

731500

**BUDDHA, COMPARE,
AND OTHER EMP
ENVIRONMENT DATA REDUCTION CODES**

JUNE 1971

**U. S. ARMY HARRY DIAMOND LABORATORIES
WASHINGTON, D. C.**

PREPARED BY:

**David L. Jones
Deborah H. Stump**

**BRADDOCK, DUNN AND McDONALD, INC.
WASHINGTON, D.C. OFFICE**

UNDER CONTRACT NUMBER DAAK02-70-C-0258

**This project has been funded by the
Defense Atomic Support Agency
under Nuclear Weapons Effects
Research Subtask Numbers
EA-091 and EA-094**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Braddock, Dunn and McDonald, Inc. 8027 Leesburg Pike McLean, Virginia 22101		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE BUDDHA, COMPARE, AND OTHER EMP ENVIRONMENT DATA REDUCTION CODES			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) FINAL REPORT Jan 1971 - June 1971			
5. AUTHOR(S) (First name, middle initial, last name) David L. Jones and Deborah H. Stump			
6. REPORT DATE June 1971		7a. TOTAL NO. OF PAGES 203	7b. NO. OF REFS 2
8a. CONTRACT OR GRANT NO. DAAK02-70-C-0258		9a. ORIGINATOR'S REPORT NUMBER(S) BDM/W-TR71-66-0258	
b. PROJECT NO.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
c.			
d.			
10. DISTRIBUTION STATEMENT Distribution of this document is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY U.S. Army Harry Diamond Laboratories Washington, D.C. 20438	
13. ABSTRACT The programs documented in this report are part of a comprehensive package of digital computer codes which calculate the electromagnetic pulse (EMP) environment that is produced by a nuclear explosion. Each new program is described in the form of a user's guide which includes a discussion on how the program operates, instructions on how to execute the program, and sample output generated by the program. For each modified program the discussion tells how and why it was modified.			

DD FORM 1473

REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified

Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Electromagnetic Pulse (EMP)						
Digital Computer Code						
Graphical Display of Output						

Unclassified

Security Classification

SUMMARY

This is the final report for contract DAAK02-70-C-0258 and covers the period from January 1971 through June 1971. It is a summary of all programs written and/or modified under this contract during that period by Braddock, Dunn and McDonald, Inc. for the U.S. Army Mobility Equipment Research and Development Center and subsequently, for U.S. Army Harry Diamond Laboratories. Each new program is described in the form of a user's guide which includes a discussion on how the program operates, instructions on how to execute the program, and sample output generated by the program. For each modified program the discussion tells how and why it was modified.

FOREWORD

The programs documented in this report are part of a comprehensive package of digital computer codes which calculate the electromagnetic pulse (EMP) environment that is produced by a nuclear explosion. This work is a continuation of an extensive technical effort formerly conducted at the U.S. Army Mobility Equipment Research and Development Center (MERDC), Fort Belvoir, Virginia, and presently conducted at the U.S. Army Harry Diamond Laboratories, Washington, D.C. The modification of previously documented computer codes and the writing of new codes has necessitated this documentation.

The programming tasks were performed under contract DAAK02-70-C-0258 by Mr. Jeffrey A. Borbely (project leader), Mr. David L. Jones, and Mrs. Deborah H. Stump of Braddock, Dunn and McDonald, Inc. Technical supervisor was Mr. William T. Wyatt, Jr. formerly of the Physics Division, Electromagnetic Effects Laboratory, MERDC, and presently of Branch 1030, HDL.

TABLE OF CONTENTS

	<u>Page</u>
I. NEW CODES AND MAJOR MODIFICATIONS	
A. COMPARE	1
1. Introduction	1
2. Plot Definition and Control	
3. Subroutines LINEUP and LYNEUP	3
4. Subroutines LTCONE and LTKONE	4
5. Subroutines TIMAXS and TYMAXS	4
6. Subroutine COEF	5
7. Subroutine SCALE	6
8. Subroutine LINES	6
9. Subroutines GREEK, SEPAP8, and ISHIFT	6
10. Description of Input	7
a. Type 1	7
b. Type 2	9
c. Type 3	12
d. Type 4	14
e. Type 5	17
11. Sample Output	21
12. Program Listing	41
B. BUDDHA	77
1. Introduction	77
2. Input Revisions	77
3. Data Revisions	81
4. Plot Additions	81
5. Subroutine LINES	83
6. Subroutine LOGPLT	83
7. CALCOMP Subroutines	84
8. Sample Output	85
9. Program Listing	89
C. GREEK	115
1. Introduction	115
2. Calling Procedure	115
3. Data Control	116
4. Subroutine SEPAR8	116
5. Function Subroutine ISHIFT	116
6. Special Considerations	117
7. Sample Output	118
8. Program Listing	120

TABLE OF CONTENTS (continued)

	<u>Page</u>
II. MINOR MODIFICATIONS	
A. ELECTRA	125
1. Description of Modifications	125
2. Program Listing	127
B. ORESTES	153
1. Description of Modifications	153
2. Program Listing	154
C. REDACT	181
1. Description of Modifications	181
2. Program Listing	182
D. REDACTO	191
1. Description of Modifications	191
2. Program Listing	192

I. NEW CODES AND MAJOR MODIFICATIONS

A. COMPARE

1. Introduction

Program COMPARE is a plot program which draws from one to ten parallel curves on the same plotting area to facilitate data comparison. Given a set of EMP field values which are a function of three independent parameters (range, time, and theta), the user may elect to hold any one of the three parameters constant, have each of the ten curves represent a different value of the second parameter, and plot the EMP field values versus the remaining parameter. Each of the curves may be scaled independently or the data for all the curves may be scaled as one set.

2. Plot Definition and Control

The EMP field data is supplied to program COMPARE by reading the output tapes from program ELECTRA the ground burst EMP code.^{1,2} In order to determine which data is to be read from the tape a series of input data cards are read by program COMPARE. The detailed structure of these cards

1. Jones, D. L. and D. H. Stump, ELECTRA, ORESTES, and Supporting Graphic Display Codes, U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, February 1971, pp. 1-36.

2. Borbely, J.A. and D. L. Jones, ELECTRA, An Electromagnetic Pulse Fortran Program (User's Guide), U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, October 1969.

will be discussed in Section I.A.10. The first data card determines which type of plot is desired. That is, it establishes which parameter will be held constant, which parameter (designated z-axis) will change only from curve to curve, and which parameter (designated x-axis) will be plotted against the field data. This first card will contain one of the following ten-character keywords:

<u>PLOT TYPE</u>	<u>KEY WORD</u>	<u>Z-AXIS</u>	<u>X-AXIS</u>	<u>CONSTANT</u>
1	TIME-THETA	TIME	THETA	RANGE
2	RANGE-THETA	RANGE	THETA	TIME
3	TIME-RANGE	TIME	RANGE	THETA
4	RANGE-TIME	RANGE	TIME	THETA
5	THETA-TIME	THETA	TIME	RANGE

These five key words are used as switches to control the execution of five separate sections of program COMPARE. The THETA-RANGE-TIME combination is not implemented in this version of COMPARE.

After the program enters one of the five sections, the additional data items needed to set up that type of plot are read from cards. These data items include the value of the constant parameter, the values of the z-axis parameters, and the values or limits of the x-axis parameters. In addition controls for scaling and for x-axis spacing are also set by data card. Once these parameters have been set, the EMP data is read from tape, edited and ordered accordingly, and then transferred to the appropriate plotting subroutine (LINEUP, LYNEUP, LTCONE; LTKONE, TIMAXS, or TYMAXS).

3. Subroutines LINEUP and LYNEUP

Subroutine LINEUP is called by program COMPARE to produce plots of types 1 and 2 as defined in Section i.A.2. It calls subroutine SCALE, treating the EMP field data for all curves as one set, and receives one maximum, one minimum, and one delta y scale factor. LINEUP then draws each curve separately, scaling each data set against the maximum, the minimum, and the delta y scale factor. Each curve has its own set of axes appropriately annotated.

If the plot is of type 1, the constant parameter is range and is annotated by the statement ALL CURVES ARE FOR $R = (\text{range})$. Each curve represents a different value of either source time or retarded time and is labeled either $T = (\text{source time})$ or $\tau = (\text{retarded time})$. The x-axis shows the values of theta which may be spaced according to actual value or spaced at equal intervals as a user option.

If the plot is of type 2, the constant parameter is either source time or retarded time and is annotated either by the statement ALL CURVES ARE FOR $T = (\text{source time})$ or by ALL CURVES ARE FOR $\tau = (\text{retarded time})$. Each curve represents a different value of range and is labeled $R = (\text{range})$. The x-axis again shows the values of theta as described above.

Subroutine LYNEUP differs from LINEUP only in the area of data scaling. In subroutine LYNEUP the EMP field data set for each curve is supplied to subroutine SCALE separately. The resulting maximum, minimum, and scale factor for each curve are then reflected in the annotation on its set of axes.

4. Subroutines LTCONE and LTKONE

Subroutine LTCONE is called by program COMPARE to produce plots of type 3 as defined in Section 1.A. 2. In a manner similar to that described for subroutine LINEUP in Section 1.A.3., subroutine LTCONE calls subroutine SCALE, treating the EMP field data for all curves as one set, and then draws the curves separately with individually annotated axes.

For a type 3 plot the constant parameter is theta and is annotated by the statement ALL CURVES ARE FOR $\theta = (\text{theta})$. Each curve represents a different value of source time and is labeled $T = (\text{source time})$. The x-axis shows values of retarded range R' where $R' = R - cxT$, $R = \text{actual range}$, $T = \text{source time}$, and $c = \text{speed of light}$. Since each curve represents a different value of T , different retarded ranges must be calculated for each curve and then each curve must be shifted to align with the annotated x-axis at the bottom of the plot.

Subroutine LTKONE differs from LTCONE only in the area of data scaling. In subroutine LTKONE the EMP field data set for each curve is supplied to subroutine SCALE separately. The resulting maximum, minimum, and scale factor for each curve are then reflected in the annotation on its set of axes.

5. Subroutines TIMAXS and TYMAXS

Subroutines TIMAXS and TYMAXS are called by program COMPARE to produce plots of types 4 and 5 as defined in Section 1.A.2. Subroutine

TIMAXS calls subroutine SCALE, treating the EMP field data for all curves as one set, and receives one maximum, one minimum, and one delta y scaling factor. TIMAXS then draws each curve separately, scaling the data against the maximum, minimum, and scale factor established for the total data set. Each curve has its own set of axes appropriately annotated.

Subroutine TYMAXS supplies subroutine SCALE with the data for each curve separately and then plots each curve according to its own maximum, minimum, and delta y scale factor. Each curve again has its own set of axes appropriately annotated.

If the plot is of type 4, the constant parameter is theta and is annotated by the statement ALL CURVES ARE FOR $\theta = (\text{theta})$. Each curve represents a different value of range and is labeled $R = (\text{range})$. The x-axis shows the values of either source time or retarded time in units of nanoseconds and is labeled either T IN NSEC or τ IN NSEC. The time values may be spaced according to actual value or at equal intervals as a user option.

If the plot is of type 5, the constant parameter is range and is annotated by the statement ALL CURVES ARE FOR $R = (\text{range})$. Each curve represents a different value of theta and is labeled $\theta = (\text{theta})$. The x-axis is the same as described above for type 4 plots.

6. Subroutine COEF

Subroutine COEF helps set up the y-axis labeling for the six plotting subroutines described in Sections 1.A.3., 1.A.4., and 1.A.5.

Given a delta y scale factor, subroutine COEF breaks the factor apart into a coefficient between one and ten and a power of ten. This coefficient is then used to determine the number of intervals into which the y-axis will be divided. The power of ten is used along with the minimum y value to determine the annotation to be written beside each tic mark on the y-axis.

7. Subroutine SCALE

Subroutine SCALE performs data scaling for the six plotting subroutines described in Sections 1.A.3., 1.A.4., and 1.A.5. Given a data array, subroutine SCALE scans the array to find the maximum and minimum values. It then defines the delta y scale factor by the following expression:

$$\Delta y = (\text{maximum } y - \text{minimum } y) / \text{length of plot axis}$$

8. Subroutine LINES

Subroutine LINES is a plot routine designed for drawing data curves. Given two data arrays which are to be plotted against each other, subroutine LINES performs a curve fit to the data in a manner such that the curve can be represented by a series of equally spaced dots, line segments or both.

9. Subroutines GREEK, SEPAR8, and ISHIFT

These three subroutines work together to draw the letters of the Greek alphabet in either upper case or lower case form. They will be described in detail in Section 1.C. of this report. They are called by program COMPARE to supply τ 's and θ 's for labeling the plots.

10. Description of Input

The card input for program COMPARE can best be described by breaking it up into five distinct groups even though several of the groups will have the same card types.

a. Type 1

The input for plots of type 1 as defined in Section 1.A.2. consists of the five card types (1A, 1B, 1C, 1D, and 1E) shown below:

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
1A	1-10	Select type of plot	CHOICE	Alpha-betic	Must equal TIME-THETA
	11-80	Not used			Blank
1B	1-10	Number of curves	NZ	Integer	Determines number of time values to appear on card 1C below. Must be ≤ 10 in present version
	11-80	Not used			Blank
1C	1-10	Time value for 1st curve	Z(1)	Floating	May be source time or retarded time in seconds (See TCOL below)
	11-20	Time value for 2nd curve	Z(2)	Floating	May be source time or retarded time in seconds (See TCOL below)
	21-30	Time value for 3rd curve	Z(3)	Floating	May be source time or retarded time in seconds (See TCOL below)
	31-40	Time value for 4th curve	Z(4)	Floating	May be source time or retarded time in seconds (See TCOL below)
	41-50	Time value for 5th curve	Z(5)	Floating	May be source time or retarded time in seconds (See TCOL below)

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
IC cont.	51-60	Time value for 6th curve	Z(6)	Floating	May be source time or retarded time in seconds (See TCOL below)
	61-70	Time value for 7th curve	Z(7)	Floating	May be source time or retarded time in seconds (See TCOL below)
	71-80	Time value for 8th curve	Z(8)	Floating	May be source time or retarded time in seconds (See TCOL below)
ID	1-10	Time increment	DT	Floating	Used as tolerance factor in matching time values
	11-20	Range increment	DR	Floating	Used as tolerance factor in matching range values
	21-30	Constant parameter	CONVAR	Floating	Range value for all curves
	31-40	Time indicator	TCOL	Floating	TCOL=0, retarded time TCOL≠0, source time
	41-50	Length of x-axis	XLEN	Floating	6 inches recommended
	51-60	Length of y-axis	YLEN	Floating	This is the combined total length of all the y-axes. Each individual axis length will equal: $(YLEN/NZ) - .25$ YLEN must be ≤ 10 .
	61-80	Not used			Blank
IE	1-4	EMP field to be plotted	AVAR	Alphabetic	Must equal BPHI, ETHE, ERAD, or SIGT.
	5-10	Not used			Blank
	11-20	Minimum θ value	THETMN	Floating	Minimum theta value for x-axis data.

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
1E cont.	21-30	Maximum θ value	THETMX	Floating	Maximum theta value for x-axis data.
	31-40	x-axis spacing control	INCX	Integer	INCS=1, spaced according to data; INCX=2, equally spaced
	41-50	y-axis scaling control	ISCALE	Integer	ISCALE=0, scale all curves as one set; ISCALE $\neq$ 0, scale each curve separately.
	51-80	Not used			Blank

A standard deck for type 1 plots will consist of one type 1A, one type 1B, one type 1C (a second type 1C is required if NZ is greater than eight), one type 1D, and as many type 1E cards as desired. The last type 1E card must be followed by a blank card which signals the end of that set of type 1 plots.

b. Type 2

The input for plots of type 2 as defined in Section 1.A.2. consists of the five card types (2A, 2B, 2C, 2D, 2E) shown below:

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
2A	1-10	Select type of plot	CHOICE	Alphabetic	Must equal RANG-THETA
	11-80	Not used			Blank
2B	1-10	Number of curves	NZ	Integer	Determines number of range values to appear on card 2C below. Must be ≤ 10 in present version.

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
2B cont.	11-80	Not used			Blank
2C	1-10	Range value for 1st curve	Z(1)	Float- ing	Range in meters
	11-20	Range value for 2nd curve	Z(2)	Float- ing	Range in meters
	21-30	Range value for 3-d curve	Z(3)	Float- ing	Range in meters
	31-40	Range value for 4th curve	Z(4)	Float- ing	Range in meters
	41-50	Range value for 5th curve	Z(5)	Float- ing	Range in meters
	51-60	Range value for 6th curve	Z(6)	Float- ing	Range in meters
	61-70	Range value for 7th curve	Z(7)	Float- ing	Range in meters
	71-80	Range value for 8th curve	Z(8)	Float- ing	Range in meters
2D	1-10	Time increment	DT	Float- ing	Used as tolerance factor in matching time values
	11-20	Range increment	DR	Float- ing	Used as tolerance factor in matching range values
	21-30	Constant parameter	CONVAR	Float- ing	Time value for all curves. May be source time or retarded time
	31-40	Time indicator	TCOL	Float- ing	TCOL=0, retarded time TCOL≠0, source time
	41-50	Length of x-axis	XLEN	Float-	6 inches recommended

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
2D cont.	51-60	Length of y-axis	YLEN	Floating	This is the combined total length of all the y-axes. Each individual axis length will equal: $(YLEN/NZ) - .25$ YLEN must be ≤ 10 .
	61-80	Not used			Blank
2E	1-4	EMP field to be plotted	AVAR	Alphabetic	Must equal BPHI, ETHE, ERAD, or SIGT.
	5-10	Not used			Blank
	11-20	Minimum θ value	THETMN	Floating	Minimum theta value for x-axis data.
	21-30	Maximum θ value	THETMX	Floating	Maximum theta value for x-axis data.
	31-40	x-axis spacing control	INCX	Integer	INCX=1, spaced according to data; INCX=2, equally spaced
	41-50	y-axis scaling control	ISCALE	Integer	ISCALE=0, scale all curves as one set; ISCALE $\neq$ 0, scale each curve separately.
	51-80	Not used			Blank

A standard deck for type 2 plots will consist of one type 2A, one type 2B, one type 2C (a second type 2C is required if NZ is greater than eight), one type 2D, and as many type 2E cards as desired. The last type 2E card must be followed by a blank card which signals the end of that set of type 2 plots.

c. Type 3

The input for plots of type 3 as defined in Section 1.A.2.
consists of the six card types (3A, 3B, 3C, 3D, 3E, and 3F) shown below:

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
3A	1-10	Select type of plot	CHOICE	Alpha-betic	Must equal TIME-RANGE
	11-80	Not used			Blank
3B	1-10	Number of curves	NZ	Integer	Determines number of time values to appear on card 3C below. Must be ≤ 10 in present version.
	11-80	Not used			Blank
3C	1-10	Time value for 1st curve	Z(1)	Float-ing	Source time value in seconds
	11-20	Time value for 2nd curve	Z(2)	Float-ing	Source time value in seconds
	21-30	Time value for 3rd curve	Z(3)	Float-ing	Source time value in seconds
	31-40	Time value for 4th curve	Z(4)	Float-ing	Source time value in seconds
	41-50	Time value for 5th curve	Z(5)	Float-ing	Source time value in seconds
	51-60	Time value for 6th curve	Z(6)	Float-ing	Source time value in seconds
	61-70	Time value for 7th curve	Z(7)	Float-ing	Source time value in seconds
	71-80	Time value for 8th curve	Z(8)	Float-ing	Source time value in seconds

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
3D	1-10	Time increment	DT	Float-ing	Used as tolerance factor in matching time values
	11-20	Range increment	DR	Float-ing	Used as tolerance factor in matching range values
	21-30	Length of x-axis	XLEN	Float-ing	6 inches recommended
	31-40	Length of y-axis	YLEN	Float-ing	This is the combined total length of all the y-axes. Each individual axis length will equal $(YLEN/NZ) - .25 YLEN$ must be ≤ 10 .
	41-80	Not used			Blank
3E	1-10	Number of range values	NRANGE	Integer	Number of range values to be calculated for X-axis.
	11-20	First range value	RMIN	Float-ing	First range value, basis for calculating all other range values
	21-30	Increment between range values	RINC	Float-ing	Each range value will equal the previous value plus RINC
	31-80	Not used			Blank
3F	1-4	EMP field to be plotted	AVAR	Alpha-betic	Must equal BPHI, ETHE, ERAD, or SIGT
	5-10	Not used			Blank
	11-20	Constant parameter	CONVAR	Float-ing	Theta value for all curves
	21-30	X-axis spacing control	INCX	Integer	INCX=1, spaced according to data; INCX=2, equally spaced

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
3F cont.	31-40	y-axis scaling control	ISCALE	Integer	ISCALE=0, scale all curves as one set; ISCALE $\neq$ 0, scale each curve separately.
	41-80	Not used			Blank

A standard deck for type 3 plots will consist of one type 3A, one type 3B, one type 3C (a second type 3C is required if NZ is greater than eight), one type 3D, one type 3E, and as many type 3F cards as desired. The last type 3F card must be followed by a blank card which signals the end of that set of type 3 plots.

d. Type 4

The input for plots of type 4 as defined in Section 1.A.2. consists of the seven card types (4A, 4B, 4C, 4D, 4E, 4F, and 4G) shown below:

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
4A	1-10	Select type of plot	CHOICE	Alphabetic	Must equal RANGE-TIME
	11-80	Not used			Blank
4B	1-10	Number of curves	NZ	Integer	Determines number of range values to appear on card 4C below. Must be ≤ 10 in present version.
	11-80	Not used			Blank

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
4C	1-10	Range value for 1st curve	Z(1)	Float-ing	Range in meters
	11-20	Range value for 2nd curve	Z(2)	Float-ing	Range in meters
	21-30	Range value for 3rd curve	Z(3)	Float-ing	Range in meters
	31-40	Range value for 4th curve	Z(4)	Float-ing	Range in meters
	41-50	Range value for 5th curve	Z(5)	Float-ing	Range in meters
	51-60	Range value for 6th curve	Z(6)	Float-ing	Range in meters
	61-70	Range value for 7th curve	Z(7)	Float-ing	Range in meters
	71-80	Range value for 8th curve	Z(8)	Float-ing	Range in meters
4D	1-10	Time increment	DT	Float-ing	Used as tolerance factor in matching time values
	11-20	Range increment	DR	Float-ing	Used as tolerance factor in matching range values
	21-30	Length of x-axis	XLEN	Float-ing	6 inches recommended
	31-40	Length of y-axis	YLEN	Float-ing	This is the combined total length of all the y-axes. Each individual axis length will equal $(YLEN/NZ) - .25 YLEN$ must be ≤ 10 .
	41-80	Not used			Blank.

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
4E	1-10	Number of different DELT's	NDELTS	Integer	This controls the number of type 4F cards to be read.
	11-20	Minimum time value	TMIN	Floating	First time value, basis for calculating all other time values. May be source or retarded time (See TCOL below).
	21-80	Not used			Blank
4F	1-10	Number of times to use the DELT value found on this card	NDEL	Integer	Each time value for the x-axis is equal to the previous value plus DELT. NDEL tells how many times to add DELT.
	11-20	Increment between time values.	DELT	Floating	See description of NDEL above.
	21-80	Not used			Blank
4G	1-4	EMP field to be plotted	AVAR	Alphabetic	Must equal BPHI, ETHE, ERAD, or SIGT.
	5-10	Not used			Blank
	11-20	Constant parameter	CONVAR	Floating	Theta value for all curves
	21-30	x-axis spacing control	INCX	Integer	INCX=1, spaced according to data; INCS=2, equally spaced.
	31-40	y-axis scaling control	ISCALE	Integer	ISCALE=0, scale all curves as one set; ISCALE $\neq$ 0, scale each curve separately.
	41-50	Time indicator	TCOL	Floating	TCOL=0, retarded time TCOL $\neq$ 0, source time
	51-80	Not used			Blank

A standard deck for type 4 plots will consist of one type 4A, one type 4B, one type 4C (a second type 4C is required if NZ is greater than eight), one type 4D, one type 4E, as many type 4F as specified by the NDELTS parameter, and as many type 4G cards as desired. The last type 4G card must be followed by a blank card which signals the end of that set of type 4 plots.

e. Type 5

The input of plots of type 5 as defined in Section 1.A.2. consists of the seven card types (5A, 5B, 5C, 5D, 5E, 5F, and 5G) shown below:

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
5A	1-10	Select type of plot	CHOICE	Alpha-betic	Must equal THETA-TIME
	11-80	Not used			Blank
5B	1-10	Time increment	DT	Float-ing	Used as tolerance factor in matching time values
	11-20	Range increment	DR	Float-ing	Used as tolerance factor in matching range values
	21-30	Length of x-axis	XLEN	Flota-ing	6 inches recommended
	31-40	Length of y-axis	YLEN	Float-ing	This is the combined total length of all the y-axes. Each individual axis length will equal $(YLEN/NZ) - .25$ YLEN must be ≤ 10 .
	41-50	Constant parameter	CONVAR	Float-ing	Range value for all curves.

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
5B cont.	51-80	Not used			Blank
5C	1-10	Number of different DELT's	NDELTS	Integer	This controls the number of type 5D cards to be read
	11-20	Minimum time value	TMIN	Floating	First time value, basis for calculating all other time values. May be source or retarded time (see TCOL below).
	21-80	Not used			Blank
5D	1-10	Number of times to use the DELT value found on this card	NDELTS	Integer	Each time value for the x-axis is equal to the previous value plus DELT. NDELTS tells how many times to add DELT.
	11-20	Increment between time values.	DELT	Floating	See description of NDELTS above.
	21-80	Not used			Blank
5E	1-10	Number of curves	NZ	Integer	Determines number of theta values to appear on card 5F below. Must be ≤ 10 in present version.
	11-80	Not used			Blank
5F	1-10	Theta value for 1st curve	Z(1)	Floating	Theta value in degrees
	11-20	Theta value for 2nd curve	Z(2)	Floating	Theta value in degrees
	21-30	Theta value for 3rd curve	Z(3)	Floating	Theta value in degrees

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
5F cont.	31-40	Theta value for 4th curve	Z(4)	Floating	Theta value in degrees
	41-50	Theta value for 5th curve	Z(5)	Floating	Theta value in degrees
	51-60	Theta value for 6th curve	Z(6)	Floating	Theta value in degrees
	61-70	Theta value for 7th curve	Z(7)	Floating	Theta value in degrees
	71-80	Theta value for 8th curve	Z(8)	Floating	Theta value in degrees
5G	1-4	EMP field to be plotted	AVAR	Alphabetic	Must equal BPH1, ETHE, ERAD, or SIGT
	5-10	Not used			Blank
	11-20	Time indicator	TCOL	Floating	TCOL=0, retarded time TCOL≠0, source time
	21-30	x-axis spacing control and theta redefinition control	INCX	Integer	INCX=1, spaced according to data; INCX=2, equally spaced. If INCX=3, a new set of 5E-5F cards are read to redefine the theta values.
	31-40	y-axis scaling control	ISCALE	Integer	ISCALE=0, scale all curves as one set; ISCALE≠0, scale each curve separately
	41-80	Not used			Blank

A standard deck for type 5 plots will consist of one type 5A, one type 5B, one type 5C, as many type 5D as specified by the NDELTS parameter, one type 5E, one type 5F (a second type 5F is required if NZ is greater than eight), and as many type 5G cards as desired. If the last type 5G card sets

INCX=3, then the series 5E-5G may be repeated. The last type 5G card must be followed by a blank card which signals the end of that set of type 5 plots.

Plots of types 1 through 5 may be requested in any order desired as long as all the cards needed for a particular type are grouped together. To stop execution of the plot cycle and close the plot file, a card with the word STOP in columns 1-4 must be placed after the last blank card.

11. Sample Output

Description of Sample Output

<u>Figure</u>	<u>Plot Type</u>	<u>Time Values</u>	<u>Y-axis Scaling</u>	<u>X-axis Spacing</u>
1	1	Retarded	As one set	By value
2	1	Retarded	As one set	Equally spaced
3	1	Source	As one set	Equally spaced
4	1	Source	Separately	Equally spaced
5	2	Retarded	As one set	By value
6	2	Retarded	As one set	Equally spaced
7	2	Source	As one set	Equally spaced
8	2	Source	Separately	Equally spaced
9	3	Source	Separately	By actual value*
10	3	Source	Separately	By retarded value
11	4	Retarded	As one set	By value
12	4	Retarded	Separately	By value
13	4	Source	As one set	By value
14	4	Source	Separately	By value
15	5	Retarded	As one set	By value
16	5	Retarded	Separately	By value
17	5	Source	As one set	By value
18	5	Source	Separately	By value

*Note: The actual value data points should have the same relative spacing as the retarded value data points. The following modification to program COMPARE should eliminate the discrepancy: The first statement following statement 1220 should read $X(I) = I * RINC$.

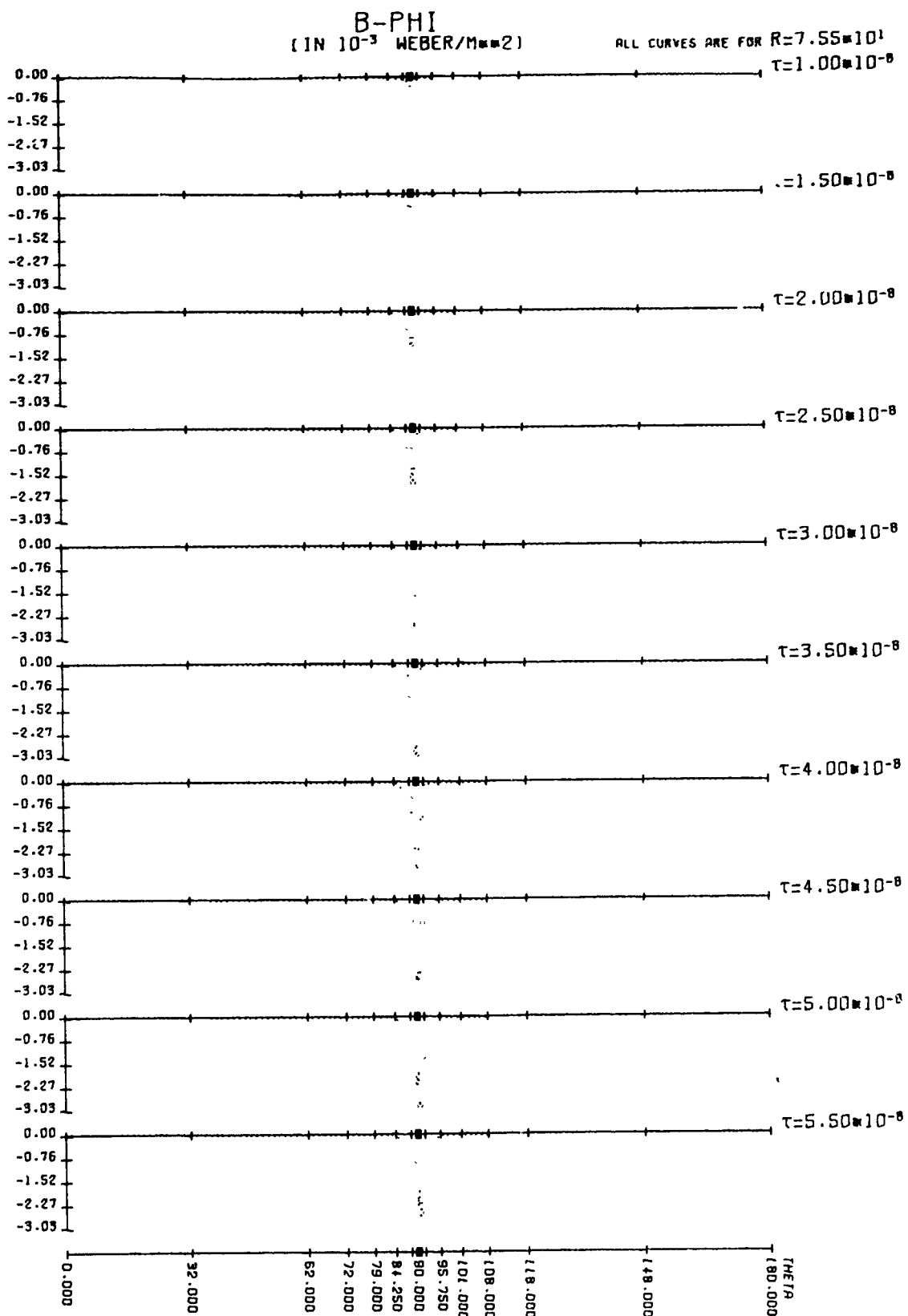


Figure 1.

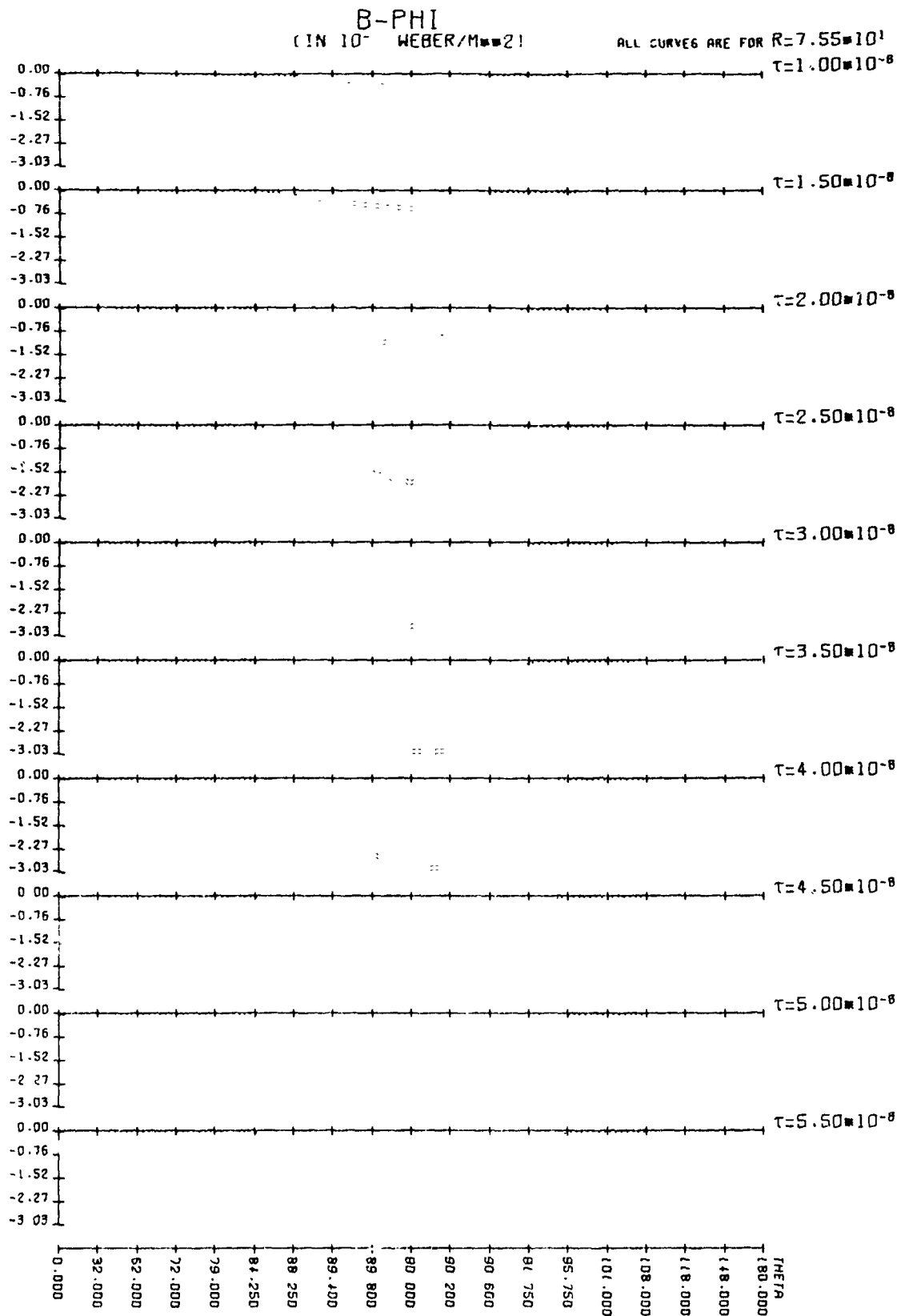


Figure 2.

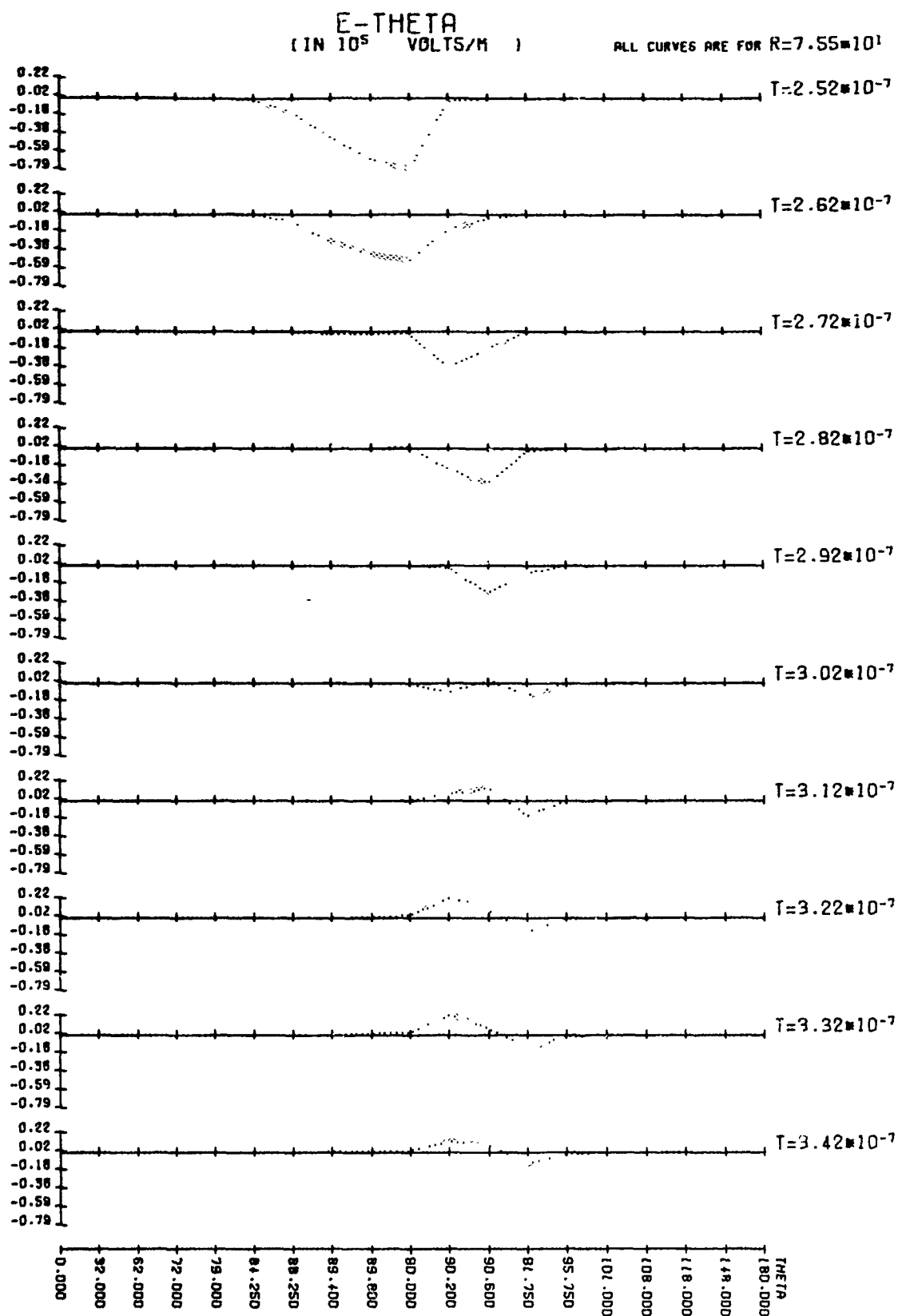


Figure 3.

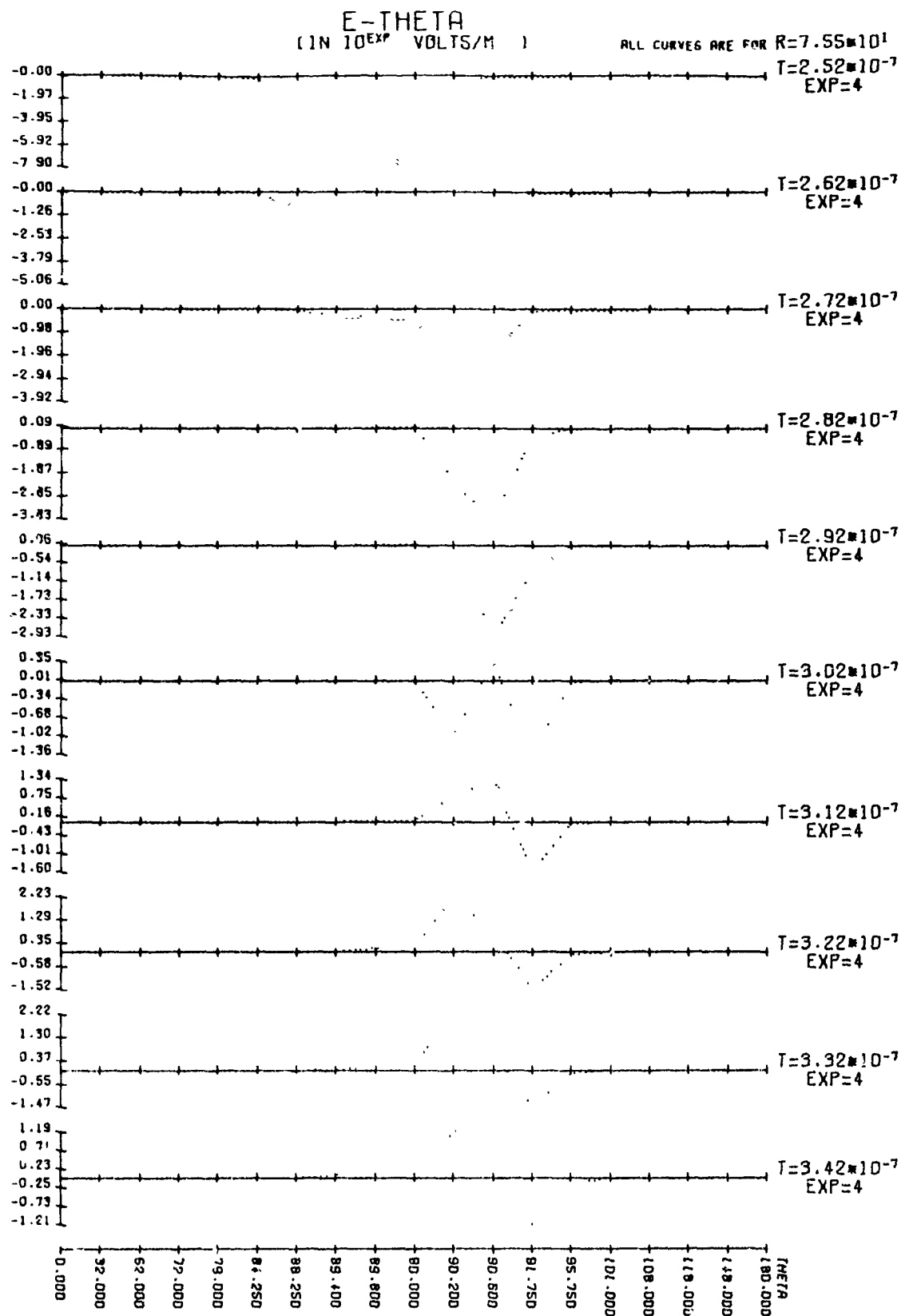


Figure 4.

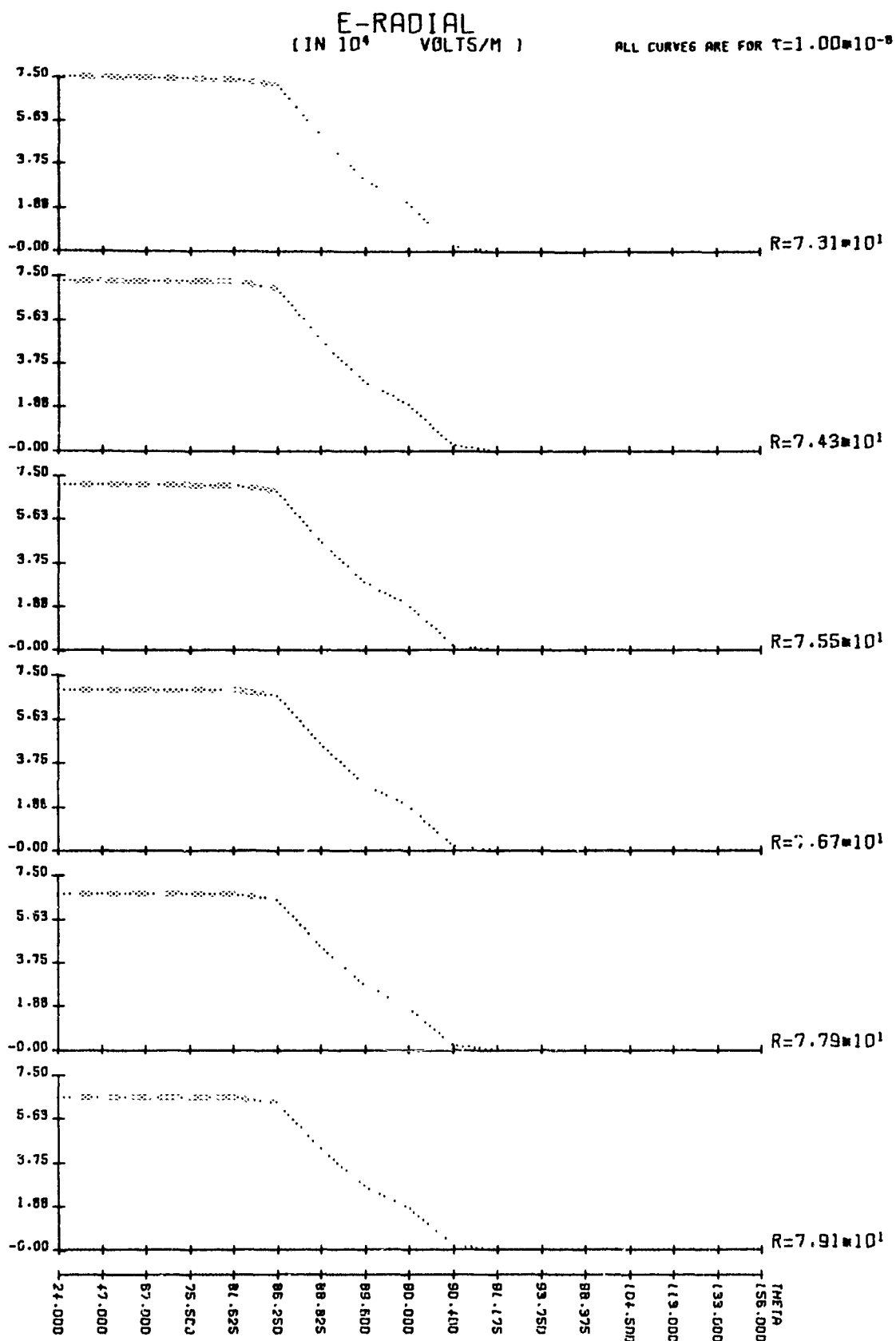


Figure 5.

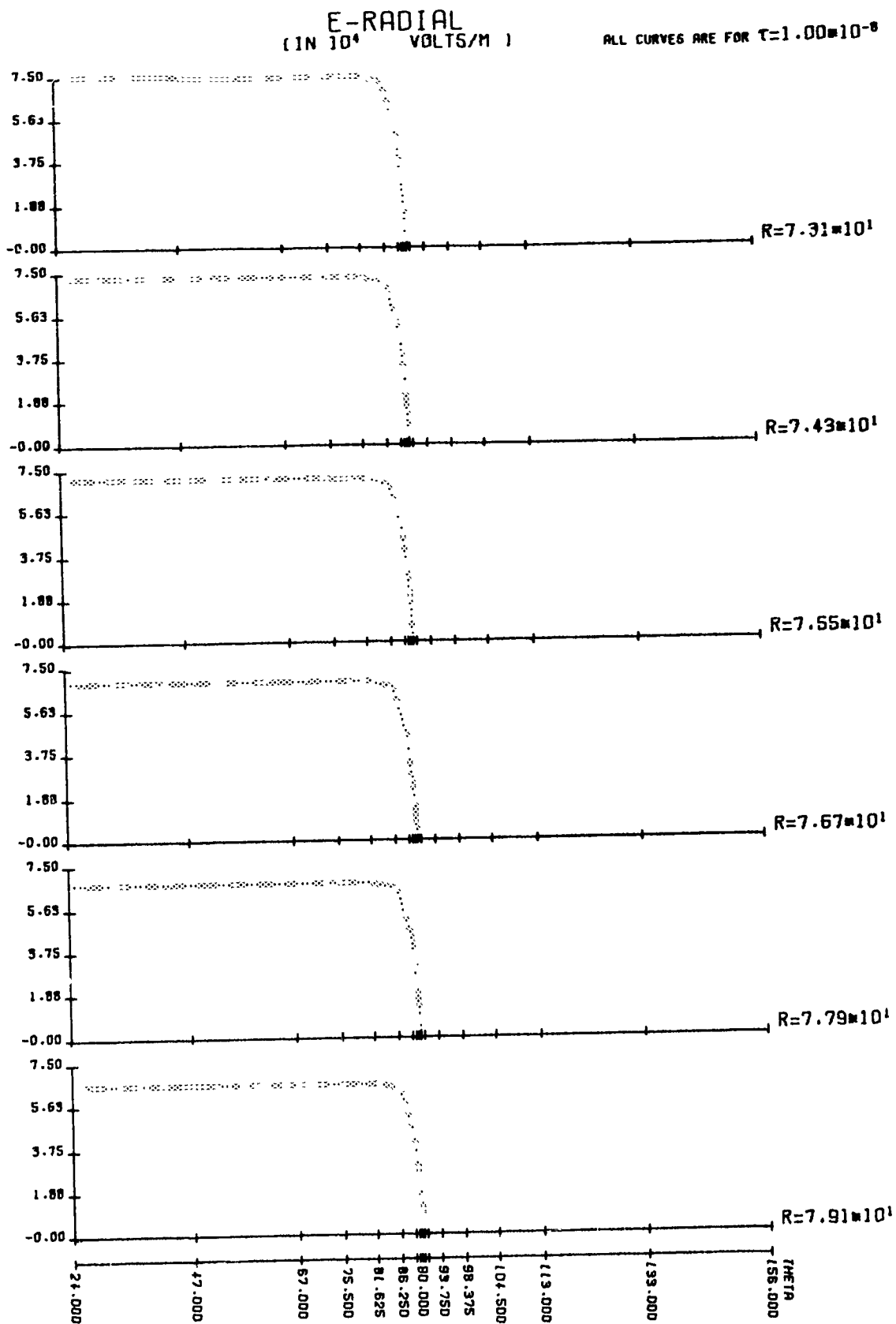


Figure 6.

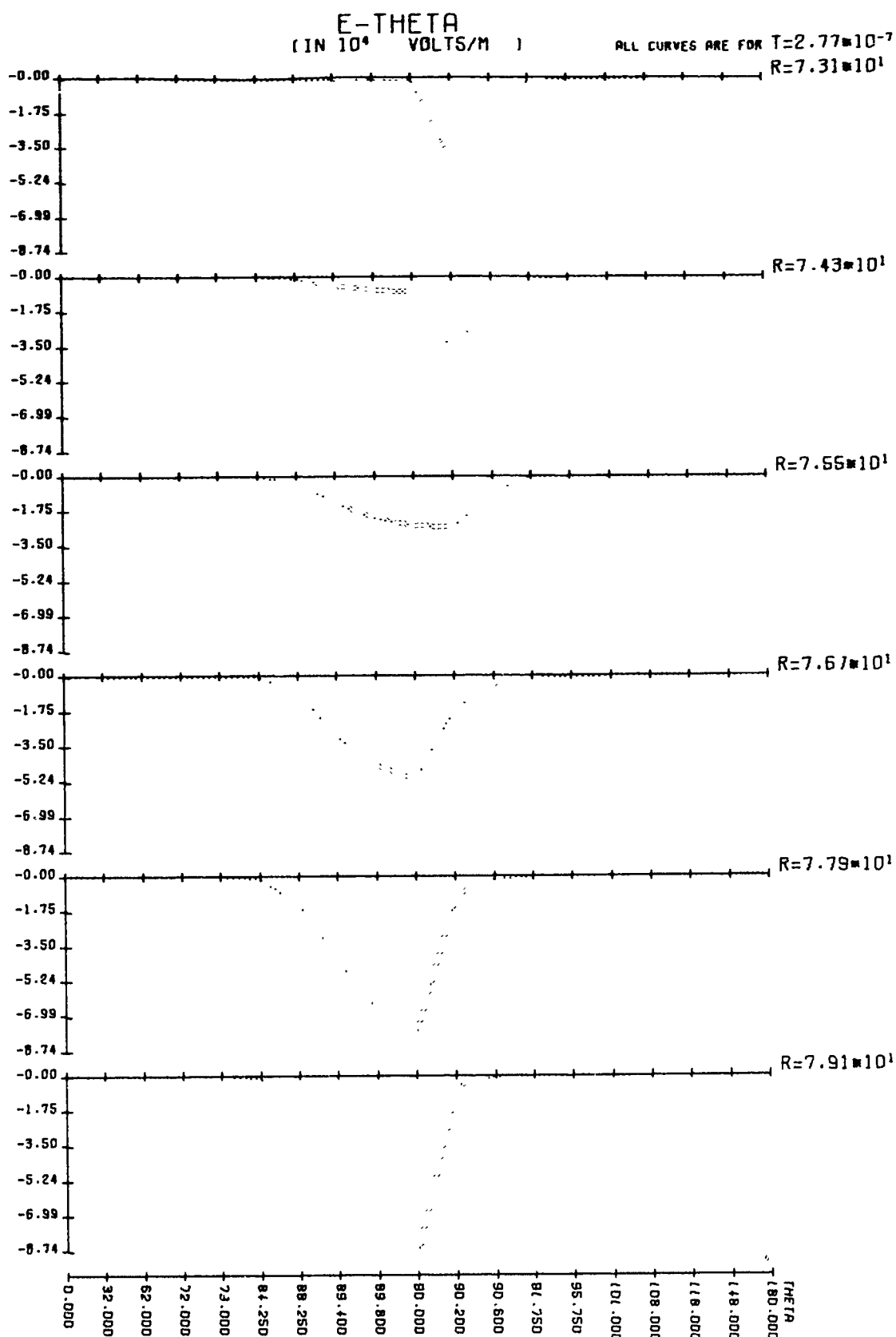


Figure 7.

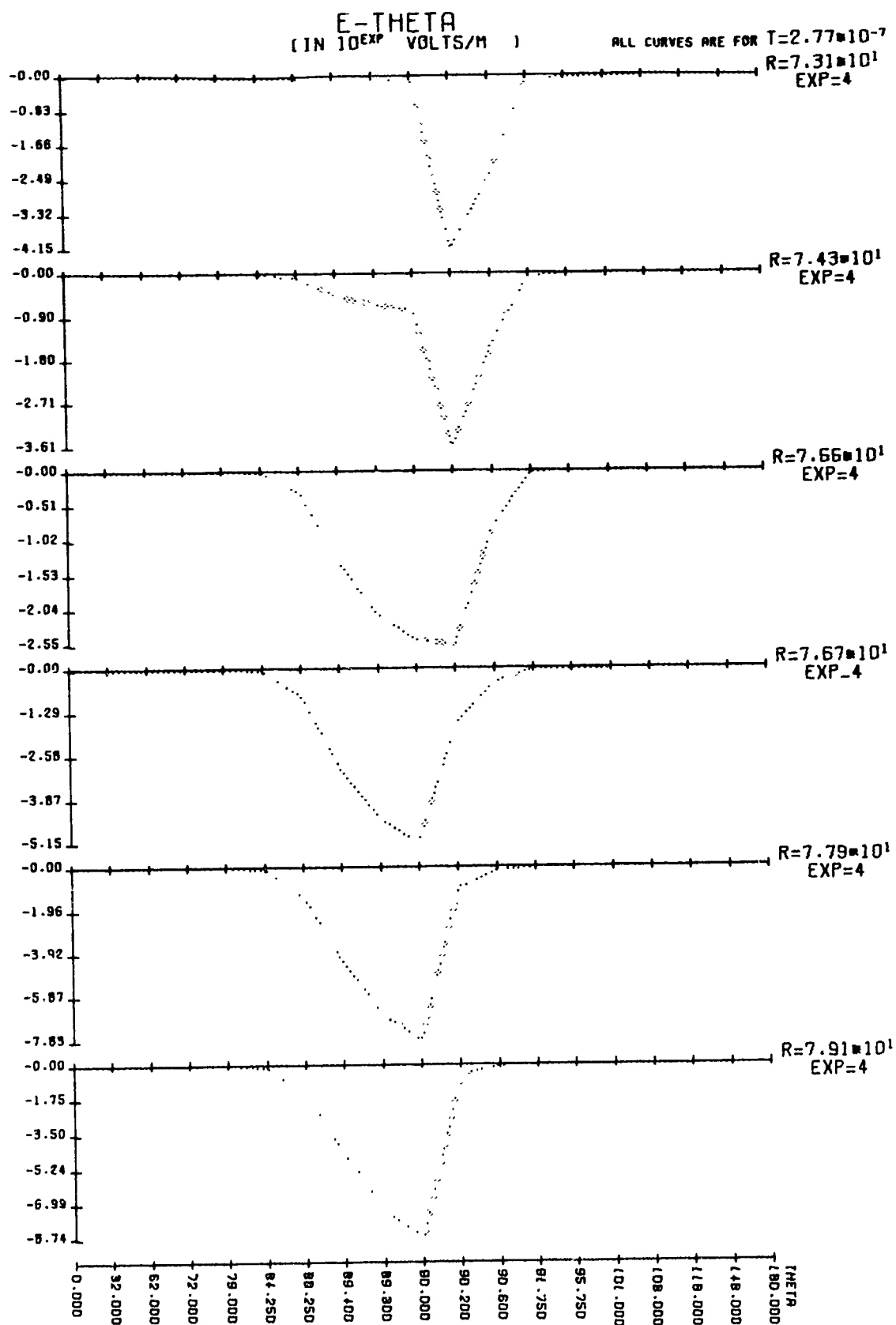


Figure 8.

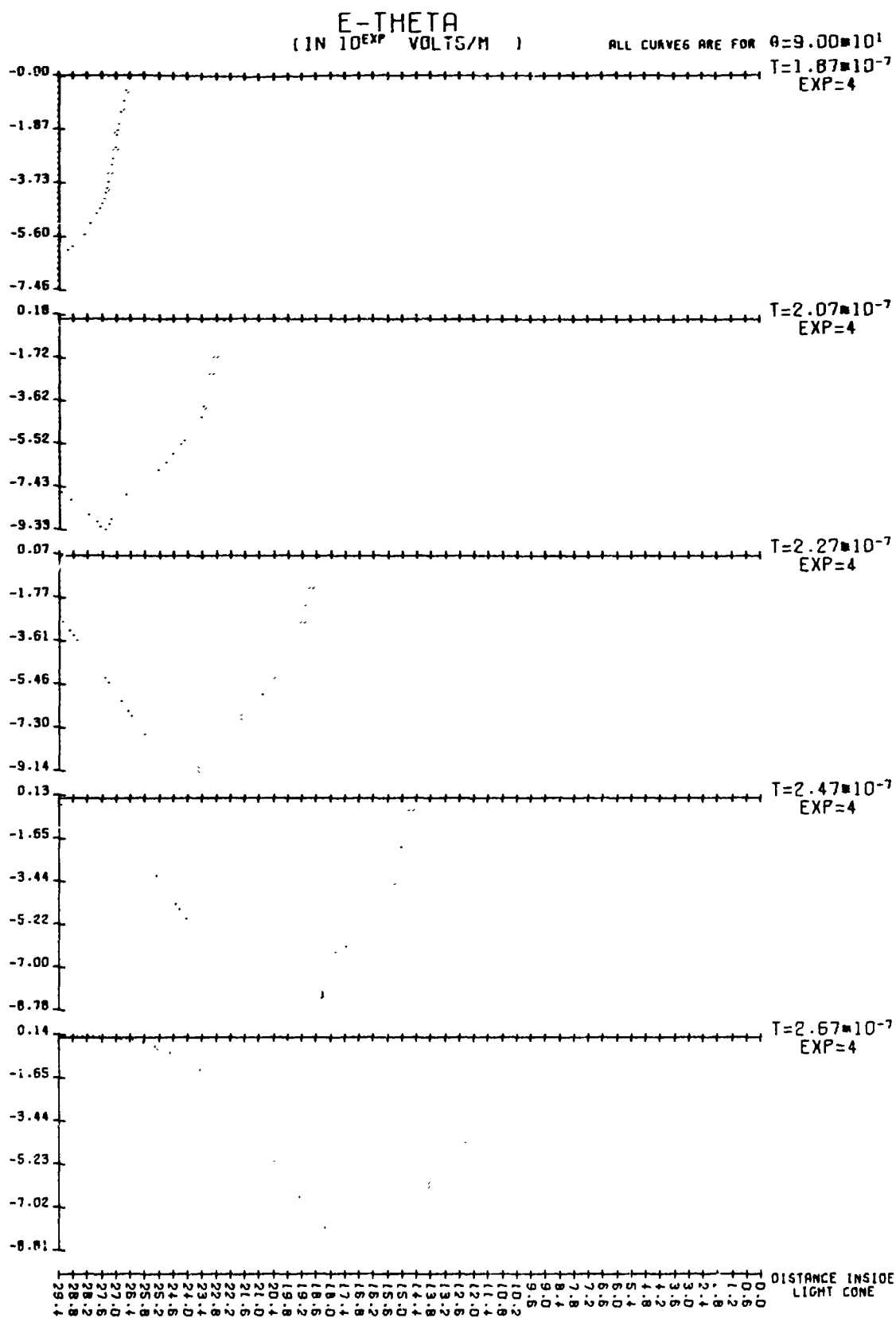


Figure 9.

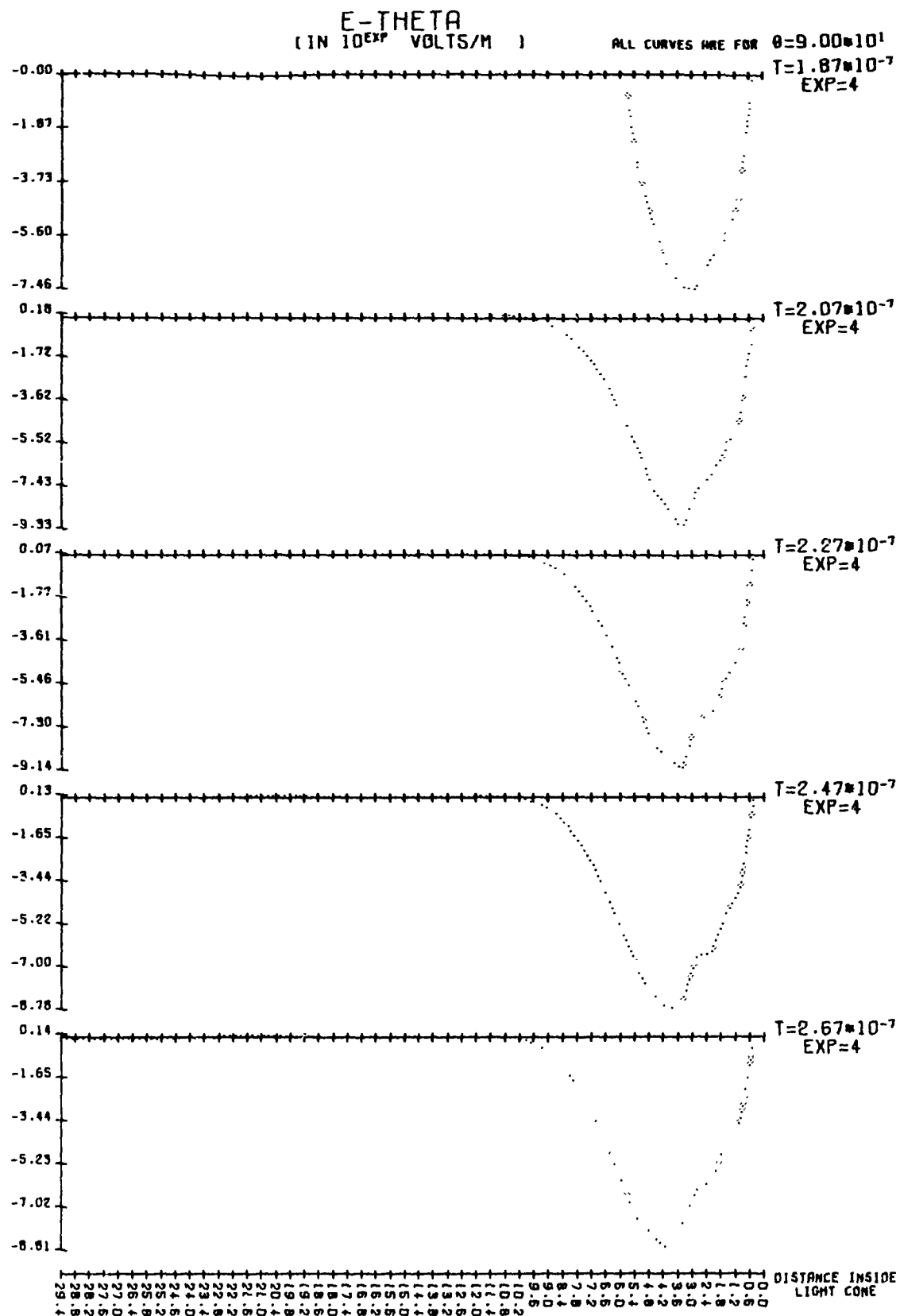


Figure 10.

E-RADIAL (IN 10^5 VOLTS/M)

ALL CURVES ARE FOR $\theta = 9.00 \times 10^1$

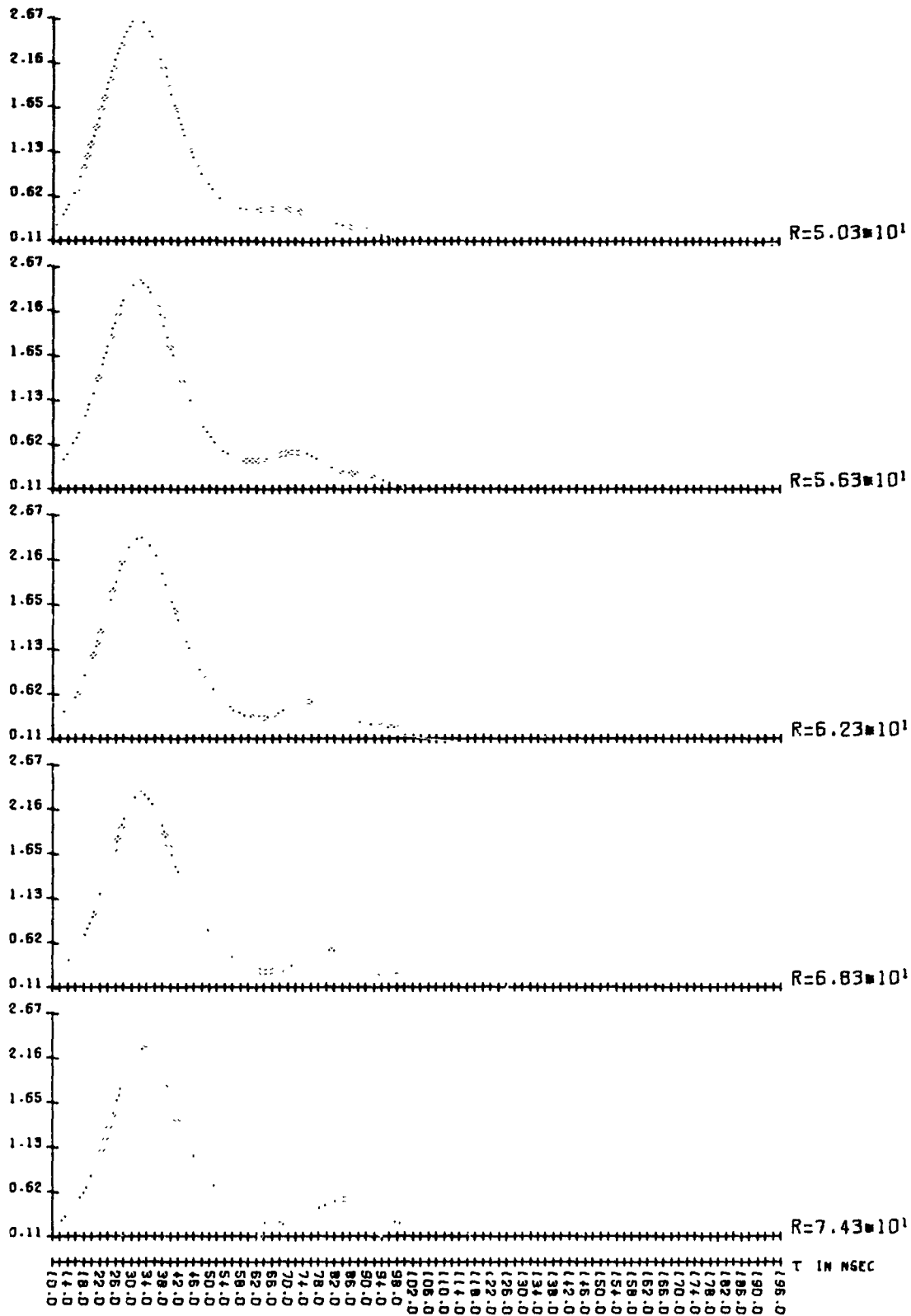


Figure 11.

E-RADIAL (IN 10^{EXP} VOLTS/M)

ALL CURVES ARE FOR $\theta = 9.00 \times 10^1$

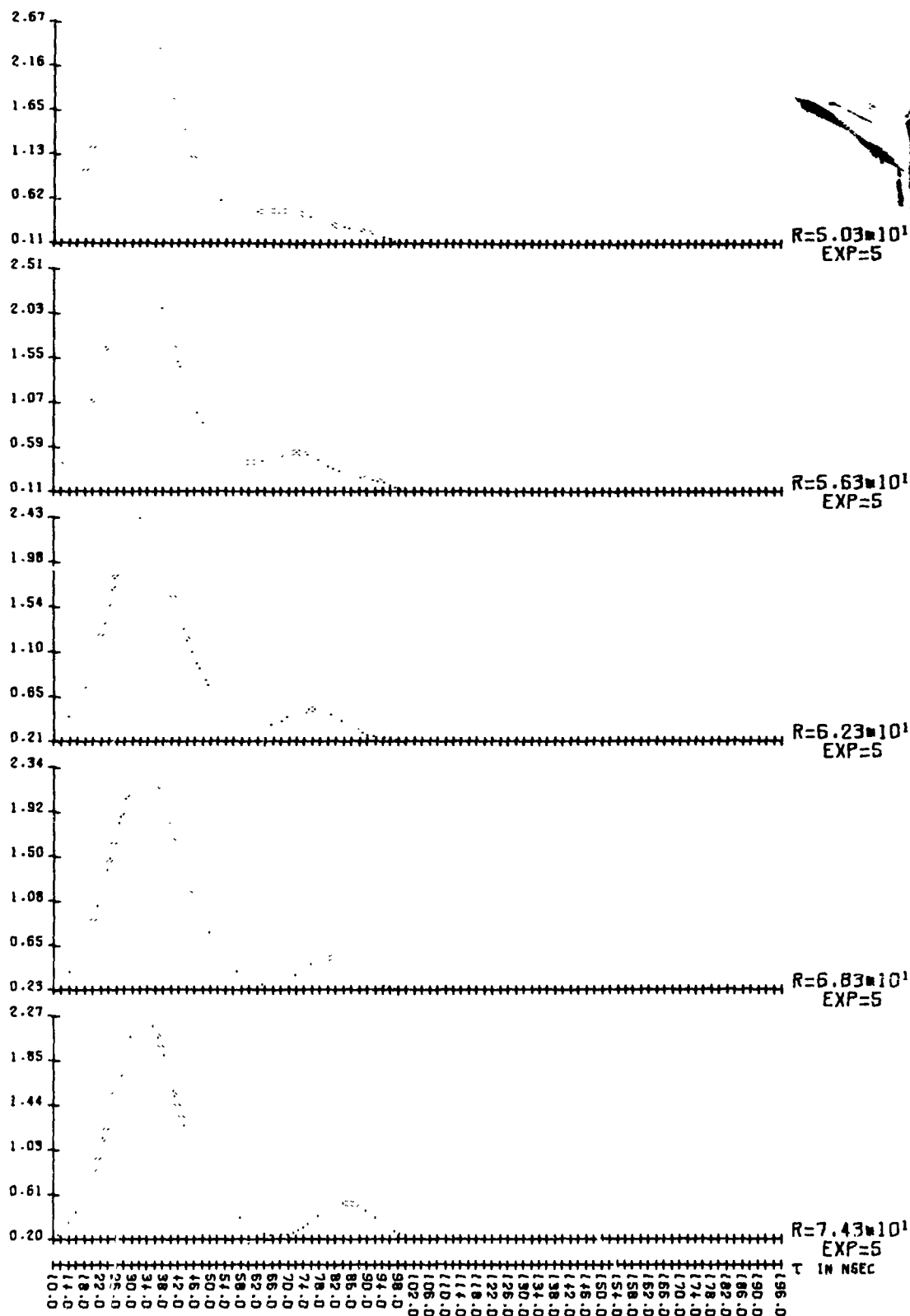


Figure 12.

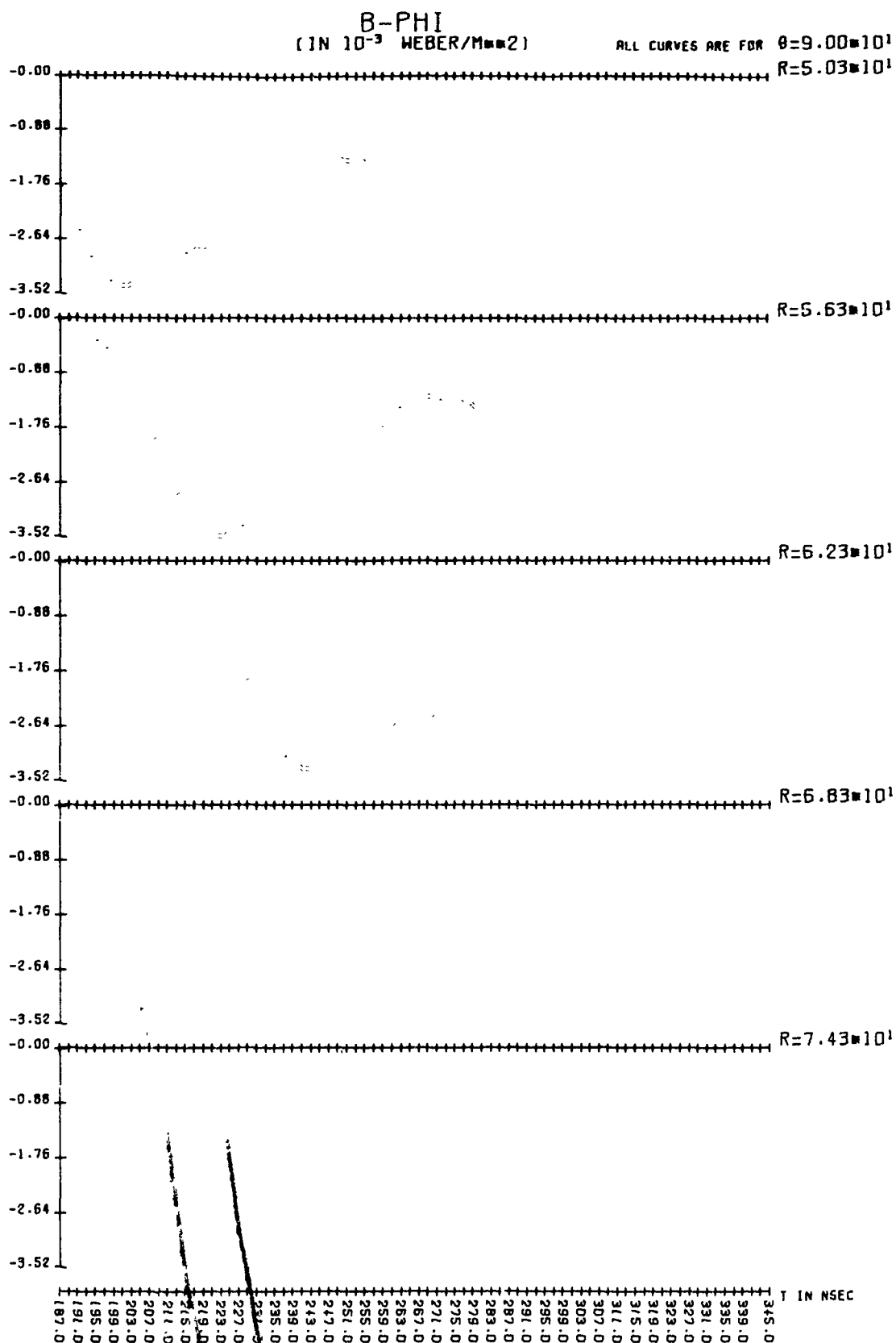


Figure 13.

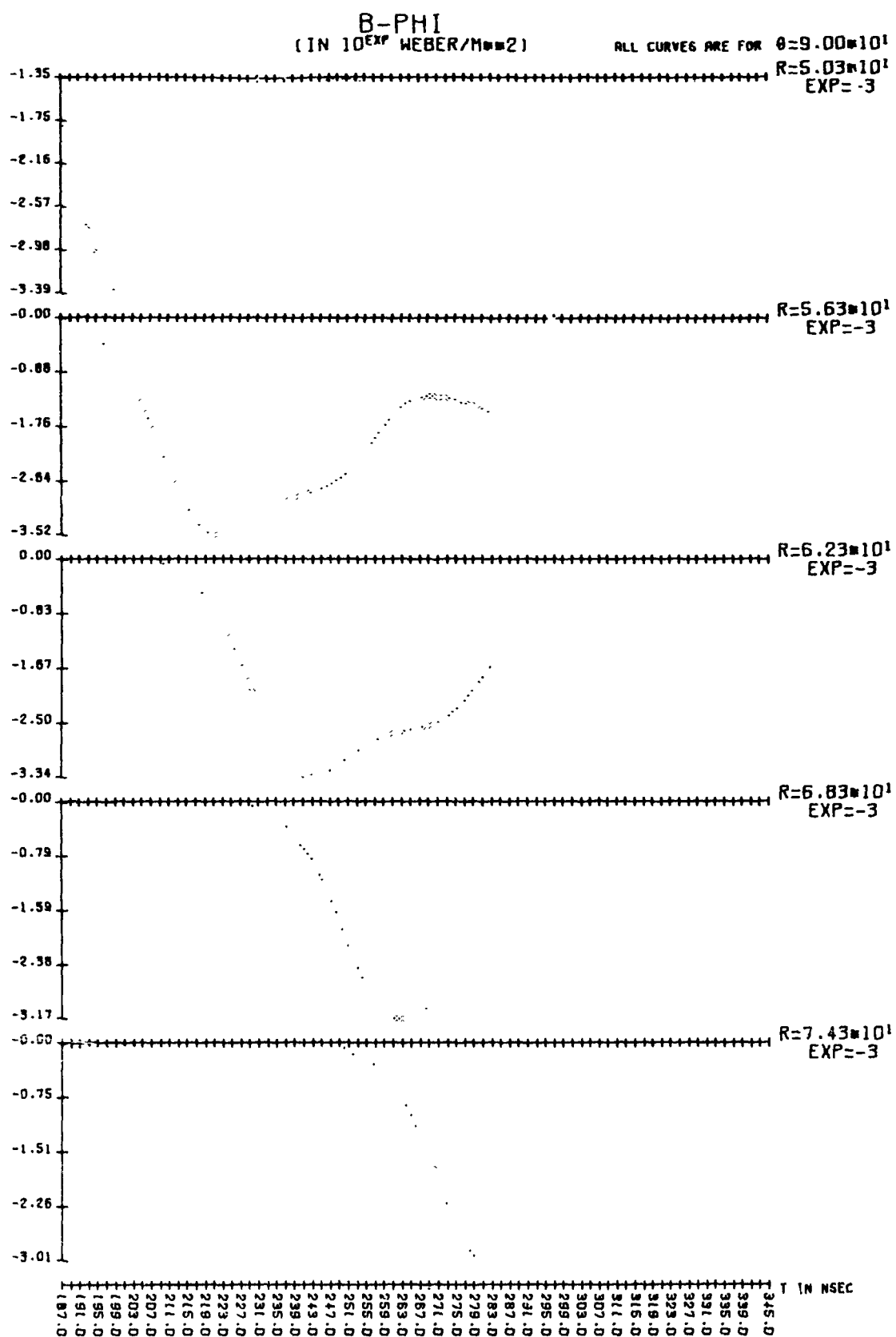


Figure 14.

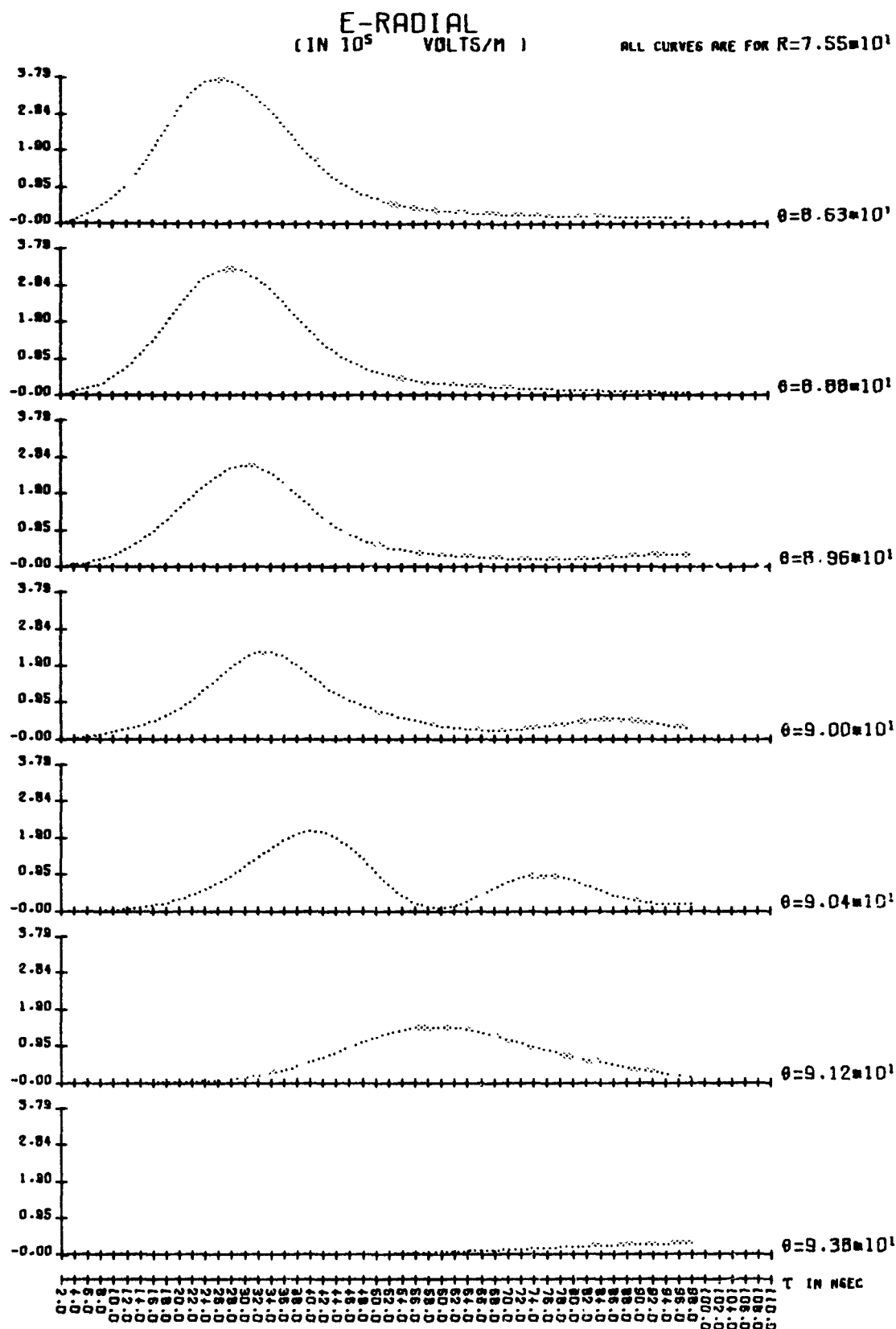


Figure 15.

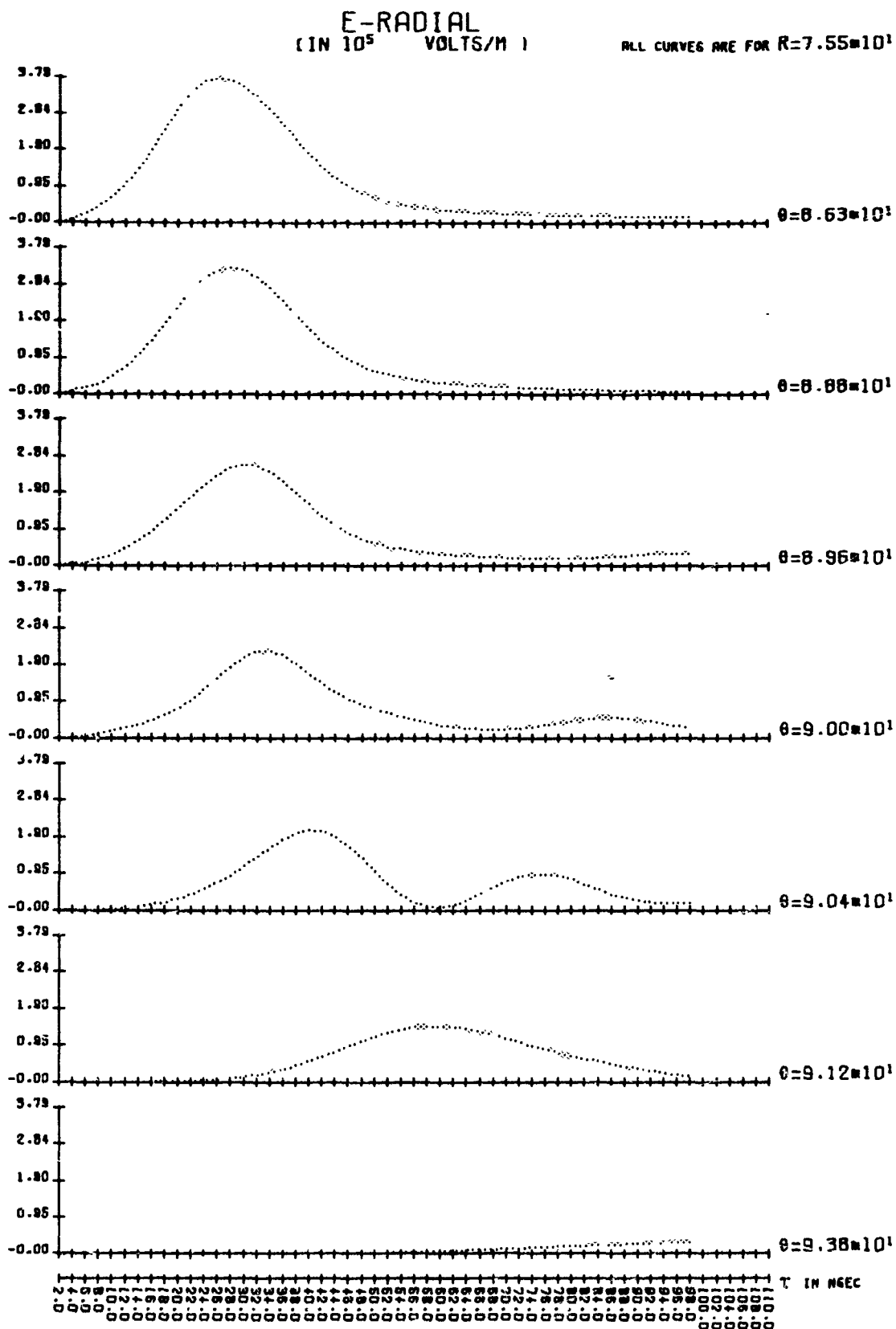


Figure 17.

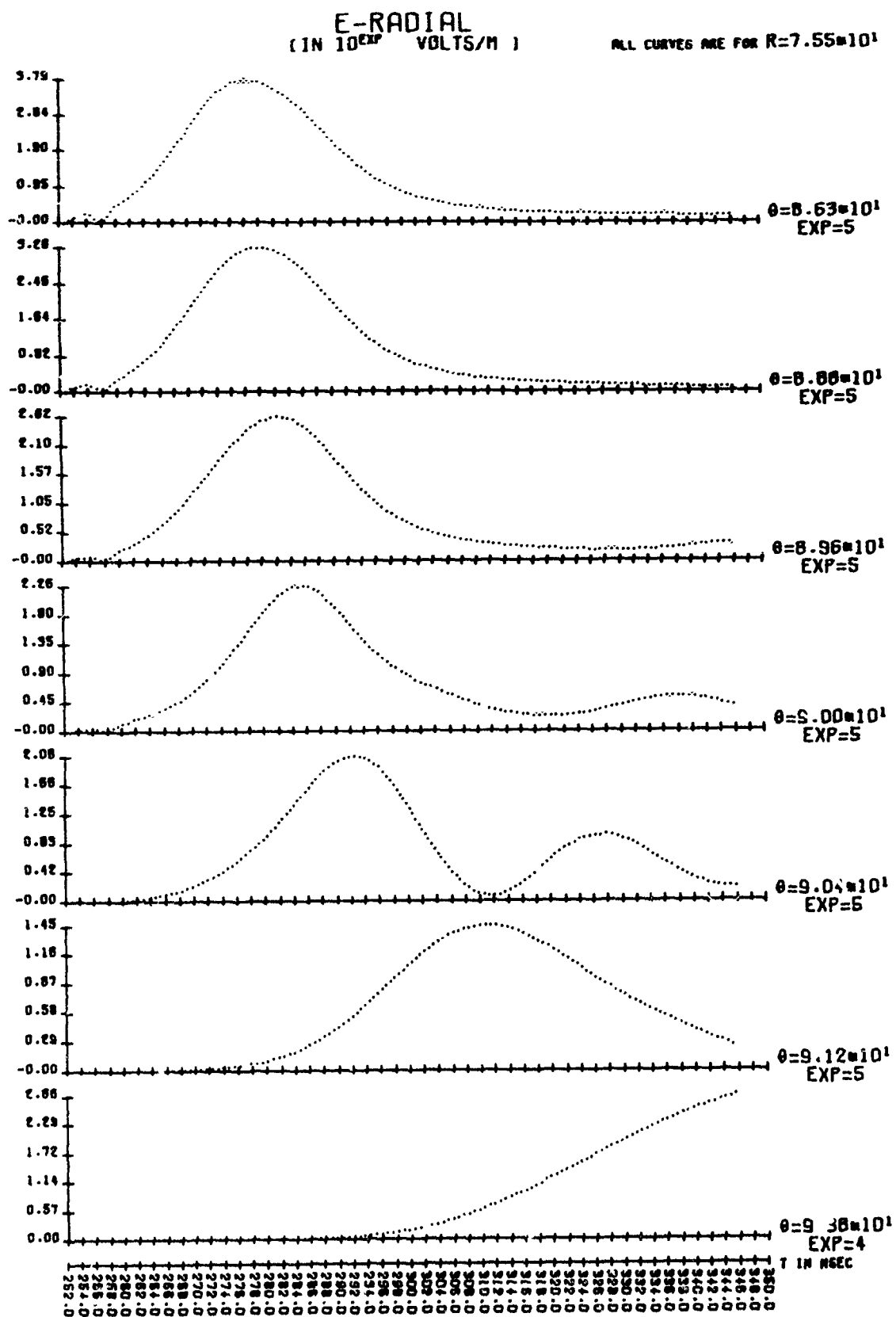


Figure 18.

12. Program Listing


```

000214      WRITE(2) HEADFR
000221      WRITE(2) OUTARR
000226      LABEL=LABEL(IC)
000230      LUN=UNITS(IC)
000232      ITHETA=0
000233      ICOL=1
000234      LIMIT=MAX*1
000236      IF(IC.NE.4) GO TO 245
000240      ICOL=6
000241      LIMIT=MAX
000242      DO 25 I=1,LIMIT
000244          IPI=1+3
000246          TEMPO=ITARR(IP3*ICOL,
000248              IPI*TEMP*1+ITHETA) GO TO 250
000252          IF(TEMP.GT.ITHETA) GO TO 250
000254          ITHETA=ITHETA+1
000256          IF(ITHETA.EQ.1) ITHETA=1
000258          XNUM(ITHETA)=TEMP
000260          CONTINUE
000262          IF(IC.EQ.4) GO TO 280
000264          IF(XNUM(I).GE.90.) GO TO 265
000266          CONTINUE
000268          GO TO 280
000270          IPI=1
000272          IPI=190+1
000274          IPI=190+1
000276          IPI=190+1
000278          DO 27 J=190,ITHETA
000280              XNUM(IPUT+1)=XNUM(IPUT)
000282              IPI=IPUT+1
000284          CONTINUE
000286          XNUM(I90)=90.
000288          ITHETA=ITHETA+1
000290          IPI=0
000292          IPI=190
000294          IPI=190
000296          IPI=190
000298          IPI=190
000300          IPI=190
000302          IPI=190
000304          IPI=190
000306          IPI=190
000308          IPI=190
000310          IPI=190
000312          IPI=190
000314          IPI=190
000316          IPI=190
000318          IPI=190
000320          IPI=190
000322          IPI=190
000324          IPI=190
000326          IPI=190
000328          IPI=190
000330          IPI=190
000332          IPI=190
000334          IPI=190
000336          IPI=190
000338          IPI=190
000340          IPI=190
000342          IPI=190
000344          IPI=190
000346          IPI=190

000346      READ(IUNIT) OUTARR
000348      IF(EOF,IUNIT) 500,320
000350      IF(OUTARR(1+6).LT.CONVAM*H2) GO TO 300
000352      IF(OUTARR(1+6).GT.CONVAM*H2) GO TO 300
000354      T=OUTARR(1+1,ITIM)
000356      DO 33 L=LSTART,NZ
000358          IF(IT-IT.GT.Z(L)) GO TO 330

```

CONSTANT = RANGE

X-AXIS = THETA 4-AXIS = TIME


```

000617      GO TO 500
000618      DO 450 M=1,ITHETA
000619      INDEX=INDEX+1
000620      T=ITMIN+2*M
000621      Y(INDEX)=OUTARR(IR,ICVM)
000622      450 CONTINUE
000623      IF(IL,LT,NZ) GO TO 350
000624      500 IF(INCX,FQ,2) GO TO 600
000625      DO 550 I=1,ITHETA
000626      X(I)=XNUM(I)
000627      550 CONTINUE
000628      GO TO 700
000629      600 DO 650 I=1,ITHETA
000630      X(I)=I
000631      650 CONTINUE
000632      700 IF(ISCALF.EQ.0) GO TO 750
000633      CALL LYNEUP
000634      GO TO 800
000635      750 CALL LINEUP
000636      800 CALL PLOT(XLEN+2.5*0.1,-3)
000637      IUNIT=2
000638      IWOFF=1
000639      GO TO 220
000640
000641      X-AXIS = RET.RANGE
000642      Z-AXIS = TIME
000643      CONSTANT = THETA
000644
000645      1000 READ 2, NZ
000646      READ 4, (Z(I),I=1,NZ)
000647      TMAX=Z(1)
000648      DO 1005 I=2,NZ
000649      IF(TMAX,LT,Z(I)) TMAX=Z(I)
000650      1005 CONTINUE
000651      READ 4, DT,DR,XLFN,YLFN
000652      HORS=5*DR
000653      READ 2, NRANGE,RMIN,RINC
000654      XNUM(1)=RMIN
000655      DO 1100 M=2,NRANGE
000656      XNUM(M)=XNUM(M-1)+RINC
000657      1100 CONTINUE
000658      DELX=(NRANGE-1)*RINC
000659      NPZ=NRANGE*NZ
000660      NX=NRANGE
000661      XI AB=XLARL(ITYPE)
000662      LABTYP=LABTYP(ITYPE)
000663      1000 READ 5, AVAR,CONVAR,INCA,ISCALE
000664      IF(INCX,FQ,0) GO TO 100
000665      DO 1025 I=1,NB7
000666      Y(I)=0.0
000667      1005 CONTINUE
000668      DO 1040 J=1,4
000669      IC=J
000670      IF(AVAR,FQ,VAR(J)) GO TO 1050
000671      1040 CONTINUE
000672      1050 REWIND 1
000673      READ(1) NDUM,NDUM,LMAX
000674      READ(1) HEADER
000675      READ(1) OUTARR
000676      ICVM=ICV(IC)

```

```

001056      LAD=LABEL(IC)
001057      LON=LUNITS(IC)
001058      INDEX=0
001059      T=LASTEN.
001060      T=0
001061      ICUL=6
001062      LIMIT=LMAX
001063      IF (IC.EQ.4) GO TO 1055
001064      ICUL=1
001065      LIMIT=LMAX+1
001066      IF (CONVAR.EQ.90.) IM=2
001067      IF (IR.EQ.2) GO TO 1075
001068      GO 1060 T=1,LIMIT
001069      T=1+3
001070      IF (OUTARR(IR,ICOL).EQ.CONVAR) GO TO 1070
001071      CONTINUE
001072      PRINT 9, CONVAR
001073      GO TO 1070
001074      1065 READ(1) OUTARR
001075      IF (EOF.1) 1200,1070
001076      T=OUTARR(1,1)
001077      IF (T.EQ.TLAST) GO TO 1100
001078      DO 1080 I=1,N7
001079      IF (T-DT.GT.Z(L)) GO TO 1080
001080      IF (T-DT.GE.Z(L)) GO TO 1100
001081      CONTINUE
001082      IF (T-DT.GT.TMAX) GO TO 1200
001083      GO TO 1065
001084      1100 R=OUTARR(1,6)
001085      T=LAST
001086      DO 1110 M=1,NRANGE
001087      IF (R-HDR.GT.XNUM(M)) GO TO 1110
001088      IF (R-HDR.GE.XNUM(M)) GO TO 1130
001089      CONTINUE
001090      GO TO 1065
001091      1130 INDEX=(L-1)*NRANGE+M
001092      Y(INDEX)=OUTARR(IR,ICOL)
001093      GO TO 1065
001094      1200 IF (INX.EQ.2) GO TO 1220
001095      DO 1210 I=1,NRANGE
001096      X(I)=XNUM(I)
001097      CONTINUE
001098      GO TO 1240
001099      1220 DO 1230 I=1,NRANGE
001100      X(I)=1
001101      CONTINUE
001102      1240 IF (ISCALF.EQ.2) GO TO 1250
001103      CALL LTKONE
001104      GO TO 1300
001105      1250 CALL LTCONE
001106      1300 CALL PLOT(XLEN*2.5,0.,-3)
001107      GO TO 1030

```

C X-AXIS = TIME Z-AXIS = RANGE CONSTANT = THETA
 2000 READ 2, NZ
 001236 READ 4, (Z(I),I=1,NZ)
 001244 READ 4, DT,DR,XLFN,YLEN
 001257

```

001273 HDR=5*DR
001275 READ 2, NDELTS, TMIN
001277 NX=1
001279 XNUM(1)=TMIN
001300 DO 2010 I=1, NDELTS
001310 READ 2, NDELTS, DELT
001320 DO 2012 J=1, NDELTS
001322 NX=NX+1
001324 XNUM(NX)=XNUM(NX-1)+DELT
001327 CONTINUE
001334 REMIND 1
001336 IUNIT=1
001337 IWRITE=0
001340 NXZ=NX*N7
001342 XLAB=XLABEL(IWRITE)
001345 LABTYPE=LABTYPE(IWRITE)
001346 KONTYPE=KONTYPE(IWRITE+1)
001350 READ 5, AVAR, CONVAR, INCA, ISCALE, TCOL
001352 IF (INCA.FO.0) GO TO 100
001354 REMIND 2
001357 DO 2032 I=1, N7
001371 NPTS(I)=0
001373 CONTINUE
001374 DO 2035 I=1, NXZ
001376 Y(I)=0.0
001400 CONTINUE
001401 DO 2040 J=1, 14
001403 IC=J
001405 IF (AVAR.FO.VAR(J)) GO TO 2050
001406 CONTINUE
001410 READ(IUNIT) NNDUM, LMAX
001412 READ(IUNIT) HEADER
001423 READ(IUNIT) OUTARR
001430 IF (IWRITE.FO.1) GO TO 2055
001435 WRITE(2) NNDUM, LMAX
001437 WRITE(2) HEADER
001450 WRITE(2) OUTARR
001455 ICVH=ICV(IC)
001457 LAB=LABEL(IC)
001466 LUN=LUNITS(IC)
001467 ITIM=ITIME(IC)
001471 IF (ICOL.NE.0.) ITIM=1
001473 TI=0
001474 IR=0
001475 ICOL=6
001476 LIMIT=LMAX
001477 IF (IC.FO.4) GO TO 2060
001479 ICOL=1
001501 LIMIT=LMAX+1
001502 IF (CONVAR.FO.99.) IM=2
001504 IF (IM.FO.2) GO TO 2075
001507 DO 2065 I=1, LIMIT
001511 IR=I+3
001513 IF (OUTARR(IR,ICOL).EQ.CONVAR) GO TO 2075
001515 CONTINUE
001522 PRINT 9, CONVAR
001524 GO TO 2030
001532 2070 READ(IUNIT) OUTARR
001533

```

```

001540 IF (EOF, IUNIT) 2120, 2075
001541 R=OUTARR(1,6)
001542 DO 2080 M=1, N7
001543 IF (R-HDR.GT.Z(M)) GO TO 2080
001544 IF (R-HDR.GE.Z(M)) GO TO 2080
001545 GO TO 2070
001546 CONTINUE
001547 GO TO 2070
001548 T=OUTARR(1, ITIM)
001549 DO 2100 I=1, NX
001550 IF (T-OT.GT.XNUM(I)) GO TO 2100
001551 IF (T-OT.GE.XNUM(I)) GO TO 2110
001552 GO TO 2070
001553 CONTINUE
001554 IF (M.EQ.N7.AND.L.EQ.NX) GO TO 2120
001555 GO TO 2070
001556 INDEX=(M-1)*NX+L
001557 Y(IX)=OUTARR(IR, ICVM)
001558 IF (I=ROT.EQ.0) WRITE(2) OUTARR
001559 NPTS(M)=NPTS(M)+1
001560 IL=IL+1
001561 IF (IL.LT.NXZ) GO TO 2070
001562 IF (INCX.FO.2) GO TO 2130
001563 DO 2130 I=1, NX
001564 X(I)=XNUM(I)
001565 CONTINUE
001566 GO TO 2150
001567 DO 2140 I=1, NX
001568 X(I)=I
001569 CONTINUE
001570 IF (ISCALF.EQ.0) GO TO 2160
001571 CALL TMAXS
001572 GO TO 2200
001573 CALL TIMAXS
001574 GO TO 2200
001575 CALL PLOT(XLEN*2.5, 0.0, -3)
001576 IUNIT=2
001577 IWRITE=1
001578 GO TO 2030
001579
001670
001671 X-AXIS = TIME      Z-AXIS = THETA      CONSTANT = RANGE
001672
001673 3000 READ 4, DT, DR, XLEN, YLEN, CONVAR
001674 WRITE(5, 500P)
001675 READ 2, NDELT, TMIN
001676 NX=1
001677 XNUM(1)=TMIN
001678 DO 3010 I=1, NDELT
001679 READ 2, NDELT, DELT
001680 DO 3010 J=1, NDELT
001681 NX=NX+1
001682 XNUM(NX)=XNUM(NX-I)*DELT
001683 CONTINUE
001684 P=IND 1
001685 IUNIT=1
001686 IWRITE=0
001687 KONTYPE=LALTY(2)
001688 LABTYPE=LALTY(ITYPE)
001689 READ 2, N7

```


[illegible]

000	000
001	001
002	002
003	003
004	004
005	005
006	006
007	007
008	008
009	009
010	010
011	011
012	012
013	013
014	014
015	015
016	016
017	017
018	018
019	019
020	020
021	021
022	022
023	023
024	024
025	025
026	026
027	027
028	028
029	029
030	030
031	031
032	032
033	033
034	034
035	035
036	036
037	037
038	038
039	039
040	040
041	041
042	042
043	043
044	044
045	045
046	046
047	047
048	048
049	049
050	050
051	051
052	052
053	053
054	054
055	055
056	056
057	057
058	058
059	059
060	060
061	061
062	062
063	063
064	064
065	065
066	066
067	067
068	068
069	069
070	070
071	071
072	072
073	073
074	074
075	075
076	076
077	077
078	078
079	079
080	080
081	081
082	082
083	083
084	084
085	085
086	086
087	087
088	088
089	089
090	090
091	091
092	092
093	093
094	094
095	095
096	096
097	097
098	098
099	099

```

SUBROUTINE LIHUP
  COMMON /X(2000),Y(2000),TIME(1),IHEIA(20),OUTAB(21,7),IQUF(1024),
  1 HADER(8),NCJUVF,NCJINT,YLFW,XLFW,LAB(6),LUN,LANTYP,XLAB,CNVAO,
  2 KONTYP,DEL,NP(5),ICOL
  COMMON/KOFF/YMIN,DELY,YMINCO,DELYCO,NSO(5),IP,HEIGHT
  COMMON/XYSLOT/XLSLOT(200),YLSLOT(200),YNU(5),YMAXQS(5),XCJUV(200)
  IF=01/20.
  CVAP=CNVAP+01F
  NVAP=0.
  100 IF(CVAP*.65+.1.) GO TO 150
  CVAP=CVAP*.15.
  NVAP=NVAP+.1.
  GO TO 160
  150 IF(CVAP*.11+.17.) GO TO 175
  CVAP=CVAP*.10.
  NVAP=NVAP+.1.
  GO TO 150
  175 CALL LIMPS(0.6,0,-2)
  OSLX=(X(NCJINT)-Y(1))/XLFW
  XMIN=X(1)
  DO 200 J=1,NCJINT
    XLSLOT(J)=(X(J)-YMIN)/OFLX
  CONTINUE
  NCJ=NCJUVF*NCJINT
  HEIGHT=YLEN*NCJUVF-.25
  CALL SCALE(Y,HEIGHT,NCJ,1)
  YMIN=Y(NCJ+1)
  DELY=Y(NCJ+2)
  YMAX=YF(1)+OFLY+YMIN
  CALL GOF
  OMF=IP
  XL2=XLEN/2.
  CALL SYMROL(XL2-.75,10.75,.15,LA+.0,.11)
  CALL SYMROL(XL2-1.,1.5,.15,M(TN 15,.,.6)
  CALL NUMERO(XL2-.5,10.55,.17,OOH,0,.-1)
  CALL SYMROL(XL2-6.,10.5,.15,M,5.,10)
  CALL SYMROL(XL2+.3,10.5,.15,.,.1)
  CALL SYMROL(XLEN-1.2,11.5,0.7,1,SMALL CUPVES AND FOR,J.,.19)
  IF(KONTYP.EQ.240) GO TO 250
  IF(ICOL.EQ.1.) GO TO 250
  CALL GOFEX(XLEN+.1,10.5,.1,19,0.,C)
  CALL SYMROL(XLEN+.2,10.5,.1,14,0.,1)
  GO TO 260
  250 CALL SYMROL(XLEN+.1,10.5,.1,KONTYP,0.,2)
  260 CALL NUMERO(XLEN+.3,10.5,.1,CVAP,16.,2)
  CALL SYMROL(XLEN+.7,10.5,.1,XM(1),.3)
  CALL NUMERO(XLEN+1.,10.55,.17,PVAP,.,.-1)
  YSHIFT=YMIN/DELY
  IF(YMIN*.5+.0.) YSHIFT=0.5
  IF(YMAX*.5+.0.) YSHIFT=HEIGHT
  DO 300 I=1,MCJ
    Y(I)=(Y(I)-YMIN)/OFLY
  CONTINUE
  YPOS=YLEN*.375+HEIGHT
  NCJ=NCJINT+1
  DO 600 I=1,NCJUVF

```

```

000230      GO 400 J=1,NPQINT
000231      ICI=ICI+1
000232      YPLNT(J)=Y(ICT),YPOS
000233      CONTINUE
000234      CALL LINES(XPLNT(I),YPLNT(I),J)
000235      CALL LINES(XPLNT,I,YPLNT,I)
000236      YPOS=YPOS+YSCALEI
000237      OTIME=0.0
000238      AXTIME=TIME(I)+OIF
000239      IF(CAMPII,GF,1) GO TO 450
000240      AXTIME=AXTIME+IC
000241      OTIME=OTIME+1
000242      GO TO 420
000243      IF(CAMPII,LI,1J) GO TO 460
000244      AXTIME=AXTIME/10
000245      OTIME=OTIME+1
000246      GO TO 450
000247      YLAB=YPOS+3.5
000248      IF(LABYD,CO,200=) GO TO 465
000249      IF(ICAL,IF,0) GO TO 465
000250      CALL ZDEF(XLEN+1,YLAB,1,1,0,0)
000251      CALL SYM9L(XLEN+1,YLAB,1,1,1,1)
000252      GO TO 470
000253      CALL SYM9L(XLEN+1,YLAB,1,1,1,1,0,2)
000254      CALL NUMBE(XLEN+1,YLAB,1,1,1,1,0,2)
000255      CALL SYM9L(XLEN+1,YLAB,1,1,1,1,0,3)
000256      CALL NUMBE(XLEN+1,YPOS+0.75,0.07,OTIME,0,-1)
000257      CALL PLNT(XPLNT(XPLNT),YPOS+3)
000258      DO 500 K=1,NPQINT
000259      KX=NP1-K
000260      CALL SYM9L(XPLNT(KX),YPOS+0.07,13,0,-2)
000261      CALL PLNT(XPLNT(KX),YPOS+3)
000262      CONTINUE
000263      YPOS=YPOS-HEIGHT-20
000264      CONTINUE
000265      YINC=HEIGHT/NSPACE
000266      YSTART=YLEN+375
000267      YMAX=YINC+OELYC
000268      YINC=JELYN/NSPACE
000269      HTIC=NSPACE+1
000270      GO 620 IF=1,NTIC
000271      YXPOS(I)=-.35
000272      CONTINUE
000273      YNUM(I)=YMAX
000274      IF(YMAX,LI,0) YXPOS(I)=-.42
000275      GO 630 J=2,NTIC
000276      YNUM(J)=YNUM(J-1)-YINC
000277      IF(YNUM(J),LI,0) YXPOS(J)=-.42
000278      CONTINUE
000279      GO 700 IF=1,NCPDEF
000280      CALL PLNT(C,YSTART,3)
000281      CALL SYM9L(C,YSTART,1,1,1,1,0,2)
000282      CALL NUMBE(YXPOS(I),YSTART,0.37,YNUM(I),0,2)
000283      CALL PLNT(C,YSTART,1)
000284      GO 650 J=2,NTIC
000285      YSTART=YSTART-YINC
000286      CALL SYM9L(C,YSTART,1,1,1,1,0,3)
000287      CALL NUMBE(YXPOS(J),YSTART,0.07,YNUM(J),0,2)

```

```

000463      CALL PLOT(0.,YSTART,3)
000466      CONTINUE
000471      YSTART=YSTART+.26
000473      700 CONTINUE
000475      CALL PLOT(XPLOT(NPINT),YSTART,3)
000500      DO 800 K=1,NPINT
000502      IF (THETA(K).EQ.90.) X90=XPLOT(K)
000506      KK=NP1-K
000510      CALL SYMQL(XPLOT(KK),YSTART,.07,13,0.,-2)
000514      CALL PLOT(XPLOT(KK),YSTART,3)
000517      900 CONTINUE
000522      YSTART=YSTART+.07
000524      XLAST=XPLOT(1)
000525      X90=XPLOT(NPINT)
000526      DO 900 J=1,NPINT
000530      XTEST=XPLOT(J)
000532      IF (J.EQ.1.OR.J.EQ.NPINT) GO TO 880
000541      IF (XTEST.EQ.X90) GO TO 880
000543      IF (XTEST.GT.X90) GO TO 950
000546      IF (XTEST-XLAST.LT.(.09.09.X90-XTEST.LI-.09) GO TO 970
000557      GO TO 880
000560      850 IF (XTEST-XLAST.LI-.09.09.X90-X90) X90=XTEST.LI-.09) GO TO 900
000572      880 CALL NUMREP(XTEST-.035,YSTART,.07,THETA(J),270.,3)
000601      XLAST=XTEST
000603      900 CONTINUE
000606      CALL SYMQL(XLEN+.1,YSTART,.07,XLAST,270.,5)
000614      RETURN
000615      END

```

```

000002      SUBROUTINE LYNEUP
000002      COMMON X(200),Y(200),TIME(10),THETA(200),OUTARS(21,7),TRUF(1024),
000002      1 HEADPR(8),NCURVE,NPOINT,YLEN,XL,N,LAB,OT,LUN,LARTV,XLA,CONVAR,
000002      2 CONTYP,DEL,NPIS(10),ICOL
000002      COMMON/KOEF/ YMIN,DELY,YMICO,DELYCO,NSPASE,IP,HEIGHT
000002      COMMON/XYPLNT/ XPLNT(200),YPLNT(200),YNUX(6),YXPOQS(6),XCURVE(200)
000002      JTF=OT/2J.
000002      CVAO=CONVAR+DTF
000002      PVAO=0.
000002      100 IF(CVIR.GE.1.) GO TO 150
000002      CVAO=CVAP*13.
000002      PVAR=PVAO-1.
000002      GO TO 100
000002      150 IF(CVAR.LT.10.) GO TO 175
000002      CVAO=CVAP/10.
000002      PVAR=PVAR+1.
000002      GO TO 150
000002      175 CALL LINES(G,6,C,-2)
000002      DELX=(X(NPOINT)-X(1))/XLEN
000002      XMI=X(1)
000002      DO 200 J=1,NPOINT
000002      XPLNT(J)=(X(J)-XMIN)/DELX
000002      CONTINUE
000002      NP1=NPOINT+1
000002      HEIGHT=YLEN/NCURVE -.25
000002      XL2=YLEN/2.
000002      YPOQS=YLEN+.375-HEIGHT
000002      YSTART=YLEN+.375
000002      IS=0
000002      CALL SYMQL(XL2-.75,10.75,.15,LA,0,.10)
000002      CALL SYMQL(XL2-1.,1.,5.,2.6H(IN 10,0,5)
000002      CALL SYMQL(XL2-.4,10.55,.07,3HEXP(0,3)
000002      CALL SYMQL(XL2-0.1,10.55,.1LUN,C,.10)
000002      CALL SYMQL(XL2+.9,10.55,.11-),0.,1)
000002      CALL SYMQL(XLEN-1.23,1.5,.07,1-HALL CURVES ARE FOR,0,.18)
000002      IF(KOPTYO.EQ.2HQR=) GO TO 250
000002      IF(ICOL.NE.0.) GO TO 250
000002      CALL GREEK(XLEN+.1,10.55,.1,19,6,.10)
000002      CALL SYMQL(XLEN+.2,10.55,.1,1H=, .1)
000002      GO TO 260
000002      250 CALL SYMQL(XLEN+.1,10.55,.1,KOPTYO,6,.2)
000002      CALL NUMBER(XLEN+.3,10.55,.1,CVAP,6,.12)
000002      CALL SYMQL(XLEN+.7,10.55,.1,3H+10,3,3)
000002      CALL YUMREP(XLEN+.1,10.55,.07,PVAR,6,.1)
000002      DO 700 II=1,NCURVE
000002      DO 260 J=1,NPOINT
000002      YPLNT(J)=Y(IS+J)
000002      CONTINUE
000002      CALL SCALE(YPLNT,HEIGHT,NPOINT,1)
000002      YMIN=YPLNT(NPOINT+1)
000002      DELY=YPLNT(NPOINT+2)
000002      YMAX=HEIGHT*DELY+YMIN
000002      CALL COEF
000002      PONE=IP
000002      YSHIFT=-YMIN/DELY
000002      IF(YMIN.GE.0.) YSHIFT=0.0
000002      IF(YMAX.LE.0.) YSHIFT=HEIGHT
000002      0.0222

```



```

000503 YSTART=YSTART-.20
000505 ISEIS=NDP(IH)
000506
000511 700 CONTINUE
000514 CALL PLOT(XPL(I,NDP(IH)),YSTART,3)
000516 GO 900 K=1,NDP(IH)
000522 IF(THETA(K).EQ.90.) X90=XPL(I,K)
000524 KK=NDP1-K
000530 CALL SYM03L(XPL(I,KK),YSTART,.07,13,0.,-2)
000533 CALL PLOT(XPL(I,KK),YSTART,3)
000536
000540 800 CONTINUE
000542 YSTART=YSTART-.07
000544 XLAST=XPL(I,1)
000546 X01=XPL(I,NDP(IH))
000548 DO 900 J=1,NDP(IH)
000550 XTEST=XPL(I,J)
000552 IF(J.EQ.1.00.J.EQ.NDP(IH)) GO TO 980
000554 IF(XTEST.EQ.X90) GO TO 990
000556 IF(XTEST.GT.X90) GO TO 950
000558 IF(XTEST-XLAST.LI.0.09.02.X90-XTEST.LI.0.09) GO TO 900
000560 GO TO 920
000562
000574 850 IF(XTEST-XLAST.LI.0.09.02.X01-XTEST.LI.09) GO TO 930
000576 890 CALL NUMREP(XTEST-.035,YSTART,.07,THETA(J),270.,3)
000578 XLAST=XTEST
000580 900 CONTINUE
000582 CALL SYM03L(XLEN+.1,YSTART,.07,XLAR,270.,5)
000584 RETURN
000586 END
000631

```

```

SUBROUTINE LTCONE
COMMON X(200),Y(200),TIME(10),XNUM(200),OUTARR(21,7),IBUF(1024),
1 HEADP(8),NCURVE,NPOINT,YLEN,XLEN,LAB,CT,LUN,LABTYP,XLAB,CONVAR,
2 KUN,TYP,DEL,NPTS(10),ICOL
COMMON/KOEF/ YMIN,DELY,YMINCO,DELYCO,NSPACE,IP,HEIGHT
COMMON/XPLOT/ XPLOT(200),YPLT(200),YNUM(6),YXPOS(6),XCURV(200)
DATA C/3.058/

DTF=DT/20.
CV=CONVAR+DTF
PVARE=
100 IF(CVAR,GE,1.) GO TO 150
CVAR=CVAR+1.
PVAREPVAR-1.
GO TO 100
150 IF(CVAR,LT,10.) GO TO 175
CVARECVAR/10.
PVAREPVAR+1.
GO TO 150
175 CALL LINCS(0,0,0,-2)
DELX=X(NPC,NT)-X(1)/XLEN
XMIN=X(1)
DO 200 J=1,NPOINT
XPLOT(J)=(X(J)-XMIN)/DELX
CONVINE
DELX=DEL/XLEN
NCP=NCURVE*NPOINT
HEIGHT=YLEN/NCURVE *-25
CALL SCALE(Y,HEIGHT,NCP,1)
YMIN=Y(NCP,1)
DELY=Y(NCP,2)
YMAX=HEIGHT+DELY+YMIN
CALL COEF
DO 300 IP
XL2=XLFN/2.
CALL SYMOL(XL2-.75,1-.75,.15,LAR,0.,10)
CALL SYMOL(XL2-1.,10.5,1.16M(IN 10, .16)
CALL NUMBER(XL2-2.4,1-.65,.07,POV,9,1)
CALL SYMOL(XL2-3.1,1.5,1.1LUN,0.1)
CALL SYMOL(XL2+.9,1-.6,1.1M,0.1)
CALL SYMOL(XLEN-1.28,16.6,.07,18HALL CURVES ARE FOR 0.0,18)
CALL GEPK(XLFN+.1,10.6,1.18,0.0)
CALL SYMOL(XLEN+.2,1.5,1.1M,0.1)
CALL NUMBER(XLFN+.3,1-.6,1.1CVAR,0.2)
CALL SYMOL(XLEN+.7,1-.6,1.13M,0.3)
CALL NUMBER(XLEN+.1,1.65,.07,PVAR,0.1)
YMIN=YMIN/DELY
IF(YMIN,GE,0.) YSHIFT=0.0
IF(YMAX,LE,0.) YSHIFT=-1.0
DO 300 I=1,NCP
Y(I)=(Y(I)-YMIN)/DELY
CONTINUE
Y0=YLEN+.375-HEIGHT
Y0=EXPINT.1
DO 300 I=1,NCP
C=C+Y(I)-Y0
I=0

```

```

000223      IN=(I+1)*NPOINT
000224      DO 350 J=1,NPOINT
000225      IF (XNUM(J).GT.CT) GO TO 400
000226      L=L+1
000227      350 CONTINUE
000228      NPL=NPOINT-L
000229      XADJ=XMIN+CT-X(L)
000230      J=NPL+1
000231      INNPL=IN-NPL
000232      DO 410 J=J,NPOINT
000233      IJ=J+INNPL
000234      XCURV(IJ)=(X(J)-XADJ)/DELX
000235      IF (XCURV(IJ).LT.C.) XCURV(IJ)=0.
000236      YPLOT(IJ)=Y(IJ)+YPOS
000237      410 CONTINUE
000238      CALL LINES(XCURV(IJ),YPLOT(IJ),0)
000239      CALL LINES(XCURV(IJ),YPLOT(IJ),L)
000240      YPOPYC=YPOS+YSHIFT
000241      PTIME=0.
000242      AMPTIME=TIME(I)+DTF
000243      420 IF (AMPTIME.E.1.) GO TO 450
000244      AMPTIME=AMPTIME+1.
000245      PTIME=PTIME-1.
000246      GO TO 420
000247      450 IF (AMPTIME.LT.10.) GO TO 460
000248      AMPTIME=AMPTIME/10.
000249      PTIME=PTIME+1.
000250      GO TO 450
000251      460 YLAB=YPOPYC+.725
000252      CALL SYMROL(XLEN+.1,YLAB,.1,LABTYP,0,.2)
000253      CALL NUMER(XLEN+.3,YLAB,.1,AMPTIME,0,.2)
000254      CALL SYMROL(XLEN+.7,YLAB,.1,SH*10,0,.3)
000255      CALL NUMER(XLEN+.1,YPOPYC+.75,.7,PTIME,0,.1)
000256      CALL PLOT(XPLOT(NPOINT),YPOPYC,3)
000257      DO 500 K=1,NPOINT
000258      KK=NPI-K
000259      CALL SYMROL(XPLOT(KK),YPOPYC+.07,13,.1,-2)
000260      CALL PLOT(XPLOT(KK),YPOPYC,3)
000261      500 CONTINUE
000262      YPOS=YPOS-HEIGHT-.20
000263      510 CONTINUE
000264      YINC=HEIGHT/NSPACE
000265      YSTART=YLEN+.375
000266      YMAX=YMINC+DELYCO
000267      YINC=DELYCO/NSPACE
000268      NTIC=NSPACE+1
000269      DO 620 I=1,NTIC
000270      YXPOS(I)=-.35
000271      620 CONTINUE
000272      YNUM(I)=YMAX
000273      IF (YMAX.LT.C.) YXPOS(I)=-.42
000274      DO 630 J=2,NTIC
000275      YNUM(J)=YNUM(J-1)+YINC
000276      IF (YNUM(J).LT.0.) YXPOS(J)=-.42
000277      630 CONTINUE
000278      DO 700 II=1,NCURVE
000279      CALL PLOT(0,YSTART,3)
000280      CALL SYMROL(0,YSTART,.1,13,90,.1,-2)

```

```

000450 CALL NUMBER(YNXPQS(1),YSTART,.07,YNUM(1),.5,.2)
000454 CALL PLOT(0.,YSTART,3)
000457 DO 65 J=2,NTIC
000461 YSTART=YSTART+YINCH
000463 CALL SYMROL(0.,YSTART,.1,13,90,.-2)
000467 CALL NUMBER(YNXPQS(JJ),YSTART,.07,YNUM(JJ),0.,2)
000474 CALL PLOT(0.,YSTART,3)
000477 650 CONTINUE
000482 YSTART=YSTART+.20
000486 700 CONTINUE
000494 CALL PLOT(XPLOT(NPOINT),YSTART,3)
000496 DO 800 K=1,NPOINT
000501 KK=NPI-K
000505 CALL SYMROL(XPLOT(KK),YSTART,.07,13,.-2)
000509 CALL PLOT(XPLOT(KK),YSTART,3)
000513 800 CONTINUE
000517 YSTART=YSTART+.07
000521 YNORM=XNUM(NPOINT)
000524 XLAST=XPLOT(1)
000527 XPNT=XPLOT(NPOINT)
000531 DO 900 J=1,NPOINT
000533 XNUM=XNORM-XNUM(J)
000536 XTEST=XPLOT(J)
000540 IF (J.EQ.1.OR.J.EQ.NPOINT) GO TO 880
000542 IF (XTEST-XLAST.LT.0.0)OR(XPNT-XTEST.LT.0.0) GO TO 900
000546 880 CALL NUMBER(YTEST-.035,YSTART,.07,XNUM,270,.)
000550 XLAST=XTEST
000554 900 CONTINUE
000557 CALL SYMROL(XLEN+.1,YSTART,.07,15,HDISTANCE INSIDE,0.,15)
000561 CALL SYMROL(XLEN+.275,YSTART-.105,.07,10,HEIGHT CONE,0.,10)
000564 RETURN
000567 END
000571
000573
000576
000579
000582
000585
000588
000591
000594
000597
000600
000603
000606
000609
000612
000615

```

```

SUBROUTINE LTRONE
COMMON X(200),Y(200),TIME(10),XNUM(1200),OUTAPP(21,7),IBUF(1024),
1 HEADERR(4),NCURVE,NPOINT,YLEN,XLEN,LAB,DT,LUN,LABTYP,XLAB,COLVAR,
2 KONTYP,NEL,NPTS(10),ICUL
COMMON/KOEF/ YMIN,DELY,YMINCO,BELCO,NSPACE,IP,HEIGHT
COMMON/XYPLOT/ XPLOT(200),YPLOT(200),YNUM(6),YNXPAS(6),XCURV(200)
DATA C/3.0E8/

DTF=DT/20.
CVAR=CONVAR+DTF
PVAR=P.
100 IF(CVAR,GE.1.) GO TO 150
CVAR=CVAR+.10.
PVAR=PVAR+.1.
GO TO 100
150 IF(CVAR,LT.10.) GO TO 175
CVAR=CVAR/10.
PVAR=PVAR/1.
GO TO 150
175 CALL LINES(0,4,0,-2)
DELX=(X(NPOINT)-X(1))/XLEN
DO 200 J=1,NPOINT
XPLOT(J)=(X(J)-XMIN)/DELX
CONTINUE
DELX=DELX/XLEN
NPI=NPOINT+1
HEIGHT=YLEN/NCURV. *.25
XL2=XLEN/2.
YPOS=YLEN*.375-HEIGHT
YS=TART=YLEN*.375
IS=0
CALL SYMROL(XL2-.75,1-.75,15,LAR,0,10)
CALL SYMROL(XL2-1,10,6,1,6H(IN 10,2,6)
CALL SYMROL(XL2-0.4,1,65,07,3HEXP,2,3)
CALL SYMROL(XL2-0.1,1,6,0,1,LUN,0,12)
CALL SYMROL(XL2+.9,1,6,1,1,0,1)
CALL SYMROL(XLEN-1,28,10,6,07,18HALL CURVES ARE FOR,0,18)
CALL GREEK(XLEN+.1,10,6,1,8,0,0)
CALL SYMROL(XLEN+.2,1,5,1,1H,0,1)
CALL NUMER(XLEN+.3,1,6,0,1,CVAR,0,2)
CALL SYMROL(XLEN+.7,1,6,1,3H10,0,3)
CALL NUMER(XLEN+1,1,65,07,PVAR,0,1)
DO 250 J=1,NPOINT
YPLOT(J)=Y(YS+J)
CONTINUE
CALL SCALE(YPLOT,HEIGHT,NPOINT,1)
YMIN=YPLOT(NPOINT+1)
DELY=YPLOT(NPOINT+2)
YMAX=HEIGHT+DELY+YMIN
CALL CDEF
POM=IP
YSHIFT=YMIN/DELY
IF(YMIN,GE.0.) YSHIFT=0.0
IF(YMAX,LE.0.) YSHIFT=HEIGHT
CT=C+TIME(1)

```

```

000221 L=0
000222 DO 300 J=1,NPOINT
000223 IF (XNUM(J).GT.CT) GO TO 350
000224 L=L+1
000225 300 CONTINUE
000226 350 NPL=NPOINT-L
000227 XADJ=XMIN-CT-X(L)
000228 J1=NPL+1
000229 NPJ=NPOINT+J1
000230 NPJL=NPJ-NPL
000231 DO 400 J=J1,NPOINT
000232 TJ=NPJ-J
000233 TJJ=NPJL-J
000234 XCURV(J)=(X(J)-XADJ)/DELX
000235 IF (XCURV(J).LT.0.) XCURV(J)=0.
000236 YPLOT(J)=(YPLT(IJL)-YMIN)/DELY*YPOS
000237 400 CONTINUE
000238 CALL LINES(XCURV(J1),YPLT(J1),0)
000239 CALL LINES(XCURV(J1),YPLT(J1),L)
000240 YPOPS=YPOS*YSHIFT
000241 PTIME=0.0
000242 AMPTIME=TIME(I1)*DTF
000243 IF (AMPTIME.GE.1.) GO TO 450
000244 AMPTIME=AMPTIME*10.
000245 PTIME=PTIME-1.
000246 GO TO 420
000247 450 IF (AMPTIME.LT.10.) GO TO 460
000248 AMPTIME=AMPTIME/10.
000249 PTIME=PTIME+1.
000250 GO TO 450
000251 460 VLAB=YPOPS+.25
000252 CALL SYMROL(XLEN+.1,VLAB,.1,LABTY,0,.2)
000253 CALL NUMRER(XLEN+.3,VLAB,.1,AMPTIME,0,.2)
000254 CALL SYMROL(XLEN+.7,VLAB,.1,3M*10,0,.3)
000255 CALL NUMRER(XLEN+.1,YPOPS+.075,.07,PTIME,0,.1)
000256 CALL SYMROL(XLEN+.35,YPOPS-.125,.1,4EXP,0,.4)
000257 CALL NUMRER(XLEN+.75,YPOPS-.125,.1,POMIN,0,-1)
000258 CALL PLOT(XPLT(NPOINT),YPOPS,3)
000259 DO 500 K=1,NPOINT
000260 KK=NP1-K
000261 CALL SYMROL(XPLT(KK),YPOPS,.07,13,0,-2)
000262 CALL PLOT(XPLT(KK),YPOPS,3)
000263 500 CONTINUE
000264 YPOS=YPOS-HEIGHT-.20
000265 VINC=HEIGHT/NSPACE
000266 YMAX=YMINCO+DFLYCO
000267 YINC=DFLYCO/NSPACE
000268 NTIC=NSPACE+1
000269 DO 620 I=1,NTIC
000270 YXPOS(I)=.35
000271 620 CONTINUE
000272 YNUM(I)=YMAX
000273 IF (YMAX.LT.0.) YXPOS(I)=-.42
000274 DO 630 J=2,NTIC
000275 YNUM(I)=YNUM(J-1)-YINC
000276 IF (YNUM(I).LT.0.) YXPOS(J)=-.42
000277 630 CONTINUE
000278 CALL PLOT(0.,YSTART,3)
000279 000400
000401
000402
000403
000404
000405
000406
000407
000408
000409
000410
000411
000412
000413
000414
000415
000416
000417
000418
000419
000420
000421
000422
000423
000424
000425
000426
000427
000428
000429
000430
000431
000432
000433
000434
000435
000436
000437
000438
000439
000440
000441
000442
000443
000444
000445
000446
000447
000448
000449
000450
000451
000452
000453

```

```

000445 CALL SYMBOL(0.,YSTART,.1,13,90,-2)
000446 CALL NUMER(YNXPOS(1),YSTART,.07,XNUM(1),.5,.2)
000447 DO 650 JJ=2,NTIC
000448 YSTART=YSTART+YINCH
000449 CALL SYMBOL(0.,YSTART,.1,13,90,-2)
000450 CALL NUMER(YNXPOS(JJ),YSTART,.07,XNUM(JJ),.5,.2)
000451 CALL PLOT(0.,YSTART,3)
000452 CONTINUE
000453 YSTART=YSTART+.25
000454 IS=IS+NPOINT
000455 IF IS=NPOINT
000456 700 CONTINUE
000457 CALL PLOT(XPLOT(NPOINT),YSTART,3)
000458 DO 800 K=1,NPOINT
000459 KK=NPI-K
000460 CALL SYMBOL(XPLOT(KK),YSTART,.07,13,.5,-2)
000461 CALL PLOT(XPLOT(KK),YSTART,3)
000462 CONTINUE
000463 YSTART=YSTART+.07
000464 XNUM=XNUM(NPOINT)
000465 XLAST=XPLOT(1)
000466 XPNT=XPLOT(NPOINT)
000467 DO 900 J=1,NPOINT
000468 XNUM=XNUM-XNUM(J)
000469 XTEST=XPLOT(J)
000470 IF(J.EQ.1.OR.J.EQ.NPOINT) GO TO 890
000471 IF(XTEST-XLAST.LT.0.05.OR.XPNT-XTEST.LT.5) GO TO 900
000472 800 CALL NUMER(IXTEST=.035,YSTART,.07,XNUM,270,.1)
000473 XLAST=XTEST
000474 CONTINUE
000475 CALL SYMBOL(XLEN,.1,YSTART,.07,150,150,INCL,0.15)
000476 CALL SYMBOL(XLEN,.275,YSTART,.105,.07,100,100,CON,0.15)
000477 RETURN
000478 END

```

```

000002 SUBROUTINE TMAXS
000002 COMMON X(200),Y(200),Z(10),XNUM(200),OUTADR(21,7),IBUF(1024),
000002 1 HEADPR(a),NCURVE,NPOINT,CYCLE,XLEN,LAB*DT,LUN,LABTYP,XLAB,CONVAR,
000002 2 KONTYP,DEL,NPTS(10),FCUL
000002 COMMON/KDEF/ YMIN,DELY,YMINCO,DELYCO,NSPACE,IP,HEIGHT
000002 COMMON/XYPLOT/ XPLT(200),YPLT(200),YNUM(6),VNXPOS(6),XCURV(200)
000002
000002 I,TF=DT/20,
000002 CVAR=CONVAR*DTF
000002 PVAR=..
000006 I10 IF(CVAP,GE,1.) GO TO 150
000007 CVAR=CVAR*10.
000012 PVAR=PVAR-1.
000013 GO TO 100
000015 I50 IF(CVAP,LT,10.) GO TO 175
000015 CVAR=CVAR/10.
000021 PVAR=PVAR+1.
000023 GO TO 150
000024 I75 CALL LINES(0,6,0,-2)
000027 DFLX=(X(NPOINT)-X(1))/XLEN
000033 XMIN=X(1)
000034 DO 200 J=1,NPOINT
000035 XPLT(J)=(X(J)-XMIN)/DELX
000040 CONTINUE
000042 NCP=NCURVE*NPOINT
000044 HFIGT=YLEN/NCURVE -.25
000047 YMAX=Y(1)
000051 YMIN=Y(1)
000052 DO 250 I=1,NCURVE
000053 NPT=NPTS(I)
000055 ICT=(I-1)*NPOINT
000060 DO 260 J=1,NPT
000062 ICT=ICT+1
000064 YPLT(J)=Y(ICT)
000066 CONTINUE
000070 CALL SCALE(YPLT,HEIGHT,NPT,1)
000073 YTEST=YPLT(NPT*2)*HEIGHT*YPLT(NPT+1)
000077 IF(YMAX,LT,YTEST) YMAX=YTEST
000102 IF(YMIN,GT,YPLT(NPT+1)) YMIN=YPLT(NPT+1)
000106 CONTINUE
000111 DFLY=(YMAX-YMIN)/HEIGHT
000114 CALL COEF
000115 POK=IP
000117 XL2=XLEN/2.
000127 CALL SYMROL(XL2-.75,1,.75,.15,LAB,0,.10)
000135 CALL SYMROL(XL2-1.,10,6,1,6H(IN 10,.,6)
000143 CALL NUMBER(XL2-0.4,1,.65,.07,POW,0,.1)
000151 CALL SYMROL(XL2-0.1,1,.6,1,LUN,0,.1)
000157 CALL SYMROL(XL2-0.9,1,.6,1,1H,0,.1)
000167 IF(KONTYP,EO,2H ) GO TO 260
000175 CALL SYMROL(XLEN-1.23,10,6,.07,18HALL CUVES ARE FOR,0,.18)
000183 CALL SYMROL(XLEN+.1,1,.6,1,KONTYP,0,.2)
000191 GO TO 270
000197 CALL SYMROL(XLEN-1.28,10,6,.07,18HALL CURVES ARE FOR,0,.18)
000204 CALL GREFK(XLFN+.1,10,6,1,18,0,.0)
000212 CALL SYMROL(XLEN+.2,1,.6,1,1H,0,.1)
000220 CALL NUMBER(XLEN+.3,1,.6,1,CVAR,0,.2)

```

```

000226 CALL SYMROL(XLEN+.7,1.0,6.1,3H*10.0,0.3)
000227 CALL NUMRER(XLEN+1,1.65,0.07,PVAR,0.0,-1)
000228 YSHIFT=YMIN/DELY
000229 IF(YMIN,GE,0.0) YSHIFT=0.0
000230 IF(YMAX,LE,0.0) YSHIFT=HEIGHT
000231 DO 300 I=1,NCP
000232 Y(I)=(Y(I)-YMIN)/DELY
000233 300 CONTINUE
000234 ICT=J
000235 YPOS=YLEN+.375-HFIGHT
000236 NPI=NPOINT+1
000237 DO 600 I=1,NCURVE
000238 ICT=(I-1)*NPOINT
000239 NPT=NPTS(I)
000240 DO 400 J=1,NPT
000241 ICT=ICT+1
000242 YPLOT(J)=Y(ICT)*YPOS
000243 400 CONTINUE
000244 CALL LINES(XPLOT(I),YPLOT(I),0)
000245 CALL LINES(XPLOT,I,YPLOT,NPT)
000246 YPOPS=YPOS+YSHIFT
000247 PZ=0.0
000248 AMPZ=Z(I)*DTF
000249 420 IF(AMPZ,GE,1.0) GO TO 450
000250 AMPZ=AMPZ*10.0
000251 PZ=PZ+1.0
000252 GO TO 420
000253 450 IF(AMPZ,LT,1.0) GO TO 460
000254 AMPZ=AMPZ/10.0
000255 PZ=PZ+1.0
000256 GO TO 450
000257 460 YLAB=YPOPS+.25
000258 IF(LATYP,EG,2H) GO TO 470
000259 CALL SYMROL(XLEN+.1,YLAB+.1,LABTYP,0.0,2)
000260 GO TO 480
000261 470 CALL GREFK(XLEN+.1,YLAB+.1,8.0,0.0)
000262 CALL SYMROL(XLEN+.2,YLAB+.1,1H=0.0,1)
000263 CALL NUMRER(XLEN+.3,YLAB+.1,AMPZ,0.0,2)
000264 CALL SYMROL(XLEN+.7,YLAB+.1,3H*10.0,0.3)
000265 CALL NUMRER(XLEN+1,YPOPS+.075,0.07,PZ,0.0,-1)
000266 CALL PLOT(XPLOT(NPOINT),YPOPS,3)
000267 DO 500 K=1,NPOINT
000268 KK=NPI-K
000269 CALL SYMROL(XPLOT(KK),YPOPS,0.07,13,5,-2)
000270 CALL PLOT(XPLOT(KK),YPOPS,3)
000271 500 CONTINUE
000272 YPOS=YPOS-HEIGHT-.20
000273 600 CONTINUE
000274 YINCH=HEIGHT/NSPACE
000275 YSTART=YLEN+.375
000276 YMAX=YMINCO+DFLYCO
000277 YINC=DELYCO/NSPACE
000278 NTIC=NSPACE+1
000279 DO 620 I=1,NTIC
000280 YNPOS(I)=YSTART
000281 620 CONTINUE
000282 YNLM(I)=YMAX
000283 IF(YMAX,LT,0.0) YNXPPOS(I)=-.42

```

```

000442      DO 63 J=2,NTIC
000443      YNUM(J)=YNUM(J-1)+YINC
000444      IF(YNUM(J).LT.0.0) YNXPUS(J)=-.42
000445      CONTINUE
000446      DO 700 II=1,NCURVE
000447      CALL PLOT(0,YSTART,3)
000448      CALL SYMOL(0,YSTART,.1,13,90,.2)
000449      CALL NUMBER(YNXPUS(1),YSTART,.07,YNUM(1),.5,.2)
000450      CALL PLOT(0,YSTART,3)
000451      DO 650 JJ=2,NTIC
000452      YSTART=YSTART+YINC
000453      CALL SYMOL(0,YSTART,.1,13,90,.2)
000454      CALL NUMBER(YNXPUS(JJ),YSTART,.07,YNUM(JJ),.5,.2)
000455      CALL PLOT(0,YSTART,3)
000456      CONTINUE
000457      YSTART=YSTART+.20
000458      CONTINUE
000459      CALL PLOT(XPLOT(NPOINT),YSTART,3)
000460      KK=NP1-K
000461      DO 800 K=1,NPOINT
000462      CALL SYMOL(XPLOT(KK),YSTART,.07,13,.5,.2)
000463      CALL PLOT(XPLOT(KK),YSTART,3)
000464      CONTINUE
000465      YSTART=YSTART+.07
000466      XLAST=XPLOT(1)
000467      XPNT=XPLOT(NPOINT)
000468      DO 900 J=1,NPOINT
000469      XNUM=XNUM(J)+1.E9
000470      XTEST=XPLOT(J)
000471      IF(J.EQ.1.OR.J.EQ.NPOINT) GO TO 880
000472      IF(XTEST-XLAST.LT.0.03.UR.XPNT-XTEST.LT.0.09) GO TO 900
000473      CALL NUMBER(XTEST-.035,YSTART,.07,XNUM,270.0)
000474      XLAST=XTEST
000475      CONTINUE
000476      IF(TCOL.EQ.0.) GO TO 1000
000477      CALL SYMOL(XLEN+.1,YSTART,.07,9HT IN NSFC,0.09)
000478      RETURN
000479      CALL GREFK(XLEN+.1,YSTART,.1,19,0.0)
000480      CALL SYMOL(XLEN+.3,YSTART,.07,THIN NSEC,.5,.7)
000481      RETURN
000482      END

```

```

SUBROUTINE TYMAX
COMMON X(200),Y(200),Z(10),XNUM(200),OUTAHR(21,7),IRUF(1024),
1 HEADER(4),NCURVE,NPOINT,YLEN,XLEN,LAB,DT,LUN,LAB,YP,XLAB,CONVAR,
2 KONTYP,DEL,NPTS(10),TCUL
COMMON/KOEF/ YMIN,DELY,YMINCO,DELYCO,NSPACE,IP,HEIGHT
COMMON/XPLOT/ XPLOT(200),YPLT(200),YNUM(6),YXPOS(6),XCURV(200)

DTF=DT/20,
CVAR=CONVAR+DTF
DVAR=.
100 IF(CVAR.GE.1.) GO TO 150
CVAR=CVAR+10,
PVAR=PVAR-1,
GO TO 100
150 IF(CVAR.LT.1.) GO TO 175
CVAR=CVAR/10,
PVAR=PVAR+1,
GO TO 150
175 CALL LINES(0,4,2,-2)
DELX=(X(NPOINT)-X(1))/XLEN
YMIN=X(1)
DO 200 J=1,NPOINT
XPLOT(J)=(X(J)-XMIN)/DELX
200 CONTINUE
NPT=NPOINT+1
HEIGHT=YLEN/NCURVE+.5
XL2=XLEN/2,
VDJS=XLEN+.375*HEIGHT
VSTAK=XLEN+.375
CALL SYMROL(XL2-.75*.15*LAB,0,.10)
CALL SYMROL(XL2-.10*.6*.1*HIN,0,.06)
CALL SYMROL(XL2-.4*.1*.85*.07*3HEXP,5,.03)
CALL SYMROL(XL2-.1*.1*.6*.1*LUN,0,.15)
CALL SYMROL(XL2+.9*.1*.85*.1*H,0,.1)
IF(KONTYP.EQ.2H) GO TO 220
CALL SYMROL(XLEN-1.23*.1*.6*.07,18HALL CURVES ARE FOR,0,.18)
CALL SYMROL(XLEN+.1*.1*.85*.1*KONTYP,0,.12)
GO TO 240
220 CALL SYMROL(XLEN-.128*.10*.6*.07,18HALL CUOVES ARE FOR,0,.13)
CALL GREK(XLEN+.1*.10*.6*.1*8,0,.1)
CALL SYMROL(XLEN+.2*.1*.85*.1*H=0,.1)
240 CALL NUMBER(XLEN+.3*.1*.85*.1*CVAR,0,.12)
CALL SYMROL(XLEN+.7*.1*.85*.1*3H*1.0,0,.3)
CALL NUMBER(XLEN+.1*.1*.85*.07*PVAR,0,.1)
DO 700 IT=1,NCURVE
NPT=NPTS(IT)
ICT=(IT-1)*NPOINT
DO 250 J=1,NPT
YPLT(J)=Y(ICT+J)
250 CONTINUE
CALL SCALE(YPLT,HEIGHT,NPT,1)
YMIN=YPLT(NPT+1)
DELY=YPLT(NPT+2)
YMAX=HEIGHT+DELY+YMIN
CALL COEF
DOW=IP
YSHIFT=YMIN/DELY

```

```

000232 IF(YMIN-GE,0.) YSHIFT=0.0
000233 IF(YMAX-LE,0.) YSHIFT=HEIGHT
000234 DO 400 J=1,NPT
000240 YPLOT(J)=(YPL0T(J)-YMIN)/DELY *YPOS
000242 400 CONTINUE
000246 CALL LINES(XPLOT(1),YPL0T(1),0)
000251 CALL LINES(XPLOT,YPL0T,NPT)
000253 YPOPS=YPOS*YSHIFT
000256 P7=0.0
000260 AMPZ=Z(I1)+D7F
000261 IF(AMPZ-GE,1.) GO TO 450
000264 AMPZ=AMP7*10.
000267 P7=PZ-1.
000270 GO TO 420
000272 450 IF(AMPZ-LT,10.) GO TO 450
000275 AMPZ=AMPZ/10.
000276 PZ=PZ+1.
000280 GO TO 450
000281 YLAB=YPOPS*.225
000283 IF(LATYP.EQ,0.) GO TO 470
000285 CALL SYMROL(XLEN,.1,YLAB,.1,LABTYP,0.02)
000287 GO TO 480
000290 470 CALL GREK(XLEN,.1,YLAB,.1,0.0)
000293 CALL SYMROL(XLEN,.2,YLAB,.1,1H=0.0,1)
000296 CALL NUMBER(XLEN,.3,YLAB,.1,AMPZ,0.02)
000299 CALL SYMROL(XLEN,.7,YLAB,.1,3H=10.0,3)
000302 CALL NUMBER(XLEN,.1,YPOPS*.075,.07,PZ,0.0-1)
000305 CALL SYMROL(XLEN,.35,YPOPS-.125,.1,4HEXP=0.4)
000308 CALL NUMBER(XLEN,.75,YPOPS-.125,.1,POW,0.0-1)
000311 CALL PLOT(XPLOT(NPOINT),YPOPS,3)
000314 DO 500 K=1,NPOINT
000317 KK=NPI-K
000320 CALL SYMROL(XPLOT(KK),YPOPS,.07,13,0.0-2)
000323 CALL PLOT(XPLOT(KK),YPOPS,3)
000326 500 CONTINUE
000329 YPOS=YPOS-HEIGHT-.20
000332 YINCH=HEIGHT/NSPACE
000335 YMAX=YMINCO*DFLYCO
000338 YINC=DFLYCO/NSPACE
000341 NTIC=NSPACE*1
000344 DO 620 I=1,NTIC
000347 YNAPOS(I)=-.35
000350 620 CONTINUE
000353 YNUM(I)=YMAX
000356 IF(YMAX-LT,0.0) YNXPUS(I)=-.42
000359 DO 630 J=2,NTIC
000362 YNUM(J)=YNUM(J-1)-YINC
000365 IF(YNUM(J).LT,0.0) YNXPUS(J)=-.42
000368 630 CONTINUE
000371 CALL PLOT(C,YSTART,3)
000374 CALL SYMROL(C,YSTART,.1,13,0.0-2)
000377 CAL' NUMBER(YNXPUS(1),YSTART,.07,YNUM(1),0.02)
000380 CALL PLOT(C,YSTART,3)
000383 DO 650 J=2,NTIC
000386 YSTART=YSTART-YINCH
000389 CALL SYMROL(C,YSTART,.1,13,0.0-2)
000392 CALL NUMA( YNXPUS(J),YSTART,.07,YNUM(J),0.02)
000395 CALL PLOT(C,YSTART,3)

```

```

000511 650 CONTINUE
000514 700 YSTART=YSTART+.25
000516 700 CONTINUE
000520 CALL PLOT(XPLOT(NPOINT),YSTART+3)
000523 DO 800 K=1,NPOINT
000525 KK=NP1-K
000527 CALL SYMBOL(XPLOT(KK),YSTART+.07,i3,*,*-2)
000533 CALL PLOT(XPLOT(KK),YSTART+3)
000536 800 CONTINUE
000541 YSTART=YSTART+.07
000543 XLAST=XPLOT(1)
000544 XPNT=XPLOT(NPOINT)
000545 DO 900 J=1,NPOINT
000547 XNUM=XNUM(J)+1.E9
000551 XTEST=XPLOT(J)
000553 IF (J.EQ.1.OR.J.EQ.NPOINT) GO TO 880
000555 IF (XTEST-XLAST.LT.C.09)GR.XPNT=XTEST.LT.i.09) GO TO 900
000557 880 CALL NUMBER(XTEST-.035,YSTART+.07,XNUMH,270,i1)
000559 XLAST=XTEST
000562 900 CONTINUE
000564 IF (TCOL.EQ.0.) GO TO 1000
000567 CALL SYMBOL(XLEN+.1,YSTART+.07,9HT IN NSFC+0.05)
000570 RETURN
000573 1000 CALL GREFK(XLPN+i,YSTART+.1,19.0.00)
000575 CALL SYMBOL(XLEN+.3,YSTART+.07,7HIN NSEC.i.0.7)
000577 RETURN
000579 END

```

```

000002 SUBROUTINE COFF
000002 COMMON/KREF/ YMIN,DELY,YMINC,DELYC,NRET,I,HEIGHT
000002 DIMENSION YAMP(10),NT(10)
000002 DATA YAMP/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0/
000002 DATA NT/5,5,5,4,5,4,5,4,5,5/
000002 I=0
000002 DELYC=DELY*HEIGHT
000002 100 IF(DELYC.LT.10.) GO TO 200
000002 DELYC=DELYC/10.
000002 I=I+1
000002 GO TO 100
000002 200 IF(DELYC.GE.1.) GO TO 300
000002 DELYC=DELYC*10.
000002 I=I-1
000002 GO TO 200
000002 300 YMINC=YMIN*10.0*(-1)
000002 DELYC=HEIGHT*DELY*10.0*(-1)
000002 DO 400 K=1,9
000002 IF(DELYC.LE.YAMP(K)) GO TO 500
000002 400 CONTINUE
000002 K=10
000002 500 NRET=NT(K)
000002 RETURN
000002 END

```


000125	115	YA=XB
000126		SX=AMINI(S,-Z,ZLEFT+ZEND(IJ))
000127		IF(SX.GT.0.0) GO TO 120
000128		IJ=IJ+1
000129		IF(IJ.LE.JJ) GO TO 121
000130		IJ=1
000131		ZLEFT=Z
000132		JDOT1=0
000133		JDOT2=0
000134		GO TO 115
000135	120	CONTINUE
000136		XR=XA+SX*XFAC
000137		YB=YA+SY*YFAC
000138		JJ=JK(IJ)
000139		GO TO JJ*(130,140,150,160)
000140	130	CONTINUE
000141		S=S-SX
000142		ZLEFT=ZLEFT-SX
000143		IF(S.LE.0.0) GO TO 200
000144		GO TO 115
000145	140	CONTINUE
000146		CALL PLOT (XA,YA,3)
000147		CALL PLOT (XB,YB,2)
000148		GO TO 130
000149	150	CONTINUE
000150		IF(JDOT1.NE.0) GO TO 130
000151		CALL PLOT (XA,YA,3)
000152		CALL PLOT (XB,YB,2)
000153		JDOT1=1
000154		GO TO 130
000155	160	CONTINUE
000156		IF(JDOT2.NE.0) GO TO 130
000157		CALL PLOT (XA,YA,3)
000158		CALL PLOT (XB,YB,2)
000159		JDOT2=1
000160		GO TO 130
000161	200	CONTINUE
000162		RETURN
000163	210	CONTINUE
000164		DO 22, I=1,NPTS
000165	220	CALL PLOT (X(I),Y(I),2)
000166		CONTINUE
000167		RETURN
000168		END

*FOLLOWING VARIABLES EQUIVALENT BUT NOT REFERENCED
INTENS

```

000007 SUBROUTINE SCALE(ARRAY,AXLEN,NPTS,IMIX)
000007 DIMENSION ARRAY(1)
000007 AMAX=ARRAY(1)
000007 AMIN=AMAX
000011 DO 1 0 I=2,NPTS
000013 AMAX=AMAX1(AMAX,ARRAY(I))
000017 AMIN=AMIN1(AMIN,ARRAY(I))
000023 100 CONTINUE
000025 ARRAY(NPTS+1)=AMIN
000026 IF (AMAX.EQ.AMIN) GO TO 200
000030 ARRAY(NPTS+2)=(AMAX-AMIN)/AXLEN
000033 RETURN
000041 END
000042

```

SUBROUTINE GREEK(XPOS,YPOS,SCALE,LETTER,T,ETA,ICAP)

LETTER CODES

1 ALPHA	7 ETA	13 NU	19 TAU
2 BETA	8 THETA	14 XI	20 UPSILON
3 GAMMA	9 IOTA	15 OMICRON	21 PHI
4 DELTA	10 KAPPA	16 PI	22 CHI
5 EPSILON	11 LAMBDA	17 RHO	23 PSI
6 ZETA	12 MU	18 SIGMA	24 OMEGA

LOWER CASE -- ICAP=1 UPPER CASE -- ICAP=1

DIMENSION XYINCH(32),IPOINT(159),JPOINT(54),KPOINT(47),ISTART(24)
 EQUIVALENCE (IPOINT(57),JPOINT(1)),(IPOINT(113),KPOINT(1))
 DATA XYINCH/
 1 .03125,.0625,.09375,.125,.15625,.1875,.21875,.25,.28125,.3125,
 2 .34375,.375,.40625,.4375,.46875,.5,.53125,.5625,.59375,.625,
 3 .65625,.6875,.71875,.75,.78125,.8125,.84375,.875,.90625,.9375,
 4 .96875,1.0/
 DATA IPOINT/
 1 312523522146164213409,10400541024405100558,02620565106613661664
 2 21572344525412740304,71420000000000000000,00003670573107413778
 3 27722742374237122668,17931361056113211668,205722552353235021458
 4 164412444446415100004,253321711564146113528,134614521461126710728
 5 9573 3734770000000004,2333207615712711768,27731070206225523518
 6 2245262144006402434,90500560263056551678,21271266046401610548
 7 00511444443124121404,00155755000000000000,003737517520720778
 8 1577117507236500558,004602452046214417468,50400000000000000000
 9 03101172037267002468,04250711372167222728,24652462624000000000
 A 0310664017137506773,127715751771254426538,17461542124064063428
 B 1464053002040600000,0525034704430641144,12401441564400000000
 C 32404026725560005,013503770576221244,8,26421224040000000000
 D 432252004002120447,07461346164725222728,201220472146224644508
 E 027367046045203403,044012441752226023638,63670000000000000000
 F 137175127420720723,117505730371026600648,556201062103561534
 G 05041450444204320414,16405240000000000000,13271666206422613558
 H 23522462043164113404,104005410343014600545,00550161036405610678
 I 53670000000000000004,1127306402660165006438,006107260743222642404
 J 0000364047047712754,167521742372246724648/
 DATA JPOINT/
 1 22611451255275624014,26400000000000000000,8,253123701667136710678
 2 05661364016100550024,014403430541140134,8,16412043224623523558
 3 226126641566536700004,262703670260016500648,06212271232134114408
 4 44400000000000000004,002703670466048404,8,03500541040144017418
 5 21442351236026461673,14272066226524632668,285724542252205114508
 6 11505510352615400574,07670163045056611678,14611537504000000000
 7 03217347357217414,4042340244123334448,0310270036603635548
 8 7511150135016522664,2472571153751400004,03572640162005600560528
 9 1460342064144013424,14441457144154224,8,224025422746305230568
 A 27624646463670000000,00004400200147244,8,24003040061462546008
 B 372177247627427114,2766266324612160006,8,2102457265527527448
 C 26432441214004002378,24400000000000000000,8,263225770077023702408
 D 0404440000000000000,00001477344440000,8,27322770077023702408
 E 00274274522417638,17235755000000000000,8,003200772577004025408
 F 4345 000000000000000,0037047702370240000,8,04402500314327007778
 G 25373177226047600000,153723752557334673,428,315530502645234214408

```

H.1340664203450150000554,006201670473067513778,16770422045642025609
I.222255560000000000000000,103716771337,34010018,56400000000000000000
J.0037477623702490000000,044002142777253731778/
DATA KPOINT/
1 1223274025007140000000,0000004002001477244,8,24,00704000000000000000
2 00000440020002770378,03771540277732713,378,3040260072400000000000
3 00000440020002770378,0377260267724377,778,003700740036277627378
4 27746220656062021608,212221560003004000018,2741270367400000000000
5 16373275273306731628,315530502645234216408,1340064203450150000554
6 00520167047306751378,567700000000000000008,00000440020002770378
7 31777372740250071404,003721772478267427718,2765266224621572578
8 0237240000045000008,2732247700771760044,8,2640474500000000000000
9 00327177277307214378,144010060040000000008,00260072017504770778
A 1275147167521724778,2775307230661430144,8,1000404000000000000000
A 143220712367256426618,26562552350206461458,124506460350015300568
C 06161640367667112728,14720737177133713438,0700574000000000000000
D 003704770272667030008,244004000040020024778,2437707700000000000000
E 00270257046575312518,165121532465266734678,103720771437144010008
F 604000000000000000008,000501400440015504418,006502720676137714778
G 217625727265276126558,214026406745000000008/
DATA ISTART/185,687,1192,1494,1895,2198,25100,28103,32109,34111,
1 35114,37116,40119,43121,46125,53130,55132,59135,64137,67139,70143
2,75149,77152,80156/
LFTTR=LETTER+24*ICAP
SINTHE=SIN(THETA*.01745)
COSTHE=COS(THETA*.01745)
DO 120 J=1,24
JSTAR=ISTART(J)
IF(LETTER.EQ.J) GO TO 130
120 CONTINUE
130 IF(JSTAR.GT.10000) GO TO 140
JSTAR=JSTAR+100
IF(LETTR.GT.24) JSTAR=JSTAR-JSTAR*100
GO TO 150
140 JSTAR=JSTAR/1000
IF(LETTR.GT.24) JSTAR=JSTAR-JSTAR*1000
150 IFLAG=0
200 JPEN=2
CALL SEPARB(IPOINT(JSTAR),IX,IY,IPEN,IFLAG,LAST)
X=XYINCH(IX+1)*COSTHE-XYINCH(IY+1)*SINTHE
Y=XYINCH(IY+1)*COSTHE+XYINCH(IX+1)*SINTHE
IF(IPEN.EQ.0) JPEN=3
CALL PLOT(X*SCALE+XPOS,Y*SCALE+YPOS,JPEN)
IFLAG=1
IF(LAST.NE.1) GO TO 200
CALL PLOT(XPOS,YPOS,3)
RETURN
END

```

```

SUBROUTINE SEPAR3 (IU,IX,IY,IPEN,IFLAG,LAST)
  DIMENSION ID(1)
  LAST=I
  IF (IFLAG.EQ.0) LOC=12
  LOCW=(LOC-12)/6+1
  ID=ISHIFT(ID(LOCW),MOD(LOC,6))
  IY=IDD.AND.31
  IPEN=IDD.AND.32
  IDU=ISHIFT(IDU,-6)
  IX=IDD.AND.31
  LOC=LOC+12
  IDU=IDD.AND.32
  IF (IDD.NF.0) LAST=1
  RETURN
END

```

```

000011
000011
000011
000013
000020
000024
000036
000037
000042
000050
000051
000053
000055
000056

```

ISHIFT

1 1123111-62400000002
2 56110 56270 63320 22631
3 0400000001 + 45063
4

IDENT ISHIFT
ENTRY ISHIFT
VFD 36/01ISHIFT,24/2
ISHIFT
BSS 1
SA1 B1
SA2 B2
SB3 X2
LX6 B3,X1
EQ ISHIFT
END
STORAGE USED
6400 ASSEMBLY

10 STATEMENTS
0.137 SECONDS
1 SYMROLS
3 REFERENCES

B. BUDDHA

1. Introduction

Program BUDDHA was originally designed to process a maximum of two data tapes according to record number, to plot right and left hand axes on a linear scale, and to overlay curves.¹ The present version of BUDDHA has been expanded to process multiple tapes according to record name and/or record number and to plot right and left hand axes on a log scale.

2. Input Revisions

There are three additional data card types and card type B² has three new variables defined.

Type C is the tape card, type D is the record card, and type E is the log card. A data set consists of a tape card followed by as many record cards as specified on the tape card. When all the record cards have been read, another tape card is read and the cycle continues until a blank tape card is read. An axis card is then read which is followed by a log card and as many curve cards as specified on the axis card. When all the curve cards are read another axis card and a log card are read and the cycle continues until two blank cards are read which indicate all input data has been processed.

1. Jones, D. L. and D. H. Stump, ELECTRA, ORESTES, AND SUPPORTING GRAPHIC DISPLAY CODES, U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, February 1971, pp. 169-195.

2. Ibid, p. 169.

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENT	PROGRAM NAME	TYPE	DESCRIPTION
B	1-20	Not used	KARD(NS,3)		Same as report description
	21-30				Data tape number (1, 2,3, etc.)
	31-50				Same as report description
	51-54				Blank
	55-60		DATE	Alpha-numeric	Name of record to be plotted
	61-65		IRUN	Integer	Run number
	66-70		IALT	Integer	Altitude Number
	71-75		IZ(NS)	Integer	Light intensity
	76-80	Not used			Blank
C	1-10		ITAPE	Integer	Tape number (1, 2, 3, etc.)
	11-20		NDATES	Integer	Number of record names to be processed for a particular tape.
	21-80	Not used			Blank

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENT	PROGRAM NAME	TYPE	DESCRIPTION
D	1-6	Not used	TDATE(1, ITAPE)	Alpha-numeric	Record name on data tape, where 1 runs from 1 to NDATES.
	4-10		NRUNS	Integer	Blank
	11-20				Total number of runs for a particular record name
	21-30				Total number of altitudes for a particular record name
	31-40	Not used	MAXVAL	Integer	Number of maximum value records
	41-80				Blank
E	1-4	Not used	ILOGO (INC)	Alpha-numeric	LOGO indicates a log axis
	5-8		IPRT	Alpha-numeric	Blank
	9-10				Print control for debugging
	11-16	Not used	ILINE	Alpha-numeric	Blank
	17-20				Grid line indicator
	21-25		DECADE (INC,1)	Floating	Number of decades for left hand axis

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENT	PROGRAM NAME	TYPE	DESCRIPTION
	26-30		DECADE (INC,2)	Float- ing	Number of decades for right hand axis
	31-80	Not used			Blank

3. Data Revisions

In the original program after all data cards were processed, the curve cards were ordered according to record number and the random access files were opened. The appropriate data tape was read and data was written on the corresponding random access file.³ In the present version, a tape card and as many record cards as specified on the tape card are read. For each record card that is read, the total number of records and the starting block number for that particular record name are calculated. The total number of records and the starting block number are then used to calculate the appropriate record number that corresponds to the record name on that curve card. The data tape number on that curve card is multiplied by ten thousand and is then added to the record number. The curve cards are then ordered according to record number and the random access files are opened.

The reading of the data tape or tapes has one modification. Because multiples of ten thousand based on the tape number are added to the record number, all the records from a particular tape are processed and then a new tape is processed. Data from odd number tapes are stored on one random access file and data from even number tapes are stored on another random access file.

4. Plot Additions

Program BUDDHA has been expanded to plot ordinate axis data on a log scale. The data points are plotted on a positive log axis, a negative

3. Ibid., pp. 170-171

log axis, or on a split positive and negative log axis depending upon the behavior of the data. When the left hand axis array contains all positive data, the data points are scanned to determine the positive maximum and the positive minimum. If the number of cycles is defined by the user, the positive minimum is redefined as the difference between the positive maximum and the number cycles. If the number of cycles is not defined by the user, the number of cycles is set equal to the difference between the positive maximum and the positive minimum. The negative maximum, the negative minimum, and the number of cycles are determined in a similar manner when the data array contains all negative data. Both sets of variables are determined when the array contains positive and negative data. The process is identical for the right hand axis data if appropriate.

The length of the log axis is determined by the behavior of both the right and left hand axis data. The axes are 8.5 inches if the data arrays contain all positive or all negative data. If one array contains positive and negative data, the axes are split into a positive axis and a negative axis which are 4.18 inches long. The log axes are drawn using subroutine LOGPLT (see I.B.6). The time axis is located at the middle of the plot when the data is positive and negative and at the bottom when the data is either all positive or all negative. The time array is annotated with the minimum and maximum cutoff times as in the original program.⁴

4. Ibid., pp. 171-172

In order to begin plotting, the data array is scanned to determine the starting point and the end point of the positive and negative data for a particular curve. Any zeroes that are encountered are omitted. If the data oscillates positive and negative more than ten times, the variable is not plotted and message is printed. The logs of the data points are taken, converted into inches, and are plotted using subroutine LINES (see 1.B.5) until all curves are drawn. Any points that are less than the defined minimum are set equal to the minimum. The ordinate axis is labeled and a title is written at the bottom of the plot. The right hand axis is then labeled if appropriate. An enclosing box is drawn around the plot and the paper is advanced until all plots are made, after which the plot file is closed.

5. Subroutine LINES

Subroutine LINES is a plot routine designed for drawing data curves and annotating plots. Given two data arrays which are to be plotted against each other, subroutine LINES performs a curve fit to the data in a manner such that the curve can be represented by a series of equally spaced dots, line segments or both. Given the X-coordinate and the Y-coordinate, subroutine LINES draws the appropriate symbol at the bottom of the plot.

6. Subroutine LOGPLT

Subroutine LOGPLT is a special plot routine designed to draw and annotate positive and negative log axes. The log axis is annotated with tic marks at the power of ten and intermediate tic marks at 2, 4, 6, and 8

between the powers of ten. For negative data the power of ten is preceded by a negative sign. The tic marks are perpendicular to the axis line and are bisected by it. A positive angle, in degrees, produces a positive log axis and a negative angle produces a negative log axis. A switch, IFR, is used to determine a right or left hand axis. If IFR is not equal to one a left hand axis is drawn and if IFR is equal to one a right hand axis is drawn.

7. CALCOMP Subroutines

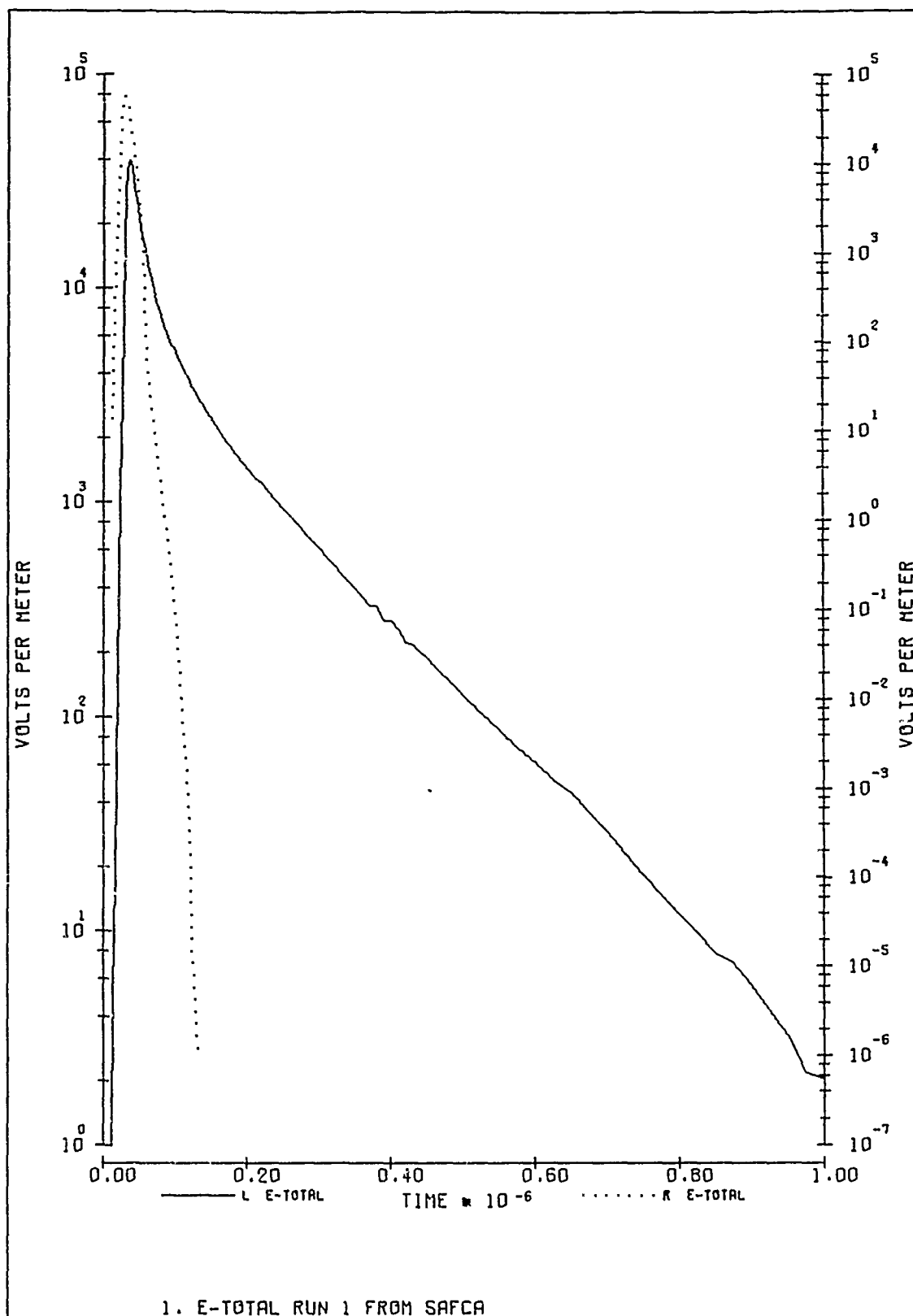
There are three subroutines from the CALCOMP plotting package that are called by subroutine LOGPLT. The subroutines are NUMBER, PLOT, and SYMBOL.

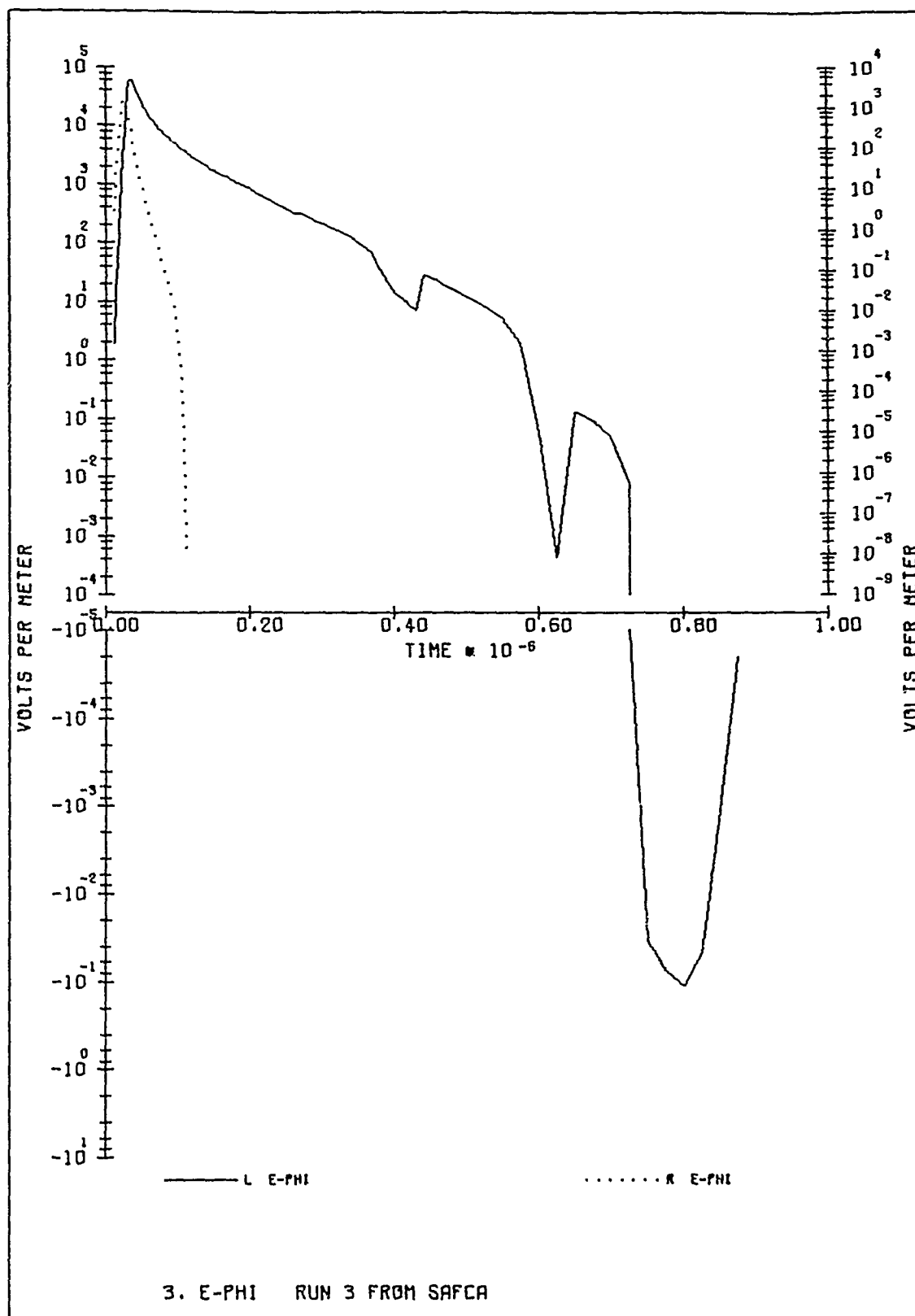
NUMBER - writes floating point numbers.

PLOT - converts pen movement from inches to actual plotter commands.

SYMBOL - writes alphanumeric text and special symbols at any angle desired.

8. Sample Output





9. Program Listing

```

PROGRAM RUDDHA (INPUT,OUTPUT,TAPE1,TAPE2,TAPE3=TAPE1,TAPE5,
1  TAPE51,TAPE52)
  DIMENSION NC( 20),AKND( 50,11),IPT1(50),IPT2(500),DATA(7000),
1  DATAR(1),DATAL(1),IIEH(2000),TIMEL(2000),IV(22),LABEL(22),
3  ITHR(1P24),IHEADR(5),IWINC( 20),IWX( 20),DATREC(400),
3  TIMREC(400),IZ( 50)
  DIMENSION WORX(400),WURKY(400),NPTS(11),ILOGO(20),DECADE(20,2),
1  *START(10),WRLOCK(10,10),DATE(10,1),*NSTART(10,10)
  DIMENSION LUNITS(2,22)
  DOUBLE PRECISION LUNITX(22)
  INTEGER PITELL

  DATA IV /10METH, 10HERA, 10HFEH, 10HEPH, 10METH,
1  10HFEV, 10HENO, 10HEFA, 10HEFF,
2  10MCDA, 10MCPH, 10MCTH, 10MRA,
3  10MWH, 10MWT, 10MCEL, 10MSUM,
4  10MSTG, 10MBRK, 10HFA, 10HEFA,
5  10HELM, 10HPR1, 10HPR2, 10HPR3, 10HPR4, 10HPR5, 10HPR6, 10HPR7, 10HPR8, 10HPR9, 10HPR10, 10HPR11, 10HPR12, 10HPR13, 10HPR14, 10HPR15, 10HPR16, 10HPR17, 10HPR18, 10HPR19, 10HPR20, 10HPR21, 10HPR22, 10HPR23, 10HPR24, 10HPR25, 10HPR26, 10HPR27, 10HPR28, 10HPR29, 10HPR30, 10HPR31, 10HPR32, 10HPR33, 10HPR34, 10HPR35, 10HPR36, 10HPR37, 10HPR38, 10HPR39, 10HPR40, 10HPR41, 10HPR42, 10HPR43, 10HPR44, 10HPR45, 10HPR46, 10HPR47, 10HPR48, 10HPR49, 10HPR50, 10HPR51, 10HPR52, 10HPR53, 10HPR54, 10HPR55, 10HPR56, 10HPR57, 10HPR58, 10HPR59, 10HPR60, 10HPR61, 10HPR62, 10HPR63, 10HPR64, 10HPR65, 10HPR66, 10HPR67, 10HPR68, 10HPR69, 10HPR70, 10HPR71, 10HPR72, 10HPR73, 10HPR74, 10HPR75, 10HPR76, 10HPR77, 10HPR78, 10HPR79, 10HPR80, 10HPR81, 10HPR82, 10HPR83, 10HPR84, 10HPR85, 10HPR86, 10HPR87, 10HPR88, 10HPR89, 10HPR90, 10HPR91, 10HPR92, 10HPR93, 10HPR94, 10HPR95, 10HPR96, 10HPR97, 10HPR98, 10HPR99, 10HPR100, 10HPR101, 10HPR102, 10HPR103, 10HPR104, 10HPR105, 10HPR106, 10HPR107, 10HPR108, 10HPR109, 10HPR110, 10HPR111, 10HPR112, 10HPR113, 10HPR114, 10HPR115, 10HPR116, 10HPR117, 10HPR118, 10HPR119, 10HPR120, 10HPR121, 10HPR122, 10HPR123, 10HPR124, 10HPR125, 10HPR126, 10HPR127, 10HPR128, 10HPR129, 10HPR130, 10HPR131, 10HPR132, 10HPR133, 10HPR134, 10HPR135, 10HPR136, 10HPR137, 10HPR138, 10HPR139, 10HPR140, 10HPR141, 10HPR142, 10HPR143, 10HPR144, 10HPR145, 10HPR146, 10HPR147, 10HPR148, 10HPR149, 10HPR150, 10HPR151, 10HPR152, 10HPR153, 10HPR154, 10HPR155, 10HPR156, 10HPR157, 10HPR158, 10HPR159, 10HPR160, 10HPR161, 10HPR162, 10HPR163, 10HPR164, 10HPR165, 10HPR166, 10HPR167, 10HPR168, 10HPR169, 10HPR170, 10HPR171, 10HPR172, 10HPR173, 10HPR174, 10HPR175, 10HPR176, 10HPR177, 10HPR178, 10HPR179, 10HPR180, 10HPR181, 10HPR182, 10HPR183, 10HPR184, 10HPR185, 10HPR186, 10HPR187, 10HPR188, 10HPR189, 10HPR190, 10HPR191, 10HPR192, 10HPR193, 10HPR194, 10HPR195, 10HPR196, 10HPR197, 10HPR198, 10HPR199, 10HPR200, 10HPR201, 10HPR202, 10HPR203, 10HPR204, 10HPR205, 10HPR206, 10HPR207, 10HPR208, 10HPR209, 10HPR210, 10HPR211, 10HPR212, 10HPR213, 10HPR214, 10HPR215, 10HPR216, 10HPR217, 10HPR218, 10HPR219, 10HPR220, 10HPR221, 10HPR222, 10HPR223, 10HPR224, 10HPR225, 10HPR226, 10HPR227, 10HPR228, 10HPR229, 10HPR230, 10HPR231, 10HPR232, 10HPR233, 10HPR234, 10HPR235, 10HPR236, 10HPR237, 10HPR238, 10HPR239, 10HPR240, 10HPR241, 10HPR242, 10HPR243, 10HPR244, 10HPR245, 10HPR246, 10HPR247, 10HPR248, 10HPR249, 10HPR250, 10HPR251, 10HPR252, 10HPR253, 10HPR254, 10HPR255, 10HPR256, 10HPR257, 10HPR258, 10HPR259, 10HPR260, 10HPR261, 10HPR262, 10HPR263, 10HPR264, 10HPR265, 10HPR266, 10HPR267, 10HPR268, 10HPR269, 10HPR270, 10HPR271, 10HPR272, 10HPR273, 10HPR274, 10HPR275, 10HPR276, 10HPR277, 10HPR278, 10HPR279, 10HPR280, 10HPR281, 10HPR282, 10HPR283, 10HPR284, 10HPR285, 10HPR286, 10HPR287, 10HPR288, 10HPR289, 10HPR290, 10HPR291, 10HPR292, 10HPR293, 10HPR294, 10HPR295, 10HPR296, 10HPR297, 10HPR298, 10HPR299, 10HPR300, 10HPR301, 10HPR302, 10HPR303, 10HPR304, 10HPR305, 10HPR306, 10HPR307, 10HPR308, 10HPR309, 10HPR310, 10HPR311, 10HPR312, 10HPR313, 10HPR314, 10HPR315, 10HPR316, 10HPR317, 10HPR318, 10HPR319, 10HPR320, 10HPR321, 10HPR322, 10HPR323, 10HPR324, 10HPR325, 10HPR326, 10HPR327, 10HPR328, 10HPR329, 10HPR330, 10HPR331, 10HPR332, 10HPR333, 10HPR334, 10HPR335, 10HPR336, 10HPR337, 10HPR338, 10HPR339, 10HPR340, 10HPR341, 10HPR342, 10HPR343, 10HPR344, 10HPR345, 10HPR346, 10HPR347, 10HPR348, 10HPR349, 10HPR350, 10HPR351, 10HPR352, 10HPR353, 10HPR354, 10HPR355, 10HPR356, 10HPR357, 10HPR358, 10HPR359, 10HPR360, 10HPR361, 10HPR362, 10HPR363, 10HPR364, 10HPR365, 10HPR366, 10HPR367, 10HPR368, 10HPR369, 10HPR370, 10HPR371, 10HPR372, 10HPR373, 10HPR374, 10HPR375, 10HPR376, 10HPR377, 10HPR378, 10HPR379, 10HPR380, 10HPR381, 10HPR382, 10HPR383, 10HPR384, 10HPR385, 10HPR386, 10HPR387, 10HPR388, 10HPR389, 10HPR390, 10HPR391, 10HPR392, 10HPR393, 10HPR394, 10HPR395, 10HPR396, 10HPR397, 10HPR398, 10HPR399, 10HPR400, 10HPR401, 10HPR402, 10HPR403, 10HPR404, 10HPR405, 10HPR406, 10HPR407, 10HPR408, 10HPR409, 10HPR410, 10HPR411, 10HPR412, 10HPR413, 10HPR414, 10HPR415, 10HPR416, 10HPR417, 10HPR418,
```

```

000003 NFRAME=1
000004 IFU=0
000005 INC=1
000006 NS=0
000007 IDA1=
000008 IDA2=
000009 PRINT 4
000010 NTAPES=0
000011
000012 READ 17, ITAPE, NDATES
000013 IF (ITAPE.LE.0) GO TO 110
000014 NSTART(1,ITAPE)=1
000015
000016 DO 1,5 I=1, NDATES
000017 READ 18, TDATE(I,ITAPE), NRUNS, MAXALT, MAXVAL, NGRND, NALT
000018 PRINT 19, TDATE(I,ITAPE), NRUNS, MAXALT, MAXVAL, NGRND, NALT
000019 NPEC=MAXALT*MAXVAL
000020 NBLOCK(I,ITAPE)=NREC
000021 IF (I.EQ.NDATES) GO TO 105
000022 NSTART(I+1,ITAPE)=NREC+NRUNS*NSTART(I,ITAPE)
000023 CONTINUE
000024 PRINT 17, (NBLOCK(I,ITAPE), NSTART(I,ITAPE), I=1, NDATES)
000025 GO TO 110
000026
000027 READ AXIS DATA CARO
000028
000029 I10 READ 1, (NC(INC,I), I=1,9), TMINC(INC), TMAXC(INC)
000030 READ 12, TLOGO(INC), IPT, ILINE, DECADE(INC,1), DECADE(INC,2)
000031 POINT 3, (NC(INC,I), I=1,9), TMINC(INC), TMAXC(INC)
000032 NCT=NC(INC,1)
000033 IF (NCT.LE.0) GO TO 150
000034
000035 READ CURVE DATA CARO
000036
000037 DO 130 I=1, NCT
000038 NS=NS+1
000039 READ 2, (KARD(NS,J), J=1,4), KARD(NS,10), DATE, IRUN, IALT, IZ(NS)
000040 ITAPE=KARD(NS,3)
000041 IF (IZ(NS).LE.0) IZ(NS)=16
000042 IF (NTAPES.LT.KARD(NS,3)) NTAPES=KARD(NS,3)
000043 IF (KARD(NS,2).GT.0) GO TO 113
000044 I1=1
000045 IF (DATE.EQ.TDATE(I,ITAPE)) GO TO 112
000046 I1=I1+1
000047 GO TO 111
000048 I12 KARD(NS,2)=NBLOCK(I,ITAPE)*(IRUN-1)+IALT*NSTART(I,ITAPE)-1
000049 I15 KARD(NS,2)=KARD(NS,2)+ITAPE*10000
000050 PRINT 11, (KARD(NS,J), J=1,4), DATE, IRUN, IALT, KARD(NS,10)
000051 KARD(NS,5)=NS
000052 KARD(NS,11)=INC
000053 DO 115 K=6,9
000054 KARD(NS,K)=0
000055 CONTINUE
000056 IMOD=MOD(KARD(NS,3)+2)
000057 IF (IMOD.EQ.0) GO TO 120
000058 IDA1=IDA1+1
000059 GO TO 130
000060 IDA2=IDA2+1
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112
000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170
000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185
000186
000187
000188
000189
000190
000191
000192
000193
000194
000195
000196
000197
000198
000199
000200
000201
000202
000203
000204
000205
000206
000207
000208
000209
000210
000211
000212
000213
000214
000215
000216
000217
000218
000219
000220
000221
000222
000223
000224
000225
000226
000227
000228
000229
000230
000231
000232
000233
000234
000235
000236
000237
000238
000239
000240
000241
000242
000243
000244
000245
000246
000247
000248
000249
000250
000251
000252
000253
000254
000255
000256
000257
000258
000259
000260
000261
000262
000263
000264
000265
000266
000267
000268
000269
000270
000271
000272
000273
000274
000275
000276
000277
000278
000279
000280
000281
000282
000283
000284
000285
000286
000287
000288
000289
000290
000291
000292
000293
000294
000295
000296
000297
000298
000299
000300
000301
000302
000303
000304
000305
000306
000307
000308
000309
000310
000311
000312
000313
000314
000315
000316
000317
000318
000319
000320
000321
000322
000323
000324
000325
000326
000327
000328
000329
000330
000331
000332
000333
000334
000335
000336
000337
000338
000339
000340
000341
000342
000343
000344
000345
000346
000347
000348
000349
000350
000351
000352
000353
000354
000355
000356
000357
000358
000359
000360
000361
000362
000363
000364
000365

```



```

001120 349 IF (IMAXEQ.EQ.0) GO TO 350
001121   DATREC(IMAX)=DATREC(IMAX)-DATREC(IMAXEQ)* (TMAX-TIMREC(IMAXEQ)) /
001122   I (TIMREC(IMAX)-TIMREC(IMAXEQ)) +DATREC(IMAXEQ)
001123   TIMREC(IMAX)=TMAX
001124   350 IF (KARD(JP,4).EQ.10ML ) GO TO 365
          COLLECT DATA FOR RIGHT HAND AXIS
          DO 365 J=IMIN,IMAX
            NPVR=NPVR+1
            DATAR(NPVR)=DATREC(J)
            TIMER(NPVR)=TIMREC(J)
          360 CONTINUE
            TIME=TIMER(NPVR)
            IFR=1
            GO TO 380
          365 DO 375 J=IMIN,IMAX
            NPVL=NPVL+1
            DATAL(NPVL)=DATREC(J)
            TIMEL(NPVL)=TIMREC(J)
          370 CONTINUE
            TIME=TIMEL(NPVL)
          380 IF (STM.LT.TIME) STM=TIME
            KARD(JP,8)=IMAX-IMIN+1
          400 CONTINUE
            IF (ILOGO(IM).EQ.LOGO) GO TO 700
          COLLECT DATA FOR LEFT HAND AXIS
          DO 375 J=IMIN,IMAX
            NPVL=NPVL+1
            DATAL(NPVL)=DATREC(J)
            TIMEL(NPVL)=TIMREC(J)
          370 CONTINUE
            TIME=TIMEL(NPVL)
          380 IF (STM.LT.TIME) STM=TIME
            KARD(JP,8)=IMAX-IMIN+1
          400 CONTINUE
            IF (ILOGO(IM).EQ.LOGO) GO TO 700
          FIND AND ADJUST MAX VALUES
          YMAGL=0.0
          DO 405 J=1,NPVL
            IF (ABS(DATAL(J)).GT.YMAGL) YMAGL=ABS(DATAL(J))
          405 CONTINUE
          CALL ADJUST(YMAGL,IPL,NL)
          YOL=(YMAGL*10.**IPL)/4.25
          IF (TMAX.EQ.0.) GO TO 407
          YMAG=TMAX
          CALL ADJUST(TMAG,IT,NT)
          DT=(TMAX-TMIN)/6.0
          TMAG=TMAG/(10.**IT)
          TMIN=TMIN
          TMINAX=TMIN/(10.**IT)
          GO TO 409
          407 TMAG=STM
          CALL ADJUST(TMAG,IT,NT)
          DT=(TMAG*10.**IT)/6.0
          T=0.
          TMINAX=T.
          409 IF (IFR.NE.1) GO TO 415
          YMAGR=Y.
          DO 41 J=1,NPVR
            IF (ABS(DATAR(J)).GT.YMAGR) YMAGR=ABS(DATAR(J))
          410 CONTINUE

```

```

001303 CALL ADJUST(YMAGR,IPR,NH)
001304 DOR=(YMAGR*10.*IPR)/.25
001305 IL=0
001306 IR=0
001307 LFFILL = 0
001308 RTTELL = 0
001309 IUNITL=1
001310 IUNITR=1
001311 DO 467 I=1,NCT
001312 ID=IP+I
001313 ISYMBL=ISYMBL+1
001314 ISM=KAPD(IP+9)
001315 JLABEL=KARO(IP,15)
001316 IF(KAPD(IP,4).EQ.10HL ) GO TO 43~
001317
001318
001319
001320
001321
001322
001323
001324
001325
001326
001327
001328
001329
001330
001331
001332
001333
001334
001335
001336
001337
001338
001339
001340
001341
001342
001343
001344
001345
001346
001347
001348
001349
001350
001351
001352
001353
001354
001355
001356
001357
001358
001359
001360
001361
001362
001363
001364
001365
001366
001367
001368
001369
001370
001371
001372
001373
001374
001375
001376
001377
001378
001379
001380
001381
001382
001383
001384
001385
001386
001387
001388
001389
001390
001391
001392
001393
001394
001395
001396
001397
001398
001399
001400
001401
001402
001403
001404
001405
001406
001407
001408
001409
001410
001411
001412
001413
001414
001415
001416
001417
001418
001419
001420
001421
001422
001423
001424
001425
001426
001427
001428
001429
001430
001431
001432
001433
001434
001435
001436
001437
001438
001439
001440
001441
001442
001443
001444
001445
001446
001447
001448
001449
001450
001451
001452
001453
001454
001455
001456
001457
001458
001459
001460
001461
001462
001463
001464
001465
001466
001467
001468
001469
001470
001471
001472
001473
001474
001475
001476
001477
001478
001479
001480
001481
001482
001483
001484
001485
001486
001487
001488
001489
001490
001491
001492
001493
001494
001495
001496
001497
001498
001499
001500
001501
001502
001503
001504
001505
001506
001507
001508
001509
001510
001511
001512
001513
001514
001515
001516
001517
001518
001519
001520
001521
001522
001523
001524
001525
001526
001527
001528
001529
001530
001531
001532
001533
001534
001535
001536
001537
001538
001539
001540
001541
001542
001543
001544
001545
001546
001547
001548
001549
001550
001551
001552
001553
001554
001555
001556
001557
001558
001559
001560
001561
001562
001563
001564
001565
001566
001567
001568
001569
001570
001571
001572
001573
001574
001575
001576
001577
001578
001579
001580
001581
001582
001583
001584
001585
001586
001587
001588
001589
001590
001591
001592
001593
001594
001595
001596
001597
001598
001599
001600
001601
001602
001603
001604
001605
001606
001607
001608
001609
001610
001611
001612
001613
001614
001615
001616
001617
001618
001619
001620
001621
001622
001623
001624
001625
001626
001627
001628
001629
001630
001631
001632
001633
001634
001635
001636
001637
001638
001639
001640
001641
001642
001643
001644
001645
001646
001647
001648
001649
001650
001651
001652
001653
001654
001655
001656
001657
001658
001659
001660
001661
001662
001663
001664
001665
001666
001667
001668
001669
001670
001671
001672
001673
001674
001675
001676
001677
001678
001679
001680
001681
001682
001683
001684
001685
001686
001687
001688
001689
001690
001691
001692
001693
001694
001695
001696
001697
001698
001699
001700
001701
001702
001703
001704
001705
001706
001707
001708
001709
001710
001711
001712
001713
001714
001715
001716
001717
001718
001719
001720
001721
001722
001723
001724
001725
001726
001727
001728
001729
001730
001731
001732
001733
001734
001735
001736
001737
001738
001739
001740
001741
001742
001743
001744
001745
001746
001747
001748
001749
001750
001751
001752
001753
001754
001755
001756
001757
001758
001759
001760
001761
001762
001763
001764
001765
001766
001767
001768
001769
001770
001771
001772
001773
001774
001775
001776
001777
001778
001779
001780
001781
001782
001783
001784
001785
001786
001787
001788
001789
001790
001791
001792
001793
001794
001795
001796
001797
001798
001799
001800
001801
001802
001803
001804
001805
001806
001807
001808
001809
001810
001811
001812
001813
001814
001815
001816
001817
001818
001819
001820
001821
001822
001823
001824
001825
001826
001827
001828
001829
001830
001831
001832
001833
001834
001835
001836
001837
001838
001839
001840
001841
001842
001843
001844
001845
001846
001847
001848
001849
001850
001851
001852
001853
001854
001855
001856
001857
001858
001859
001860
001861
001862
001863
001864
001865
001866
001867
001868
001869
001870
001871
001872
001873
001874
001875
001876
001877
001878
001879
001880
001881
001882
001883
001884
001885
001886
001887
001888
001889
001890
001891
001892
001893
001894
001895
001896
001897
001898
001899
001900
001901
001902
001903
001904
001905
001906
001907
001908
001909
001910
001911
001912
001913
001914
001915
001916
001917
001918
001919
001920
001921
001922
001923
001924
001925
001926
001927
001928
001929
001930
001931
001932
001933
001934
001935
001936
001937
001938
001939
001940
001941
001942
001943
001944
001945
001946
001947
001948
001949
001950
001951
001952
001953
001954
001955
001956
001957
001958
001959
001960
001961
001962
001963
001964
001965
001966
001967
001968
001969
001970
001971
001972
001973
001974
001975
001976
001977
001978
001979
001980
001981
001982
001983
001984
001985
001986
001987
001988
001989
001990
001991
001992
001993
001994
001995
001996
001997
001998
001999
002000

```

```

001502 CALL LINES (1,2,IZ(IP),-ISYMBL)
001506 CALL LINES (WORKT(1),-ORXY(1),0)
001511 CALL LINES (WORKT,WORKY,M)
001514 TEMPX=0.5+3.0*FLOAT(MOD(ISYMBL+1,2))
001524 TEMPY=-0.14*FLOAT((ISYMBL+1)/2)
001530 CALL LINES (TEMPX,TEMPY+0.035,0)
001534 TEMPX=TEMPX+0.6
001536 CALL LINES (TEMPX,TEMPY+0.035,1)
001543 CALL SYMBOL (TEMPX+0.07,TEMPY+0.07,1HL,0.0,0.1)
001551 CALL SYMBOL (TEMPX+0.21,TEMPY+0.07,LABEL(JS),0.0,0.10)
001560 CALL SYMBOL (TEMPX+0.98,TEMPY+0.07,JLABEL,0.0,0.10)
001566 IUNITL=ISM
001570 460 CONTINUE

1550 WRITE NEGATIVE AMPLITUDE VALUE ON LEFT HAND AXIS
EXP=IPL
Y=0
DY=4.25/NL
N=2*NL*1
IF(EXP.NF.0.) GO TO 500
CALL NUMBER(-0.7,-0.05,1,-YMAGL,0.0,2)
GO TO 520
500 CALL NUMBER(-0.7,0.05,1,-YMAGL,0.0,2)
IF(EXP.GT.0.0) GO TO 510
CALL SYMBOL(-0.6,-0.15,1,1H,0.0,1)
CALL NUMBER(-0.5,-0.15,1,EXP,0.0,-1)
CALL SYMBOL(-0.2,-0.15,1,1H,0.0,1)
GO TO 520
510 CALL SYMBOL(-0.6,-0.15,1,2H,0.0,2)
CALL NUMBER(-0.4,-0.15,1,EXP,0.0,-1)
CALL SYMBOL(-0.2,-0.15,1,1H,0.0,1)
520 CALL PLOT (0.0,0.3)

1560 DRAW LEFT HAND AXIS
DO 537 I=1,N
CALL SYMBOL(0.,Y,1,13,90.,-2)
CALL PLOT (0.,Y+DY)
Y=Y+DY
530 CONTINUE

1570 WRITE POSITIVE AMPLITUDE VALUE ON LEFT HAND AXIS
CALL SYMBOL(-0.2,2.3,25,1,LUNITS(1,IUNITL),90.0,20)
IF(EXP.NF.0.) GO TO 540
CALL NUMBER(-0.0,0.05,1,1,YMAGL,0.0,2)
GO TO 560
540 CALL NUMBER(-0.6,0.55,1,1,YMAGL,0.0,2)
IF(EXP.GT.0.0) GO TO 550
CALL SYMBOL(-0.6,0.35,1,1H,0.0,1)
CALL NUMBER(-0.5,0.35,1,EXP,0.0,-1)
CALL SYMBOL(-0.2,0.35,1,1H,0.0,1)
GO TO 560
550 CALL SYMBOL(-0.6,0.35,1,2H,0.0,2)
CALL NUMBER(-0.4,0.35,1,EXP,0.0,-1)
CALL SYMBOL(-0.2,0.35,1,1H,0.0,1)
560 CALL PLOT (0.0,2.25,3)

```

```

001752      DRAW AND LABEL TIME AXIS
001754      N=NT+1
001755      X=X*
001757      DX=6./NT
001760      EXP=IT
001764      TINC=(MAG-TMINAX)/NT
001765      DO 57 I=1,N
001772      T=TINC*(I-1)+TMINAX
001776      CALL SYMROL(X,4.25+.1,1,1,3,0,-2)
002004      CALL NUMMER(X-.1,4.1,1,1,7,0,0,2)
002007      CALL PLOT (X,4.25,3)
002011      X=X+DX
002014      570 CONTINUE
002020      CALL SYMROL(2.5+3.9+.1,9*TIME * 10+0.0,9)
002024      CALL NUMMER(3.45+3.95+.07,EXP,0.0,-1)
002026      IF(IFR,NF,1) GO TO 660

WRITE NEGATIVE AMPLITUDE VALUE ON RIGHT HAND AXIS

EXP=IPR
Y=Y*
NY=6.25/NR
N=2*NR+1
IF(EXP,NF,0.) GO TO 600
CALL NUMMER( 6.2,-0.05,.1,-YMAGR,0.0,2)
GO TO 620
600 CALL NUMMER( 6.2,0.05,.1,-YMAGR,0.0,2)
IF(EXP,GT,0.) GO TO 610
CALL SYMROL( 6.3,-0.15,.1,1H(,0.0,1)
CALL NUMMER( 6.4,-0.15,.1,EXP,0.0,-1)
CALL SYMROL( 6.7,-0.15,.1,1H(,0.0,1)
GO TO 620
610 CALL SYMROL( 6.3,-0.15,.1,2H(+,0.0,2)
CALL NUMMER( 6.5,-0.15,.1,EXP,0.0,-1)
CALL SYMROL( 6.7,-0.15,.1,1H(,0.0,1)
620 CALL PLOT (6.0,0,3)

DRAW RIGHT HAND AXIS

DO 630 I=1,N
CALL SYMROL(6.,Y,.1,1,3,90,-2)
CALL PLOT (6.,Y,3)
Y=Y+DY
630 CONTINUE

WRITE POSITIVE AMPLITUDE VALUE ON RIGHT HAND AXIS

CALL SYMROL(6.58,3.25+.1,LUNITS(1,IUNITR),90.0,20)
IF(EXP,NF,0.) GO TO 640
CALL NUMMER( 6.3,8.45 ,.1,YMAGR,0.0,2)
GO TO 660
640 CALL NUMMER( 6.3,8.55 ,.1,YMAGR,0.0,2)
IF(EXP,GT,0.) GO TO 650
CALL SYMROL( 6.3,8.35,.1,1H(,0.0,1)
CALL NUMMER( 6.4,8.35 ,.1,EXP,0.0,-1)
CALL SYMROL( 6.7,8.35 ,.1,1H(,0.0,1)

```

```

002165      GO TO 660
002166      CALL SYMROL( 6.3,8.35      ,1.2H(*.0.0.2)
002172      CALL NUMRER( 6.5,8.35      ,1.1EXP(0.0*-1)
002176      CALL SYMROL( 6.7,8.35      ,1.1H(*.0.0.1)
002202      CONTINUE
002202      CALL SYMROL(0.5,-1.1,1.1,HEADR,0.0,5)
002206      GO TO 190
002206
002207      IF (TMAX.EQ.0.0) GO TO 705
002210      TMAG=TMAG
002211      CALL ADJUST(TMAG,IT,NT)
002214      NT=(TMAX-TMIN)/6.0
002217      TMAG=TMAG/(10.**IT)
002224      TMIN=TMIN
002225      TMINAX=TMIN/10.**IT
002232      GO TO 710
002232
002232      TMAG=STM
002234      CALL ADJUST(TMAG,IT,NT)
002236      NT=(TMAG*10.**IT)/6.0
002244      TMINAX=0.
002245      TMIN=TMIN
002246      CONTINUE
002246      PDEC=DECADE(IM,1)
002246
002250      DETERMINE POSITIVE MAX AND POSITIVE MTN
002252      DO 800 J=1,NPVL
002254      IF (DATAL(J).LE.0.0) GO TO 800
002256      POSMAX=ALOG10(DATAL(J))
002261      GO TO 850
002261      CONTINUE
002264      IP0S=0
002265      GO TO 950
002265      IP0S=1
002266      DO 9 JJ=1,NPVL
002270      TEMPL=DATAL(JJ)
002272      IF (TEMPL.LE.0.0) GO TO 900
002273      TEMPL=ALOG10(TEMPL)
002276      IF (POSMTN.GT.EMPL) POSMIN=EMPL
002276      IF (POSMAX.LT.EMPL) POSMAX=EMPL
002304      CONTINUE
002304
002307      IF (PDEC.GT.0.0) GO TO 425
002311      IPMAX=POSMAX
002313      IPMIN=POSMIN
002315      IF (IPMAX.GE.2) IPMAX=IPMAX+1
002320      POSMAX=IPMAX
002321      IF (IPMIN.LT.2) IPMIN=IPMIN-1
002324      POSMIN=IPMIN
002326      PDEC=ABS(POSMAX-POSMIN)
002330      GO TO 950
002330
002331      POSMIN=POSMAX-PDEC
002333      IPMAX=POSMAX
002335      IPMIN=POSMIN
002337      IF (IPMAX.GE.2) IPMAX=IPMAX+1

```

```

002342 POSMAX=IPMAX
002343 IF (IPMIN.LT.0) IPMIN=IPMIN-1
002347 POSMIN=IPMIN

      DETERMINE NEGATIVE MAX AND NEGATIVE MIN

002351 950 DEC=DECADE(IM,1)
002352 DO 1000 J=1,NPVL
002353 IF (DATAR(J).GE.0.0) GO TO 1000
002355 YNEGMN=ALOG10(ABS(DATAR(J)))
002357 YNEGMX=ALOG10(ABS(DATAR(J)))
002363 GO TO 1050
002365 1000 CONTINUE
002370 INEG=N
002373 IF (IPOS.NE.0) GO TO 1150
002374 PRINT 13,KAP0(IM,1),KARD(IM,2)
002375 GO TO 2000
002404 1050 INEG=1
002405 DO 1100 JJ=1,NPVL
002406 IFMPL=DATAR(JJ)
002410 IF (TEMPL.GE.0.0) GO TO 1100
002412 IFMPL=ALOG10(ABS(TEMPL))
002413 IF (YNEGMN.GT.TEMPL) YNEGMN=TEMPL
002417 IF (YNEGMX.LT.TEMPL) YNEGMX=TEMPL
002422 1100 CONTINUE
002425 IF (DEC.GT.0.0) GO TO 1125
002430 NFGMAX=YNEGMX
002432 NFGMIN=YNEGMN
002434 IF (NEGMX.GE.0) NEGMX=NEGMX+1
002436 YNEGMX=NEGMX
002441 IF (NEGMN.LT.0) NEGMN=NEGMN-1
002442 YNEGMN=NEGMN
002445 DEC=ABS(YNEGMX-YNEGMN)
002447 GO TO 1140
002451 1125 YNEGMN=YNEGMX_DEC
002452 NFGMAX=YNEGMX
002454 NFGMIN=YNEGMN
002456 IF (NEGMX.GE.0) NEGMX=NEGMX+1
002460 YNEGMX=NEGMX
002463 IF (NEGMN.LT.0) NEGMN=NEGMN-1
002465 YNEGMN=NEGMN
002470 1150 IF (IFR.NE.1) GO TO 1290

      DETERMINE POSITIVE MAX AND POSITIVE MIN FOR RIGHT HAND AXIS

002472 1150 IF (IFR.NE.1) GO TO 1290

      PDECR=DECADE(IM,2)
002474 DO 1160 J=1,NPVR
002476 IF (DATAR(J).LE.0.0) GO TO 1160
002500 POSMXR=ALOG10(DATAR(J))
002502 POSMNR=ALOG10(DATAR(J))
002504 GO TO 1165
002507 1160 CONTINUE
002507 IPOS=N
002512 GO TO 1185
002513 1165 IPOSR=1

```



```

003100 C 1330 IF (IPOS,NE.1) GO TO 1335
003102 IF (INFOR,NE.1) GO TO 1340

                                RIGHT HAND AXIS HAS POSITIVE AND NEGATIVE DATA

003104 CALL PLOT (0.,4.39,-3)
003106 DELTAP=POEC/4.18
003110 DELTPR=POECR/4.18
003112 DELTNR=DECR/4.18
003113 YSHIFP=.14
003115 YSHIFN=-.14
003116 CALL LOGPLT (4.18,90.,ILINE,POEC,POSMIN,DELTAP,IPRT,0)
003124 CALL LOGPLT (4.18,90.,ILINE,POECR,POSMNR,DELTPR,IPRT,1)
003126 CALL LOGPLT (4.18,-90.,ILINE,DECR,YNEN,DELNR,IPRT,1)
003146 CALL PLOT (0.,0.,3)
003151 YORG=-4.39
003153 YIN=.0
003154 GO TO 1400

                                RIGHT HAND AXIS HAS ALL NEGATIVE DATA

003154 C 1335 CALL PLOT (0.,4.39,-3)
003157 DELTAP=POEC/4.18
003161 DELTNR=POECR/4.18
003163 YSHIFP=.14
003165 YSHIFN=-.14
003166 CALL LOGPLT (4.18,90.,ILINE,POEC,POSMIN,DELTAP,IPRT,0)
003175 CALL LOGPLT (4.18,-90.,ILINE,DECR,YNEN,DELNR,IPRT,1)
003205 CALL PLOT (0.,0.,3)
003210 YORG=-4.39
003212 YIN=.0
003213 GO TO 1400

                                RIGHT HAND AXIS HAS ALL POSITIVE DATA

003213 C 1340 CALL PLOT (0.,0.,3)
003216 DELTAP=POEC/8.5
003220 DELTPR=POECR/8.5
003222 YSHIFP=.14
003223 CALL LOGPLT (8.5,90.,ILINE,POEC,POSMIN,DELTAP,IPRT,0)
003243 CALL LOGPLT (8.5,90.,ILINE,POECR,POSMNR,DELTPR,IPRT,1)
003244 CALL PLOT (0.,0.,3)
003250 YORG=-0.15
003251 YIN=.0
003251 GO TO 1400

                                LEFT HAND AXIS HAS ALL NEGATIVE DATA

003251 C 1350 IF (IFR,FO.1) GO TO 1360
003253 CALL PLOT (0.,8.5,-3)
003256 DELTAN=DEC/8.5
003260 CALL LOGPLT (8.5,-90.,ILINE,DEC,YNEN,DELTAN,IPRT,0)
003270 YSHIFN=.0
003271 YORG=-8.5
003273 YIN=.5
003274 CALL PLOT (0.,0.,3)
003277 GO TO 1400

```

```

003300 1360 IF (IPR8.NE.1) GO TO 1370
003302 IF (IPR8.NE.1) GO TO 1383

      RIGHT HAND AXIS HAS POSITIVE AND NEGATIVE DATA

003304 CALL PLOT (0.,4.39,-3)
003306 DELTAN=DFC/4.18
003310 DELTPR=PDECR/4.18
003312 DELTNR=DFCR/4.18
003314 YSHIFP=.14
003316 YSHIFN=-.14
003318 CALL LOGPLT (4.18,-90.,ILINE,DEC,YNEGMN,DELTAN,IPRT,0)
003320 CALL LOGPLT (4.18,90.,ILINE,PDECR,POSMMR,DELTPR,IPRT,1)
003322 CALL LOGPLT (4.18,-90.,ILINE,DEC,YNEGN,DELTPR,IPRT,1)
003324 CALL PLOT (0.,0.,3)
003326 YORG=-4.39
003328 YIN=0.
003330 GO TO 1400

      RIGHT HAND AXIS HAS ALL NEGATIVE DATA

003332 1370 CALL PLOT (0.,8.5,-3)
003334 DELTAN=DFC/8.5
003336 DELTPR=PDECR/8.5
003338 YSHIFN=0.
003340 CALL LOGPLT (8.5,-90.,ILINE,DEC,YNEGMN,DELTAN,IPRT,0)
003342 CALL LOGPLT (8.5,-90.,ILINE,DEC,YNEGN,DELTPR,IPRT,1)
003344 CALL PLOT (0.,0.,3)
003346 YORG=-8.5
003348 YIN=0.
003350 GO TO 1400

      RIGHT HAND AXIS HAS ALL POSITIVE DATA

003352 1380 CALL PLOT (0.,4.39,-3)
003354 DELTAN=DFC/4.18
003356 DELTPR=PDECR/4.18
003358 YSHIFP=.14
003360 YSHIFN=-.14
003362 CALL LOGPLT (4.18,-90.,ILINE,DEC,YNEGMN,DELTAN,IPRT,0)
003364 CALL LOGPLT (4.18,90.,ILINE,PDECR,POSMMR,DELTPR,IPRT,1)
003366 CALL LOGPLT (4.18,-90.,ILINE,DEC,YNEGN,DELTPR,IPRT,1)
003368 CALL PLOT (0.,0.,3)
003370 YORG=-4.39
003372 YIN=0.
003374 GO TO 1400

      DRAW AND LABEL TIME AXIS

003376 1400 N=NT+1
003378 X=0.
003380 DX=0.5/NT
003382 EXP=IT
003384 TINC=(TMAG-TMTNAX)/NT
003386 ON 1410 T=1,N
003388 T=TINC*(I-1)+TMINAX
003390 CALL SYMROL (X,YIN,1,13,0,-2)
003392 CALL NUMBER (X=1,YIN=.15,1,1,0,0,2)
003394 CALL PLOT (X,YIN,3)
003396 X=X+DX
003398 CONTINUE
003400 CALL SYMROL (2.5,YIN=.35,1,9,TIME * 10,0,0,0)
003402
003404
003406
003408
003410
003412
003414
003416
003418
003420
003422
003424
003426
003428
003430
003432
003434
003436
003438
003440
003442
003444
003446
003448
003450
003452
003454
003456
003458
003460
003462
003464
003466
003468
003470
003472
003474
003476
003478
003480
003482
003484
003486
003488
003490
003492
003494
003496
003498
003500
003502
003504
003506
003508
003510
003512
003514
003516
003518
003520
003522
003524
003526
003528
003530
003532
003534
003536
003538
003540
003542
003544
003546
003548
003550
003552
003554
003556
003558
003560
003562
003564
003566
003568
003570
003572
003574
003576
003578
003580
003582
003584
003586
003588
003590
003592
003594
003596
003598
003600
003602
003604
003606
003608
003610
003612
003614
003616
003618
003620
003622
003624
003626
003628
003630
003632
003634
003636
003638
003640
003642
003644
003646
003648
003650
003652
003654
003656
003658
003660
003662
003664
003666
003668
003670
003672
003674
003676
003678
003680
003682
003684
003686
003688
003690
003692
003694
003696
003698
003700
003702
003704
003706
003708
003710
003712
003714
003716
003718
003720
003722
003724
003726
003728
003730
003732
003734
003736
003738
003740
003742
003744
003746
003748
003750
003752
003754
003756
003758
003760
003762
003764
003766
003768
003770
003772
003774
003776
003778
003780
003782
003784
003786
003788
003790
003792
003794
003796
003798
003800
003802
003804
003806
003808
003810
003812
003814
003816
003818
003820
003822
003824
003826
003828
003830
003832
003834
003836
003838
003840
003842
003844
003846
003848
003850
003852
003854
003856
003858
003860
003862
003864
003866
003868
003870
003872
003874
003876
003878
003880
003882
003884
003886
003888
003890
003892
003894
003896
003898
003900
003902
003904
003906
003908
003910
003912
003914
003916
003918
003920
003922
003924
003926
003928
003930
003932
003934
003936
003938
003940
003942
003944
003946
003948
003950
003952
003954
003956
003958
003960
003962
003964
003966
003968
003970
003972
003974
003976
003978
003980
003982
003984
003986
003988
003990
003992
003994
003996
003998
004000

```

```

003557 CALL NUMBER (3.45,YIN-.30*.07*EXP(0.2*-1))
003558 IL=0
003559 IP=0
003560 LEFTLL=0
003561 RTITELL=0
003562 IUNITL=1
003563 IUNITR=1
003564 NFILES=0
003565
003566 DO 1900 I=1,NCT
003567 IP=IP+1
003568 ISYMBL=ISYMBL+1
003569 ISM=KARD(IP,9)
003570 JLABEL=KARD(IP,10)
003571 PRINT 5, KARD(IP,1),KARD(IP,2),KARD(IP,8)
003572 IF (KARD(IP,4).EQ.10MP) GO TO 1750
003573
003574
003575
003576
003577
003578
003579
003580
003581
003582
003583
003584
003585
003586
003587
003588
003589
003590
003591
003592
003593
003594
003595
003596
003597
003598
003599
003600
003601
003602
003603
003604
003605
003606
003607
003608
003609
003610
003611
003612
003613
003614
003615
003616
003617
003618
003619
003620
003621
003622
003623
003624
003625
003626
003627
003628
003629
003630
003631
003632
003633
003634
003635
003636
003637
003638
003639
003640
003641
003642
003643
003644
003645
003646
003647
003648
003649
003650
003651
003652
003653
003654
003655
003656
003657
003658
003659
003660
003661
003662
003663
003664
003665
003666
003667
003668
003669
003670
003671
003672
003673
003674
003675
003676
003677
003678
003679
003680
003681
003682
003683
003684
003685
003686
003687
003688
003689
003690
003691
003692
003693
003694
003695
003696
003697
003698
003699
003700
003701
003702
003703
003704
003705
003706
003707
003708
003709
003710
003711
003712
003713
003714
003715
003716
003717
003718
003719
003720
003721
003722
003723
003724
003725
003726
003727
003728
003729
003730
003731
003732
003733
003734
003735
003736
003737
003738
003739
003740
003741
003742
003743
003744
003745
003746
003747
003748
003749
003750
003751
003752
003753
003754
003755
003756
003757
003758
003759
003760
003761
003762
003763
003764
003765
003766
003767
003768
003769
003770
003771
003772
003773
003774
003775
003776
003777
003778
003779
003780
003781
003782
003783
003784
003785
003786
003787
003788
003789
003790
003791
003792
003793
003794
003795
003796
003797
003798
003799
003800
003801
003802
003803
003804
003805
003806
003807
003808
003809
003810
003811
003812
003813
003814
003815
003816
003817
003818
003819
003820
003821
003822
003823
003824
003825
003826
003827
003828
003829
003830
003831
003832
003833
003834
003835
003836
003837
003838
003839
003840
003841
003842
003843
003844
003845
003846
003847
003848
003849
003850
003851
003852
003853
003854
003855
003856
003857
003858
003859
003860
003861
003862
003863
003864
003865
003866
003867
003868
003869
003870
003871
003872
003873
003874
003875
003876
003877
003878
003879
003880
003881
003882
003883
003884
003885
003886
003887
003888
003889
003890
003891
003892
003893
003894
003895
003896
003897
003898
003899
003900
003901
003902
003903
003904
003905
003906
003907
003908
003909
003910
003911
003912
003913
003914
003915
003916
003917
003918
003919
003920
003921
003922
003923
003924
003925
003926
003927
003928
003929
003930
003931
003932
003933
003934
003935
003936
003937
003938
003939
003940
003941
003942
003943
003944
003945
003946
003947
003948
003949
003950
003951
003952
003953
003954
003955
003956
003957
003958
003959
003960
003961
003962
003963
003964
003965
003966
003967
003968
003969
003970
003971
003972
003973
003974
003975
003976
003977
003978
003979
003980
003981
003982
003983
003984
003985
003986
003987
003988
003989
003990
003991
003992
003993
003994
003995
003996
003997
003998
003999
004000

```

```

003662 IF (DATAL(LPTS),LT,0.0) GO TO 1680
003664 IF (DATAL(LPTS),GT,0.0) GO TO 1525
003666 MOUNT=MOUNT+1
003667 GO TO 1560
003670 JCOUNT=JCOUNT+1
003672 1480 CONTINUE
003675 1500 NPIS(NFILE) = JCOUNT
003677 1525 MSTART(NFILE) = ISTART
003701 ISTART=ISTART+MOUNT+JCOUNT
003703 IF (ISTART,GT,LL) GO TO 1550
003706 NFILE=NFILE+1
003707 IF (NFILE,LE,10) GO TO 1420
003711 NFILES=NFILES + 1
003712 PRINT 14, KARD (IP,1), KARD(IP,2)
003725 IF (NFILES,GE, NCT) GO TO 2000
003727 GO TO 1940
003728 1550 CONTINUE
003729 C
003725 CALL LINES (1,2,IZ(IP),-ISYMBL)
003732 DO 1750 II=1,NFILE
003734 K=MSTART(II)
003736 J=NPIS(II)+K-1
003740 MFIRST=0
003741 IF (DATAL(K),LT,0.0) GO TO 1600
003743 1560 IF (DATAL(K),GT,0.0) GO TO 1575
003746 K=K+1
003747 IF (K,GT,J) GO TO 1700
003752 GO TO 1560
003754 1575 YMIN=POSIN
003755 YP2=DELTAP
003757 YSHIFT=YSHIFT
003757 GO TO 1625
003757 1600 YMIN=YNEGMN
003761 YP2=-DELTAN
003762 YSHIFT=YSHIFN
003764 M=0
003765 DO 1675 IK=K,J
003767 TEMPL=DATAL(IK)
003771 IF (TEMPL,EQ,0.0) GO TO 1675
003772 M=M+1
003773 IF (MFIRST,EQ,0) MFIRST=M
003775 TEMPL=ALOG10(ABS(TEMPL))
004001 IF (TEMPL,LT,YMIN) TEMPL=YMIN
004004 WORKY(M)=(TEMPL(IK)-T4)/DT
004010 WORKY(M)=(TEMPL-YMIN)/YP2+YSHIFT
004014 1675 CONTINUE
004017 IF (II,NF,1) GO TO 1690
004021 CALL LINES (WORKY (MFIRST),WORKY(MFIRST),0)
004024 CALL LINES (WORKY(WORKY,M))
004027 IF (II,EQ,NFILE) GO TO 1700
004031 CALL LINES (WORKY(M),YSHIFT,1)
004034 CALL LINES (WORKY(M),-YSHIFT,0)
004041 CONTINUE
004043 TEMPX=0.5+3.5*FLOAT(MOD(ISYMBL+1,2))
004054 TEMPY=-0.14*FLOAT((ISYMBL+1)/2)+YORG
004061 CALL LINES (TEMPX,TEMPY,0.035,0)
004065 CALL LINES (TEMPX+.60,TEMPY+.035,1)
004074 CALL SYMROL (TEMPX+.67,TEMPY,0.07,HL,0.0,1)

```

```

004102 CALL SYMROL(TEMPX+.07,TEMPY.0.07,LABEL(ICH),0.0,10)
004103 CALL SYMROL(TEMPX+1.58,TEMPY.0.07,LABEL(ACH),0.0,10)
004104 TUNITL=ISM
004105 GO TO 1900
004106
004107 1750 IP=IR+1
004108 L=LEFTLL +1
004109 LL=LEFTLL +KARD(IP*8)
004110 LEFTLL=LL
004111 ISTART=I
004112 NFILE=1
004113 ICOUNT=0
004114 MOUNT=0
004115 DO 1770 IPTS=ISTART,LL
004116 IF (DATAR(LPTS).LT.0.) GO TO 1810
004117 IF (DATAR(LPTS).NE.0.) GO TO 1780
004118 MOUNT=MOUNT+1
004119 1770 CONTINUE
004120 1780 ISTART=ISTART+MOUNT
004121 MOUNT=0
004122 DO 1800 LPTS=ISTART,LL
004123 IF (DATAR(LPTS).GT.0.) GO TO 1790
004124 IF (DATAR(LPTS).LT.0.) GO TO 1810
004125 MOUNT=MOUNT+1
004126 GO TO 1800
004127 1790 ICOUNT=ICOUNT+1
004128 1800 CONTINUE
004129 NPTS(NFILE)=ICOUNT
004130 MSTART(NFILE)=ISTART
004131 GO TO 1860
004132 IF (ICOUNT.LE.0) GO TO 1820
004133 NPTS(NFILE)=ICOUNT
004134 MSTART(NFILE)=ISTART
004135 NFILE=NFILE+1
004136 IF (NFILE.LE.10) GO TO 1820
004137 NFILES=NFILES+1
004138 IF (NFILES.GE.NCT) GO TO 2000
004139 GO TO 1900
004140 1820 JCOUNT=0
004141 ISTART=ISTART+MOUNT+ICOUNT
004142 IF (ISTART.GT.LL) GO TO 1860
004143 MOUNT=0
004144 DO 1840 IPTS=ISTART,LL
004145 IF (DATAR(LPTS).LT.0.) GO TO 1830
004146 IF (DATAR(LPTS).GT.0.) GO TO 1850
004147 MOUNT=MOUNT+1
004148 GO TO 1840
004149 1830 JCOUNT=JCOUNT+1
004150 1840 CONTINUE
004151 1850 NPTS(NFILE)=JCOUNT
004152 MSTART(NFILE)=ISTART
004153 ISTART=ISTART+MOUNT+JCOUNT
004154 IF (ISTART.GT.LL) GO TO 1860
004155 NFILE=NFILE+1
004156 IF (NFILE.LE.10) GO TO 1760
004157 NFILES=NFILES+1
004158 IF (NFILES.GE.NCT) GO TO 2000
004159 GO TO 1900
004160 1860 CONTINUE
004161
004162
004163
004164
004165
004166
004167
004168
004169
004170
004171
004172
004173
004174
004175
004176
004177
004178
004179
004180
004181
004182
004183
004184
004185
004186
004187
004188
004189
004190
004191
004192
004193
004194
004195
004196
004197
004198
004199
004200
004201
004202
004203
004204
004205
004206
004207
004208
004209
004210
004211
004212
004213
004214
004215
004216
004217
004218
004219
004220
004221
004222
004223
004224
004225
004226
004227
004228
004229
004230
004231
004232
004233
004234
004235
004236
004237
004238
004239
004240
004241
004242
004243
004244
004245
004246
004247

```

```

004267 CALL LINES (1,2,12(IP),-ISYMBL)
004268 DO 1895 I1=1,NFILE
004269 K=NSTART(I1)
004270 J=NPITS(I1)+K-1
004271 MFIRST=0
004272 1870 IF (DATAR(K).LT.0.0) GO TO 1860
004273 IF (DATAR(K).GT.0.0) GO TO 1875
004274 K=K+1
004275 IF (K.GT.J) GO TO 1895
004276 GO TO 1870
004277 1875 YMIN=POS(MNR)
004278 YP2=DELTPR
004279 YSHIFT=YSHIFT+YP2
004280 GO TO 1885
004281 1880 YMIN=YMIN+YNEG
004282 YP2=DELTPR
004283 YSHIFT=YSHIFT+YP2
004284 M=0
004285 DO 1890 IK=K,J
004286 TEMPR=DATAR(IK)
004287 IF (TEMPR.EQ.0.0) GO TO 1890
004288 M=M+1
004289 IF (MFIG.ST.EQ.0) MFIG=ST=M
004290 TEMPR=AL0010(ABST(TEMPR))
004291 IF (TEMPR.LT.YMIN) TEMPR=YMIN
004292 WORKT(M)=(TIMER(IK)-TM)/DT
004293 WOP.Y(M)=(TEMPR-YMIN)/YP2+YSHIFT
004294 1890 CONTINUE
004295 IF (I1.NE.1) GO TO 1892
004296 CALL LINES (WORKT(MFIRST),WORKY(MFIRST),0)
004297 1892 CALL LINES (WORKT,WORKY,M)
004298 IF (I1.EQ.NFILE) GO TO 1895
004299 CALL LINES (WORKT(M),YSHIFT+1)
004300 CALL LINES (WORKT(M),-YSHIFT+0)
004301 1895 CONTINUE
004302 TEMPR=0.5+3.5*FLOAT(MOD(ISYMBL+1,2))
004303 TEMPY=0.14*FLOAT((ISYMBL+1)/2)+YORG
004304 CALL LINES (TEMPX,TEMPY,0.035,0)
004305 CALL LINES (TEMPX+50,TEMPY+0.035,1)
004306 CALL SYMBOL (TEMPX+57,TEMPY,0.07,MR,0.0,1)
004307 CALL SYMBOL (TEMPX+57,TEMPY,0.07,LABEL(ISM),0.0,1)
004308 CALL SYMBOL (TEMPX+1.59,TEMPY,0.07,JLABEL,N.0,1)
004309 TUNITR=ISM
004310 1800 CONTINUE
004311 CALL PLOT (0.,YORG,-3)
004312 CALL SYMBOL (-.63,3.25,.1,LUNITS(I,IUNITL),90.,20)
004313 CALL SYMBOL (.5,-1.05,.1,HEADR,0.0,50)
004314 IF (IFR.NE.1) GO TO 1950
004315 CALL SYMBOL (.67,3.25,.1,LUNITS(I,IUNITP),90.,20)
004316 1950
004317
004318
004319
004320
004321
004322
004323
004324
004325
004326
004327
004328
004329
004330
004331
004332
004333
004334
004335
004336
004337
004338
004339
004340
004341
004342
004343
004344
004345
004346
004347
004348
004349
004350
004351
004352
004353
004354
004355
004356
004357
004358
004359
004360
004361
004362
004363
004364
004365
004366
004367
004368
004369
004370
004371
004372
004373
004374
004375
004376
004377
004378
004379
004380
004381
004382
004383
004384
004385
004386
004387
004388
004389
004390
004391
004392
004393
004394
004395
004396
004397
004398
004399
004400
004401
004402
004403
004404
004405
004406
004407
004408
004409
004410
004411
004412
004413
004414
004415
004416
004417
004418
004419
004420
004421
004422
004423
004424
004425
004426
004427
004428
004429
004430
004431
004432
004433
004434
004435
004436
004437
004438
004439
004440
004441
004442
004443
004444
004445
004446
004447
004448
004449
004450
004451
004452
004453
004454
004455
004456
004457
004458
004459
004460
004461
004462
004463
004464
004465
004466
004467
004468
004469
004470
004471
004472
004473
004474
004475
004476
004477
004478
004479
004480
004481
004482
004483
004484
004485
004486
004487
004488
004489
004490
004491
004492
004493
004494
004495
004496
004497
004498
004499
004500
004501
004502
004503
004504
004505
004506
004507
004508
004509
004510
004511
004512
004513
004514
004515
004516
004517
004518
004519
004520
004521
004522
004523
004524
004525
004526
004527
004528
004529
004530
004531
004532
004533
004534
004535
004536
004537
004538
004539
004540
004541
004542
004543
004544
004545
004546
004547
004548
004549
004550
004551
004552
004553
004554
004555
004556
004557
004558
004559
004560
004561
004562
004563
004564
004565
004566
004567
004568
004569
004570
004571
004572
004573
004574
004575
004576
004577
004578
004579
004580
004581
004582
004583
004584
004585
004586
004587
004588
004589
004590
004591
004592
004593
004594
004595
004596
004597
004598
004599
004600
004601
004602
004603
004604
004605
004606
004607
004608
004609
004610
004611
004612
004613
004614
004615
004616
004617
004618
004619
004620
004621
004622
004623
004624
004625
004626
004627
004628
004629
004630
004631
004632
004633
004634
004635
004636
004637
004638
004639
004640
004641
004642
004643
004644
004645
004646
004647
004648
004649
004650
004651
004652
004653
004654
004655
004656
004657
004658
004659
004660
004661
004662
004663
004664
004665
004666
004667
004668
004669
004670
004671
004672
004673
004674
004675
004676
004677
004678
004679
004680
004681
004682
004683
004684
004685
004686
004687
004688
004689
004690
004691
004692
004693
004694
004695
004696
004697
004698
004699
004700
004701
004702
004703
004704
004705
004706
004707
004708
004709
004710
004711
004712
004713
004714
004715
004716
004717
004718
004719
004720
004721
004722
004723
004724
004725
004726
004727
004728
004729
004730
004731
004732
004733
004734
004735
004736
004737
004738
004739
004740
004741
004742
004743
004744
004745
004746
004747
004748
004749
004750
004751
004752
004753
004754
004755
004756
004757
004758
004759
004760
004761
004762
004763
004764
004765
004766
004767
004768
004769
004770
004771
004772
004773
004774
004775
004776
004777
004778
004779
004780
004781
004782
004783
004784
004785
004786
004787
004788
004789
004790
004791
004792
004793
004794
004795
004796
004797
004798
004799
004800
004801
004802
004803
004804
004805
004806
004807
004808
004809
004810
004811
004812
004813
004814
004815
004816
004817
004818
004819
004820
004821
004822
004823
004824
004825
004826
004827
004828
004829
004830
004831
004832
004833
004834
004835
004836
004837
004838
004839
004840
004841
004842
004843
004844
004845
004846
004847
004848
004849
004850
004851
004852
004853
004854
004855
004856
004857
004858
004859
004860
004861
004862
004863
004864
004865
004866
004867
004868
004869
004870
004871
004872
004873
004874
004875
004876
004877
004878
004879
004880
004881
004882
004883
004884
004885
004886
004887
004888
004889
004890
004891
004892
004893
004894
004895
004896
004897
004898
004899
004900
004901
004902
004903
004904
004905
004906
004907
004908
004909
004910
004911
004912
004913
004914
004915
004916
004917
004918
004919
004920
004921
004922
004923
004924
004925
004926
004927
004928
004929
004930
004931
004932
004933
004934
004935
004936
004937
004938
004939
004940
004941
004942
004943
004944
004945
004946
004947
004948
004949
004950
004951
004952
004953
004954
004955
004956
004957
004958
004959
004960
004961
004962
004963
004964
004965
004966
004967
004968
004969
004970
004971
004972
004973
004974
004975
004976
004977
004978
004979
004980
004981
004982
004983
004984
004985
004986
004987
004988
004989
004990
004991
004992
004993
004994
004995
004996
004997
004998
004999
005000

```

DRAW ENCLOSING BOX AND THEN ADVANCE PAPER

```

004507 CALL PLOT (11., 0., -3)
004512 CONTINUE
004515 CALL PLOT (11., 0., 999)
004517 PRINT A, NFRAME
004525 STOP
004527 END

```


000127	115	YB=YA
000131		SX=AMINI(S,-Z+ZLEFT,ZEND(IJ))
000137		IF(SX.GT.C.0) GO TO 130
000142		IJ=IJ+1
000143		IF(IJ.LE.JJ) GO TO 120
000145		IJ=1
000146		ZLEFT=Z
000147		JDOT1=0
000150		JDOT2=0
000152	120	GO TO 115
000152		CONTINUE
000152		XB=XA+SK*XFACI
000155		YB=YA+SK*YFACI
000156		JJ=JK(IJ)
000160		GO TO JJJ,(130,140,150,160)
000162	130	CONTINUE
000165		S=S-SX
000167		ZLEFT=ZLEFT-SX
000170		IF(S.LE.C.0) GO TO 200
000172		GO TO 110
000172	140	CONTINUE
000172		CALL PLOT(XA,YA,3)
000175		CALL PLOT(XA,YA,2)
000177		GO TO 130
000200	150	CONTINUE
000203		IF(JDOT1.NE.0) GO TO 130
000204		CALL PLOT(XA,YA,3)
000207		CALL PLOT(XA,YA,2)
000212		JDOT1=1
000213		GO TO 130
000216	160	CONTINUE
000216		IF(JDOT2.NE.0) GO TO 130
000217		CALL PLOT(XA,YA,3)
000222		CALL PLOT(XA,YA,2)
000225		JDOT2=1
000226		GO TO 130
000231	200	CONTINUE
000234		RETURN
000234	210	CONTINUE
000234		DO 220 I=1,NPTS
000234		CALL PLOT(X(I),Y(I),2)
000242	220	CONTINUE
000247		RETURN
000247		END


```

000013 SUBROUTINE LOGPLT (SOR,THETA,ILINE,DEC,XLEAST,DELTA,IPR,IFR)
000013 DATA LINES,I2HPR /4HLINE,2HPR/
000013 1 FORMAT(1X,6HLOGPLT,6E20.8)
000013 2 FORMAT (2X,3F10.0)
000013
000013 ANGLE=0.
000013 XMOST=XLFST+DEC
000013 XNCS=ABS(XMOST-XLEAST)
000015 NCYS=XNCS+1
000017 ON=-.2
000024 TD=-.2
000024 ODM=-.2
000024 TDM=-.26
000025 XTEN=-.34
000027 XTIC=-.44
000030 IF (IFR.NE.1) GO TO 100
000032 ON=6.41
000035 TD=6.41
000036 ODM=6.41
000037 TDM=6.34
000041 XTIC=6.11
000042 XTEN=6.21
000044
000046 100 IF (THETA.LT.0.0) GO TO 110
000048 S=SOR
000050 YSHIFT=.14
000052 ON TO 125
000053 S=SOP
000054 YSHIFT=.14
000056 DPCYS=S/XNCS
000058 DO 500 K=1,NCYS
000062 XK=K
000063 FPN=XLFST+XK-1.
000065 AFIN=ABS(FPN)
000067 IF (THETA.GT.0.0) GO TO 150
000069 YRG=(XK-1.)*DPCYS+.05
000072 CD=YRG+YSHIFT
000074 YTC=CD+.05
000076 CALL PLOT (XTIC,YTC,1)
000100 CALL SYMROL (XTIC,YTC,1,15,0.0,0.02)
000107 GO TO 175
000113 150 CONTINUE
000115 CD=(XK-1.)*DPCYS+.09
000117 YRG=CD+YSHIFT
000121 175 CONTINUE
000121 CALL NUMRER (XTEN,CD,1,10.0,0.0,1)
000125 IF (AFPN.GT.9.) GO TO 250
000127 IF (AFPN.LT.0.0) GO TO 200
000134 XRG=0.0
000135 GO TO 35
000137 200 CONTINUE
000137 XRG=0.0H
000137
000141 250 GO TO 35
000141 CONTINUE
000141 IF (FPN.GE.0.) GO TO 300
000143 XRG=TDH

```

```

000144      GO TO 350
000145      CONTINUE
000146      XRG=TD
000147      350 CONTINUE
000148      IF (IPR.NF,12HPR) GO TO 375
000149      PRINT 1, XRG,YRG
000150      375 CONTINUE
000151      CALL NUMBER (XRG,YRG,.07,FPN,ANGLE,-1)
000152      400 CONTINUE
000153      TDEC=DEC
000154      THETA1=ARS(THETA)
000155      X=C.O
000156      IF (IPR.FQ,1) X=6.0
000157      FY=S*YSHIFT
000158      CALL PLOT (X,FY,3)
000159      CALL SYMROL (X,FY,.14,13,THETA1,-2)
000160      CALL PLOT (X,FY,3)
000161      DO 700 I=1,IDECC
000162      XI=I
000163      TFM=(DEC-XI+1)*DPCYS
000164      D2=TEM-.499*DPCYS
000165      D4=TEM-.398*DPCYS
000166      D6=TEM-.222*DPCYS
000167      D8=TEM-.097*DPCYS
000168      DT=ABS(TFM-1)*DPCYS
000169      IF (THETA.LT,0.0) DT=-DT
000170      Y2=D2+YSHIFT
000171      Y4=D4+YSHIFT
000172      Y6=D6+YSHIFT
000173      Y8=D8+YSHIFT
000174      YT=DT+YSHIFT
000175      585 CONTINUE
000176      IF (IPR.NF,12HPR) GO TO 585
000177      PRINT 1, Y2,Y4,Y6,Y8,YT,TEM
000178      585 CONTINUE
000179      CALL SYMROL (X,Y8,.1,13,THETA1,-2)
000180      CALL PLOT (X,Y8,3)
000181      CALL SYMROL (X,Y6,.1,13,THETA1,-2)
000182      CALL PLOT (X,Y6,3)
000183      CALL SYMROL (X,Y4,.1,13,THETA1,-2)
000184      CALL PLOT (X,Y4,3)
000185      CALL SYMROL (X,Y2,.1,13,THETA1,-2)
000186      CALL PLOT (X,Y2,3)
000187      CALL SYMROL (X,YT,.14,13,THETA1,-2)
000188      CALL PLOT (X,YT,3)
000189      700 CONTINUE
000190      RETURN
000191      END

```

C. GREEK

1. Introduction

Subroutine GREEK is the controlling routine for a set of three subroutines which together draw the letters of the Greek alphabet using the CALCOMP plotting system. Subroutine GREEK can draw all 24 Greek letters in either upper or lower case. The user may specify any height and any orientation angle within the limitations of the plotting equipment.

2. Calling Procedure

Subroutine GREEK is called in Fortran by the following statement:

CALL GREEK(XPOS, YPOS, SCALE, LETTER, THETA, ICAP) with the

arguments defined as follows:

- XPOS - the X position (in inches) of the lower left-hand corner (before rotation) of the letter.
- YPOS - the y position (in inches) of the lower left-hand corner (before rotation) of the letter.
- SCALE - the scale height(in inches) desired for the letter. Some lower case letters will be drawn slightly smaller to maintain proportionality between letters. A height which is a multiple of 31 times the plotter increment is recommended for best results.
- LETTER - an integer number (1 to 24) specifying the position in the Greek alphabet of the letter to be drawn.
- THETA - orientation angle (in degrees) at which the letter is to be drawn. 0° - letter is vertical: positive angle - rotates letter counterclockwise.
- ICAP - selector for upper or lower case letters. ICAP = 0 -- lower case, ICAP=1 -- upper case.

Subroutine GREEK was written with the assumption that the user will open the plot file and establish an origin prior to calling the subroutine.

3. Data Control

The data points for each letter were selected by first drawing the letter on a 31 x 31 grid and then approximating all curved lines by straight line segments. The coordinates of the starting and stopping points of each line segment were then recorded along with the pen up-down commands. Next, in order to minimize the amount of computer storage space needed for the subroutine, the data points were packed such that each 60 bit word contains the x and y coordinates and pen commands for five end points. When a letter is being drawn, the particular data points for that letter are then unpacked by subroutine SEPAR8 described below.

4. Subroutine SEPAR8

Subroutine SEPAR8 unpacks the 60 bit computer words and separates the data into x and y coordinates and pen commands. It performs this operation by using a series of left and right shifts and masking functions.

5. Function Subroutine ISHIFT

Function subroutine ISHIFT performs left or right shifts of 1 to 60 bits. It is written in COMPASS (Control Data Corporation assembly language) for use on a CDC 6000 series computer.

6. Special Considerations

Since the data points have been packed into 60 bit words and function subroutine ISHIFT is written in COMPASS, the subroutine GREEK package presented here can only be utilized on a CDC 6000 series computer. The package requires 633 (octal) words of core and utilizes the CALCOMP routine PLOT.

7. Sample Output

(height of letters = 0.31)

α β γ δ ε ζ η θ ι κ λ μ

ν ξ ο π ρ σ τ υ φ χ ψ ω

A B Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ

N Ξ Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω

8. Program Listing

UU UU UU U UU.U UU

1 ALPHA	7 EIA	13 NU	19 TAU
2 BETA	8 TETA	14 XI	20 UPSILON
3 GAMMA	9 IOTA	15 OMICRON	21 PHI
4 DELTA	10 KAPPA	16 PI	22 CHI
5 EPSILON	11 LAMDA	17 RHO	23 PSI
6 ZETA	12 MU	18 SIGMA	24 OMEGA

LOWER CASE -- ICAP=0 UPPER CASE -- ICAP=1

```

DIMENSION XYNCH(32),IPOINT(159),JPOINT(156),KPOINT(47),LSTART(24)
EQUIVALENCE (IPOINT(37),IPOINT(11),JPOINT(113),KPOINT(1))
DATA XYNCH/
.63125,.0025,.69375,.125,.15625,.1875,.21875,.25,.29125,.3125,
.34375,.375,.40625,.4375,.46875,.5,.53125,.5625,.59375,.625,
.65625,.6875,.71875,.75,.78125,.8125,.84375,.875,.90625,.9375,
.96875,1.0/

```

[illegible][illegible]

110000
110000
110000

000011

000011


```

SUBROUTINE SEPAR (XU,IA,IY,IPEN,IFLAG,LAST)
  DIMENSION ID(1)
  LAST=1
  IF (IFLAG.EQ.0) LOC=12
  LOC=(LOC-12)/60 +1
100 LOC=IS-IT(10(LOC),MOD(LOC,60))
  IY=100.AND.31
  IPEN=100.AND.32
  IDU=IS-IT(100,-6)
  IX=100.AND.31
  LOC=LOC+12
  IDU=100.AND.32
  IF (100.NE.0) LAST=1
  RETURN
END

```

```

000011
000011
000011
000013
000020
000034
000036
000037
000042
000050
000051
000053
000055
000056

```

ISHIFT

0	11031.11002400000000	IDENT	ISHIFT	1 SYMBOLS
1		ENTRY	ISHIFT	3 REFERENCES
2	5611, 50220	VFO	30/011SHIFT, 24/2	
	65320	BSS	1	
		SA1	B1	
		SA2	B2	
		SB3	XC	
		LX6	B3, X1	
		CU	ISHIFT	
3	0400000001 *	END	10 STATEMENTS	
4	45000	STORAGE USED	0.143 SECONDS	
		6400 ASSEMBLY		

II. MINOR MODIFICATIONS

A. ELECTRA

1. Description of Modifications

Program ELECTRA^{1,2}, the ground burst EMP code, received two minor modifications since it was last documented. All input/output (I/O) operations using tape or drum were revised in order to cut running time and to make more efficient use of data tapes.

The running time was cut by replacing all lengthy I/O lists, which used implied DO loops, with a variably dimensioned subroutine (IOSAVE) which can write or read any length array as a single operation. This modification should produce up to 90% reduction in running time for the operations which it affects.

More efficient use of data tapes was accomplished by revising the contents of the data output array (OUTARR). By removing some unnecessary data items from the array and repositioning other items, the dimensions of the array were reduced from 21x7 to 21x4 resulting in a decrease of 63 words. The words in the reduced array were then packed two into one by an assembly language packing routine (PACK). The net result was to reduce

1. Jones, D. L. and D. H. Stump, ELECTRA, ORESTES, and Supporting Graphic Display Codes, U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, February 1971, pp. 1-36.

2. Borbely, J. A. and D. L. Jones, ELECTRA, An Electromagnetic Pulse Fortran Program (User's Guide), U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, October 1969.

the number of words written to tape, at a given time step and range step, from 147 to 43. If the number of angles used for a given run is less than 18, the number of words written to tape will be reduced even more.

2. Program Listing


```

000003      611 FORMAT(*0.0. OUTPUT DATA FILES ***)
000003      612 FORMAT(*FILE*,I3.0,*,RANGE NO*,I2.0,*,S*,*,RANGE*,S*,*,NO OF RECS*)
000003      613 FORMAT(IH,12I,F10.2,I15)
000003      614 FORMAT(*NEXT VALUE OF I IS*,I16.8)
000003      615 FORMAT(*ELECTRA...INU...MAL TERMINATION*)
000003      999 FORMAT(I8F10.0)
000003      100 FORMAT(I7A10,A8,I2)
000003      1113 FORMAT(IH1,8A10)
000003      1117 FORMAT(*ELECTRA...LRRHUM TERMINATION*)
000003      1127 FORMAT(*HEADER FOR FILE *,I2.0 ON TAPE DOES NOT CHECK*)
000003      1128 FORMAT(*HEADER ON TAPE: *,8A10)
C
000003      IEHR#6
000004      DO 100 I=1,8
000006      DO 100 J=1,200
000007      ICUT(I,J)=0
000013      100 CONTINUE
000017      DO 101 I=1,21
000020      DO 101 J=1,4
000021      OUIAHR(I,J)=0.
000025      101 CONTINUE
000031      DO 105 I=1,200
000032      #PRIV(I)=0
000033      105 CONTINUE
000035      LQD#C
000036      LQAD#1
000037      NTIME#0
000040      NBLCK#0
000041      NFILES#1
000042      NSMFT#0
000043      RESIT#1.
000044      REIND ISTORE
000046      READ 999,XJ1,XJ2,XJ3,XJ4,TC,T1,T2,I3,XJRUN,CLAMDA
000050      READ 1000,HEADER,IENUM
000010      READ(S,999) DR,DT,TS,SEQ,MMIN
000126      READ(S,1) ITHETA,(IHEI(I),I=1,IHEIA)
000143      READ(S,1) ITHET,(IINET(I),I=1,IINET)
000160      READ(S,2) INSEG,KDIN,UMAX,NSTEPS,NSTART,LGND,LMAX
000202      READ(S,3) NKPRNT,(KPRIN(I),I=1,NKPRNT)
000217      IF(NSTART.LE.0) NSTART#2
000222      IF(NSTEPS.LE.0) NSTEPS#4
000225      IF(SEC.LE.0.0) SEC#0.001
000230      READ(S,2) NKOUT
000236      IF(NKOUT.LE.0) GO TO 121
000240      READ(S,999) (ORNG(I),I=1,NKOUT)
000252      GO TO 140
000253      121 M#1
000254      READ(S,1) NDROUT,ORNG(1)
000264      TEMPOR#ORNG(1)
C
000266      DO 130 JJ=1,NDROUT
000267      READ(S,1) NROUT,DROUT
000276      DO 125 I=1,NROUT
000300      K#I+M
000302      TEMPOR#TEMPOR+DROUT
000304      ORNG(KI)#TEMPOR
000306      125 CONTINUE

```

```

000310      M=M*NRKOUT
000311      130 CONTINUE
000312      IF (K*LE.200) GO TO 135
000313      PRINT 25, K
000314      STOP
000315
000316      135 NRKOUT=K
000317      140 LGNDPI=LGND*1
000318      LGM1=LGND-1
000319      LMAXPI=LMAX*1
000320      LAXP=LMAX*4
000321      NPACK=LMAXP*2 -1
000322      KOIM1=KOIM-1
000323      KOIMP1=KOIM*1
000324      DO 1225 L=1, LMAXPI
000325      TH(L)=.01745329*THEIA(L)
000326      STM(L)=SIN(TH(L))
000327      IF (L.E.LMAXPI) GO TO 1225
000328      MT(L)=.01745329*MTETA(L)
000329
000330      1225 CONTINUE
000331      XTRAP=COS(TH(LGM1))/(COS(TH(LGM1))-COS(TH(LGND)))
000332      PCAL=C*DT/(TH(LGNDPI)-TH(LGND))*SQRT(1.-(C*DT/DR)**2))*1.E-5
000333      IF (RMIN*GE*RCAL) GO TO 1227
000334      IF (RMIN*LE.5.) GO TO 1226
000335      PRINT 26, RCAL
000336      STOP
000337
000338      1226 RMIN=RCAL
000339      1227 NMIN=MIN/(C*DT)*1.
000340      RSIANT=RMIN
000341      IF (IRENUM.LL. . GO TO 55
000342      REWIND IDATA
000343      REWIND JDATA
000344      REWIND KDATA
000345      REWIND LDATA
000346
000347      C      READ THE RESTART TAPE
000348      C
000349      C
000350      CALL IOSAVE(XNES,XNM5,E5,FS,BS,E,F,8,XNE,XNM,NLOAD,NMIN,1,1)
000351      DO 1036 J=1,NLOAD
000352      CALL IOSAVE(XNES,XNMS,E5,FS,BS,E,F(1,2),F(1,2),XNE(2),XNM(2),
000353      1 NLOAD,NMIN,KDIMP1,2)
000354      CALL IOSAVE(XNES,XNMS,E5,FS,BS,E,F(1,KDIMP1),F(1,KDIMP1)*B(1,KDIMP1),
000355      1,XNE(KDIMP1),XNM(KDIMP1),NLOAD,NMIN,1,2)
000356      1036 CONTINUE
000357      C
000358      C      REWIND ISTORE
000359      C
000360      READ (IDATA) NFILES,NRKOUT,LMAX
000361      IF (NFILES.EQ.IRENUM) GO TO 1024
000362      WRITE(6,8) NFILES,IRENUM
000363      STOP
000364
000365      1024 NFILES=IRENUM*1
000366      WRITE (JDATA) NFILES,NRKOUT,LMAX
000367      C
000368      C      DATA TAPE SETUP LOOP
000369      C
000370      C
000371      READ(IDATA) HEAD,IRNUMT
000372      DO 1025 J=1*4
000373      IF (HEAD(J).NE.HEADK(J)) GO TO 1026
000374
000446      000446
000447      000446
000448      000513
000449      000544
000450      000547
000451      000551
000452      000562
000453      000564
000454      000574
000455      000576
000456      000600
000457
000458      000611
000459      000620
000460      000622

```



```

001079      NRGIN(K)=K/CDT
001079      CONTINUE
1245      PRINT 5, RMIN,DR,DI
001075      PRINT 4, LAB(1),KOUT,LAB(2),KUT,LAB(3),NMIN,LAB(4),NMAX
001075      PRINT 4, LAB(5),NTEPS,LAB(6),NSTART,LAB(7),LGND,LAB(8),LMAX
001132      PRINT 7, (KOUT(I),I=1,NKOUT)
001156      EMUCUR=1.293/RHO
001171      EMURC1=EMURC/N/76600=
001173      TEMP=XELEN/4.E-6
001175      GCONE=TEAP/COULOM*ALLHUE)
001177      DTEPS = DT/EPSCID
001202      QTEPSG = DT/EPSCID
001204      XGND = EXP(-SGND*DT/PSG)
001206      AGND = (1.0-XGND)/SGND
001212      CHUDK = CHU*DR
001215      DTOR = DT/DR
001217      C1=C*UT/UM
001220      C2=C*CI*RMIN/DR+NSEU*NU[M-900000
001222

C
C      TIME LOOP
C
DO 410 N=MIN,NMAX
  T=H*UT
  NTIME=N
  CALL SECND(TIML)
  IF (TIME-GE.15)-GO TO 420
  IF (TIME-LE.15)-NSTART=SEC) GO TO 103
  CALL RESTART
  RESTAT=RESINT+1.
  IF (LOAD-GE.6) CALL SEGMENT(1)
103  IF (LOAD-GE.6) CALL SEGMENT(1)
102  T=MDIET-HUT
  TPUT=T+DT
  NSHIFT=C1*N-NSHIFT-02
  IF (NSHIFT-LE.0) NSHIFT=0
  NSHIFT=NSHIFT+NSHIFT1
  DO 2005 L=1,LMAXP1
    FK(L)=F(L,1)
    FTEMP(L)=FTEMP(L)
    FTEMP(L)=F(L,2)
2005  CONTINUE
    IF (NSHIFT-GE.5) GO TO 104
    IF (NSHIFT-GE.5) GO TO 2010
    CALL SHIFT
    DO 2009 L=1,LMAXP1
      F(RAP(L)=3.0F(L,1)-3.0FTEMP(L)+FTEMP2(L)
2009  CONTINUE
    GO TO 120
2010  DO 2011 L=1,LMAXF1
      F(L,1)=FATRAP(L)
2011  CONTINUE
104  RSTART=HMIN+NSHIFT*DTOR
      Q=UCUR*XJAY(RSTART+IMDU)
      XNEA=XNE(1)
      XNMA=XNM(1)
      TEMP=ATACH*ALF*(XNEA+XNMA)
      ARG=UT*TEMP
001231
001233
001236
001237
001241
001244
001247
001250
001252
001255
001257
001261
001263
001271
001273
001275
001276
001300
001301
001303
001305
001306
001307
001310
001312
001322
001325
001325
001327
001332
001334
001340
001342
001346
001347
001351
001355

```

```

001356 EXPON=EXP(-ARG)
001362 IF (ARG .GE. 1.E-7) GO TO 16
001364 FACTOR = ARG
001366 GO TO 17
001368 16 FACTOR = 1. - EXPON
001366 17 XNE(1)=EXPON*XNEA+FACTOR*Q/TEMP
001370 TEMP=XIOCOM*(XNEA*XNMA)
001375 ARG = DT*TEMP
001377 EXPON = EXP(-ARG)
001401 IF (ARG .GE. 1.E-7) GO TO 16
001406 FACTOR = ARG
001407 GO TO 20
001410 18 FACTOR = 1. - EXPON
001412 20 XNM(1)=EXPON*XNMA
001414 IF (TEMP .LE. 0.0) GO TO 21
001416 XN(1)=XNM(1)*FACTOR*ATTACH*XNEA/TEMP
001422 21 CONTINUE
001424 20 LOADM=LOAD-1
    IF (LOADM.LT.0) LOADM=0
C
C      RANGE LOOP
C
DO 31 K=1,KDIM
R=H*UR
RMDK=R-RDK.
RPHDK=R+RDK
RPHDKC=RPHDK/C
RCL=R/C
OTUR=DT/R
OTURR=OTUR/DR
IF (RPHDK.GE.C*IMHJ) GO TO 300
KPL=K+1
Q=GCUL*AJAY(RPHDK,IMHJ)
XNLA=XNE(KPL)
XNMA=XNM(KPL)
TEMP=ATTACH*ALF*(XNEA*XNMA)
ARG = DT*TEMP
EXPON = EXP(-ARG)
IF (ARG .GE. 1.E-7) GO TO 1111
FACTOR = ARG
GO TO 1112
1111 FACTOR = 1. - EXPON
1112 XNE(KPL)=EXPON*XNEA+FACTOR*Q/TEMP
1112 TEMP=XIOCOM*(XNEA*XNMA)
ARG = DT*TEMP
EXPON = EXP(-ARG)
IF (ARG .GE. 1.E-7) GO TO 1113
FACTOR = ARG
GO TO 1114
1113 FACTOR = 1. - EXPON
1114 XNM(KPL)=EXPON*XNMA
IF (TEMP .LE. 0.0) GO TO 1115
XNM(KPL)=XNM(KPL)+FACTOR*ATTACH*XNEA/TEMP
1115 CONTINUE
XSEC = 0.5*(XNE(K) + XNE(KPL))
XJUN = 0.5*(XNM(K) + XNM(KPL))
XJ=XJAY(X+T)

```

```

001547 CEA=CUULOM*EMOCON*ANEC
001551 CXAXP=CULOM*XIOMUB*(2.*XION*ASEC)
001555 CXAXP=CULOM*XIOMUB*(2.*XION*(KP1)+ANE(KP1))
001562 CEAN=CULOM*EMOCON*ANEC(KP1)

C C
      THETA LOOP E
001565 DO 165 L=1,LMAX
001566 EK(L)=E(L,K)
001571 LP1=L+1
001573 IF (L.GE.LGND) GO TO 167
001575 FVAL=.5*(FFACTA(L)+F(L,KP1))+F(L,KP1)*F(L,KP1)*F(L,K
      )
      SIGMA=CEA*(EMOBN(SURT(FVAL**2+EK(L)**2)*EMOBN))*.CXAXP
      TEMP = SIGMA*DTPTS
      IF (TEMP.LE.15.) GO TO 162
      X = L.*N
      A = 1./C/SIGMA
      GO TO 160
142 X = EXP(-TEMP)
      A = (1.-X)/SIGMA
      GO TO 160
150 IF (L.GT.LGND) GO TO 170
      SIGMA=CEA*(EMOBN(SURT(FVAL**2+EK(L-1)*EK(L)**2+FK(L)+F(L,KP1)**2
      ))*EMOBN))*CXAXP
      TEMP = SIGMA*DTPTS
      IF (TEMP.LE.15.) GO TO 162
      XAIR = 0.0
      AAIR = 1./C/SIGMA
      GO TO 163
162 XAIR = EXP(-TEMP)
      AAIR = (1.-XAIR)/SIGMA
163 AOA1 = XAIR*AGND
      X = (AAIR*XGND+AGND*AAIR)/AOA1
      A = AAIR*AGND/AOA1
      GO TO 160
170 IF (L.GT.LGNDP1) GO TO 180
      X = AGND
      A = AGND
      XJ = 0.0
      E(L,K)=X*EK(L)+A*(-AA*(B(LP1,K)*STM(LP1-B(L,K)*STM(L))/R/ECON(L))
      IF (ABS(E(L,K)).LT.1.E-25) E(L,K)=0.
185 CONTINUE

C C C
      THETA LOOP F
001760 DO 235 L=1,LMAXP1
001761 FK(L)=F(L,KP1)
001765 IF (L.GT.1) GO TO 240
      EVAL=.5*(EK(1)+E(1,KP1))
      GO TO 210
200 IF (L.GT.LGND) GO TO 220
      LM1=L-1
      EVAL=.5*(FFACTA(LM1)*EK(L)+E(L,KP1))+EFAC1B(LM1)*EK(LM1)*E(LM1)*K
      (P1))
210 SIG(L)=CEAN*(EMOBN(SURT(EVAL**2+FK(L)**2)*EMOBN))*CXAXP
      TEMP = SIG(L)*DTPTS

```

TH EL00P


```

000003      C
000006      FUNCTION EMOBFN(EOVERP)
000011      IF (EOVERP.GT.0.45/7) GO TO 3
000014      IF (EOVERP.GT.0.095) GO TO 2
000020      IF (EOVERP.GT.0.03) GO TO 1
000027      EMOBFN=2.092/(1.0+33.0*EOVERP)
000033      RETURN
000037      1 EMOBFN=1.80722*EOVERP*(-3*.1006+EOVERP*(3*2.685-EOVERP*1186.94))
000041      RETURN
000045      2 EMOBFN=0.169267/Sqrt(EOVERP)
000051      RETURN
000055      3 IF (EOVERP.GT.60.6) GO TO 4
000056      XLN=ALOG(EOVERP)
000057      EMOBFN=0.16212*EXP(ALN*10.04912*XLN-.51680))
000058      RETURN
000059      4 EMOBFN=0.033684*0.0279/EOVERP
000060      RETURN
000061      END

```

```

000005      FUNCTION XJAY(X,T)
1      COMMON/SOURCE/      AJ1,      XJ3,      XJ4,
2                          XJ2,      T3,
                          T1,
                          CLAUDA
C
1      COMMON /PARAMS/      C,      CMU,      EPS,
2      EPSGRD,      SUND,      ATTACH,      AELEN,
      X10MOD,      AELMOD,      X10CUM,      ALF
C
      TAU=1-R/C
      IF(TAU.GT.0.) GO TO 10
      XJAY=1.
      RETURN
10  IF(TAU.GE.12) GO TO 20
      XJAY=-1.675E5*EXP(2.58*TAU)/(1.*1.616E-4*EXP(2.58*TAU))/R**2
      RETURN
20  XJAY=-16.349E6*EXP(-.002E8*TAU))/R**2
      RETURN
      END
000005
000007
000012
000012
000013
000016
000036
000037
000047
000050

```

```

000002 SUBROUTINE RESTART
COMMON C(17,201), XNM(201), XNE(201),
1 XOUT(200), KPRIM(200), ICOUNT(200),
2 XMM(16), EFAC(16), EFACD(16),
3 ECON(17), EK(17), SIG(17),
4 FFAC(17), TH(18), STM(18),
5 HEAD(8), OUTAB(21,4),
6 FTEMP(16), FATRAP(16), THETA(18),
7 MTHETA(17)

C COMMON /INTEGER/
1 NLOAD, NTIME, LOAD,
2 IHENUM, NLOAD, KDIM, KJMPI,
3 JSTORE, LDATA, NBLOCK, DATA,
4 IDATA, LMAXP, NPACK,

C COMMON /RANDT/
1 RPHDR, R, MUC, RPR, RPHDR,
2 LGND, LGNDP, DR, HDR, LGH,
3 NFILES, NRPFI, K, KPI, KNUM,
4 CMUR, UTIPS, XTRAP, EMORCN,
5 TENDT, DT, AGND, IPDT,
6 OMUG(22)

C COMMON /SAVE/ ES(17,40), FS(18,40), DS(18,20), XNES(20),
1 XMS(20)

C 1 FORMATT(////10M NRLOA = ,15)

C NBLCK=NHLOCK+1
REWIN LUATA
CALL IOSAVE(XNES,XMS,ES,FS,DS,E,F,0,XNE,XNM,NLOAD,NTIME,1,5)
DO 400 I=1,NLOAD
CALL IOSAVE(XNES,XMS,ES,FS,DS,E(1,2),F(1,2),XNE(2),XNM(2),
1 NLOAD,NTIME,KDIM,1,0)
CALL IOSAVE(XNES,XMS,ES,FS,DS,E(1,KJMPI),F(1,KJMPI),B(1,KJMPI))
1,XNE(KJMPI),XNM(KJMPI),NLOAD,NTIME,1,6)
400 CONTINUE
END FILE LUATA
REWIN ISTORE
PRINT 1, NBLCK
REFURN
END

```



```

000227 F(L,I)=F(L,KOIMPI)
000232 H(L,I)=H(L,KOIMPI)
000235 CONTINUE
000237 F(LMAXPI,I)=F(LMAXPI,KOIMPI)
000243 H(LMAXPI,I)=H(LMAXPI,KOIMPI)
000246 XNE(I) = XNE(KOIMPI)
000250 XNM(I) = XNM(KOIMPI)
000251 IF (LOAD.LE.NLOAD) GO TO 250

C      ILOAD IS ADVANCED WHEN A NEW COMPLETE SEGMENT W/ KOIM MEMBERS
C      IS ADDED TO THE SEGMENTS BEING HELD ON DISK. NLOAD IS = TO
C      NUMBER OF SEGMENTS ON DISK + 1
C
NLOAD=NLOAD+1
DO 2=1 K=2,KOIMPI
  XNE(K)=0.
  XNM(K)=0.
  F(LMAXPI,K)=0.
  H(LMAXPI,K)=0.
  DO 2=L=1,LMAX
    E(L,K)=0.
  F(L,K)=0.
  H(L,K)=0.
240 H(L,K)=0.
      RETURN

C 250 IF (SHIFT.NE.) GO TO 260
C
C      READ IN FROM DISK THAT SEG TO COMP SAME SEG AT NEW DELTA T
C
CALL IOSAVE(XNE,XNM,ES,FS,SS,E(1,2),F(1,2),H(1,2),XNM(2),
1 NLOAD,NTIME,KOIMPI)
CALL IOSAVE(XNE,XNM,ES,FS,SS,E(1,KOIMPI),F(1,KOIMPI),H(1,KOIMPI),
1 XNE(KOIMPI),XNM(KOIMPI),NLOAD,NTIME+1,7)
RETURN

C 260 KOIM=2,KOIM-2
CALL IOSAVE(XNE,XNM,ES,FS,SS,E(1,2),F(1,2),H(1,2),XNE(2),XNM(2),
1 NLOAD,NTIME,KOIMPI)
CALL IOSAVE(XNE,XNM,ES,FS,SS,E(1,KOIM),F(1,KOIM),H(1,KOIM),
1 XNE(KOIM),XNM(KOIM),NLOAD,NTIME+1,7)
IF (LOAD.LE.NLOAD) GO TO 275
DO 2=L=1,LMAX
  E(L,KOIMPI)=ES(L,KOIM)
  F(L,KOIMPI)=FS(L,KOIM)
  H(L,KOIMPI)=H(L,KOIM)
270 CONTINUE
  F(LMAXPI,KOIMPI)=FS(LMAXPI,LOAD)
  H(LMAXPI,KOIMPI)=H(LMAXPI,LOAD)
  XNE(KOIMPI)=XNE(LOAD)
  XNM(KOIMPI)=XNM(LOAD)
  RETURN

275 DO 2=L=1,LMAX
  E(L,KOIMPI)=0.0
  F(L,KOIMPI)=0.0
  H(L,KOIMPI)=0.0
280 CONTINUE
  F(LMAXPI,KOIMPI)=0.0
  H(LMAXPI,KOIMPI)=0.0

```

```

000545 XMS(LUIMP1)=0.0
000546 XMS(KUIMP1)=0.0
000547 RETURN

C
C      SAVE MIGHT MOST DEG UN DISK, PREPARE TO STEP TIME AND READ IN
C      EMLJEM COMPUTED DEG AT START OF TIME LOOP
C
000547 300 CALL IOSAVE(XMS,XNMS,ES,FS,DS,E,F,B,XNE,XNM,NLOAD,NTIME,1,B)
000548 CALL IOSAVE(XMS,XNMS,ES,FS,DS,E,F,B,XNE,XNM,NLOAD,NTIME,1,B)
000549 1 NLOAD,NTIME,KUIMP1,0)
000550 CALL IOSAVE(XMS,XNMS,ES,FS,DS,E,F,B,XNE,XNM,NLOAD,NTIME,1,B)
000551 1,XNE(KUIMP1),XNM(KUIMP1),NLOAD,NTIME,1,B)
000552 REWIND ISTORE
000553 REWIND JSTORE
000554 I=ISTORE
000555 JSTORE=JSTORE
000556 JSTORE=I
000557 IF(Load.Eu,1) RETURN
000558 LQAD=LOAD-1
000559 DO 35 L=1,LMAX
000560 ES(L,LOAD)=E(L,2)
000561 FS(L,LOAD)=F(L,2)
000562 DS(L,LOAD)=D(L,2)
000563 350 CONTINUE
000564 FS(L,LOAD)=LOAD=1
000565 FS(L,LOAD)=B(L,LOAD,1,2)
000566 XMS(LOAD)=XNE(2)
000567 XMS(LOAD)=XNM(2)
000568 LOAD=1
000569 RETURN
000570
000571 400 CALL IOSAVE(XMS,XNMS,ES,FS,DS,E,F,B,XNE,XNM,NLOAD,NTIME,1,B)
000572 CALL IOSAVE(XMS,XNMS,ES,FS,DS,E,F,B,XNE,XNM,NLOAD,NTIME,1,B)
000573 1 NLOAD,NTIME,KUIMP1,0)
000574 CALL IOSAVE(XMS,XNMS,ES,FS,DS,E,F,B,XNE,XNM,NLOAD,NTIME,1,B)
000575 1,XNE(KUIMP1),XNM(KUIMP1),NLOAD,NTIME,1,B)
000576 REWIND JSTORE
000577 RETURN
000578 END

```

```

000002      SUBROUTINE PRINTER
COMMON E(17,201), F(18,201), B(18,201), XNE(201),
1      XNM(201), KPRINT(200), ICOUNT(200,8),
2      WMT(16), EFACT(16), EFACTB(16),
3      HTH(17), EK(17), SIG(17),
4      FFACT(17), FFACTB(17), TH(18),
5      FK(18), HEADR(8), TH(18), OUTAR(12,4),
6      FTEMP(18), FTEMPB(18), FTRAP(18), THETA(18),
7      HTHETA(17)

C      **NOTE** WHEN DIMENSIONS OF OUTAKK ARE CHANGED, IROW CHECK
C      SHOULD BE CHANGED
C
000002      COMMON /INTEGER/ KUATA, ISTORE, NTIME, LOAD,
1      IRENUM, KAUUT, LMAAP1, KDIM, KDIMPL,
2      JSTORE, NSHIFT, LDATA, NBLOCK, JDATA,
3      IUATA, KUIMM1, LMAXP, NPACK,
4
C      COMMON /RANUT/ K, KUC, RPOD, RPHDK,
1      RPHORC, NSPART, DR, HDR, LGM1,
2      LGND, LGNDP1, A, KPI, KNUM,
3      NFILES, HSHFT, QCON, EMUCON, EMUCN,
4      CHUDR, DLEPS, XTAP, I, TPDT,
5      TPOT, INOUT, UT, AGND, XGND,
6      ORAG(200)

C
000002      1 FORMAT(///3110)
000002      9 FORMAT(1H1///1X,201,10X,RANGE NO,*,12,11X,PAGE *,13)
000002      10 FORMAT(11X,TIME HIGH,NSC,6X,TIME TOT SIGMA,6X,TIME B PH1,*,
1      10X,TIME E THETA,17X,IRENUM = *,13,5X,TIME E RADIAL,*)
000002      11 FORMAT(5X,*(5X,E15.8),1X,E15.8)
000002      12 FORMAT(3X,*NION = *,E15.8,15X,*CURRENT = *,E15.8)
000002      13 FORMAT(2X,*THETA,6X,NSC,13X,*SIGMA TOTAL,*,11X,*B PH1,*,14X,
1      10X,THE TA,*,23X,*THE IA/C,*,VA,*,E RADIAL,*)
000002      14 FORMAT(F4.3,2X,E15.8,3X,E15.8)
000002      15 FORMAT(101X,F8.3,7X,E15.8)
000002      16 FORMAT(F8.3,17X,F3(3X,E15.8))
000002      22 FORMAT(2X,*GROUND*,3(1H-),*GROUND*) *E15.8,5X,E15.8,1X,*GROUND*,
1      9(1H-),F8.3,7X,E15.8)
000002      23 FORMAT(F8.3,37X,2(3X,E15.8))

C      OUTAKK(1,1)=I
000002      OUTAKK(1,2)=TPOT-KUC
000004      OUTAKK(1,3)=T-RPHDK
000006      OUTAKK(1,4)=TPUT-RUC
000010      OUTAKK(2,1)=ORNG(KNUM)
000011      OUTAKK(2,2)=F(LGM1,K)*XIRAP*(F(LGM1,K)-F(LGM1,K))
000013      OUTAKK(2,4)=F(LGM1,K)*XIRAP*(F(LGM1,K)-F(LGM1,K))
000024      OUTAKK(3,1)=TPOT-KPHDK
000032      OUTAKK(3,2)=XNM(KPI)
000034      OUTAKK(3,3)=XNE(KPI)
000036      OUTAKK(3,4)=XJAY(R,TPDT)
000040      D0 31, L=1,LMAXP1
000043      LP3=L+3
000044      IF (L-6T.LGND) GO TO 300
000046      OUTAKK(LP3+3)=SIG(L)
000051

```

```

000053      J00 OUTARR(LP3+4)=B(L,N)
000060      OUTARR(LP3+1)=F(L,N)
000063      IF (L-50.LMAXP1) GO TO J10
000064      OUTARR(LP3+2)=E(L,N)
000070      CONTINUE
000073      ICOUNT(KNUM,NFILES)=ICOUNT(KNUM,NFILES)+1

C
000077      IF (KPRINT(KNUM),E4,U) GO TO 370
000100      KPRINT(KNUM)=KPRINT(KNUM)-1
000102      IPAGE = 1
000102      DO 314 J=1,NFILES
000104      IPAGE = IPAGE + ICOUNT(KNUM,J)
000111      CONTINUE
000113      PRINT 9,HEADER,KNUM,IPAGE
000125      PRINT 10,IROWN
000133      PRINT 11, OUTARR(1+J),OUTARR(1+3),OUTARR(1+4),OUTARR(3+1),OUTARR(1
1+2)
000151      PRINT 12,XNM(KP1),OUTARR(3+4)
000161      PRINT 13
000165      PRINT 14, THETA(1),OUTARR(J+3),OUTARR(4+3),OUTARR(4+4),OUTARR(4+1)
000203      DO 321 M=1,LGM1
000205      LHM+4
000207      LHMEN+3
000211      PRINT 15, THETA(M),OUTARR(LM0+2)
000220      PRINT 19, THETA(M+1),OUTARR(L+3),OUTARR(L+4),OUTARR(L+1)
000234      CONTINUE
000237      PRINT 22, OUTARR(2+J),OUTARR(2+2),THETA(LUND),OUTARR(L+2)
000252      LMD0=LMAX-1
000254      DO 331 M=LUND,LMD0
000256      LPJ=M+4
000260      MPE=M+1
000262      PRINT 23, THETA(MP1),OUTARR(LP3+4),OUTARR(LP3+1)
000273      PRINT 15, THETA(MP1),OUTARR(LPJ+2)
000303      CONTINUE
000306      LST=LMAX+4
000310      PRINT 23, THETA(LMAX+1),OUTARR(LST+4),OUTARR(LST+1)
000321      PRINT 1, MSIFT,MSFT,TIME
000333      370 IF (IROWN+1) RETURN

C
C      WRITE OUTPUT DATA ON TAPE JDATA
000335      IROWN=3
000336      ICUL=1
000337      I=J
000340      II=4
000341      J=1
000342      JJ=1
000343      LMAXP3=LMAX+3
000345      DO 400 IPACK=1,NPACK
000347      OUTARR(IK0+ICUL)=PACK(OUTARR(1+J),OUTARR(1+JJ))
000364      IROWN=IROWN+1
000365      IF (IROWN+LE+21) GO TO 385
000367      IROWN=1
000370      ICUL=ICUL+1
000371      I=I+2
000373      II=II+2
000374      IF (I+LT-LMAXP4) GO TO 400
000376      ITEST=1-LMAXP3
000400      GO TO (385,390,392), ITEST

```

```

000406
000407
000410
000411
000412
000413
000414
000415
000416
000417
000420
000422
000425
000427
000431
000432

388 II=1
   JJ=JJ+1
   GO TO 400
390 I=1
   J=J+1
   II=I+1
   JJ=JJ+1
   GO TO 400
392 I=2
   J=J+1
400 CONTINUE
   NPACAS=NPACK+2
   CALL UATAPE(OUTARR,NPACAS+2)
   RETURN
   END

```

```

000002 SUBROUTINE SHIFT
COMMON E(17,20), XNE(20),
1 KPRINT(20), ICOUNT(20),
2 EFACT(16), EFACTB(16),
3 ECT(17), SIG(17),
4 FFACT(17), TH(18), STH(18),
5 MEADER(8), HEAD(8), OUTARK(21),
6 FTEMP(18), FXTRAP(16), THETA(18),
7 MTHETA(17)

C
000002 COMMON /PARAMS/
1 EPS,
2 XION, XIONB,
3 COULOM, CHU, ATTACH, EPS,
4 XELMG, XION, ALF, XELN,
5 KDATA, LMAX, LMAXP1, LMAXP2,
6 INLOAD, INENUM, NSHIFT, KDIAM1,
7 IOATA, KDIAM2, KDIAM3, KDIAM4,
8 KDIAM5, KDIAM6, KDIAM7, KDIAM8,
9 KDIAM9, KDIAM10, KDIAM11, KDIAM12,
10 KDIAM13, KDIAM14, KDIAM15, KDIAM16,
11 KDIAM17, KDIAM18, KDIAM19, KDIAM20,
12 KDIAM21, KDIAM22, KDIAM23, KDIAM24,
13 KDIAM25, KDIAM26, KDIAM27, KDIAM28,
14 KDIAM29, KDIAM30, KDIAM31, KDIAM32,
15 KDIAM33, KDIAM34, KDIAM35, KDIAM36,
16 KDIAM37, KDIAM38, KDIAM39, KDIAM40,
17 KDIAM41, KDIAM42, KDIAM43, KDIAM44,
18 KDIAM45, KDIAM46, KDIAM47, KDIAM48,
19 KDIAM49, KDIAM50, KDIAM51, KDIAM52,
20 KDIAM53, KDIAM54, KDIAM55, KDIAM56,
21 KDIAM57, KDIAM58, KDIAM59, KDIAM60,
22 KDIAM61, KDIAM62, KDIAM63, KDIAM64,
23 KDIAM65, KDIAM66, KDIAM67, KDIAM68,
24 KDIAM69, KDIAM70, KDIAM71, KDIAM72,
25 KDIAM73, KDIAM74, KDIAM75, KDIAM76,
26 KDIAM77, KDIAM78, KDIAM79, KDIAM80,
27 KDIAM81, KDIAM82, KDIAM83, KDIAM84,
28 KDIAM85, KDIAM86, KDIAM87, KDIAM88,
29 KDIAM89, KDIAM90, KDIAM91, KDIAM92,
30 KDIAM93, KDIAM94, KDIAM95, KDIAM96,
31 KDIAM97, KDIAM98, KDIAM99, KDIAM100,
32 KDIAM101, KDIAM102, KDIAM103, KDIAM104,
33 KDIAM105, KDIAM106, KDIAM107, KDIAM108,
34 KDIAM109, KDIAM110, KDIAM111, KDIAM112,
35 KDIAM113, KDIAM114, KDIAM115, KDIAM116,
36 KDIAM117, KDIAM118, KDIAM119, KDIAM120,
37 KDIAM121, KDIAM122, KDIAM123, KDIAM124,
38 KDIAM125, KDIAM126, KDIAM127, KDIAM128,
39 KDIAM129, KDIAM130, KDIAM131, KDIAM132,
40 KDIAM133, KDIAM134, KDIAM135, KDIAM136,
41 KDIAM137, KDIAM138, KDIAM139, KDIAM140,
42 KDIAM141, KDIAM142, KDIAM143, KDIAM144,
43 KDIAM145, KDIAM146, KDIAM147, KDIAM148,
44 KDIAM149, KDIAM150, KDIAM151, KDIAM152,
45 KDIAM153, KDIAM154, KDIAM155, KDIAM156,
46 KDIAM157, KDIAM158, KDIAM159, KDIAM160,
47 KDIAM161, KDIAM162, KDIAM163, KDIAM164,
48 KDIAM165, KDIAM166, KDIAM167, KDIAM168,
49 KDIAM169, KDIAM170, KDIAM171, KDIAM172,
50 KDIAM173, KDIAM174, KDIAM175, KDIAM176,
51 KDIAM177, KDIAM178, KDIAM179, KDIAM180,
52 KDIAM181, KDIAM182, KDIAM183, KDIAM184,
53 KDIAM185, KDIAM186, KDIAM187, KDIAM188,
54 KDIAM189, KDIAM190, KDIAM191, KDIAM192,
55 KDIAM193, KDIAM194, KDIAM195, KDIAM196,
56 KDIAM197, KDIAM198, KDIAM199, KDIAM200,
57 KDIAM201, KDIAM202, KDIAM203, KDIAM204,
58 KDIAM205, KDIAM206, KDIAM207, KDIAM208,
59 KDIAM209, KDIAM210, KDIAM211, KDIAM212,
60 KDIAM213, KDIAM214, KDIAM215, KDIAM216,
61 KDIAM217, KDIAM218, KDIAM219, KDIAM220,
62 KDIAM221, KDIAM222, KDIAM223, KDIAM224,
63 KDIAM225, KDIAM226, KDIAM227, KDIAM228,
64 KDIAM229, KDIAM230, KDIAM231, KDIAM232,
65 KDIAM233, KDIAM234, KDIAM235, KDIAM236,
66 KDIAM237, KDIAM238, KDIAM239, KDIAM240,
67 KDIAM241, KDIAM242, KDIAM243, KDIAM244,
68 KDIAM245, KDIAM246, KDIAM247, KDIAM248,
69 KDIAM249, KDIAM250, KDIAM251, KDIAM252,
70 KDIAM253, KDIAM254, KDIAM255, KDIAM256,
71 KDIAM257, KDIAM258, KDIAM259, KDIAM260,
72 KDIAM261, KDIAM262, KDIAM263, KDIAM264,
73 KDIAM265, KDIAM266, KDIAM267, KDIAM268,
74 KDIAM269, KDIAM270, KDIAM271, KDIAM272,
75 KDIAM273, KDIAM274, KDIAM275, KDIAM276,
76 KDIAM277, KDIAM278, KDIAM279, KDIAM280,
77 KDIAM281, KDIAM282, KDIAM283, KDIAM284,
78 KDIAM285, KDIAM286, KDIAM287, KDIAM288,
79 KDIAM289, KDIAM290, KDIAM291, KDIAM292,
80 KDIAM293, KDIAM294, KDIAM295, KDIAM296,
81 KDIAM297, KDIAM298, KDIAM299, KDIAM300,
82 KDIAM301, KDIAM302, KDIAM303, KDIAM304,
83 KDIAM305, KDIAM306, KDIAM307, KDIAM308,
84 KDIAM309, KDIAM310, KDIAM311, KDIAM312,
85 KDIAM313, KDIAM314, KDIAM315, KDIAM316,
86 KDIAM317, KDIAM318, KDIAM319, KDIAM320,
87 KDIAM321, KDIAM322, KDIAM323, KDIAM324,
88 KDIAM325, KDIAM326, KDIAM327, KDIAM328,
89 KDIAM329, KDIAM330, KDIAM331, KDIAM332,
90 KDIAM333, KDIAM334, KDIAM335, KDIAM336,
91 KDIAM337, KDIAM338, KDIAM339, KDIAM340,
92 KDIAM341, KDIAM342, KDIAM343, KDIAM344,
93 KDIAM345, KDIAM346, KDIAM347, KDIAM348,
94 KDIAM349, KDIAM350, KDIAM351, KDIAM352,
95 KDIAM353, KDIAM354, KDIAM355, KDIAM356,
96 KDIAM357, KDIAM358, KDIAM359, KDIAM360,
97 KDIAM361, KDIAM362, KDIAM363, KDIAM364,
98 KDIAM365, KDIAM366, KDIAM367, KDIAM368,
99 KDIAM369, KDIAM370, KDIAM371, KDIAM372,
100 KDIAM373, KDIAM374, KDIAM375, KDIAM376,
101 KDIAM377, KDIAM378, KDIAM379, KDIAM380,
102 KDIAM381, KDIAM382, KDIAM383, KDIAM384,
103 KDIAM385, KDIAM386, KDIAM387, KDIAM388,
104 KDIAM389, KDIAM390, KDIAM391, KDIAM392,
105 KDIAM393, KDIAM394, KDIAM395, KDIAM396,
106 KDIAM397, KDIAM398, KDIAM399, KDIAM400,
107 KDIAM401, KDIAM402, KDIAM403, KDIAM404,
108 KDIAM405, KDIAM406, KDIAM407, KDIAM408,
109 KDIAM409, KDIAM410, KDIAM411, KDIAM412,
110 KDIAM413, KDIAM414, KDIAM415, KDIAM416,
111 KDIAM417, KDIAM418, KDIAM419, KDIAM420,
112 KDIAM421, KDIAM422, KDIAM423, KDIAM424,
113 KDIAM425, KDIAM426, KDIAM427, KDIAM428,
114 KDIAM429, KDIAM430, KDIAM431, KDIAM432,
115 KDIAM433, KDIAM434, KDIAM435, KDIAM436,
116 KDIAM437, KDIAM438, KDIAM439, KDIAM440,
117 KDIAM441, KDIAM442, KDIAM443, KDIAM444,
118 KDIAM445, KDIAM446, KDIAM447, KDIAM448,
119 KDIAM449, KDIAM450, KDIAM451, KDIAM452,
120 KDIAM453, KDIAM454, KDIAM455, KDIAM456,
121 KDIAM457, KDIAM458, KDIAM459, KDIAM460,
122 KDIAM461, KDIAM462, KDIAM463, KDIAM464,
123 KDIAM465, KDIAM466, KDIAM467, KDIAM468,
124 KDIAM469, KDIAM470, KDIAM471, KDIAM472,
125 KDIAM473, KDIAM474, KDIAM475, KDIAM476,
126 KDIAM477, KDIAM478, KDIAM479, KDIAM480,
127 KDIAM481, KDIAM482, KDIAM483, KDIAM484,
128 KDIAM485, KDIAM486, KDIAM487, KDIAM488,
129 KDIAM489, KDIAM490, KDIAM491, KDIAM492,
130 KDIAM493, KDIAM494, KDIAM495, KDIAM496,
131 KDIAM497, KDIAM498, KDIAM499, KDIAM500,
132 KDIAM501, KDIAM502, KDIAM503, KDIAM504,
133 KDIAM505, KDIAM506, KDIAM507, KDIAM508,
134 KDIAM509, KDIAM510, KDIAM511, KDIAM512,
135 KDIAM513, KDIAM514, KDIAM515, KDIAM516,
136 KDIAM517, KDIAM518, KDIAM519, KDIAM520,
137 KDIAM521, KDIAM522, KDIAM523, KDIAM524,
138 KDIAM525, KDIAM526, KDIAM527, KDIAM528,
139 KDIAM529, KDIAM530, KDIAM531, KDIAM532,
140 KDIAM533, KDIAM534, KDIAM535, KDIAM536,
141 KDIAM537, KDIAM538, KDIAM539, KDIAM540,
142 KDIAM541, KDIAM542, KDIAM543, KDIAM544,
143 KDIAM545, KDIAM546, KDIAM547, KDIAM548,
144 KDIAM549, KDIAM550, KDIAM551, KDIAM552,
145 KDIAM553, KDIAM554, KDIAM555, KDIAM556,
146 KDIAM557, KDIAM558, KDIAM559, KDIAM560,
147 KDIAM561, KDIAM562, KDIAM563, KDIAM564,
148 KDIAM565, KDIAM566, KDIAM567, KDIAM568,
149 KDIAM569, KDIAM570, KDIAM571, KDIAM572,
150 KDIAM573, KDIAM574, KDIAM575, KDIAM576,
151 KDIAM577, KDIAM578, KDIAM579, KDIAM580,
152 KDIAM581, KDIAM582, KDIAM583, KDIAM584,
153 KDIAM585, KDIAM586, KDIAM587, KDIAM588,
154 KDIAM589, KDIAM590, KDIAM591, KDIAM592,
155 KDIAM593, KDIAM594, KDIAM595, KDIAM596,
156 KDIAM597, KDIAM598, KDIAM599, KDIAM600,
157 KDIAM601, KDIAM602, KDIAM603, KDIAM604,
158 KDIAM605, KDIAM606, KDIAM607, KDIAM608,
159 KDIAM609, KDIAM610, KDIAM611, KDIAM612,
160 KDIAM613, KDIAM614, KDIAM615, KDIAM616,
161 KDIAM617, KDIAM618, KDIAM619, KDIAM620,
162 KDIAM621, KDIAM622, KDIAM623, KDIAM624,
163 KDIAM625, KDIAM626, KDIAM627, KDIAM628,
164 KDIAM629, KDIAM630, KDIAM631, KDIAM632,
165 KDIAM633, KDIAM634, KDIAM635, KDIAM636,
166 KDIAM637, KDIAM638, KDIAM639, KDIAM640,
167 KDIAM641, KDIAM642, KDIAM643, KDIAM644,
168 KDIAM645, KDIAM646, KDIAM647, KDIAM648,
169 KDIAM649, KDIAM650, KDIAM651, KDIAM652,
170 KDIAM653, KDIAM654, KDIAM655, KDIAM656,
171 KDIAM657, KDIAM658, KDIAM659, KDIAM660,
172 KDIAM661, KDIAM662, KDIAM663, KDIAM664,
173 KDIAM665, KDIAM666, KDIAM667, KDIAM668,
174 KDIAM669, KDIAM670, KDIAM671, KDIAM672,
175 KDIAM673, KDIAM674, KDIAM675, KDIAM676,
176 KDIAM677, KDIAM678, KDIAM679, KDIAM680,
177 KDIAM681, KDIAM682, KDIAM683, KDIAM684,
178 KDIAM685, KDIAM686, KDIAM687, KDIAM688,
179 KDIAM689, KDIAM690, KDIAM691, KDIAM692,
180 KDIAM693, KDIAM694, KDIAM695, KDIAM696,
181 KDIAM697, KDIAM698, KDIAM699, KDIAM700,
182 KDIAM701, KDIAM702, KDIAM703, KDIAM704,
183 KDIAM705, KDIAM706, KDIAM707, KDIAM708,
184 KDIAM709, KDIAM710, KDIAM711, KDIAM712,
185 KDIAM713, KDIAM714, KDIAM715, KDIAM716,
186 KDIAM717, KDIAM718, KDIAM719, KDIAM720,
187 KDIAM721, KDIAM722, KDIAM723, KDIAM724,
188 KDIAM725, KDIAM726, KDIAM727, KDIAM728,
189 KDIAM729, KDIAM730, KDIAM731, KDIAM732,
190 KDIAM733, KDIAM734, KDIAM735, KDIAM736,
191 KDIAM737, KDIAM738, KDIAM739, KDIAM740,
192 KDIAM741, KDIAM742, KDIAM743, KDIAM744,
193 KDIAM745, KDIAM746, KDIAM747, KDIAM748,
194 KDIAM749, KDIAM750, KDIAM751, KDIAM752,
195 KDIAM753, KDIAM754, KDIAM755, KDIAM756,
196 KDIAM757, KDIAM758, KDIAM759, KDIAM760,
197 KDIAM761, KDIAM762, KDIAM763, KDIAM764,
198 KDIAM765, KDIAM766, KDIAM767, KDIAM768,
199 KDIAM769, KDIAM770, KDIAM771, KDIAM772,
200 KDIAM773, KDIAM774, KDIAM775, KDIAM776,
201 KDIAM777, KDIAM778, KDIAM779, KDIAM780,
202 KDIAM781, KDIAM782, KDIAM783, KDIAM784,
203 KDIAM785, KDIAM786, KDIAM787, KDIAM788,
204 KDIAM789, KDIAM790, KDIAM791, KDIAM792,
205 KDIAM793, KDIAM794, KDIAM795, KDIAM796,
206 KDIAM797, KDIAM798, KDIAM799, KDIAM800,
207 KDIAM801, KDIAM802, KDIAM803, KDIAM804,
208 KDIAM805, KDIAM806, KDIAM807, KDIAM808,
209 KDIAM809, KDIAM810, KDIAM811, KDIAM812,
210 KDIAM813, KDIAM814, KDIAM815, KDIAM816,
211 KDIAM817, KDIAM818, KDIAM819, KDIAM820,
212 KDIAM821, KDIAM822, KDIAM823, KDIAM824,
213 KDIAM825, KDIAM826, KDIAM827, KDIAM828,
214 KDIAM829, KDIAM830, KDIAM831, KDIAM832,
215 KDIAM833, KDIAM834, KDIAM835, KDIAM836,
216 KDIAM837, KDIAM838, KDIAM839, KDIAM840,
217 KDIAM841, KDIAM842, KDIAM843, KDIAM844,
218 KDIAM845, KDIAM846, KDIAM847, KDIAM848,
219 KDIAM849, KDIAM850, KDIAM851, KDIAM852,
220 KDIAM853, KDIAM854, KDIAM855, KDIAM856,
221 KDIAM857, KDIAM858, KDIAM859, KDIAM860,
222 KDIAM861, KDIAM862, KDIAM863, KDIAM864,
223 KDIAM865, KDIAM866, KDIAM867, KDIAM868,
224 KDIAM869, KDIAM870, KDIAM871, KDIAM872,
225 KDIAM873, KDIAM874, KDIAM875, KDIAM876,
226 KDIAM877, KDIAM878, KDIAM879, KDIAM880,
227 KDIAM881, KDIAM882, KDIAM883, KDIAM884,
228 KDIAM885, KDIAM886, KDIAM887, KDIAM888,
229 KDIAM889, KDIAM890, KDIAM891, KDIAM892,
230 KDIAM893, KDIAM894, KDIAM895, KDIAM896,
231 KDIAM897, KDIAM898, KDIAM899, KDIAM900,
232 KDIAM901, KDIAM902, KDIAM903, KDIAM904,
233 KDIAM905, KDIAM906, KDIAM907, KDIAM908,
234 KDIAM909, KDIAM910, KDIAM911, KDIAM912,
235 KDIAM913, KDIAM914, KDIAM915, KDIAM916,
236 KDIAM917, KDIAM918, KDIAM919, KDIAM920,
237 KDIAM921, KDIAM922, KDIAM923, KDIAM924,
238 KDIAM925, KDIAM926, KDIAM927, KDIAM928,
239 KDIAM929, KDIAM930, KDIAM931, KDIAM932,
240 KDIAM933, KDIAM934, KDIAM935, KDIAM936,
241 KDIAM937, KDIAM938, KDIAM939, KDIAM940,
242 KDIAM941, KDIAM942, KDIAM943, KDIAM944,
243 KDIAM945, KDIAM946, KDIAM947, KDIAM948,
244 KDIAM949, KDIAM950, KDIAM951, KDIAM952,
245 KDIAM953, KDIAM954, KDIAM955, KDIAM956,
246 KDIAM957, KDIAM958, KDIAM959, KDIAM960,
247 KDIAM961, KDIAM962, KDIAM963, KDIAM964,
248 KDIAM965, KDIAM966, KDIAM967, KDIAM968,
249 KDIAM969, KDIAM970, KDIAM971, KDIAM972,
250 KDIAM973, KDIAM974, KDIAM975, KDIAM976,
251 KDIAM977, KDIAM978, KDIAM979, KDIAM980,
252 KDIAM981, KDIAM982, KDIAM983, KDIAM984,
253 KDIAM985, KDIAM986, KDIAM987, KDIAM988,
254 KDIAM989, KDIAM990, KDIAM991, KDIAM992,
255 KDIAM993, KDIAM994, KDIAM995, KDIAM996,
256 KDIAM997, KDIAM998, KDIAM999, KDIAM1000,
257 KDIAM1001, KDIAM1002, KDIAM1003, KDIAM1004,
258 KDIAM1005, KDIAM1006, KDIAM1007, KDIAM1008,
259 KDIAM1009, KDIAM1010, KDIAM1011, KDIAM1012,
260 KDIAM1013, KDIAM1014, KDIAM1015, KDIAM1016,
261 KDIAM1017, KDIAM1018, KDIAM1019, KDIAM1020,
262 KDIAM1021, KDIAM1022, KDIAM1023, KDIAM1024,
263 KDIAM1025, KDIAM1026, KDIAM1027, KDIAM1028,
264 KDIAM1029, KDIAM1030, KDIAM1031, KDIAM1032,
265 KDIAM1033, KDIAM1034, KDIAM1035, KDIAM1036,
266 KDIAM1037, KDIAM1038, KDIAM1039, KDIAM1040,
267 KDIAM1041, KDIAM1042, KDIAM1043, KDIAM1044,
268 KDIAM1045, KDIAM1046, KDIAM1047, KDIAM1048,
269 KDIAM1049, KDIAM1050, KDIAM1051, KDIAM1052,
270 KDIAM1053, KDIAM1054, KDIAM1055, KDIAM1056,
271 KDIAM1057, KDIAM1058, KDIAM1059, KDIAM1060,
272 KDIAM1061, KDIAM1062, KDIAM1063, KDIAM1064,
273 KDIAM1065, KDIAM1066, KDIAM1067, KDIAM1068,
274 KDIAM1069, KDIAM1070, KDIAM1071, KDIAM1072,
275 KDIAM1073, KDIAM1074, KDIAM1075, KDIAM1076,
276 KDIAM1077, KDIAM1078, KDIAM1079, KDIAM1080,
277 KDIAM1081, KDIAM1082, KDIAM1083, KDIAM1084,
278 KDIAM1085, KDIAM1086, KDIAM1087, KDIAM1088,
279 KDIAM1089, KDIAM1090, KDIAM1091, KDIAM1092,
280 KDIAM1093, KDIAM1094, KDIAM1095, KDIAM1096,
281 KDIAM1097, KDIAM1098, KDIAM1099, KDIAM1100,
282 KDIAM1101, KDIAM1102, KDIAM1103, KDIAM1104,
283 KDIAM1105, KDIAM1106, KDIAM1107, KDIAM1108,
284 KDIAM1109, KDIAM1110, KDIAM1111, KDIAM1112,
285 KDIAM1113, KDIAM1114, KDIAM1115, KDIAM1116,
286 KDIAM1117, KDIAM1118, KDIAM1119, KDIAM1120,
287 KDIAM1121, KDIAM1122, KDIAM1123, KDIAM1124,
288 KDIAM1125, KDIAM1126, KDIAM1127, KDIAM1128,
289 KDIAM1129, KDIAM1130, KDIAM1131, KDIAM1132,
290 KDIAM1133, KDIAM1134, KDIAM1135, KDIAM1136,
291 KDIAM1137, KDIAM1138, KDIAM1139, KDIAM1140,
292 KDIAM1141, KDIAM1142, KDIAM1143, KDIAM1144,
293 KDIAM1145, KDIAM1146, KDIAM1147, KDIAM1148,
294 KDIAM1149, KDIAM1150, KDIAM1151, KDIAM1152,
295 KDIAM1153, KDIAM1154, KDIAM1155, KDIAM1156,
296 KDIAM1157, KDIAM1158, KDIAM1159, KDIAM1160,
297 KDIAM1161, KDIAM1162, KDIAM1163, KDIAM1164,
298 KDIAM1165, KDIAM1166, KDIAM1167, KDIAM1168,
299 KDIAM1169, KDIAM1170, KDIAM1171, KDIAM1172,
300 KDIAM1173, KDIAM1174, KDIAM1175, KDIAM1176,
301 KDIAM1177, KDIAM1178, KDIAM1179, KDIAM1180,
302 KDIAM1181, KDIAM1182, KDIAM1183, KDIAM1184,
303 KDIAM1185, KDIAM1186, KDIAM1187, KDIAM1188,
304 KDIAM1189, KDIAM1190, KDIAM1191, KDIAM1192,
305 KDIAM1193, KDIAM1194, KDIAM1195, KDIAM1196,
306 KDIAM1197, KDIAM1198, KDIAM1199, KDIAM1200,
307 KDIAM1201, KDIAM1202, KDIAM1203, KDIAM1204,
308 KDIAM1205, KDIAM1206, KDIAM1207, KDIAM1208,
309 KDIAM1209, KDIAM1210, KDIAM1211, KDIAM1212,
310 KDIAM1213, KDIAM1214, KDIAM1215, KDIAM1216,
311 KDIAM1217, KDIAM1218, KDIAM1219, KDIAM1220,
312 KDIAM1221, KDIAM1222, KDIAM1223, KDIAM1224,
313 KDIAM1225, KDIAM1226, KDIAM1227, KDIAM1228,
314 KDIAM1229, KDIAM1230, KDIAM1231, KDIAM1232,
315 KDIAM1233, KDIAM1234, KDIAM1235, KDIAM1236,
316 KDIAM1237, KDIAM1238, KDIAM1239, KDIAM1240,
317 KDIAM1241, KDIAM1242, KDIAM1243, KDIAM1244,
318 KDIAM1245, KDIAM1246, KDIAM1247, KDIAM1248,
319 KDIAM1249, KDIAM1250, KDIAM1251, KDIAM1252,
320 KDIAM1253, KDIAM1254, KDIAM1255, KDIAM1256,
321 KDIAM1257, KDIAM1258, KDIAM1259, KDIAM1260,
322 KDIAM1261, KDIAM1262, KDIAM1263, KDIAM1264,
323 KDIAM1265, KDIAM1266, KDIAM1267, KDIAM1268,
324 KDIAM1269, KDIAM1270, KDIAM1271, KDIAM1272,
325 KDIAM1273, KDIAM1274, KDIAM1275, KDIAM1276,
326 KDIAM1277, KDIAM1278, KDIAM1279, KDIAM1280,
327 KDIAM1281, KDIAM1282, KDIAM1283, KDIAM1284,
328 KDIAM1285, KDIAM1286, KDIAM1287, KDIAM1288,
329 KDIAM1289, KDIAM1290, KDIAM1291, KDIAM1292,
330 KDIAM1293, KDIAM1294, KDIAM1295, KDIAM1296,
331 KDIAM1297, KDIAM1298, KDIAM1299, KDIAM1300,
332 KDIAM1301, KDIAM1302, KDIAM1303, KDIAM1304,
333 KDIAM1305, KDIAM1306, KDIAM1307, KDIAM1308,
334 KDIAM1309, KDIAM1310, KDIAM1311, KDIAM1312,
335 KDIAM1313, KDIAM1314, KDIAM1315, KDIAM1316,
336 KDIAM1317, KDIAM1318, KDIAM1319, KDIAM1320,
337 KDIAM1321, KDIAM1322, KDIAM1323, KDIAM1324,
338 KDIAM1325, KDIAM1326, KDIAM1327, KDIAM1328,
339 KDIAM1329, KDIAM1330, KDIAM1331, KDIAM1332,
340 KDIAM1333, KDIAM1334, KDIAM1335, KDIAM1336,
341 KDIAM1337, KDIAM1338, KDIAM1339, KDIAM1340,
342 KDIAM1341, KDIAM1342, KDIAM1343, KDIAM1344,
343 KDIAM1345, KDIAM1346, KDIAM1347, KDIAM1348,
344 KDIAM1349, KDIAM1350, KDIAM1351, KDIAM1352,
345 KDIAM1353, KDIAM1354, KDIAM1355, KDIAM1356,
346 KDIAM1357, KDIAM1358, KDIAM1359, KDIAM1360,
347 KDIAM1361, KDIAM1362, KDIAM1363, KDIAM1364,
348 KDIAM1365, KDIAM1366, KDIAM1367, KDIAM1368,
349 KDIAM1369, KDIAM1370, KDIAM1371, KDIAM1372,
350 KDIAM1373, KDIAM1374, KDIAM1375, KDIAM1376,
351 KDIAM1377, KDIAM1378, KDIAM1379, KDIAM1380,
352 KDIAM1381, KDIAM1382, KDIAM1383, KDIAM1384,
353 KDIAM1385, KDIAM1386, KDIAM1387, KDIAM1388,
354 KDIAM1389, KDIAM1390, KDIAM1391, KDIAM1392,
355 KDIAM1393, KDIAM1394, KDIAM1395, KDIAM1396,
356 KDIAM1397, KDIAM1398, KDIAM1399, KDIAM1400,
357 KDIAM1401, KDIAM1402, KDIAM1403, KDIAM1404,
358 KDIAM1405, KDIAM1406, KDIAM1407, KDIAM1408,
359 KDIAM1409, KDIAM1410, KDIAM1411, KDIAM1412,
360 KDIAM1413, KDIAM1414, KDIAM1415, KDIAM1416,
361 KDIAM1417, KDIAM1418, KDIAM1419, KDIAM1420,
362 KDIAM1421, KDIAM1422, KDIAM1423, KDIAM1424,
363 KDIAM1425, KDIAM1426, KDIAM1427, KDIAM1428,
364 KDIAM1429, KDIAM1430, KDIAM1431, KDIAM1432,
365 KDIAM1433, KDIAM1434, KDIAM1435, KDIAM1436,
366 KDIAM1437, KDIAM1438, KDIAM1439, KDIAM1440,
367 KDIAM1441, KDIAM1442, KDIAM1443, KDIAM1444,
368 KDIAM1445, KDIAM1446, KDIAM1447, KDIAM1448,
369 KDIAM1449, KDIAM1450, KDIAM1451, KDIAM1452,
370 KDIAM1453, KDIAM1454, KDIAM1455, KDIAM1456,
371 KDIAM1457, KDIAM1458, KDIAM1459, KDIAM1460,
372 KDIAM1461, KDIAM1462, KDIAM1463, KDIAM1464,
373 KDIAM1465, KDIAM1466, KDIAM1467, KDIAM1468,
374 KDIAM1469, KDIAM1470, KDIAM1471, KDIAM1472,
375 KDIAM1473, KDIAM1474, KDIAM1475, KDIAM1476,
376 KDIAM1477, KDIAM1478, KDIAM1479, KDIAM1480,
377 KDIAM1481, K
```

```

000075      CEAN=CCULON*EMOCON*ANE(1)
000077      DO 6 L=L1,LMAXP1
000101      FK(L)=F(L,2)
000103      IF(L.GF.LGNDUP1) GO TO 500
000105      IF(L.GT.1) GO TO 350
000110      EVAL=.5*(E(1,1)+E(1,2))
000113      GO TO 400
000115      LMI=L-1
350      EVAL=.5*(EFACTA(LM1)*E(L,1)+E(L,2))*EFACTB(LM1)*E(LM1,1)+E(LM1,2)
      1))
400      SIG(L)=CEAN*(EMORFN(SUM1(EVAL**2*FK(L)**2)*EMOBCN1)+CXXXPX
      TEMP=SIG(L)*DTEPS
      IF(TEMP.LE.15.) GO TO 450
      X=L*.0
      GO TO 550
450      X=EXP(-TEMP)
      A=(1.-X)/SIG(L)
      GO TO 550
500      IF(L.GT.LGNDUP1) GO TO 550
      X=AGND
      A=AGND
550      F(L,1)=X*FK(L)-A*(MPUX*(L,2)-K*G(L,1))/RPNDR/CMUDR
      IF(ABS(F(L,1)).LT.1E-23) F(L,1)=0.
600      CONTINUE
      R(LMAXP1,1)=H(LMAXP1,2)
      DO 65 L=L1,LMAX
      E(L,1)=E(L,2)
      B(L,1)=B(L,2)
650      CONTINUE
      DO 700 K=2,KDIM
      KPI=K+1
      XNE(K)=XNE(KPI)
      XNM(K)=XNM(KPI)
      F(LMAXP1,K)=F(LMAXP1,KPI)
      H(LMAXP1,K)=H(LMAXP1,KPI)
      DO 700 L=L1,LMAX
      E(L,K)=E(L,KPI)
      F(L,K)=F(L,KPI)
      B(L,K)=B(L,KPI)
700      CONTINUE
      IF(LGND.GT.0) GO TO 800
      XNE(KDIMP1)=0.
      XNM(KDIMP1)=0.
      F(LMAXP1,KDIMP1)=0.
      B(LMAXP1,KDIMP1)=0.
      DO 750 L=L1,LMAX
      E(L,KDIMP1)=0.
      F(L,KDIMP1)=0.
      B(L,KDIMP1)=0.
750      CONTINUE
      RETURN
800      XNE(KDIMP1)=XNES(L,0)
      XNM(KDIMP1)=XNMS(L,0)
      F(LMAXP1,KDIMP1)=F(LMAXP1,LOAD)
      B(LMAXP1,KDIMP1)=BS(LMAXP1,LOAD)
      DO 850 L=L1,LMAX
      E(L,KDIMP1)=ES(L,LOAD)
000075
000077
000101
000103
000105
000110
000113
000115
000125
000141
000143
000145
000146
000150
000151
000155
000160
000161
000165
000166
000167
000200
000203
000204
000206
000210
000211
000213
000215
000217
000221
000222
000227
000234
000236
000243
000250
000256
000262
000265
000266
000267
000272
000275
000276
000301
000303
000306
000310
000313
000314
000321
000326
000330

```

F(L*QJ)M21)=FS(L*LUW)
B(L*QJ)M21)=FS(L*LUW)
850 CONTINUE
RETURN
END

000335
000342
000350
000352
000352


```

000006      SUBROUTINE DATAPE (OUTARR,NPACKS,IOP1)
      COMMON /INTEGER/  KDATA,  ISTORE,
1      NLEAD,  LMAX,  LMAX1,
2      IRENUM,  NOUT,  LDATA,
3      JSTORE,  NSHIFT,  KUIRM1,
4      IDATA
      DIMENSION OUTARR(NPACKS)
      C
      GO TO (100+200), IOP1
000007      100 READ(IDATA) OUTARR
000008      IF(EOF(IDATA) 250+200
000009      200 WRITE(JDATA) OUTARR
000010      200 RETURN
000011      250 IOP1=C
000012      250 RETURN
000013      END
000014
000015      NTIME,
000016      KUIRM,
000017      NSLOCK,
000018      LMAXP2,
000019      LOAD,
000020      KUIRM1,
000021      JDATA,
000022      NPACK,

```

		BLOCK DATA			
	C ***	TEST PROBLEM	***		
000002	C	COMMUN /PARAMS/ 1 EPSGND, 2 XIUMOB,	C, SUND, AELMGE,	COULOM, RMU, XIOCOM,	CMU, ATTACH, ALF, EPS, XELEN,
000002	C	DATA 1 EPSGND, 2 XIUMOB, 3 4 14.14556E-11, 5 2.7E-4, END	C, SUND, AELMGE, 3.88, 5.E-3, 4.51,	COULOM, RMU, XIOCOM, 1.8E-19, 1.493, 2.5E-12,	CMU, ATTACH, ALF/ 12.566E-7, 1.8E, 4.5E-13/ EPS, XELEN, 1.4,

B. ORESTES

1. Description of Modifications

Program ORESTES¹ received two minor modifications since it was last documented. All input/output (I/O) operations using tape or drum were revised in order to cut running time and to make more efficient use of data tapes.

The running time was cut by replacing all lengthy I/O lists, which used implied DO loops, with a variably dimensioned subroutine (IOSAVE) which can write or read any length array as a single operation. This modification should produce up to 90% reduction in running time for the operations which it affects.

More efficient use of data tapes was accomplished by revising the contents of the data output array (OUTARR). By removing some unnecessary data items from the array and repositioning other items, the dimensions of the array were reduced from 27x11 to 25x9 resulting in a decrease of 72 words. The words in the reduced array were then packed two into one by an assembly language packing routine (PACK). The net result was to reduce the number of words written to tape, at a given time step and range step, from 297 to 114. If the number of angles used for a given run is less than 24, the number of words written to tape will be reduced even more.

1. Jones, D. L. and D. H. Stump, ELECTRA, ORESTES, and Supporting Graphic Display Codes, U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, February 1971, pp. 37-69.

2. Program Listing

[illegible]

```

PROGRAM QKSTES(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE7,TAPE8,
1 TAPES=INPUT,TAPE5=OUTPUT)

COMMON E(23,23), F(24,25), H(24,25), XNE(24,25),
1 XNM(24,25), CEX(24), CAXPX(24), CXXPX(24),
2 CEXM(24), KOUT(16), KPRINT(16), ICOUNT(16,10),
3 NDEGM(16), DHTM(23), EFACTA(23), EFACTB(23),
4 HTH(23), ECON(23), EK(23), SIG(24),
5 FFACTA(23), FFACTB(23), TH(24), STH(24),
6 HEADR(16), FTH(24), MEAD(8), OUTARR(25,9),
7 FTEMP(24), FTEMP2(24), FXTRAP(24), THETA(24),
8 HTHETA(23)

COMMON/SOURCE/ XJ1, XJ2, XJ3, XJ4,
1 XJ5, XJ6, XJ7, XJ8, XJ9, XJ10, XJ11, XJ12, XJ13,
2 XJ14, XJ15, XJ16, XJ17, XJ18, XJ19, XJ20, XJ21, XJ22, XJ23, XJ24, XJ25, XJ26, XJ27, XJ28, XJ29, XJ30, XJ31, XJ32, XJ33, XJ34, XJ35, XJ36, XJ37, XJ38, XJ39, XJ40, XJ41, XJ42, XJ43, XJ44, XJ45, XJ46, XJ47, XJ48, XJ49, XJ50, XJ51, XJ52, XJ53, XJ54, XJ55, XJ56, XJ57, XJ58, XJ59, XJ60, XJ61, XJ62, XJ63, XJ64, XJ65, XJ66, XJ67, XJ68, XJ69, XJ70, XJ71, XJ72, XJ73, XJ74, XJ75, XJ76, XJ77, XJ78, XJ79, XJ80, XJ81, XJ82, XJ83, XJ84, XJ85, XJ86, XJ87, XJ88, XJ89, XJ90, XJ91, XJ92, XJ93, XJ94, XJ95, XJ96, XJ97, XJ98, XJ99, XJ100, XJ101, XJ102, XJ103, XJ104, XJ105, XJ106, XJ107, XJ108, XJ109, XJ110, XJ111, XJ112, XJ113, XJ114, XJ115, XJ116, XJ117, XJ118, XJ119, XJ120, XJ121, XJ122, XJ123, XJ124, XJ125, XJ126, XJ127, XJ128, XJ129, XJ130, XJ131, XJ132, XJ133, XJ134, XJ135, XJ136, XJ137, XJ138, XJ139, XJ140, XJ141, XJ142, XJ143, XJ144, XJ145, XJ146, XJ147, XJ148, XJ149, XJ150, XJ151, XJ152, XJ153, XJ154, XJ155, XJ156, XJ157, XJ158, XJ159, XJ160, XJ161, XJ162, XJ163, XJ164, XJ165, XJ166, XJ167, XJ168, XJ169, XJ170, XJ171, XJ172, XJ173, XJ174, XJ175, XJ176, XJ177, XJ178, XJ179, XJ180, XJ181, XJ182, XJ183, XJ184, XJ185, XJ186, XJ187, XJ188, XJ189, XJ190, XJ191, XJ192, XJ193, XJ194, XJ195, XJ196, XJ197, XJ198, XJ199, XJ200, XJ201, XJ202, XJ203, XJ204, XJ205, XJ206, XJ207, XJ208, XJ209, XJ210, XJ211, XJ212, XJ213, XJ214, XJ215, XJ216, XJ217, XJ218, XJ219, XJ220, XJ221, XJ222, XJ223, XJ224, XJ225, XJ226, XJ227, XJ228, XJ229, XJ230, XJ231, XJ232, XJ233, XJ234, XJ235, XJ236, XJ237, XJ238, XJ239, XJ240, XJ241, XJ242, XJ243, XJ244, XJ245, XJ246, XJ247, XJ248, XJ249, XJ250, XJ251, XJ252, XJ253, XJ254, XJ255, XJ256, XJ257, XJ258, XJ259, XJ260, XJ261, XJ262, XJ263, XJ264, XJ265, XJ266, XJ267, XJ268, XJ269, XJ270, XJ271, XJ272, XJ273, XJ274, XJ275, XJ276, XJ277, XJ278, XJ279, XJ280, XJ281, XJ282, XJ283, XJ284, XJ285, XJ286, XJ287, XJ288, XJ289, XJ290, XJ291, XJ292, XJ293, XJ294, XJ295, XJ296, XJ297, XJ298, XJ299, XJ300, XJ301, XJ302, XJ303, XJ304, XJ305, XJ306, XJ307, XJ308, XJ309, XJ310, XJ311, XJ312, XJ313, XJ314, XJ315, XJ316, XJ317, XJ318, XJ319, XJ320, XJ321, XJ322, XJ323, XJ324, XJ325, XJ326, XJ327, XJ328, XJ329, XJ330, XJ331, XJ332, XJ333, XJ334, XJ335, XJ336, XJ337, XJ338, XJ339, XJ340, XJ341, XJ342, XJ343, XJ344, XJ345, XJ346, XJ347, XJ348, XJ349, XJ350, XJ351, XJ352, XJ353, XJ354, XJ355, XJ356, XJ357, XJ358, XJ359, XJ360, XJ361, XJ362, XJ363, XJ364, XJ365, XJ366, XJ367, XJ368, XJ369, XJ370, XJ371, XJ372, XJ373, XJ374, XJ375, XJ376, XJ377, XJ378, XJ379, XJ380, XJ381, XJ382, XJ383, XJ384, XJ385, XJ386, XJ387, XJ388, XJ389, XJ390, XJ391, XJ392, XJ393, XJ394, XJ395, XJ396, XJ397, XJ398, XJ399, XJ400, XJ401, XJ402, XJ403, XJ404, XJ405, XJ406, XJ407, XJ408, XJ409, XJ410, XJ411, XJ412, XJ413, XJ414, XJ415, XJ416, XJ417, XJ418, XJ419, XJ420, XJ421, XJ422, XJ423, XJ424, XJ425, XJ426, XJ427, XJ428, XJ429, XJ430, XJ431, XJ432, XJ433, XJ434, XJ435, XJ436, XJ437, XJ438, XJ439, XJ440, XJ441, XJ442, XJ443, XJ444, XJ445, XJ446, XJ447, XJ448, XJ449, XJ450, XJ451, XJ452, XJ453, XJ454, XJ455, XJ456, XJ457, XJ458, XJ459, XJ460, XJ461, XJ462, XJ463, XJ464, XJ465, XJ466, XJ467, XJ468, XJ469, XJ470, XJ471, XJ472, XJ473, XJ474, XJ475, XJ476, XJ477, XJ478, XJ479, XJ480, XJ481, XJ482, XJ483, XJ484, XJ485, XJ486, XJ487, XJ488, XJ489, XJ490, XJ491, XJ492, XJ493, XJ494, XJ495, XJ496, XJ497, XJ498, XJ499, XJ500, XJ501, XJ502, XJ503, XJ504, XJ505, XJ506, XJ507, XJ508, XJ509, XJ510, XJ511, XJ512, XJ513, XJ514, XJ515, XJ516, XJ517, XJ518, XJ519, XJ520, XJ521, XJ522, XJ523, XJ524, XJ525, XJ526, XJ527, XJ528, XJ529, XJ530, XJ531, XJ532, XJ533, XJ534, XJ535, XJ536, XJ537, XJ538, XJ539, XJ540, XJ541, XJ542, XJ543, XJ544, XJ545, XJ546, XJ547, XJ548, XJ549, XJ550, XJ551, XJ552, XJ553, XJ554, XJ555, XJ556, XJ557, XJ558, XJ559, XJ560, XJ561, XJ562, XJ563, XJ564, XJ565, XJ566, XJ567, XJ568, XJ569, XJ570, XJ571, XJ572, XJ573, XJ574, XJ575, XJ576, XJ577, XJ578, XJ579, XJ580, XJ581, XJ582, XJ583, XJ584, XJ585, XJ586, XJ587, XJ588, XJ589, XJ590, XJ591, XJ592, XJ593, XJ594, XJ595, XJ596, XJ597, XJ598, XJ599, XJ600, XJ601, XJ602, XJ603, XJ604, XJ605, XJ606, XJ607, XJ608, XJ609, XJ610, XJ611, XJ612, XJ613, XJ614, XJ615, XJ616, XJ617, XJ618, XJ619, XJ620, XJ621, XJ622, XJ623, XJ624, XJ625, XJ626, XJ627, XJ628, XJ629, XJ630, XJ631, XJ632, XJ633, XJ634, XJ635, XJ636, XJ637, XJ638, XJ639, XJ640, XJ641, XJ642, XJ643, XJ644, XJ645, XJ646, XJ647, XJ648, XJ649, XJ650, XJ65
```



```

000305 TEMPOR=TEMPOR+DROUT
000307 ORNG(K)=TEMPOR
000311 125 CONTINUE
000313 H=H+DROUT
000314 130 CONTINUE
000316 IF (A.LE.16) GO TO 135
000321 PRINT 25,K
000323 STOP
000325 135 NKOUT=K
000327 140 LGNDPI=LGND+1
000331 LGND=LGND-1
000333 LMAXP1=LMAX+1
000335 LMAXP2=LMAX+2
000337 NPACKE=(LMAXP2*9)/2-1
000341 IREM=MOD(LMAXP2*9,2)
000343 KDIHPI=KDIH+1
000345 KDIHMI=KDIH-1
000351 DELMIN=3.14
000353 DO 125 L=1,LMAXP1
000355 TH(L)=.01745329*THEIA(L)
000357 STH(L)=SIN(TH(L))
000360 IF(L.EQ.1) GO TO 124
000362 DELTRE=TH(L)-TH(L-1)
000364 IF (DELMIN.GT.0) DELTRE=DELTRE
000366 IF (L.EQ.LMAXP1) GO TO 125
000370 122* TH(L)=.01745329*THEIA(L)
000372 1225 CONTINUE
000374 IF(LGND.GT.LMAXP1) GO TO 122
000376 XTRAP=COS(TH(LGM1))/COS(TH(LGM))
000378 RCAL=COUT/(DELMIN*SGMT(I)-(C*UT/DR)*2)) *1.E-5
000380 IF (IRMIN.E.KCAL) GO TO 1227
000382 IF (IRMIN.LE.0) GO TO 1246
000384 PRINT 26, KCAL
000386 STOP
000388 1226 RMIN=KCAL
000390 1227 NMINE=MIN/(C*DT)*1.
000392 RSFART=RMIN
000394 IF (IRENUM.LE.0) GO TO 95
000396 REWIND IDATA
000398 REWIND JDATA
000400 REWIND KDATA
000402 REWIND LUATA
000404
000406 C READ THE RESTART TAPE
000408 C
000410 CALL IOSAVE(XNES,XNMS,E5,FS,BS,E,F,B,XNE,XNM,NLOAD,NMIN,1,1)
000412 DO 1036 J=1,NLOAD
000414 CALL IOSAVE(XNES,XNMS,E5,FS,BS,E(1,2),F(1,2),A(1,2),XNE(1,2),XNM(1,2),NLOAD,NMIN,KDIHMI,2)
000416 CALL IOSAVE(XNES,XNMS,E5,FS,BS,E(1,KDIHPI),F(1,KDIHPI),A(1,KDIHPI),NLOAD,NMIN,1,2)
000418 1036 CONTINUE
000420 C
000422 REWIND ISTORE
000424 C
000426 READ (IDATA) NFILES,NKOUT,LHAX
000428 IF (NFILES.EQ.1) GO TO 1024
000430 WRITE(6,9) NFILES,IRENUM
000432

```

```

000617 STOP
000621 1024 NFILES=IRENUM+1
000623 WRITE (JDATA) NFILES,NKOUT,LMAX
C
C DATA TAPE SETUP LOOP
C
000634 READ(IIDATA) HEAD,IRENUM
000643 DO 1025 J=1,4
000645 IF (HEAD(J).NE.HEADEN(J)) GO TO 1026
000647 1025 CONTINUE
000651 GO TO 1029
C
C THE HEADER READ IN DOES NOT MATCH HEAD ON DATA TAPE
C
C PRINT HEADER,HEAD,EXIT
C
000652 1026 PRINT 1013,HEADER
000660 PRINT 1027,I
000666 PRINT 1028,HEAD
000674 PRINT 1017
000700 STOP
000702 WRITE(JDATA) HEAD,IRENUM
000711 READ(IIDATA) THETA,TIMEIA
000720 WRITE(JDATA) THETA,TIMEIA
000727 NPACKS=NPACK+2
000731 IF (IREM.NE.0) NPACKS=NPACKS+1
000734 IOPT=1
000735 1031 CALL DATAPE(OUTARR,NPACKS,IOPT)
000740 IF (IOPT) 1031,66,1031
C
000742 65 IF (IRENUM.LT.0) GO TO 66
000744 REWIND JDATA
000746 WRITE (JDATA) NFILES,NKOUT,LMAX
000757 WRITE(JDATA) HEADER,IRENUM
000766 WRITE(JDATA) THETA,TIMEIA
000775 66 IF (IRENUM.GT.0) GO TO 70
C
C INITIALIZE ARRAYS TO ZERO ON FIRST PASS
C
001000 DO 68 K=1,KDIMPI
001001 XNE(LMAXPI,K)=0.
001004 XNM(LMAXPI,K)=0.
001007 F(LMAXPI,K)=0.
001012 B(LMAXPI,K)=0.
001014 DO 66 L=1,LMAX
001016 XNE(L,K)=0.
001021 XNM(L,K)=0.
001024 E(L,K)=0.
001027 F(L,K)=0.
001031 B(L,K)=0.
001034 68 CONTINUE
001040 DO 69 L=1,LMAXPI
001042 SIG(L)=0.0
001043 FTEMP(L)=0.0
001044 69 CONTINUE
001046 70 DO 1235 L=1,LMAX
001050 LP1=L+1
001052 ECUN(L)=CHU*(TH(LP1)-TH(L))*SIN(TH(L))
001060 FFACIA(L)=(TH(L)-TH(L))/(TH(LP1)-TH(L))

```

```

001086 FFAC19(L)=1-FFACTA(L)
001087 IF (L.EQ.LMAX) GO TO 1235
001088 DHTH(L)=DTH(LP1)-DTH(L)
001089 EFACTA(L)=(DTH(LP1)-DTH(L))/DHTH(L)
001090 EFACTA(L)=1-EFACTA(L)
001091
001101 CONTINUE
001102 IF (LGND.LE.LMAX) ECUN(LGND)=.5*ECUN(LGND)
001103 COT=C*OT
001104 HDI=C*OT
001105 HDH=.5*OT
001106 DO 1245 K=1,NKOUT
001107 KOUT(K)=(ORNG(K)+HDK-MIN)/DR+.5
001108 R=KMIN-HDK+KOUT(K)*DK
001109 XREG(K)=R/CDT
001110
001110 CONTINUE
001111 WRITE(6,5) RMIN,DR,DT,HCAL,SCALE,RMO
001112 WRITE(6,6) LAB(1),NKOUT,LAB(2),KDIM,LAB(3),NMIN,LAB(4),NMAX
001113 WRITE(6,6) LAB(5),NSTEPS,LAB(6),NSTAKT,LAB(7),LGND,LAB(8),LMAX
001114 WRITE(6,7) (KOUT(I),I=1,NKOUT)
001115 EMUCUN=1.293/RMO
001116 EMUCN=EMUCUN/76000
001117 TEMP=XLEN/34.E-6
001118 OCUN=TEMP/COULOM*ALRVEI
001119 DTEPS = OT/EPG
001120 DTEPSG = OT/EPGND
001121 XGND = EXP(-SGND*DTEPSG)
001122 AGND = (1.0-XGND)/SGND
001123 CMUDK = CMU*DR
001124 DTDR = OT/DR
001125 C1=C*OT/UR
001126 C2=.5*C1*MNIN/DR*NSEG*KUIM-.900000
001127
001128 TIME LOOP
001129 DO 41, N=NMIN,NMAX
001130 T=N*OT
001131 NTIME=N
001132 CALL SECOND(TIME)
001133 IF (TIME-GE.IS) GO TO 42
001134 IF (TIME-LE.NESTRT+SEC) GO TO 103
001135 CALL RESTART
001136 PESTMT=RESTRT+1.
001137 IF (LOAD.NE.0) CALL SEGMENT(1)
001138 TMDI=T-MDT
001139 TMDI=T-MDT
001140 TPUT=T*OT
001141 NSHIFT=C1*N-NSHFTT-SC
001142 IF (NSHIFT.LI.0) NSHIFT=y
001143 NSHIFT=NSHIFT-NSHIFT
001144 DO 2005 L=1,LMAXP1
001145 FK(L)=F(L,1)
001146 FTEM,2(L)=FTEMP(L)
001147 FTEMP(L)=F(L,2)
001148 CONTINUE
001149 IF (NSHIFT.EQ.0) GO TO 104
001150 IF (NSHIFT.EQ.0) GO TO 2010
001151 CALL SHIFT
001152 DO 2009 L=1,LMAXP1
001153
001304
001306
001312
001314
001317
001323
001325
001330
001332
001334
001336
001344
001346
001350
001351
001353
001354
001356
001360
001361
001362
001363

```

```

001395      FXIRAP(L)=3.*F(L,1)-3.*TEMP(L)*FTEMP2(L)
001396      CONTINUE
001397      GO TO 120
001400      DO 20 L=1,LMAXP1
001401      F(L,1)=FXIRAP(L)
001402      CONTINUE
001405      RSTART=RMIN+NSHFTT*U
001407      R=STARI-HDR
001413      CALL AJAY(TMDT,RSI,HT,IMHDT,KSTAR,XJRH,XJT,Q)
001415      DO 110 L=1,LMAXP1
001423      XNE=XNE(L,1)
001425      TEMP=ATTACH*ALF*(XNE*XNM(L,1))
001427      ARG=DT*TEMP
001433      EXPON=EXP(-ARG)
001434      IF (ARG.GE. 1.E-7) GO TO 16
001440      XNE(L,1)=(1.0-ARG)*XNE+DT*Q(L)
001442      GO TO 17
001450      16 FACTOR = 1. - EXPON
001452      XNE(L,1)=EXPON*XNE+FACTOR*Q(L)/TEMP
001457      17 TEMP=XIUCOM*(XNE*XNM(L,1))
001463      ARG = DT*TEMP
001464      EXPON = EXP(-ARG)
001466      IF (ARG.GE. 1.E-7) GO TO 18
001470      XNM(L,1)=XNM(L,1)*(1.-ARG)+DT*ATTACH*XNE
001472      GO TO 21
001501      18 FACTOR = 1. - EXPON
001503      XNM(L,1)=EXPON*XNM(L,1)
001507      IF (TEMP.LE. 0.0) GO TO 21
001514      XNM(L,1)=XNM(L,1)+FACTOR*ATTACH*XNE/TEMP
001517      21 CONTINUE
001521      110 LOADMI=LOAD-1
001522      IF (LOADMI.LT.0) LOADMI=0
001523      C
001524      C
001525      RANGE LOOP
001526      DO 370 K=1,KDIM
001527      R=K*DR
001528      RMDR=P-HDR
001531      RPHDR=K*HNR
001532      RPHDR=K*HNR
001533      RPHDR=K*HNR/C
001535      ROC=R/C
001536      DTOR=DT/R
001540      DTORUR=DTOR/DR
001542      IF (RPHDR.GE.C*THHDI) GO TO 380
001545      K01=K+1
001547      CALL AJAY(TMDT,RPHDR,HT,XJRH,XJT,Q)
001556      DO 1115 L=1,LMAXP1
001560      XNE=XNE(L,KP1)
001564      TEMP=ATTACH*ALF*(XNE*XNM(L,KP1))
001571      ARG = DT*TEMP
001573      EXPON = EXP(-ARG)
001576      IF (ARG.GE. 1.E-7) GO TO 1111
001600      XNE(L,KP1)=(1.0-ARG)*XNE+DT*Q(L)
001611      GO TO 1112
001613      1111 FACTOR = 1. - EXPON
001614      XNE(L,KP1)=EXPON*XNE + FACTOR*Q(L)/TEMP

```

```

001622      TEMP=XI*COM*(XNEX + XNM(L,KP1))
001623      ARG = DT*TEMP
001624      EXPON = EXP(-ARG)
001625      IF (ARG .GE. 1.E-7) GO TO 113
001626      XNM(L,KP1)=XNM(L,KP1)*(1.0-ARG)*DT*ATTACH*XNEX
001627      GO TO 1115
001628
001629      113 FACTOR = 1. - EXPON
001630      114 XNM(L,KP1)=EXPON*XNM(L,KP1)
001631      115 XNM(L,KP1)=XNM(L,KP1) + FACTOR*ATTACH*XNEX/TEMP
001632      CONTINUE
001633
001634      DO 1120 L=1,LMAXP1
001635      CEXN(L)=COULOM*EMOCUN*XNE(L,KP1)
001636      CX2XPX(L)=COULOM*XIONOB*(2.*XNM(L,KP1) + XNE(L,KP1))
001637      XSEC=0.5*(XNE(L,K)+XNE(L,KP1))
001638      XIUN=0.5*(XNM(L,K)+XNM(L,KP1))
001639      CF(L)=COULOM*EMOCUN*XSEC
001640      CX2XPX(L)=COULOM*XIONOB*(2.*XION + XSEC)
001641      1120 CONTINUE
001642
001643      C
001644      C
001645      C      THETA LOOP E
001646      DO 185 L=1,LMAX
001647      EK(L)=E(L,K)
001648      LP1 = L+1
001649      IF (L.GE.LGND) GO TO 150
001650      FVAL=.5*(FFACTA(L)*(F(LP1,KP1)+F(LP1,K))+FFACTB(L)*(F(L,KP1)+F(L,K)
001651      ))
001652      CEXH1=FFACTA(L)*CEX(LP1) + FFACTB(L)*CEX(L)
001653      CX2XHT=FFACTA(L)*CX2XPX(LP1) + FFACTB(L)*CX2XPX(L)
001654      SIGMA=CEXHT*(EMOBN(SQRT(FVAL)*2. + EK(L)*2)*EMOBCN) + CX2XHT
001655      TEMP = SIGMA*DTEPS
001656      IF (TEMP.LE.15.) GO TO 142
001657      X = 0.0
001658      A = 1.0/SIGMA
001659      GO TO 180
001660      142 IF (TEMP.GE.1.E-5) GO TO 145
001661      X=1.-TEMP*.5*TEMP**2
001662      A=(1.-5*TEMP)*DTEPS
001663      GO TO 180
001664      145 X=EXP(-TEMP)
001665      A = (1.0-X)/SIGMA
001666      GO TO 180
001667      150 IF (L.GT.LGND) GO TO 170
001668      SIGMA=CEX(L)*(EMOBN(SQRT(.25*(EK(L-1) + EK(L))*2. + (FK(L) +
001669      F(L,KP1))*2))*EMOBCN) + CX2XPX(L)
001670      TEMP = SIGMA*DTEPS
001671      IF (TEMP.LE.15.) GO TO 162
001672      XAIR = 0.0
001673      AAIR = 1.0/SIGMA
001674      GO TO 165
001675      162 IF (TEMP.GE.1.E-5) GO TO 163
001676      XAIR=1.-TEMP*.5*TEMP**2
001677      AAIR=(1.-5*TEMP)*DTEPS
001678      GO TO 165
001679      163 XAIR=EXP(-TEMP)
001680      AAIR = (1.0-XAIR)/SIGMA
001681      165 AOA1 = AAIR*AGND
001682      X = (AAIR*XGND+AGND*XAIR)/AOA1
001683
001735
001736
001737
001738
001739
001740
001741
001742
001743
001744
001745
001746
001747
001748
001749
001750
001751
001752
001753
001754
001755
001756
001757
001758
001759
001760
001761
001762
001763
001764
001765
001766
001767
001768
001769
001770
001771
001772
001773
001774
001775
001776
001777
001778
001779
001780
001781
001782
001783
001784
001785
001786
001787
001788
001789
001790
001791
001792
001793
001794
001795
001796
001797
001798
001799
001800
001801
001802
001803
001804
001805
001806
001807
001808
001809
001810
001811
001812
001813
001814
001815
001816
001817
001818
001819
001820
001821
001822
001823
001824
001825
001826
001827
001828
001829
001830
001831
001832
001833
001834
001835
001836
001837
001838
001839
001840
001841
001842
001843
001844
001845
001846
001847
001848
001849
001850
001851
001852
001853
001854
001855
001856
001857
001858
001859
001860
001861
001862
001863
001864
001865
001866
001867
001868
001869
001870
001871
001872
001873
001874
001875
001876
001877
001878
001879
001880
001881
001882
001883
001884
001885
001886
001887
001888
001889
001890
001891
001892
001893
001894
001895
001896
001897
001898
001899
001900

```

```

002125      A = AAIR*AGND/AOA1
002127      GO TO 180
002127      170 IF (L.GT.LGNDP1) GO TO 180
002133      X = AGND
002134      A = AGND
002135      180 E(L,K)=X*EK(L) + A*(-XJH(L) + (B(LP1,K)*STM(LP1) - B(L,K)*STM(L))
002156      1/R/ECON(L))
002167      IF(ABS(E(L,K)).LT.1.E-25) E(L,K)=0.
                                TH ELOOP
C
C
C      THETA LOOP F
002172      DO 235 L=2,LMAX
002173      FK(L)=F(L,KP1)
002177      IF (L.GT.1) GO TO 200
002202      EVAL=5*(EK(1)+E(1,KP1))
002206      GO TO 210
002207      200 IF (L.GT.LGND) GO TO 220
002213      LM1 = L-1
002214      EVAL=5*(EFACTA(LM1)*EK(L)+E(L,KP1))*EFACB(LM1)*(EK(LM1)+E(LM1,K
                                (P1)))
002231      210 SIG(L)=CEXN(L)*(EMOBFN(SQRT(EVAL**2+FK(L)**2)*EMOBCN))+CXXXPX(L)
002245      TEMP = SIG(L)*DTEPS
002247      IF (TEMP.LE.15.) GO TO 213
002251      X = 0.0
002252      A = 1.0/SIG(L)
002254      GO TO 230
002255      213 IF (TEMP.GE.1.E-5) GO TO 215
002260      X=1.-TEMP+.5*TEMP**2
002264      A=(1.-5*TEMP)*DTEPS
002267      GO TO 230
002273      215 X=EXP(-TEMP)
002277      A = (1.0-X)/SIG(L)
002303      GO TO 230
002304      220 IF (L.GT.LGNDP1) GO TO 230
002305      X = AGND
002306      230 F(L,KP1)=X*FK(L)-A*(RPDM*B(L,KP1)-R*B(L,K))/RPHDR/CHUDR-A*XJT(L)
002335      IF(ABS(F(L,KP1)).LT.1.E-25) F(L,KP1)=0.
                                TH FLOOP
C
C
C      THETA LOOP B
002340      DO 245 L=2,LMAX
002341      LM1 = L-1
002343      B(L,K)=B(L,K)*DTOR*(E(L,K)-E(LM1,K))/DNTH(LM1)-DTORDR*(RPHDR*F(L,K
                                1P1)-RPHDR*F(L,K))
002371      IF(ABS(B(L,K)).LT.1.E-25) B(L,K)=0.
                                TH BLOOP
C
C
C      OUTPUT EDITING SECTION
002377      KNUM = RANGE NO.
C      K = INDEX OF RANGE LOOP
C      KOUT(KNUM) = INDEX OF RANGE LOOP FOR RANGE KNUM
C      N = INDEX OF TIME LOOP
C      NBEGIN(KNUM) = INDEX OF TIME LOOP FOR RANGE KNUM

```


000003	C	FUNCTION EMOBFI(EOVERP)
000006		IF (EOVERP.GT.0.457) GO TO 3
000011		IF (EOVERP.GT.0.0905) GO TO 2
000014		IF (EOVERP.GT.0.0303) GO TO 1
000020		EMOBF1= C.092/(1.0+3.0*EOVERP)
000027		RETURN
000033		1 EMOBF1=1.009202*EOVERP*(1-34.1606*EOVERP*(352.685-EOVERP*1186.44))
000037		RETURN
000041		2 EMOBF1=0.169267/5001*(EOVERP)
000051		RETURN
000055		3 IF (EOVERP.GT.60.0) GO TO 4
000056		XLN=ALOG(EOVERP)
		EMOBF1=0.16212*EXP(XLN*(0.64912*XLN-0.51686))
		RETURN
		4 EMOBF1=0.03684+0.6579*EOVERP
		RETURN
		END

000003	FUNCTION TIMEP(TAU)	AJ1	AJ2	AJ3	AJ4
	COMMON/SOURCE/	T0	T1	T2	T3
	1	XJUN	CLADA		
	2				
	C				
000003	10	IF(TAU-GE,T2) GO TO 20			
000006		TIMEP=- (7.348E26*EAP (2*E0*TAU)) / (1.01*816E-4*EXP (2.5E8*TAU))			
000024		RETURN			
000024	20	TIMEP=- (2.807E28*EAP (-2.003E8*TAU))			
000032		RETURN			
000032		END			

```

000002 SUBROUTINE RESTART
COMMON E(23,251), F(24,251), XNE(24,251),
1 XNM(24,251), CAZAPX(24), CXYPX(24),
2 CEXN(24), KOUT(16), DPTH(23), EFACTH(23),
3 NBEGIN(16), HTH(23), ECON(23), EK(23), SIG(24),
4 FFACTA(23), FFACB(23), TH(24), STM(24),
5 FK(24), HEAD(8), OUTARR(25),
6 FTEMP(24), FTHETA(23),
7 HTHETA(23)
8

C
000002 COMMON /INTEGER/ KDATA, ISTORE, NTIME, LOAD,
1 NLOAD, LMAX, LMAXP1, KDIM, KUIMPI,
2 IRENUM, KOUT, LDATA, NBLOCK, JDATA,
3 JSTORE, NSHAFT, KDIHM1, LMAXP2, NPACK,
4 IREM, IDATA

C
000002 COMMON /RAND/ R, HOC, RPDR, MPHDR,
1 RPHURC, HSTART, UR, MDR, LGM1,
2 LGND, LGNDP1, KPA, KPA, KNUY,
3 NFILS, NSMPT1, K, EHOCN, EMOBCN,
4 CHUDK, DIEPS, ATRAP, 1, TPDI,
5 TPMDT, THRT, DT, AGND, XGND,
6 ORNG(16), XNM(24), XJT(24), Q(24)

C
000002 COMMON /SAVE/ ES(24,20), FS(24,20), BS(24,20), XNES(24,20),
1 XNM2(24,20)

C
000002 1 FORMAT(///10H NBLOCK = ,I5)
000002 NBLOCK=NBLOCK+1
000002 REWIND LOATA
000004 CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,5)
000006 DO 400 I=1,NLOAD
000006 CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),XNE(1,2),XNM(1
000024 1,2),NLOAD,NTIME,KUIHM1,0)
000026 C/LL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KUIMPI),F(1,KUIMPI),B(1,KUIMPI)
000053 1,XNE(1,KUIMPI),XNM(1,KUIMPI),NLOAD,NTIME,1,6)
400 CONTINUE
000105 END FILE LOATA
000110 REWIND ISTORE
000112 PRINT 1, NBLOCK
000114 RETURN
000122 END
000123

```

```

000003      SUBROUTINE SEGMENT(ISECT)
COMMON      E(23, 51), F(24, 251), H(24, 251), ANE(24, 251),
1          XNM(24, 251), CEX(24), CA2XPA(24), KPRINT(16), COUNT(16, 10),
2          CEX(24), KOUT(16), UMTM(23), EFACTB(23), EFACTA(23),
3          NREGIN(16), ECOM(23), EK(23), SIG(24),
4          HTH(23), FFACTB(23), FFACTA(23), IM(24), STM(24),
5          FFACTB(23), FFACTA(23), HEAD(4), OUTA(25, 9),
6          FTEMP(24), FTEMP2(24), FATMAP(24), THETA(24),
7          HTHETA(23)
8
C          COMMON /INTEGER/ KUNTA, ISTORE, NTIME, LOAD,
1          NLOAD, LMAX, LMAP1, KJIM, KUIMP1,
2          IRENUM, YADU, LDATA, NBLOCK, JDATA,
3          JSTORE, NSHIFT, KUIMP1, LMAP2, NPACK,
4          IREM, IDATA
C
000003      COMMON /SAVE/ ES(23, 20), FS(24, 20), BS(24, 20), XNES(24, 20),
1          XNMS(24, 20)
C
000003      GO TO (100, 200, 300, 500), ISECT
C
C      READ IN FROM DISK LEFT MOST SEG TO START NEW LEFT MUST SEG AT
C      NEW DELTA T
C
100 CALL IOSAVE(XNES, XNMS, ES, FS, BS, E, F, B, XNE, XNM, NLOAD, NTIME, 1, 7)
CALL IOSAVE(XNES, XNMS, ES, FS, BS, E, F, B, XNE, XNM, NLOAD, NTIME, 1, 7)
1, 2) NLOAD, NTIME, KUIMP1, 7)
CALL IOSAVE(XNES, XNMS, ES, FS, BS, E, F, B, XNE, XNM, NLOAD, NTIME, 1, 7)
1, 2) NLOAD, NTIME, KUIMP1, 7)
1, 2) NLOAD, NTIME, KUIMP1, 7)
RETURN
C
200 IF (LOAD, E, 0) LOAD=1
C
C      SAVE SEGMENT JUST COMPUTED ON DISK TO FREE MEMORY
C
CALL IOSAVE(XNES, XNMS, ES, FS, BS, E, F, B, XNE, XNM, NLOAD, NTIME, 1, 8)
CALL IOSAVE(XNES, XNMS, ES, FS, BS, E, F, B, XNE, XNM, NLOAD, NTIME, 1, 8)
1, 2) NLOAD, NTIME, KUIMP1, 8)
C
IF (LOAD, E, 1) GO TO 420
LOAD=LOAD-1
DO 21 L=1, LMAX
ES(L, LOAD)=E(L, 2)
FS(L, LOAD)=F(L, 2)
BS(L, LOAD)=B(L, 2)
XNES(L, LOAD)=XNE(L, 2)
XNMS(L, LOAD)=XNM(L, 2)
210 CONTINUE
FS(LMAX, LOAD)=F(LMAX, 1, 2)
BS(LMAX, LOAD)=B(LMAX, 1, 2)
XNES(LMAX, LOAD)=XNE(LMAX, 1, 2)
XNMS(LMAX, LOAD)=XNM(LMAX, 1, 2)
C
C      LOAD IS INCREMENTED WHEN THE RANGE LOOP TERMINATES
C      NORMALLY. LOAD = THE NUMBER OF SEGMENTS ON DISK.
C

```

```

000235      220 LOAD=LOAD+1
000237      DO 23 L=1,LMAX
000240      XNE(L,1)=XNE(L,KOIMP1)
000244      XNM(L,1)=XNM(L,KOIMP1)
000247      E(L,1)=E(L,KOIMP1)
000252      F(L,1)=F(L,KOIMP1)
000255      B(L,1)=B(L,KOIMP1)
000260      230 CONTINUE
000262      XNE(LMAXP1,1)=XNE(LMAXP1,KOIMP1)
000266      XNM(LMAXP1,1)=XNM(LMAXP1,KOIMP1)
000271      F(LMAXP1,1)=F(LMAXP1,KOIMP1)
000274      B(LMAXP1,1)=B(LMAXP1,KOIMP1)
000277      IF (LOAD.LE.NLOAD) GO TO 250

C      NLOAD IS ADVANCED WHEN A NEW COMPLETE SEGMENT W/ KOIM MEMBERS
C      IS ADDED TO THE SEGMENTS BEING HELD ON DISK. NLOAD IS = TO
C      NUMBER OF SEGMENTS ON DISK + 1
C
      NLOAD=NLOAD+1
      DO 24 K=2,KOIMP1
000302      XNE(LMAXP1,K)=0.
000303      XNM(LMAXP1,K)=0.
000305      F(LMAXP1,K)=0.
000310      B(LMAXP1,K)=0.
000313      240 R(L,K)=0.
000316      B(LMAXP1,K)=0.
000320      DO 24 L=1,LMAX
000322      XNE(L,K)=0.
000325      XNM(L,K)=0.
000328      E(L,K)=0.
000330      F(L,K)=0.
000333      240 R(L,K)=0.
000335      B(L,K)=0.
000338      RETURN
000344      250 IF (NSHIFT.NE.2) GO TO 250
000345

C      READ IN FROM DISK NEXT SEG TO COMP SAME SEG AT NEW DELTA T
C
      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),R(1,2),XNE(1,2),XNM(1
000346      1,2),NLOAD,NTIME,KOIMP1,7)
000375      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KOIMP1),F(1,KOIMP1),B(1,KOIMP1)
000430      1,XNE(1,KOIMP1),XNM(1,KOIMP1),NLOAD,NTIME,1,7)
      RETURN

C      KOIMP2=KOIMP-2
000431      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),R(1,2),XNE(1,2),XNM(1
000433      1,2),NLOAD,NTIME,KOIMP2,9)
000462      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KOIM),F(1,KOIM),B(1,KOIM),XNE(1
000515      1,KOIM),XNM(1,KOIM),NLOAD,NTIME,1,7)
000520      IF (LOAD.EQ.NLOAD) GO TO 275
000522      DO 27 L=1,LMAX
000531      E(L,KOIMP1)=ES(L,LOAD)
000536      F(L,KOIMP1)=FS(L,LOAD)
000544      XNE(L,KOIMP1)=XNES(L,LOAD)
000551      XNM(L,KOIMP1)=XNMS(L,LOAD)
000557      270 CONTINUE
000561      F(LMAXP1,KOIMP1)=FS(LMAXP1,LOAD)
000567      B(LMAXP1,KOIMP1)=BS(LMAXP1,LOAD)
000574      XNE(LMAXP1,KOIMP1)=XNES(LMAXP1,LOAD)
000602      XNM(LMAXP1,KOIMP1)=XNMS(LMAXP1,LOAD)

```



```

000002 SUBROUTINE PRINTER
COMMON E(23,251), F(24,251), B(24,251), XNE(24,251),
1 CXXPX(24), CXXAPX(24),
2 KOUT(16), COUNT(16,10),
3 EFACTA(23), EFACTB(23),
4 EK(23), SIG(24),
5 FFACTA(23), TH(24), STH(24),
6 FK(24), HEAD(8), OUTARR(24,9),
7 FTEMP(24), FATRAP(24), THETA(24),
8 HTHETA(23)

C **NOTE ** WHEN DIMENSIONS OF OUTARR ARE CHANGED, IROW CHECK
C SHOULD BE CHANGED
C
000002 COMMON /INTEGER/
1 NLOAD, KDATA, ISTORE, NTIME, LOAD,
2 IRENUM, KOUT, LMAXP1, KDIH, KDIHP1,
3 JSTORE, NSHIFT, LDATA, NBLOCK, JDATA,
4 IREM, IDATA, KDIHM1, LMAXP2, NPACK,

C COMMON /RANUT/
1 RPHDR, R, RUC, RPCR, RPHDR,
2 LGNU, KSIAMT, UK, HDR, LGH1,
3 NFILES, NSHIFT, K, KPI, KNUV,
4 CMUDR, DTGPS, XTRAP, EMOCN, EMOCY,
5 TPDIT, TMDIT, DT, AGND, XGND,
6 ORNG(16), XJUT(24), U(24)

C
000002 1 FORMAT(///3116)
000002 9 FORMAT(1H1,///1X,B1,12X,RANGE NU, *12,11X,PAGE *,13)
000002 10 FORMAT(11X,TIME MINUSEC,*,6X,TIME TOT SIGMA,*,6X,TIME B PH1,*,
1 1X,TIME E THETA,*,17X,IRENUM = *,13,5X,TIME E RAUVAL,*)
000002 11 FORMAT(15X,*,5X,E15.8),1X,E15.8)
000002 12 FORMAT(13X,NTION = *,E15.8,15X,CURRENT = *,E15.8)
000002 13 FORMAT(12X,THETA,*,10X,NSC,*,9X,NUON,*,7X,XJR,*,9X,XJT,*,10X,*,U*,
17X,SIGMA TOTAL,*,3X,*,PH1,*,7X,E THETA,*,6X,THETA/2,*,3X,
2 E K-DIAL,*)
000002 14 FORMAT (F0.3,2E12.4,*,12X,E12.4)
000002 15 FORMAT (32X,E12.4,*,6X,F0.3,E12.4)
000002 19 FORMAT(F8.3,17X,3D,E15.8)
000002 22 FORMAT (2X,GROUND,*,2X,(1H-),E12.4,2X,27(1H-),GROUND,*,1X,2E12.4,
1 3X,*,3,E12.4)
000002 23 FORMAT(F4.3,2E12.4,*,12X,E12.4,*,12X,2E12.4)

C
000002 OUTARR(1,1)=1
000002 OUTARR(1,2)=1-RPHDR
000004 OUTARR(1,3)=OUTARR(1,2)
000006 OUTARR(1,4)=TPUT-ROL
000007 OUTARR(1,5)=TPUT-RPHDR
000011 OUTARR(1,7)=TPUT-RUC
000012 OUTARR(2,1)=ORNG(KNUV)
000014 IF (LOND.GT.LMAXP1) GO TO 200
000016 OUTARR(1,8)=B(LGH1,K)*XIRAP*(B(LGH1,K)-B(LGH1,K))
000021 OUTARR(1,9)=F(LGH1,K)*XIRAP*(F(LGH1,K)-F(LGH1,K))
000031 OUTARR(1,9)=F(LGH1,K)*XIRAP*(F(LGH1,K)-F(LGH1,K))
000037 200 DO 310 L=1,LMAXP1
000041 LPI=L+1
000043 LPE=L+2

```

```

000045      IF (L-GE.LMND) GO TO 300
000050      OUTARR(LP1,3)=SIG(L)
000052      300 OUTARR(LP1,4)=B(L,K)
000057      OUTARR(LP1,5)=F(L,K)
000062      OUTARR(LP1,6)=Q(L)
000063      OUTARR(LP1,7)=XNM(L,LP1)
000067      OUTARR(LP1,8)=XJT(L)
000070      OUTARR(LP1,9)=XNE(L,LP1)
000074      IF (L-GE.LMAXP1) GO TO 310
000076      OUTARR(LP1,7)=E(L,K)
000102      OUTARR(LP2,1)=XJRH(L)
000104      CONTINUE
000107      ICOUNT(KNUM,NFILES)=ICOUNT(KNUM,NFILES)+1
C
000112      IF (KPRINT(KNUM).EQ.0) GO TO 340
000113      KPRINT(KNUM)=KPRINT(KNUM)+1
000115      IPAGE = 0
000117      DO 314 J=1,NFILES
000123      IPAGE = IPAGE + ICOUNT(KNUM,J)
000125      314 CONTINUE
000127      PRINT 9,HEADER,KNUM,IPAGE
000137      PRINT 10,IENUM
000145      PRINT 11,(OUTARR(1,J),J=2,5),OUTARR(1,7)
000162      PRINT 13,(OUTARR(1,2),OUTARR(2,2),OUTARR(2,8),OUTARR(2,9),OUTARR(2,6),
000166      1,OUTARR(2,3),J=3,5)
000213      LIMIT=LMG1
000215      IF (LMND-GE.LMAX) LIMIT=LMAX
000220      DO 32 M=1,LIMIT
000222      LM=M+2
000224      LM=M+1
000225      PRINT 15,OUTARR(L,1),MTHETA(M),OUTARR(LMO,7)
000237      PRINT 14,MTHETA(LMO),OUTARR(L,2),OUTARR(L,8),OUTARR(L,9),OUTARR(L,
000237      1,6),OUTARR(L,J),J=3,5)
000265      320 CONTINUE
000270      IF (LMND-GE.LMAX) GO TO 370
000273      PRINT 22,OUTARR(L+1,1),OUTARR(1,8),OUTARR(1,9),MTHETA(LMO),OUTARR
000310      1(L,7)
000312      IF (LMND-GE.LMAX) GO TO 370
000314      LMHO=LMAX+1
000315      DO 34 M=LND,LMO
000317      LM=M+1
000321      LM=M+2
000323      PRINT 23,MTHETA(L),OUTARR(LP3,2),OUTARR(LP3,8),OUTARR(LP3,9),
000350      1 OUTARR(LP3,6),OUTARR(LP3,J),J=4,5)
000362      PRINT 15,OUTARR(MP3,1),MTHETA(L),OUTARR(LP3,7)
000366      330 CONTINUE
000371      340 LST=LMAX+2
000371      LST=LMAX+1
000371      PRINT 23,MTHETA(LST),OUTARR(LST,2),OUTARR(LST,8),OUTARR(LST,9),
000416      1 OUTARR(LST,6),OUTARR(LST,J),J=4,5)
000430      370 PRINT 1,NSHIFT,NSHIFT,TIME
000430      380 IF (IENUM.LI.0) RETURN
C
C      WRITE OUTPUT DATA ON TAPE JDATA
000432      IRUN=3
000433      ICUL=1

```

```

000434      I=J
000435      II=4
000436      J=1
000437      JJ=1
000440      DO 400 (PACK=1,NPACK
000442      OUTARR(I*II*ICOL)=PACK(OUTARR(I,J),OUTARR(II,JJ))
000457      IROW=I*II*J+1
000460      IF (I*II*J*J) GO TO 305
000462      IROW=I
000463      ICOL=ICOL+1
000464      I=I+2
000466      II=II+2
000467      IF (I*II*J*J) GO TO 400
000471      ITC=I-LMAXP1
000473      GO TO (308,398,392), ITC
000501      388 II=1
000502      JJ=JJ+1
000504      GO TO 400
000504      390 I=1
000505      JJ=J+1
000507      II=II+1
000510      JJ=JJ+1
000511      GO TO 400
000512      392 I=4
000513      J=J+1
000515      400 CONTINUE
000520      NPACKS=NPACK+2
000522      IF (I*II*J*J) GO TO 410
000523      OUTARR(I*II*ICOL)=OUTARR(LMAXP2+9)
000530      NPACKS=NPACKS+1
000531      410 CALL DATAPL(OUTARR,NPACKS,2)
000534      RETURN
000535      END

```

```

000002 SUBROUTINE SHIFT
COMMON E(23,25), F(24,25), R(24,25), ANE(24,25),
1 XNM(24,25), CEX(24), CXXAPX(24),
2 CEXN(24), KOUT(10), COUNT(16,10),
3 NREGIN(16), DITH(23), EFACT(23),
4 HTM(23), ECON(23), FK(23), SIG(24),
5 FFACT(23), FFACTB(23), TH(24), STM(24),
6 FK(24), HEADR(10), TH(24), OUTARR(25),
7 FTEMP(24), FTEMP2(24), FATHAP(24),
8 HTMETHA(23)

C
000002 COMMON /PARAMS/
1 EPSGND, SUND, C, COULOM, CMU, EPS,
2 XIONR, AELAGE, RHO, ATTACH, XELE,
3 ITMETH, ITHRET, SCALE, ALF, QCON

C
000002 COMMON /INTEGEN/
1 NLOAD, KUATA, ISTORE, NTIME, LOAD,
2 IRENUM, LMAX, LMAAP1, KDIH, KDIKPI,
3 JUSTRE, NSHIFT, LMAI4, NBLOCK, JDATA,
4 IREM, IDATA, KDIH1, LMAIP2, NPACK

C
000002 COMMON /HANDT/
1 RPHDR, R, RUC, RPHDR, RPHDR,
2 LGND, LMAIP1, UK, LGM1,
3 NFILES, NSHIFT, K, KPI, KNUM,
4 CHUDR, STEPS, XTMAP, EMOCON, EMOBCN,
5 TPHDT, INROT, UT, AGND, XGND,
6 ORNG(16), XJH(24), XJT(24), Q(24)

C
000002 COMMON /SAVE/ ES(23,20), FS(24,20), PS(24,20), XNES(24,20),
1 XNMST(24,20)
R=KSTART+HDR
RPHDR=R+HDR
RPHDR=R+HDR
CALL XJAY(TPHDT, RPHDR, I, R, XJH, XJT, Q)
DO 1115 L=1, LMAIP1
XNEX=XNM(L,2)
XNM=XNM(L,2)
TEMP=ATTACH + ALF*(ANE(X)*XNM)
ARG = DT*TEMP
EXPON = EXP(-ARG)
IF (ARG .GE. 1.E-7) GO TO 1111
XNE(L,1)=1.0-ARG, ANEX=DT*Q(L)
GO TO 1112
1111 FACTOR = 1. - EXPON
XNE(L,1)=EXPON*XNEX + FACTOR*Q(L)/TEMP
1112 TEMP=XIUCUM*(XNEX+XNM),
ARG = DT*TEMP
EXPON = EXP(-ARG)
IF (ARG .GE. 1.E-7) GO TO 113
XNM(L,1)=XNM*(1. - ARG) + DT*ATTACH*XNEX
GO TO 1115
1113 FACTOR = 1. - EXPON
1114 XNM(L,1)=EXPON*XNM
1115 CONTINUE

```

```

000111 DO 1120 L=1,LMAXP1
000112 CEAN(L)=COULOM*CHOCUN*XNE(L,1)
000113 CXXPX(L)=COULOM*CHOCUN*(2.*XNM(L,1) + ANE(L,1))
000114 1120 CONTINUE
000115 DO 245 L=2,LMAX
000116 FK(L)=F(L,2)
000117 IF (L,GT,1) GO TO 240
000118 EVAL=.5*(E(L,1)+E(L,2))
000119 GO TO 210
000120 200 IF (L,GT,LGND) GO TO 220
000121 LM=L-1
000122 EVAL=.5*(EFACTA(LM1)*(E(L,1)+E(L,2))+EFACTB(LM1)*(E(LM1,1)+E(LM1,2)
000123 1))
000124 210 SIG(L)=CEAN(L)*(CHOCUN*(CHOCUN*(EVAL*(FK(L)*2)+EFACTB(LM1)*2)+EFACTA(LM1)*2)
000125 TEMP = SIG(L)*UTEPS
000126 IF (TEMP,LE,15.) GO TO 213
000127 X = 0.0
000128 A = 1.0/TEMP(L)
000129 GO TO 230
000130 213 IF (TEMP,GE,1.E-5) GO TO 215
000131 X=1.-TEMP*.5*TEMP**2
000132 A=(1.-.5*TEMP)*UTEPS
000133 GO TO 230
000134 215 XEXP(-TEMP)
000135 A = (1.-X)/SIG(L)
000136 GO TO 230
000137 220 IF (L,GT,LGNDP1) GO TO 230
000138 X = AGNU
000139 A = AGNU
000140 230 F(L,1)=X*FK(L)-A*(CHOCUN*(L,2)-R*B(L,1))/RPMON/CHUDR-A*XJT(L)
000141 IF (ABS(F(L,1)),LT,1.E-20) F(L,1)=0.0
000142 235 CONTINUE
000143 B(LMAXP1,1)=B(LMAXP1,2)
000144 DO 650 L=1,LMAX
000145 E(L,1)=E(L,2)
000146 B(L,1)=B(L,2)
000147 650 CONTINUE
000148 DO 700 K=2,KDIM
000149 KPI=K+1
000150 F(LMAXP1,K)=F(LMAXP1,KPI)
000151 B(LMAXP1,K)=B(LMAXP1,KPI)
000152 XNE(LMAXP1,K)=XNE(LMAXP1,KPI)
000153 XNM(LMAXP1,K)=XNM(LMAXP1,KPI)
000154 DO 700 L=1,LMAX
000155 E(L,K)=E(L,KPI)
000156 F(L,K)=F(L,KPI)
000157 B(L,K)=B(L,KPI)
000158 XNE(L,K)=XNE(L,KPI)
000159 XNM(L,K)=XNM(L,KPI)
000160 700 CONTINUE
000161 IF (LOAD,GT,0) GO TO 800
000162 F(LMAXP1,KDIMP1)=0.
000163 B(LMAXP1,KDIMP1)=0.
000164 XNE(LMAXP1,KDIMP1)=0.0
000165 XNM(LMAXP1,KDIMP1)=0.0
000166 DO 750 L=1,LMAX
000167 E(L,KDIMP1)=0.
000168 F(L,KDIMP1)=0.
000169 750 CONTINUE
000170 800
000171 F(LMAXP1,KDIMP1)=0.
000172 B(LMAXP1,KDIMP1)=0.
000173 XNE(LMAXP1,KDIMP1)=0.0
000174 XNM(LMAXP1,KDIMP1)=0.0
000175 DO 750 L=1,LMAX
000176 E(L,KDIMP1)=0.
000177 F(L,KDIMP1)=0.
000178 750 CONTINUE
000179 800
000180 F(LMAXP1,KDIMP1)=0.
000181 B(LMAXP1,KDIMP1)=0.
000182 XNE(LMAXP1,KDIMP1)=0.0
000183 XNM(LMAXP1,KDIMP1)=0.0
000184 DO 750 L=1,LMAX
000185 E(L,KDIMP1)=0.
000186 F(L,KDIMP1)=0.
000187 750 CONTINUE
000188 800
000189 F(LMAXP1,KDIMP1)=0.
000190 B(LMAXP1,KDIMP1)=0.
000191 XNE(LMAXP1,KDIMP1)=0.0
000192 XNM(LMAXP1,KDIMP1)=0.0
000193 DO 750 L=1,LMAX
000194 E(L,KDIMP1)=0.
000195 F(L,KDIMP1)=0.
000196 750 CONTINUE
000197 800
000198 F(LMAXP1,KDIMP1)=0.
000199 B(LMAXP1,KDIMP1)=0.
000200 XNE(LMAXP1,KDIMP1)=0.0
000201 XNM(LMAXP1,KDIMP1)=0.0
000202 DO 750 L=1,LMAX
000203 E(L,KDIMP1)=0.
000204 F(L,KDIMP1)=0.
000205 750 CONTINUE
000206 800
000207 F(LMAXP1,KDIMP1)=0.
000208 B(LMAXP1,KDIMP1)=0.
000209 XNE(LMAXP1,KDIMP1)=0.0
000210 XNM(LMAXP1,KDIMP1)=0.0
000211 DO 750 L=1,LMAX
000212 E(L,KDIMP1)=0.
000213 F(L,KDIMP1)=0.
000214 750 CONTINUE
000215 800
000216 F(LMAXP1,KDIMP1)=0.
000217 B(LMAXP1,KDIMP1)=0.
000218 XNE(LMAXP1,KDIMP1)=0.0
000219 XNM(LMAXP1,KDIMP1)=0.0
000220 DO 750 L=1,LMAX
000221 E(L,KDIMP1)=0.
000222 F(L,KDIMP1)=0.
000223 750 CONTINUE
000224 800
000225 F(LMAXP1,KDIMP1)=0.
000226 B(LMAXP1,KDIMP1)=0.
000227 XNE(LMAXP1,KDIMP1)=0.0
000228 XNM(LMAXP1,KDIMP1)=0.0
000229 DO 750 L=1,LMAX
000230 E(L,KDIMP1)=0.
000231 F(L,KDIMP1)=0.
000232 750 CONTINUE
000233 800
000234 F(LMAXP1,KDIMP1)=0.
000235 B(LMAXP1,KDIMP1)=0.
000236 XNE(LMAXP1,KDIMP1)=0.0
000237 XNM(LMAXP1,KDIMP1)=0.0
000238 DO 750 L=1,LMAX
000239 E(L,KDIMP1)=0.
000240 F(L,KDIMP1)=0.
000241 750 CONTINUE
000242 800
000243 F(LMAXP1,KDIMP1)=0.
000244 B(LMAXP1,KDIMP1)=0.
000245 XNE(LMAXP1,KDIMP1)=0.0
000246 XNM(LMAXP1,KDIMP1)=0.0
000247 DO 750 L=1,LMAX
000248 E(L,KDIMP1)=0.
000249 F(L,KDIMP1)=0.
000250 750 CONTINUE
000251 800
000252 F(LMAXP1,KDIMP1)=0.
000253 B(LMAXP1,KDIMP1)=0.
000254 XNE(LMAXP1,KDIMP1)=0.0
000255 XNM(LMAXP1,KDIMP1)=0.0
000256 DO 750 L=1,LMAX
000257 E(L,KDIMP1)=0.
000258 F(L,KDIMP1)=0.
000259 750 CONTINUE
000260 800
000261 F(LMAXP1,KDIMP1)=0.
000262 B(LMAXP1,KDIMP1)=0.
000263 XNE(LMAXP1,KDIMP1)=0.0
000264 XNM(LMAXP1,KDIMP1)=0.0
000265 DO 750 L=1,LMAX
000266 E(L,KDIMP1)=0.
000267 F(L,KDIMP1)=0.
000268 750 CONTINUE
000269 800
000270 F(LMAXP1,KDIMP1)=0.
000271 B(LMAXP1,KDIMP1)=0.
000272 XNE(LMAXP1,KDIMP1)=0.0
000273 XNM(LMAXP1,KDIMP1)=0.0
000274 DO 750 L=1,LMAX
000275 E(L,KDIMP1)=0.
000276 F(L,KDIMP1)=0.
000277 750 CONTINUE
000278 800
000279 F(LMAXP1,KDIMP1)=0.
000280 B(LMAXP1,KDIMP1)=0.
000281 XNE(LMAXP1,KDIMP1)=0.0
000282 XNM(LMAXP1,KDIMP1)=0.0
000283 DO 750 L=1,LMAX
000284 E(L,KDIMP1)=0.
000285 F(L,KDIMP1)=0.
000286 750 CONTINUE
000287 800
000288 F(LMAXP1,KDIMP1)=0.
000289 B(LMAXP1,KDIMP1)=0.
000290 XNE(LMAXP1,KDIMP1)=0.0
000291 XNM(LMAXP1,KDIMP1)=0.0
000292 DO 750 L=1,LMAX
000293 E(L,KDIMP1)=0.
000294 F(L,KDIMP1)=0.
000295 750 CONTINUE
000296 800
000297 F(LMAXP1,KDIMP1)=0.
000298 B(LMAXP1,KDIMP1)=0.
000299 XNE(LMAXP1,KDIMP1)=0.0
000300 XNM(LMAXP1,KDIMP1)=0.0
000301 DO 750 L=1,LMAX
000302 E(L,KDIMP1)=0.
000303 F(L,KDIMP1)=0.
000304 750 CONTINUE
000305 800
000306 F(LMAXP1,KDIMP1)=0.
000307 B(LMAXP1,KDIMP1)=0.
000308 XNE(LMAXP1,KDIMP1)=0.0
000309 XNM(LMAXP1,KDIMP1)=0.0
000310 DO 750 L=1,LMAX
000311 E(L,KDIMP1)=0.
000312 F(L,KDIMP1)=0.
000313 750 CONTINUE
000314 800
000315 F(LMAXP1,KDIMP1)=0.
000316 B(LMAXP1,KDIMP1)=0.
000317 XNE(LMAXP1,KDIMP1)=0.0
000318 XNM(LMAXP1,KDIMP1)=0.0
000319 DO 750 L=1,LMAX
000320 E(L,KDIMP1)=0.
000321 F(L,KDIMP1)=0.
000322 750 CONTINUE
000323 800
000324 F(LMAXP1,KDIMP1)=0.
000325 B(LMAXP1,KDIMP1)=0.
000326 XNE(LMAXP1,KDIMP1)=0.0
000327 XNM(LMAXP1,KDIMP1)=0.0
000328 DO 750 L=1,LMAX
000329 E(L,KDIMP1)=0.
000330 F(L,KDIMP1)=0.
000331 750 CONTINUE
000332 800
000333 F(LMAXP1,KDIMP1)=0.
000334 B(LMAXP1,KDIMP1)=0.
000335 XNE(LMAXP1,KDIMP1)=0.0
000336 XNM(LMAXP1,KDIMP1)=0.0
000337 DO 750 L=1,LMAX
000338 E(L,KDIMP1)=0.
000339 F(L,KDIMP1)=0.
000340 750 CONTINUE
000341 800
000342 F(LMAXP1,KDIMP1)=0.
000343 B(LMAXP1,KDIMP1)=0.
000344 XNE(LMAXP1,KDIMP1)=0.0
000345 XNM(LMAXP1,KDIMP1)=0.0
000346 DO 750 L=1,LMAX
000347 E(L,KDIMP1)=0.
000348 F(L,KDIMP1)=0.
000349 750 CONTINUE
000350 800
000351 F(LMAXP1,KDIMP1)=0.
000352 B(LMAXP1,KDIMP1)=0.
000353 XNE(LMAXP1,KDIMP1)=0.0
000354 XNM(LMAXP1,KDIMP1)=0.0
000355 DO 750 L=1,LMAX
000356 E(L,KDIMP1)=0.
000357 F(L,KDIMP1)=0.
000358 750 CONTINUE
000359 800
000360 F(LMAXP1,KDIMP1)=0.
000361 B(LMAXP1,KDIMP1)=0.
000362 XNE(LMAXP1,KDIMP1)=0.0
000363 XNM(LMAXP1,KDIMP1)=0.0
000364 DO 750 L=1,LMAX
000365 E(L,KDIMP1)=0.
000366 F(L,KDIMP1)=0.
000367 750 CONTINUE
000368 800
000369 F(LMAXP1,KDIMP1)=0.
000370 B(LMAXP1,KDIMP1)=0.
000371 XNE(LMAXP1,KDIMP1)=0.0
000372 XNM(LMAXP1,KDIMP1)=0.0
000373 DO 750 L=1,LMAX
000374 E(L,KDIMP1)=0.
000375 F(L,KDIMP1)=0.
000376 750 CONTINUE
000377 800
000378 F(LMAXP1,KDIMP1)=0.
000379 B(LMAXP1,KDIMP1)=0.
000380 XNE(LMAXP1,KDIMP1)=0.0
000381 XNM(LMAXP1,KDIMP1)=0.0
000382 DO 750 L=1,LMAX
000383 E(L,KDIMP1)=0.
000384 F(L,KDIMP1)=0.
000385 750 CONTINUE
000386 800
000387 F(LMAXP1,KDIMP1)=0.
000388 B(LMAXP1,KDIMP1)=0.
000389 XNE(LMAXP1,KDIMP1)=0.0
000390 XNM(LMAXP1,KDIMP1)=0.0
000391 DO 750 L=1,LMAX
000392 E(L,KDIMP1)=0.
000393 F(L,KDIMP1)=0.
000394 750 CONTINUE
000395 800
000396 F(LMAXP1,KDIMP1)=0.
000397 B(LMAXP1,KDIMP1)=0.
000398 XNE(LMAXP1,KDIMP1)=0.0
000399 XNM(LMAXP1,KDIMP1)=0.0
000400 DO 750 L=1,LMAX
000401 E(L,KDIMP1)=0.
000402 F(L,KDIMP1)=0.
000403 750 CONTINUE
000404 800
000405 F(LMAXP1,KDIMP1)=0.
000406 B(LMAXP1,KDIMP1)=0.
000407 XNE(LMAXP1,KDIMP1)=0.0
000408 XNM(LMAXP1,KDIMP1)=0.0
000409 DO 750 L=1,LMAX
000410 E(L,KDIMP1)=0.
000411 F(L,KDIMP1)=0.
000412 750 CONTINUE
000413 800
000414 F(LMAXP1,KDIMP1)=0.
000415 B(LMAXP1,KDIMP1)=0.
000416 XNE(LMAXP1,KDIMP1)=0.0
000417 XNM(LMAXP1,KDIMP1)=0.0
000418 DO 750 L=1,LMAX
000419 E(L,KDIMP1)=0.
000420 F(L,KDIMP1)=0.
000421 750 CONTINUE
000422 800
000423 F(LMAXP1,KDIMP1)=0.
000424 B(LMAXP1,KDIMP1)=0.
000425 XNE(LMAXP1,KDIMP1)=0.0
000426 XNM(LMAXP1,KDIMP1)=0.0
000427 DO 750 L=1,LMAX
000428 E(L,KDIMP1)=0.
000429 F(L,KDIMP1)=0.
000430 750 CONTINUE
000431 800
000432 F(LMAXP1,KDIMP1)=0.
000433 B(LMAXP1,KDIMP1)=0.
000434 XNE(LMAXP1,KDIMP1)=0.0
000435 XNM(LMAXP1,KDIMP1)=0.0
000436 DO 750 L=1,LMAX
000437 E(L,KDIMP1)=0.
000438 F(L,KDIMP1)=0.
000439 750 CONTINUE
000440 800
000441 F(LMAXP1,KDIMP1)=0.
000442 B(LMAXP1,KDIMP1)=0.
000443 XNE(LMAXP1,KDIMP1)=0.0
000444 XNM(LMAXP1,KDIMP1)=0.0
000445 DO 750 L=1,LMAX
000446 E(L,KDIMP1)=0.
000447 F(L,KDIMP1)=0.
000448 750 CONTINUE
000449 800
000450 F(LMAXP1,KDIMP1)=0.
000451 B(LMAXP1,KDIMP1)=0.
000452 XNE(LMAXP1,KDIMP1)=0.0
000453 XNM(LMAXP1,KDIMP1)=0.0
000454 DO 750 L=1,LMAX
000455 E(L,KDIMP1)=0.
000456 F(L,KDIMP1)=0.
000457 750 CONTINUE
000458 800
000459 F(LMAXP1,KDIMP1)=0.
000460 B(LMAXP1,KDIMP1)=0.
000461 XNE(LMAXP1,KDIMP1)=0.0
000462 XNM(LMAXP1,KDIMP1)=0.0
000463 DO 750 L=1,LMAX
000464 E(L,KDIMP1)=0.
000465 F(L,KDIMP1)=0.
000466 750 CONTINUE
000467 800
000468 F(LMAXP1,KDIMP1)=0.
000469 B(LMAXP1,KDIMP1)=0.
000470 XNE(LMAXP1,KDIMP1)=0.0
000471 XNM(LMAXP1,KDIMP1)=0.0
000472 DO 750 L=1,LMAX
000473 E(L,KDIMP1)=0.
000474 F(L,KDIMP1)=0.
000475 750 CONTINUE
000476 800
000477 F(LMAXP1,KDIMP1)=0.
000478 B(LMAXP1,KDIMP1)=0.
000479 XNE(LMAXP1,KDIMP1)=0.0
000480 XNM(LMAXP1,KDIMP1)=0.0
000481 DO 750 L=1,LMAX
000482 E(L,KDIMP1)=0.
000483 F(L,KDIMP1)=0.
000484 750 CONTINUE
000485 800
000486 F(LMAXP1,KDIMP1)=0.
000487 B(LMAXP1,KDIMP1)=0.
000488 XNE(LMAXP1,KDIMP1)=0.0
000489 XNM(LMAXP1,KDIMP1)=0.0
000490 DO 750 L=1,LMAX
000491 E(L,KDIMP1)=0.
000492 F(L,KDIMP1)=0.
000493 750 CONTINUE
000494 800
000495 F(LMAXP1,KDIMP1)=0.
000496 B(LMAXP1,KDIMP1)=0.
000497 XNE(LMAXP1,KDIMP1)=0.0
000498 XNM(LMAXP1,KDIMP1)=0.0
000499 DO 750 L=1,LMAX
000500 E(L,KDIMP1)=0.
000501 F(L,KDIMP1)=0.
000502 750 CONTINUE
000503 800
000504 F(LMAXP1,KDIMP1)=0.
000505 B(LMAXP1,KDIMP1)=0.
000506 XNE(LMAXP1,KDIMP1)=0.0
000507 XNM(LMAXP1,KDIMP1)=0.0
000508 DO 750 L=1,LMAX
000509 E(L,KDIMP1)=0.
000510 F(L,KDIMP1)=0.
000511 750 CONTINUE
000512 800
000513 F(LMAXP1,KDIMP1)=0.
000514 B(LMAXP1,KDIMP1)=0.
000515 XNE(LMAXP1,KDIMP1)=0.0
000516 XNM(LMAXP1,KDIMP1)=0.0
000517 DO 750 L=1,LMAX
000518 E(L,KDIMP1)=0.
000519 F(L,KDIMP1)=0.
000520 750 CONTINUE
000521 800
000522 F(LMAXP1,KDIMP1)=0.
000523 B(LMAXP1,KDIMP1)=0.
000524 XNE(LMAXP1,KDIMP1)=0.0
000525 XNM(LMAXP1,KDIMP1)=0.0
000526 DO 750 L=1,LMAX
000527 E(L,KDIMP1)=0.
000528 F(L,KDIMP1)=0.
000529 750 CONTINUE
000530 800
000531 F(LMAXP1,KDIMP1)=0.
000532 B(LMAXP1,KDIMP1)=0.
000533 XNE(LMAXP1,KDIMP1)=0.0
000534 XNM(LMAXP1,KDIMP1)=0.0
000535 DO 750 L=1,LMAX
000536 E(L,KDIMP1)=0.
000537 F(L,KDIMP1)=0.
000538 750 CONTINUE
000539 800
000540 F(LMAXP1,KDIMP1)=0.
000541 B(LMAXP1,KDIMP1)=0.
000542 XNE(LMAXP1,KDIMP1)=0.0
000543 XNM(LMAXP1,KDIMP1)=0.0
000544 DO 750 L=1,LMAX
000545 E(L,KDIMP1)=0.
000546 F(L,KDIMP1)=0.
000547 750 CONTINUE
000548 800
000549 F(LMAXP1,KDIMP1)=0.
000550 B(LMAXP1,KDIMP1)=0.
000551 XNE(LMAXP1,KDIMP1)=0.0
000552 XNM(LMAXP1,KDIMP1)=0.0
000553 DO 750 L=1,LMAX
000554 E(L,KDIMP1)=0.
000555 F(L,KDIMP1)=0.
000556 750 CONTINUE
000557 800
000558 F(LMAXP1,KDIMP1)=0.
000559 B(LMAXP1,KDIMP1)=0.
000560 XNE(LMAXP1,KDIMP1)=0.0
000561 XNM(LMAXP1,KDIMP1)=0.0
000562 DO 750 L=1,LMAX
000563 E(L,KDIMP1)=0.
000564 F(L,KDIMP1)=0.
000565 750 CONTINUE
000566 800
000567 F(LMAXP1,KDIMP1)=0.
000568 B(LMAXP1,KDIMP1)=0.
000569 XNE(LMAXP1,KDIMP1)=0.0
000570 XNM(LMAXP1,KDIMP1)=0.0
000571 DO 750 L=1,LMAX
000572 E(L,KDIMP1)=0.
000573 F(L,KDIMP1)=0.
000574 750 CONTINUE
000575 800
000576 F(LMAXP1,KDIMP1)=0.
000577 B(LMAXP1,KDIMP1)=0.
000578 XNE(LMAXP1,KDIMP1)=0.0
000579 XNM(LMAXP1,KDIMP1)=0.0
000580 DO 750 L=1,LMAX
000581 E(L,KDIMP1)=0.
000582 F(L,KDIMP1)=0.
000583 750 CONTINUE
000584 800
000585 F(LMAXP1,KDIMP1)=0.
000586 B(LMAXP1,KDIMP1)=0.
000587 XNE(LMAXP1,KDIMP1)=0.0
000588 XNM(LMAXP1,KDIMP1)=0.0
000589 DO 750 L=1,LMAX
000590 E(L,KDIMP1)=0.
000591 F(L,KDIMP1)=0.
000592 750 CONTINUE
000593 800
000594 F(LMAXP1,KDIMP1)=0.
000595 B(LMAXP1,KDIMP1)=0.
000596 XNE(LMAXP1,KDIMP1)=0.0
000597 XNM(LMAXP1,KDIMP1)=0.0
000598 DO 750 L=1,LMAX
000599 E(L,KDIMP1)=0.
000600 F(L,KDIMP1)=0.
000601 750 CONTINUE
000602 800
000603 F(LMAXP1,KDIMP1)=0.
000604 B(LMAXP1,KDIMP1)=0.
000605 XNE(LMAXP1,KDIMP1)=0.0
000606 XNM(LMAXP1,KDIMP1)=0.0
000607 DO 750 L=1,LMAX
000608 E(L,KDIMP1)=0.
000609 F(L,KDIMP1)=0.
000610 750 CONTINUE
000611 800
000612 F(LMAXP1,KDIMP1)=0.
000613 B(LMAXP1,KDIMP1)=0.
000614 XNE(LMAXP1,KDIMP1)=0.0
000615 XNM(LMAXP1,KDIMP1)=0.0
000616 DO 750 L=1,LMAX
000617 E(L,KDIMP1)=0.
000618 F(L,KDIMP1)=0.
000619 750 CONTINUE
000620 800
000621 F(LMAXP1,KDIMP1)=0.
000622 B(LMAXP1,KDIMP1)=0.
000623 XNE(LMAXP1,KDIMP1)=0.0
000624 XNM(LMAXP1,KDIMP1)=0.0
000625 DO 750 L=1,LMAX
000626 E(L,KDIMP1)=0.
000627 F(L,KDIMP1)=0.
000628 750 CONTINUE
000629 800
000630 F(LMAXP1,KDIMP1)=0.
000631 B(LMAXP1,KDIMP1)=0.
000632 XNE(LMAXP1,KDIMP1)=0.0
000633 XNM(LMAXP1,KDIMP1)=0.0
000634 DO 750 L=1,LMAX
000635 E(L,KDIMP1)=0.
000636 F(L,KDIMP1)=0.
000637 750 CONTINUE
000638 800
000639 F(LMAXP1,KDIMP1)=0.
000640 B(LMAXP1,KDIMP1)=0.
000641 XNE(LMAXP1,KDIMP1)=0.0
000642 XNM(LMAXP1,KDIMP1)=0.0
000643 DO 750 L=1,LMAX
000644 E(L,KDIMP1)=0.
000645 F(L,KDIMP1)=0.
000646 750 CONTINUE
000647 800
000648 F(LMAXP1,KDIMP1)=0.
000649 B(LMAXP1,KDIMP1)=0.
000650 XNE(LMAXP1,KDIMP1)=0.0
000651 XNM(LMAXP1,KDIMP1)=0.0
000652 DO 750 L=1,LMAX
000653 E(L,KDIMP1)=0.
000654 F(L,KDIMP1)=0.
000655 750 CONTINUE
000656 800
000657 F(LMAXP1,KDIMP1)=0.
000658 B(LMAXP1,KDIMP1)=0.
000659 XNE(LMAXP1,KDIMP1)=0.0
000660 XNM(LMAXP1,KDIMP1)=0.0
000661 DO 750 L=1,LMAX
000662 E(L,KDIMP1)=0.
000663 F(L,KDIMP1)=0.
000664 750 CONTINUE
000665 800
000666 F(LMAXP1,KDIMP1)=0.
000667 B(LMAXP1,KDIMP1)=0.
000668 XNE(LMAXP1,KDIMP1)=0.0
000669 XNM(LMAXP1,KDIMP1)=0.0
000670 DO 750 L=1,LMAX
000671 E(L,KDIMP1)=0.
000672 F(L,KDIMP1)=0.
000673 750 CONTINUE
000674 800
000675 F(LMAXP1,KDIMP1)=0.
000676 B(LMAXP1,KDIMP1)=0.
000677 XNE(LMAXP1,KDIMP1)=0.0
000678 XNM(LMAXP1,KDIMP1)=0.0
000679 DO 750 L=1,LMAX
000680 E(L,KDIMP1)=0.
000681 F(L,KDIMP1)=0.
000682 750 CONTINUE
000683 800
000684 F(LMAXP1,KDIMP1)=0.
000685 B(LMAXP1,KDIMP1)=0.
000686 XNE(LMAXP1,KDIMP1)=0.0
000687 XNM(LMAXP1,KDIMP1)=0.0
000688 DO 750 L=1,LMAX
000689 E(L,KDIMP1)=0.
000690 F(L,KDIMP1)=0.
000691 750 CONTINUE
000692 800
000693 F(LMAXP1,KDIMP1)=0.
000694 B(LMAXP1,KDIMP1)=0.
000695 XNE(LMAXP1,KDIMP1)=0.0
000696 XNM(LMAXP1,KDIMP1)=0.0
000697 DO 750 L=1,LMAX
000698 E(L,KDIMP1)=0.
000699 F(L,KDIMP1)=0.
000700 750 CONTINUE
000701 800
000702 F(LMAXP1,KDIMP1)=0.
000703 B(LMAXP1,KDIMP1)=0.
000704 XNE(LMAXP1,KDIMP1)=0.0
000705 XNM(LMAXP1,KDIMP1)=0.0
000706 DO 750 L=1,LMAX
000707 E(L,KDIMP1)=0.
000708 F(L,KDIMP1)=0.
000709 750 CONTINUE
000710 800
000711 F(LMAXP1,KDIMP1)=0.
000712 B(LMAXP1,KDIMP1)=0.
000713 XNE(LMAXP1,KDIMP1)=0.0
000714 XNM(LMAXP1,KDIMP1)=0.0
000715 DO 750 L=1,LMAX
000716 E(L,KDIMP1)=0.
000717 F(L,KDIMP1)=0.
000718 750 CONTINUE
000719 800
000720 F(LMAXP1,KDIMP1)=0.
000721 B(LMAXP1,KDIMP1)=0.
000722 XNE(LMAXP1,KDIMP1)=0.0
000723 XNM(LMAXP1,KDIMP1)=0.0
000724 DO 750 L=1,LMAX
000725 E(L,KDIMP1)=0.
000726 F(L,KDIMP1)=0.
000727 750 CONTINUE
000728 800
000729 F(LMAXP1,KDIMP1)=0.
000730 B(LMAXP1,KDIMP1)=0.
000731 XNE(LMAXP1,KDIMP1)=0.0
000732 XNM(LMAXP1,KDIMP1)=0.0
000733 DO 750 L=1,LMAX
000734 E(L,KDIMP1)=0.
000735 F(L,KDIMP1)=0.
000736 750 CONTINUE
000737 800
000738 F(LMAXP1,KDIMP1)=0.
000739 B(LMAXP1,KDIMP1)=0.
000740 XNE(LMAXP1,KDIMP1)=0.0
000741 XNM(LMAXP1,KDIMP1)=0.0
000742 DO 750 L=1,LMAX
000743 E(L,KDIMP1)=0.
000744 F(L,KDIMP1)=0.
000745 750 CONTINUE
000746 800
000747 F(LMAXP1,KDIMP1)=0.
000748 B(LMAXP1,KDIMP1)=0.
000749 XNE(LMAXP1,KDIMP1)=0.0
000750 XNM(LMAXP1,KDIMP1)=0.0
000751 DO 750 L=1,LMAX
000752 E(L,KDIMP1)=0.
000753 F(L,KDIMP1)=0.
000754 750 CONTINUE
000755 800
000756 F(LMAXP1,KDIMP1)=0.
000757 B(LMAXP1,KDIMP1)=0.
000758 XNE(LMAXP1,KDIMP1)=0.0
000759 XNM(LMAXP1,KDIMP1)=0.0
000760 DO 750 L=1,LMAX
000761 E(L,KDIMP1)=0.
000762 F(L,KDIMP1)=0.
000763 750 CONTINUE
000764 800
000765 F(LMAXP1,KDIMP1)=0.
000766 B(LMAXP1,KDIMP1)=0.
000767 XNE(LMAXP1,KDIMP1)=0.0
000768 XNM(LMAXP1,KDIMP1)=0.0
000769 DO 750 L=1,LMAX
000770 E(L,KDIMP1)=0.
000771 F(L,KDIMP1)=0.
000772 750 CONTINUE
000773 800
000774 F(LMAXP1,KDIMP1)=0.
000775 B(LMAXP1,KDIMP1)=0.
000776 XNE(LMAXP1,KDIMP1)=0.0
000777 XNM(LMAXP1,KDIMP1)=0.0
000778 DO 750 L=1,LMAX
000779 E(L,KDIMP1)=0.
000780 F(L,KDIMP1)=0.
000781 750 CONTINUE
000782 800
000783 F(LMAXP1,KDIMP1)=0.
000784 B(LMAXP1,KDIMP1)=0.
000785 XNE(LMAXP1,KDIMP1)=0.0
000786 XNM(LMAXP1,KDIMP1)=0.0
000787 DO 750 L=1,LMAX
000788 E(L,KDIMP1)=0.
000789 F(L,KDIMP1)=0.
000790 750 CONTINUE
000791 800
000792 F(LMAXP1,KDIMP1)=0.
000793 B(LMAXP1,KDIMP1)=0.
000794 XNE(LMAXP1,KDIMP1)=0.0
000795 XNM(LMAXP1,KDIMP1)=0.0
000796 DO 750 L=1,LMAX
000797 E(L,KDIMP1)=0.
000798 F(L,KDIMP1)=0.
000799 750 CONTINUE
000800 800
000801 F(LMAXP1,KDIMP1)=0.
000802 B(LMAXP1,KDIMP1)=0.
000803 XNE(LMAXP1,KDIMP1)=0.0
000804 XNM(LMAXP1,KDIMP1)=0.0
000805 DO 750 L=1,LMAX
000806 E(L,KDIMP1)=0.
000807 F(L,KDIMP1)=0.
000808 750 CONTINUE
000809 800
000810 F(LMAXP1,KDIMP1)=0.
000811 B(LMAXP1,KDIMP1)=0.
000812 XNE(LMAXP1,KDIMP1)=0.0
000813 XNM(LMAXP1,KDIMP1)=0.0
000814 DO 750 L=1,LMAX
000815 E(L,KDIMP1)=0.
000816 F(L,KDIMP1)=0.
000817 750 CONTINUE
000818 800
000819 F(LMAXP1,KDIMP1)=0.
000820 B(LMAXP1,KDIMP1)=0.
000821 XNE(LMAXP1,KDIMP1)=0.0
000822 XNM(LMAXP1,KDIMP1)=0.0
000823 DO 750 L=1,LMAX
000824 E(L,KDIMP1)=0.
000825 F(L,KDIMP1)=0.
000826 750 CONTINUE
000827 800
000828 F(LMAXP1,KDIMP1
```

```

000402      B(L,KUIMP1)=0.
000405      XN(L,KUIMP1)=0.0
000407      XN(L,KUIMP1)=0.0
000412      750 CONTINUE
000414      RETURN
000414      800 XN(LMAXP1,KUIMP1)=XN(LMAXP1,LOAU)
000422      XN(LMAXP1,KUIMP1)=XN(LMAXP1,LOAU)
000430      F(LMAXP1,KUIMP1)=FS(LMAXP1,LOAU)
000435      B(LMAXP1,KUIMP1)=BS(LMAXP1,LOAU)
000443      DO 850 L=LMAX
000444      XN(L,KUIMP1)=XN(L,LOAU)
000452      XN(L,KUIMP1)=XN(L,LOAU)
000457      E(L,KUIMP1)=ES(L,LOAU)
000466      F(L,KUIMP1)=FS(L,LOAU)
000473      B(L,KUIMP1)=BS(L,LOAU)
000501      850 CONTINUE
000503      RETURN
000503      END

```

```

000021      SUBROUTINE IOSAVE(ANES,ANMS,ES,FS,BS,E,F,R,XNE,XNM,JSET,NMIN,KSET,
1 IUSE)
2 COMMON      EE(23,251), FF(24,251), B3(24,251), XNE(24,251),
3              CAZXPX(24), CXXXPX(24),
4              KPRINT(16), ICOUNT(16,10),
5              EFACTA(23), EFACTB(23),
6              EK(23), SIG(24),
7              TH(24), STM(24),
8              HEAD(8), OUTAHK(25,9),
9              FATHAP(24), THETA(24),
10             HTHETA(23)
11
12 C
13 COMMON /INTEGER/ KUATA, ISTORE, NTIME, LOAD,
14             NLOAD, LMAX, LMAXP1, KDIR, KUIHPI,
15             IRENUM, NKOUT, LUATA, NBLOCK,
16             JSTORE, NSHIFT, KUIMH1, LMAXP2,
17             IREM, IDATA, NPACK,
18
19 C
20 COMMON /HAND/ R, HUC, RDDR, MPHDR,
21             HPHDR, MSTRT, DR, MDR, LGM1,
22             LGND, LGNUP1, K, KPI, KNUH,
23             NFILES, NSHFT, XTHAP, EMOCN,
24             CMUDR, UTSPS, T, TPUT,
25             IPHOT, IMPUT, AGND, XGND,
26             ORNG(15), ANH(24), XJT(24), Q(24)
27
28 C
29 DIMENSION ANES(24,JSET),ANMS(24,JSET),ES(24,JSET),FS(24,JSET),BS
30             E(23,KSET),F(24,KSET),B(24,KSET),XNE(24,KSET),XNM(24,
31             KSET),DUM1(23),DUM2(24)
32 GO TO (100,200,300,400,500,600,700,800,900), IUSE
33
34 100 READ(KUATA) ICOUNT,PRINT,MINI,LOAD,NLOAD,NBLOCK,NSHFT,IRSTART,
35     1 FTEMP,FATHAP,XNES,AN,ES,FS,BS
36 200 READ(KUATA) E,F,B,ANE,XNM
37     WRITE(ISTORE) E,F,B,ANE,XNM
38     RETURN
39
40 300 WRITE(LDATA) ICOUNT,PRINT,MINI,LOAD,NLOAD,NBLOCK,NSHFT,IRSTART,
41     1 FTEMP,FATHAP,XNES,AN,ES,FS,BS
42 400 READ(JSTORE) E,F,B,ANE,XNM
43     WRITE(LDATA) E,F,B,ANE,XNM
44     RETURN
45
46 500 WRITE(LDATA) ICOUNT,PRINT,MINI,LOAD,NLOAD,NBLOCK,NSHFT,IRSTART,
47     1 FTEMP,FATHAP,XNES,AN,ES,FS,BS
48 600 IF(JSET.EQ.1) GO TO 950
49     READ(ISTORE) E,F,B,ANE,XNM
50     WRITE(LDATA) E,F,B,ANE,XNM
51     RETURN
52
53 700 READ(ISTORE) E,F,B,ANE,XNM
54     RETURN
55
56 800 WRITE(JSTORE) E,F,B,ANE,XNM
57     RETURN
58
59 900 READ(ISTORE) DUM1,E,DUM2,F,DUM3,R,DUM3,XNE,DUM3,XNM
60     RETURN
61     END

```

```

000006      SUBROUTINE DATAP(E(IOUTARR,NPACKS,IOPT))
      COMMON /INTEGEN/ KDATA, ISTORE,
1      NLOAD, LDATA, LAAAPL,
2      IHEXUM, NOUT, LDATA,
3      JSTORE, NSRIFT, KULHML,
      ISEM, IINTA
      DIMENSION OUTARR(NPACK)
      C
      GO TO (100,200), IOPT
      100 READ(10DATA) OUTARR
      IF(EOF(10DATA)) 250,200
      200 WRITE(10DATA) OUTARR
      RETURN
      250 IOPT=0
      RETURN
      END
000042
000006      LOAD,
      KULHML,
      JDATA,
      NPACK,
      NTIME,
      KDIM,
      HBLOCK,
      LMAXP2,

```

```

BLOCK DATA
C *** TEST PROBLEM ***
C
000002 COMMON /PARAMS/
1 EPSGND, CMU, EPS,
2 X10M0B, ATTACH, XELN,
3 ITHETA, SCALE, QC0Y,
C
000002 DATA
1 EPSGND, CMU, EPS,
2 X10M0B, ATTACH, XELN,
3 1.0E-19, 12.566E-7, 88.16E-11,
4 1.4650E-11, 1.0E-3, 1.0E-3,
5 2.7E-4, 4.5E-13,
END
000002

```


C. REDACT

1. Description of Modifications

Program REDACT has been modified and is now a subroutine with a driver program DATAPK. This modification was made in order to process data tapes from the modified ELECTRA which have a packed OUTARR (see II. A.1.).

DATAPK opens and closes the plot files and calculates the number of words that are packed in OUTARR. In subroutine REDACT, the column and row indicators for data editing have been modified to correspond to the modifications made in ELECTRA. In order to process the data tape, the column and row indicators for the particular variable to be plotted are used to determine the word to be unpacked. The data array is unpacked by an assembly language routine (UNPACK).

2. Program Listing

```

000003      C      COMMON ORU(1003),TMI(1003),IBUF(1024),HEADEN(4),THETA(20),
000003      C      1 THETA(20),ORU(1003),ANGLE(15),HANGLE(17),LMAXP1,LMAXP4
000003      C      DIMENSION DATA(84)
000003      C      1 FORMAT(1H1//31X,7A10,40//)
000003      C      PE=1.0 1H
000003      C      RE=1.0 1H
000003      C      READ(11) NFILES,NQNPES,LMAX
000003      C      READ(11) MEAVER
000003      C      WRITE (6,1) MEAVER
000003      C      READ (11) ANGLE,HANGLE
000003      C      INITIALIZE PLOT ROUTINE
000042      C      CALL PLOTS(1BUF,1024,10)
000045      C      CALL PLOT (0,-11,-3)
000050      C      CALL PLOT (0,1.26,-3)
000053      C      COMPUTE POINTERS FOR THETA
000053      C      LMAXP1=LMAX+1
000053      C      LMAXP4=LMAX+4
000056      C      NPA=CA=LMAXP4+2+1
000060      C      CALL REFACT (DATA,NPA,ACK)
000062      C      CLOSE PLOT FILE
000065      C      CALL PLOT (0,0.0,0.0)
000067      C      STOP
000067      C      END

```



```

000016      WRT= (6,22) ICUTS,UMAX*,MPACK,IREM
000032      READ 1,*,FOR
000040      READ(5,12) JTYPE
000046      IF(JTYPE.EQ.JTYPE) GO TO 100
000052      P=0(5,1) ITIME, (IMEFAL), I=1, ITIME(4)
000057      P=0(5,1) ITIME, (IMEFAL), I=1, ITIME(1)
000067      IIME
000077      IIME
000087      IIME
000097      IIME
000107      IIME
000117      IIME

      READ ALL DATA FROM TAPE AND WRITE PLOT DATA ON DISC OR DRUM
C
C
C
000123      101 RTIME=
000124      NPIS=
000125      IF(ISTAT.EQ.1) BACKSPACE 11
000131      TSRT=
000132      P=0(11) DATA
000140      102 IF(EQ.11) 115,1,3
000144      103 P=DATA(2)
000146      IF(R*EQ*TEMP) GO TO 112
000150      IF(ISTAT.EQ.1) GO TO 115
000152      IF(R*EQ*GT*PCARD) GO TO 115
000156      IF(R*EQ*GE*PCARD) GO TO 110
000160      GO TO 102
000161      105 WRITE(1,10) PCARD
000167      GO TO 100
000171      110 WRITE(10) DATA
000177      NPIS=PTS+1
000201      RTIME=
000202      TSRT=1
000203      GO TO 102
000205      112 WRITE(10) DATA
000213      NPIS=PTS+1
000215      GO TO 102
C
C
C
000217      115 READ 10
000221      P=0(5,2) SYM,PCARD,IM,ILUGO,DEC,IS,ILUGT,DECT,ILINE,SAB,SOR,IPR
000255      IF(SYM.EQ.ANEAR) GO TO 101
C
C
C
      CHECK INPUT, IF V... SET STANDARD VALUES
C
C
C
000261      IF(DEC.LE.0) DEC=4
000263      IF(SAB.LE.0) SAB=0
000266      IF(SOR.LE.0) SOR=0
000271      IF(ILINE.LE.0) ILINE=0
000274      IF(DECT.LE.0) DECT=2
C
C
C
      INITIALIZE PLOT COORDINATES
C
C
C
000277      SPAC=SA*2+5
000301      XANG=SAB+7
000303      IF(R*EQ*GE*1000) XANG=XANG+0.1
000307      XTIME=28
000311      XMEAS=SAB-3+25
000313      YMEAS=SOR+17
000315      ZTIME=IM

```

```

000317      IPC=KCARU
C          INITIALIZE INDICATORS AND COUNTERS
C
000321      JJ=0
C
C          IF SYM = VAY(8) , JOB IS FINISHED
C
000322      IF (SYM.EQ.VAY(8)) GO TO 110
000324      WRITE(6,6)
000327      RETURN
000330      116 WRITE(6,7) SYM,KCARU,110,1050,DEC,15,1LOG1,DFACT,ILINE,SAB,S02
C
C          DIAGNOSTICS FOR CARD INPUT
C
000362      IF (L.C.GT.0) GO TO 117
000366      WRITE(6,8)
000371      GO TO 115
000373      117 IF (SYM.EQ.VAY(1)) GO TO 118
000406      IF (SYM.EQ.VAY(2)) GO TO 119
000411      WRITE(6,9)
000414      GO TO 115
C
C          DETERMINE ROW AND COLUMN INDICATORS FOR DATA EDITING
C
000416      119 IC=110/2+3
000421      IR=110/2+4
000423      IRI=1
000424      DO 13 I=1,7
000426      IC=I
000427      YL=8-IAB(IC)
000431      IF (SYM.EQ.VAY(1)) GO TO 132
000433      130 CONTINUE
000435      WRITE(6,10)
000440      GO TO 115
000442      132 GO TO (140,170,160,145,165,150,155),IC
000455      140 IC=2
000456      IR=3
000457      IIR=3
000460      GO TO 175
000461      145 IR=IR0
000463      IT=4
000464      IF (110.EQ.LMAXP1) IR=2
000467      GO TO 175
000470      150 IC=4
000471      IR=3
000472      IT=4
000473      GO TO 175
000474      155 IR=IR4E
000476      IC=2
000477      IT=2
000500      GO TO 175
000502      160 IR=IR0
000502      IT=3
000503      GO TO 175
000503      165 IR=IR0
000505      IC=1
000506      IT=1

```



```

000735      400 CALL WRTPLT(IPLT,IMT,OUT,ALAB,YLAB,HEADR,SAB,SOR,JECT,DEC,ILINF,IPR
000754      500 IF (JTYPE.EQ.JTIME) GO TO 115
C
C      SECTION FOR GENERATING MAX VALUE VS. THETA PLOTS
C
000760      550 IF (SYM.NE.VAY(4).AND.SYM.NE.VAY(5).AND.SYM.NE.VAY(7)) GO TO 115
000773      IPR=IPR+1
000775      YMAX=ABS(UMD(1))
000777      DO 605 I=2,JJ
001000      IF (ABS(UMD(I)).GT.YMAX) YMAX=ABS(UMD(I))
001006      605 CONTINUE
001011      OUT(I,IP)=YMAX
001013      IF (IC.EQ.7) GO TO 600
001015      IF (IP.NE.ITHEI) GO TO 115
001017      CALL TPEPLT(OPUTH,ITHEI,SCM,SAB,ITP,YLAB,RCARD,HEADR)
001026      GO TO 600
001030      660 IF (IP.NE.ITHEI) GO TO 115
001032      CALL TPEPLT(OPUTH,ITHEI,SCM,SAB,ITP,YLAB,RCARD,HEADR)
001042      665 CALL PLOT(SPACE(8),-J)
001045      IPR=
001046      WRITE(4,7) SYM
C
C      RETURN FOR MORE DATA
C
001054      GO TO 115
001056      END

```

```

000013 SUBROUTINE THEPLT(ORU,ABSC,SOR,SAB,ITP,TLAB,ZRT,HEADX)
000013 DIMENSION ORD(1),ABSC(1),TLAB(1),HEADR(1)
000013 XRNQ=SZB-6
000014 XHEAD=SAB-3.25
000016 YHEAD=SOR-17
000021 CALL SCALE(ORU,SOR,ITP,1)
000023 CALL SCALE(ABSC,SAB,ITP,1)
000031 CALL NHEAD(XRNQ,0.0,1,ZRT,0.0,-1)
000036 CALL SYMHOL(HEAD,YHEAD,1,HEADR,0.0,36)
000043 CALL LINE(ABSC,ORD,ITP,4,1,1)
000053 CALL AXIS(0.0,TLAB,10,SOR,YU,ORU(1,IP*1),ORD(1,IP*2))
000072 CALL AXIS(0.0,5*HEAD,5,SAB,0,ABSC(1,IP*1),ABSC(1,IP*2))
000111 CALL PLOT(0.0,SOR,3)
000117 ISAB=SAB
000124 ZSAB=ISAB
000126 IF(ZSAB.EQ.SAB) ISAB=SAB-1.
000133 DO 200 L=1,ISAB
000135 XL=L
000136 CALL SYMHOL(XL,SOR,1,1,0.0,-2)
000142 200 CONTINUE
000150 CALL PLOT(SAB,SOR,2)
000152 ISUR=SUR
000157 ZSUR=ISUR
000161 IF(ZSUR.EQ.SOR) ISUR=SUR-1.
000166 ZSUR=ISUR
000167 CALL SYMHOL(SAB,ZSUR,1,1,0.0,-2)
000173 IS=ISUR-1
000175 DO 300 K=1,IS
000202 YK=ZSUR-K
000211 CALL SYMHOL(SAB,YK,1,1,0.0,-2)
000217 300 CONTINUE
000217 CALL PLOT(SAB,0.0,2)
000221 X91=(YK-ABSC(1,IP*1))/ABSC(1,IP*2)
000231 CALL PLOT(X91,0.0,3)
000233 CALL PLOT(X91,SUR,2)
000241 RETURN
000242 END

```

COMPASS - VER 2. 06/28/71 11.31.00.

2

			IDENT PACK	00000000
			EMPTY PACK	00000010
			EMPTY UNPACK	00000020
			WFO 42/9LPAK-18/2	00000030
			DATA 0	00000040
			SAT B1	00000050
			S44 ZLOHF	00000060
			YX1 X1*XX	00000070
			S4C B2	00000080
			LXC 3B6	00000090
			S4J ZUPHF	00000100
			YAZ A2*XJ	00000110
			BAB X.*XC	00000120
			E* PACK	00000130
			DATA 0	00000140
			SAT B1	00000150
			BAB X1	00000160
			YAY X1	00000170
			AAS 3B6	00000180
			LAB 3B6	00000190
			S4B B2	00000200
			LAT 3B6	00000210
			AAT 3B6	00000220
			LAT 3B6	00000230
			SAT B3	00000240
			UNPACK	00000250
			E*	00000260
			L11 ZUPHF	00000270
			L11 I1 +	00000280
			I1C +	00000290
			ZLOHF	00000300
			END	00000310

D. REDACTO

1. Description of Modifications

Program REDACTO has been modified and is now a subroutine with a driver program DATAPKO. This modification was made in order to process data tapes from the modified ORESTES which have a packed OUTARR (see 11.B.1).

DATAPKO opens and closes the plot files and calculates the number of words that are packed in OUTARR. In subroutine REDACTO, the column and row indicators for data editing have been modified to correspond to the modifications made in ORESTES. In order to process the data tape, the column and row indicators for the particular variable to be plotted are used to determine the word to be unpacked. The data array is unpacked by an assembly language routine (UNPACK).

2. Program Listing


```

000006      ICN5=LMAXP2/2-1
000010      IRE=MOD(LMAXP2/2,1)
000014      IF (IRFM.EQ.0) ICN5=LMAXP2/2-2
000016      WRITE (6,22) ICN5,LMAXP2,IRFM,IREM
000032      READ 1,MDR ,LGND
000036      READ (5,12) JTYPE
000040      IF (JTYPE.EQ.1) IRE=0 TO 100
000050      READ (5,11) ITHETA, (IHTA(1),I=1,ITHETA)
000054      READ (5,11) ITHETA, (IHTA(1),I=1,ITHETA)
000071      ITHETA=0
000110      IF (SYN.NE.ANEW) GO TO 100
000121      IF (SYN.NE.ANEW) GO TO 100

C      READ ALL DATA FROM TAPE AND WRITE PLOT DATA ON DISC OR DRUM
C
000125      101 RTMP=0.
000126      NPIS=0
000127      IF (ISORT.EQ.1) BACKSPACE 11
000133      ISORT=0
000134      102 REWIND 11 DATA
000142      IF (EOF(11)) 115,103
000146      103 R=DATA(2)
000150      IF (R.EQ.1) GO TO 112
000152      IF (ISORT.EQ.1) GO TO 115
000154      IF (R-MR.GT.PCARD) GO TO 115
000160      IF (R-MR.GE.PCARD) GO TO 110
000162      GO TO 102
000163      105 WRITE (6,18) RCARD
000171      GO TO 100
000173      110 NPIS=NPIS+1
000201      RTMP=0
000205      ISORT=1
000207      GO TO 102
000215      112 WRITE (18) DATA
000217      NPIS=NPIS+1
000218      GO TO 102

C      READ PLOT PARAMETER CARD
C
000221      115 REWIND 18
000223      READ (5,2) SYN,RCARD,ITM,ILOGO,DEC,IS,ILOGT,DECT,ILINE,SAB,SOR,IPR
000257      IF (SYN.EQ.ANEW) GO TO 101

C      CHECK INPUT, IF 0, SET STANDARD VALUES
C
000263      IF (DEC.LE.0) DEC=4
000265      IF (SAB.LE.0) SAB=0
000270      IF (SOR.LE.0) SOR=0.5
000273      IF (ILINE.LE.0) ILINE=1
000276      IF (DECT.LE.0) DECT=2

C      INITIALIZE PLOT COORDINATES
C
000301      SPACL=SAB*2.5
000303      XRG=SAB*7
000305      IF (RLAPU.GE.1000) ANGLE=XRG*0.1
000311      XTH=SAB*28

```

```

000313 ZHEAD=5AB-3*25
000315 ZHEAD=5AB-3*25
000317 ZHEAD=5AB-3*25
000321 ZHEAD=5AB-3*25

C INITIALIZE INDICATORS AND COUNTERS
C
C JJ=0
C
C IF SYM = VAY(9) * JUB IS FINISHED
C
000324 IF (SYM.EV.VAY(9)) GO TO 116
000326 WRITE(6,9)
000331 RETURN
C
C 116 WRITE(6,9) SYM,RCARD,IT,ILOG,DEC,IS,LOG,DEC,ILINE,SAB,SOR
C
C DIAGNOSTICS FOR CARD INPUT
C
C IF (IC.GT.0) GO TO 117
C
000364 WRITE(6,3)
000370 GO TO 115
000373
000375 117 IF (IT.GT.0) GO TO 119
000400 WRITE(6,4)
000403 GO TO 115
C
C DETERMINE ROW AND COLUMN INDICATORS FOR DATA EDITING
C
C 119 IR=1TH/2+1
C
000410 IR=1TH/2+2
000412 IR=1
000413 DO 13 1=1,8
000415 IC=1
000416 YL=8-TAB(IC)
000420 IF (SYM.EV.VAY(1)) GO TO 132
C
000422 CONTINUE
000424 130 WRITE(6,5)
000427 GO TO 115
000431 132 GO TO (14,175,163,145,165,150,155,150),IC
000445 140 IC=8
000446 ITIME=2
000447 IP=10
000451 GO TO 170
000451 145 IR=IRG
000453 ITIME=IC
000454 IF (LONG.L.LMAX) GO TO 170
000460 IF (ITHE.L.LMAXP1) GO TO 170
000462 IR=1
000463 IC=8
000464 GO TO 170
000464 150 IC=1
000465 ITIME=7
000466 IR=IRG+1
000470 GO TO 170
000471 155 IR=IRG
000473 ITIME=IC
000474 GO TO 170
000475 160 IR=IRG

```

```

000477      IC=9
000500      ITIME=5
000501      GO TO 170
000501      163 IR=IR0
000503      ITIME=3
000504      GO TO 170
000504      165 IR=IR0
000506      ITIME=IC
000507      IF (LGRO.GT.LMAX) GO TO 170
000513      IF (ITIME.LMAXP1) GO TO 170
000515      IR=1
000516      IC=9
000517      GO TO 170
000517      175 IR=IR0
000521      ITIME=2
C
C      EXIT THE DATA
C
000522      170 ISET1=1
000523      ISET2=1
000524      WRITE (6,22) ICONS,LMAXP2,IPACK,IREM
000540      IF (IREM.NE.0) GO TO 180
000543      LOGV=LOG(IR+2)
000547      IF (LOGV.EQ.0) ISET2=2
000551      IR=IR-1
000553      IR=IR+1
000554      GO TO 185
000554      180 LOGT=LOG(IRT+ITIME+2)
000554      LOGV=LOG(IR+IC+2)
000567      IF (LOGT.NE.0) ISET1=2
000571      IF (LOGV.NE.0) ISET2=2
000573      185 IUPACT=(IRT+ITIME+LMAXP2)/2-ICONS
000600      IUPALV=(IR+IC+LMAXP2)/2-ICONS
000604      DO 200 L=1,NPTS
000606      READ(10) DATA
000614      T=WORD(ISET1)
000622      IF (T.LT.15) T=15
000626      IF (JJ.EQ.1000) GO TO 250
000631      JJ=JJ+1
000633      CALL UNPACK(DATA(IUPACT),WORD(1),WORD(2))
000634      OPV(JJ)=WORD(ISET2)
000641      TM(JJ)=T
000645      IF (JJ.LE.10) PRINT *,DATA(2),OPV(JJ),TM(JJ)
000647      IF (IPACK.NE.12800) GO TO 400
000672      WRITE(6,15) JJ
000674      WRITE(6,9) OPV(JJ),TM(JJ)
000702      200 CONTINUE
000717
C
C      IF JTYPE = JTHETA, ONLY MAX VALUE PLOTS ARE MADE
C
000723      250 IF (JTYPE.EQ.JTHETA) GO TO 650
C
000725      IPL01=1
000726      IF (LOGT.NE.LOG1) GO TO 300
000730      IF (LOGV.NE.LOG0) IPL01=3
000733      GO TO 400
000734      300 IPL01=2

```

```

000735      IF (LOGU.NE.0.00) I=I+1
000740      CALL OPERATE(I,LOT,THURD,XLAB,YLAB,HEADR,SAB,SOR,DECT,DEC,LINE,IPR
000757      1,IT,MC,JJ)
          500 IF (JTYPE.EQ.JTIME) GO TO 115
          C      SECTION FOR GENERATING MAX VALUE VS. THETA PLOTS
          C
000763      650 IF (SYM.NE.VAY(4).AND.SYM.NE.VAY(5).AND.SYM.NE.VAY(7)) GO TO 115
000776      IPR=IPR+1
001000      YMAX=ABS(ORD(I))
001002      GO 625 IF Z.JJ
001003      IF (ABS(ORD(I)).GT.YMAX) YMAX=ABS(ORD(I))
001011      655 CONTINUE
001014      ORD(I)=YMAX
001016      IF (I.EQ.7) GO TO 600
001020      IF (IPR.NE.1) THEN
001022      CALL THEPLT(ORD(I),TCTA,SUR,SAB,IT,XLAB,RCARD,HEADR)
001031      GO TO 605
001033      660 IF (IPR.NE.1) THEN
001035      CALL THEPLT(ORD(I),TCTA,SUR,SAB,IT,XLAB,MCARD,HEADR)
001045      665 CALL PLOT(SPACE(0.1))
001050      IPR=J
001051      WRITE(6,7) SYM
          C      RETURN FOR MORE DATA
          C
001057      GO TO 115
001061      END

```