POTL	62	
80 CONTINUE	POTL	63	
10005 FORMAT(5H0THE I2, A6, 16H COORDINATES ARE //(12F8.3) /)	POTL	64	
WRITE(6,10006) NP	POTL	65	98
10006 FORMAT(17H0NO. OF POINTS = I4 /)	POTL	66	
C	POTL	67	
C EXPAND THE NEW COORDINATES INTO ACTUAL VARI-QUIR POINTS.	POTL	68	
CALL XPAND(NNX,XN)	POTL	69	99
CALL XPAND(NNY,YN)	POTL	70	101
IFIRST = 1	POTL	71	
IFOLST = 1	POTL	72	
IFUOST = 1	POTL	73	
90 NP=(2*NOX+1)*(2*NOY+1)	POTL	74	
C	POTL	75	
C READ IN THE FLUXES OR ADJOINTS FOR THIS PROBLEM.	POTL	76	
C	POTL	77	
C LABEL =18 CHARACTERS IDENTIFYING THESE FLUXES/ADJOINTS.	POTL	78	
C ITAPE = 1.OR 0, READ FLUXES OR ADJOINTS FROM TAPE OR CARDS.	POTL	79	
C IDUT = 0, OR 1 DETERMINES IF FLUXES/ADJOINTS WILL BE PUNCHED OR WRITTEN	POTL	80	
C TAPE RESPECTIVELY	POTL	81	
READ(5, 10007) (LABEL(I),I=1,3), ITAPE, IDUT, ICH	POTL	82	106
10007 FORMAT(3A6, 3I2)	POTL	83	
IF(ITAPE)100,140, 100	POTL	84	
100 IF(IFIRST)110,130,110	POTL	85	
110 READ(5) ID	POTL	86	119

110000	IF (ITPCK-10) 300,120,30	POTL 87	
110001	CONTINUE	POTL 88	
110002	IF I ST = 1	POTL 89	
110003	READ(5) (PHI(I),I=1,NP)	POTL 90	123
110004	GO TO 150	POTL 91	
110005	READ(5,10003) (PHI(I),I=1,NP)	POTL 92	131
110006	GO TO 150	POTL 93	
110007	CONTINUE	POTL 94	
110008	C	POTL 95	
110009	C WRITE OUT THE FLUXES	POTL 96	
110010	WRITE(6,10009) (LABLE(I),I=1,3)	POTL 97	138
110011	FORMAT(1H0,3A0)	POTL 98	
110012	WRITE(6,10010) (PHI(I),I=1,NP)	POTL 99	145
110013	FORMAT(7F15.6)	POTL 100	
110014	C	POTL 101	
110015	C DO THE INTERPOLATION	POTL 102	
110016	CALL XACON(XCY,NCX,YC,XC,ANY,NX,YN,XN,PHI,PHIN)	POTL 103	152
110017	IF (IC4.EQ.0) GO TO 170	POTL 104	
110018	READ(5,10003) (IC4X(I),I=1,IC4)	POTL 105	156
110019	READ(5,10010) (ENTR(I),I=1,IC4)	POTL 106	163
110020	DO 160 I=1,IC4	POTL 107	
110021	J = IC4X(I)	POTL 108	
110022	PHIN(J) = ENTR(I)	POTL 109	
110023	160 CONTINUE	POTL 110	
110024	170 CONTINUE	POTL 111	
110025	C	POTL 112	
110026	C	POTL 113	
110027	C	POTL 114	
110028	NP = (2*NX+1)*(2*NY+1)	POTL 115	
110029	WRITE(6,10011) NP	POTL 116	181
110030	FORMAT(25H0THE NUMBER OF NEW POINTS = I4 /)	POTL 117	
110031	WRITE(6,10012)	POTL 118	
110032	C	POTL 119	
110033	C WRITE THE NEW FLUX/ADJUNCTS	POTL 120	182
110034	FORMAT(25H0NEW FLUX/ADJUNCT ARRAY /)	POTL 121	
110035	WRITE(6,10010) (PHIN(I),I=1,NP)	POTL 122	
110036	C	POTL 123	183
110037	IF (IFDIT) 140,210,140	POTL 124	
110038	IF (IFDIST) 140,200,140	POTL 125	
110039	WRITE(10) ITPCK	POTL 126	193
110040	IFDIST = 0	POTL 127	
110041	IIAPT = 1	POTL 128	
110042	200 WRITE(10) (PHIN(I),I=1,NP)	POTL 129	195

GO TO 240	POTL 130	
210 CONTINUE	POTL 131	
IF(IFURST)220,230,220	POTL 132	
220 WRITE(7,10013)(LABEL(I),I=1,3)	POTL 133	205
10013 FORMAT(3A6)	POTL 134	
IFURST = 0	POTL 135	
230 WRITE(7,10008)(PHIN(I),I=1,NP)	POTL 136	212
240 CONTINUE	POTL 137	
C	POTL 138	
C DETERMINE IF THE NEXT PROBLEM WILL HAVE NEW SPECS. OR JUST A NEW FLUX/	POTL 139	
C ARRAY, OR IF THIS IS THE LAST PROBLEM, IGO = 2,1, OR 0 RESPECTIVELY.	POTL 140	
READ(5,10003) IGO	POTL 141	219
IF(IGO-1)250,90,10	POTL 142	
250 IF(ITAPT)260,270,260	POTL 143	
260 END FILE 10	POTL 144	224
REWIND 10	POTL 145	225
270 IF(ITAPE) 280,290,280	POTL 146	
280 REWIND 0	POTL 147	228
290 CONTINUE	POTL 148	
GO TO 310	POTL 149	
300 WRITE(6,10014)	POTL 150	231
10014 FORMAT(44H0THE WRONG FLUX TAPE HAS BEEN MOUNTED ON A-5 //)	POTL 151	
310 CONTINUE	POTL 152	
STOP	POTL 153	
END	POTL 154	

02/03/66

\$IBFTC AARG LIST,DECK,M94/2,SDO,XR7

AARG 1

72

```

      SUBROUTINE AARGH(NCX,NOY,XO,YO,NNX,NNY,XN,YN      ,PHIO,PHIN)
C SUBPROGRAM TO GENERATE A NEW FLUX/ADJOINT GUESS FOR VARI-QUIR
C
C EXPLANATION OF THE ARGUMENTS
C   (A) NOX = NO. OF OLD REGIONS REL. TO X - AXIS.
C   (B) NOY = NO. OF OLD REGIONS REL. TO Y - AXIS.
C   (C) NNX = NO. OF NEW REGIONS REL. TO X - AXIS.
C   (D) NNY = NO. OF NEW REGIONS REL. TO Y - AXIS.
C   (E) XO(I), I=1,NOX , OLD X-COORDINATES
C   (F) YO(I), I=1,NOY , OLD Y-COORDINATES
C   (G) XN(I), I=1,NXP, NEW X-POINTS
C   (H) YN(I), I=1,NYP , NEW Y-POINTS
C   (I) PHIO = OLD FLUX ARRAY
C   (J) PHIN = NEW FLUX ARRAY, IN WHICH THE ANSWERS ARE RETURNED.
C
      DIMENSION XO(1),YO(1),XN(1),YN(1),PHIO(1),PHIN(1)
      DIMENSION XTABLE(3), YTABLE(3), ZTABLE(9), NINE(3,3)
C
C CALCULATE THE TOTAL NO. OF NEW POINTS
      NXP = 2*NNX+1
      NYP = 2*NNY+1
      NP = NXP*NYP
C
C INITIALIZE THE POINT COUNTER AND START STEPPING THROUGH NEW REGIONS.
      N=0
      DO 90 I=1,NXP
      I=I
      DO 90 J=1,NYP
      J=J
      N=N+1
C
C ESTABLISH WHICH OLD REGION THE POINT (XN(I), YN(J)) FALLS IN,
C   STORE X INDEX IN K,
C   STORE Y INDEX IN L.
      Q = XN(I)
      K = INAR(Q,XO,NOX)
      Q = YN(J)
      L = INAR(Q,YO,NOY)
C
C ESTABLISH THE INDICES OF THE NINE POINTS SURROUNDING (XN(I), YN(J))
      CALL NINEP(K,L,NOY,NINE)
C

```

AARG	2	
AARG	3	
AARG	4	
AARG	5	
AARG	6	
AARG	7	
AARG	8	
AARG	9	
AARG	10	
AARG	11	
AARG	12	
AARG	13	
AARG	14	
AARG	15	
AARG	16	
AARG	17	
AARG	18	
AARG	19	
AARG	20	
AARG	21	
AARG	22	
AARG	23	
AARG	24	
AARG	25	
AARG	26	
AARG	27	
AARG	28	
AARG	29	
AARG	30	
AARG	31	
AARG	32	
AARG	33	
AARG	34	
AARG	35	
AARG	36	
AARG	37	15
AARG	38	
AARG	39	
AARG	40	
AARG	41	18
AARG	42	
AARG	43	

02/08/66

AARG - FEN SOURCE STATEMENT - IFN(S) -

C ESTABLISH THE X COORDINATES OF THESE 9 POINTS	AARG	44	20
IF(K-1)10,10,20	AARG	45	
10 XTABLE(1) = 0.	AARG	46	
XTABLE(2) = .5*X0(K)	AARG	47	
XTABLE(3) = X0(K)	AARG	48	
GO TO 30	AARG	49	
20 XTABLE(1) = X0(K-1)	AARG	50	
XTABLE(2) = X0(K-1)+.5*(X0(K)-X0(K-1))	AARG	51	
XTABLE(3) = X0(K)	AARG	52	
30 CONTINUE	AARG	53	
C	AARG	54	
C ESTABLISH THE Y COORDINATES	AARG	55	
IF(L-1)40,40,50	AARG	56	
40 YTABLE(1) = 0.	AARG	57	
YTABLE(2) = .5*Y0(L)	AARG	58	
YTABLE(3) = Y0(L)	AARG	59	
GO TO 60	AARG	60	
50 YTABLE(1) = Y0(L-1)	AARG	61	
YTABLE(2) = Y0(L-1)+.5*(Y0(L)-Y0(L-1))	AARG	62	
YTABLE(3) = Y0(L)	AARG	63	
60 CONTINUE	AARG	64	
C ESTABLISH THE FLUX/ADJACENT TABLE	AARG	65	
ITABL = 0	AARG	66	
DO 70 JJ = 1,3	AARG	67	
DO 70 II = 1,3	AARG	68	
ITABL = ITABL + 1	AARG	69	
IPT = NIN-(II,JJ)	AARG	70	
ZTABLE(ITABL) = PHIO(IPT)	AARG	71	
70 CONTINUE	AARG	72	
30 CONTINUE	AARG	73	
C	AARG	74	
C	AARG	75	
C SET THE NEW FLUX	AARG	76	
X = X0(I)	AARG	77	
Y = Y0(J)	AARG	78	
PHIN(V) = DI(XTABLE,YTABLE,ZTABLE,X,Y,3,3,INDX,INDY,1,1,1)	AARG	79	
C	AARG	80	
C END PROGRAM, GO BACK, RESET ALL INDICES.	AARG	81	
C	AARG	82	62
20 CONTINUE	AARG	83	
C	AARG	84	
THE	AARG	85	
THE	AARG	86	

02/08/66

\$IBFTC XPAN LIST,DECK,M94/2,SDO,XR7

XPAN 1

02/08/66

XPAN - FEN SOURCE STATEMENT - IFN(S) -

```

      SUBROUTINE XPAND(NC,C)
C SUBPROGRAM TO FILL OUT THE COORDINATES.
C
      DIMENSION C(1), P(100)
C
      P(1) = 0.
      P(2) = .5*C(1)
      P(3) = C(1)
C
      DO 10 I = 2, NC
      I1 = 2*I+1
      I2 = 2*I
      P(I1) = C(I)
      P(I2) = C(I-1) + .5*(C(I)-C(I-1))
10 CONTINUE
C
      DO 20 I = 1, I1
      C(I) = P(I)
20 CONTINUE
      RETURN
      END

```

```

XPAN  2
XPAN  3
XPAN  4
XPAN  5
XPAN  6
XPAN  7
XPAN  8
XPAN  9
XPAN 10
XPAN 11
XPAN 12
XPAN 13
XPAN 14
XPAN 15
XPAN 16
XPAN 17
XPAN 18
XPAN 19
XPAN 20
XPAN 21
XPAN 22

```

02/08/66

\$IBFTC NINE LIST,DECK,M94/2,SDD,XR7

NINE 1

NIJ - FFG SOURCE STATEMENT - IFN(5) -

```

SUBROUTINE NINEP( IX, JY, NY, I )
C*****NINEP
C
C GEN-ATES THE ABSOLUTE (VARI-WOIP) POINT INDICES OF A REGION GIVEN -
C IX, REGION NO. REL. TO X-AXIS
C JY, REGION NO. REL. TO Y-AXIS
C NY, TOTAL NO. OF REGIONS REL. TO Y-AXIS.
C
C THE BASEF APPEARS IN THE I ARRAY -
C
C
C      *-----*-----*
C      I(1,3)      I(2,3)      I(3,3)
C      1           1           1
C      1           1           1
C      *           *           *
C      I(1,2)      I(2,2)      I(3,2)
C      1           1           1
C      1           1           1
C      *-----*-----*
C      I(1,1)      I(2,1)      I(3,1)
C
C      DIMENSION I(3,1)
C      MINUS = 2
C      NY2P1 = 2*NY + 1
C      JY2M1 = 2*JY - 1
C      JJ = 1
C      DO 10 I1 = 1,3
C      I(11,JJ) = (2*IX-MINUS)*NY2P1 + JY2M1 + JJ - 1
C      MINUS = MINUS - 1
10 CONTINUE
C      DO 30 I1 = 1,3
C      DO 20 JJ = 2,3
C      I(11,JJ) = I(11,JJ-1) + 1
20 CONTINUE
30 CONTINUE
RETURN
END

```

NINE	2
NINE	3
NINE	4
NINE	5
NINE	6
NINE	7
NINE	8
NINE	9
NINE	10
NINE	11
NINE	12
NINE	13
NINE	14
NINE	15
NINE	16
NINE	17
NINE	18
NINE	19
NINE	20
NINE	21
NINE	22
NINE	23
NINE	24
NINE	25
NINE	26
NINE	27
NINE	28
NINE	29
NINE	30
NINE	31
NINE	32
NINE	33
NINE	34
NINE	35
NINE	36
NINE	37
NINE	38

02/08/66

\$IBFTC QUIR LIST,DECK,M94/2,SDD,XR7

QUIR 1

02/08/66

QUIR - FOR SOURCE STATEMENT - IFN(S) -

```
      SUBROUTINE QUIRT(N,Q)
      DIMENSION Q(1)
C
      DO 10 I = 2,N
C 10 Q(I) = Q(I-1)+ Q(I)
C
      RETURN
      END
```

```
QUIR  2
QUIR  3
QUIR  4
QUIR  5
QUIR  6
QUIR  7
QUIR  8
QUIR  9
```

\$IBFTC INAR LIST,DECK,M94/2,SDD,XR7

INAR 1

02/08/66

INAR - [FN SOURCE STATEMENT - IFN(S) -

```

      FUNCTION INAR(C,C7,NCO)
C SUBPROGRAM TO ESTABLISH REGION NO. REL. TO AN AXIS
C
      DIMENSION CO(1)
      I=1
      DO 40 I=1,NCO
      ISAV = I
      IF(C-CO(I))10,40,40
10  IF(I-1)30,30,20
20  IF(CO(I-1)-C)50,50,60
30  IF(C)70,50,50
40  CONTINUE
50  CONTINUE
      INAR = ISAV
      GO TO 90
60  WRITE(6,1001)
1001 FORMAT(11H0ERROR 1300 )
      GO TO 80
70  WRITE(6,1002)
1002 FORMAT(11H0ERROR 1313)
80  CALL EXIT
90  RETURN
      END

```

INAR	2	
INAR	3	
INAR	4	
INAR	5	
INAR	6	
INAR	7	
INAR	8	
INAR	9	
INAR	10	
INAR	11	
INAR	12	
INAR	13	
INAR	14	
INAR	15	
INAR	16	
INAR	17	21
INAR	18	
INAR	19	
INAR	20	23
INAR	21	
INAR	22	25
INAR	23	
INAR	24	