UC-78

MASTER

UCRL-52501

T: A CONTROLLER FOR PLOTTING DATA THE MARK I BOILING WATER REACTOR RE SUPPRESSION EXPERIMENT

This work was supported by the U.S.
Interagency Agreement DOE

LAWRENCE
LIVERMORE
LABORATORY

LAWRENCE LIVERMORE LABORATORY
University of California/Livermore, California/94550

UCRL-52501

PSEPLOT: A CONTROLLER FOR PLOTTING DATA FROM THE MARK I BOILING WATER REACTOR PRESSURE SUPPRESSION EXPERIMENT

G. S. Holman

MS. date: May 10, 1978

FOREWORD

This work was supported by the U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Division of Water Reactor Safety Research under Interagency Agreement DOE 40-550-75 with the U.S. Department of Energy.

CONTENTS

LIST OF ILLUSTRATIONS

ABSTRACT

PSEPLOT is a computer routine that was developed for the Lawrence Livermore Laboratory Octopus computer system to generate several thousand plots of engineering data in a consistent format for referencing and comparison. The time-dependent engineering data were recorded during each of 25 tests of the Mark I Pressure Suppression Experiment (PSE). Although PSEPLOT is restricted to PSE, its concept is applicable to any similar data management task.

PSEPLOT is actually three independent routines (controllees) tied together within the framework of an ORDER system controller that executes them and interfaces Octopus utility routines. The first controllee, PREPLOT, retrieves data files from archival storage via a highly structured library system developed for PSE. The second controllee, FOURPLOT, plots data in a four-per-page format. Results from each PSE test, plotted consistently page-for-page, can be quickly compared. The final controllee, POSTPLOT, automatically copies the plots into long-term storage or sends them to various Octopus output devices, or both, depending on user option. This multiple-controllee approach maximizes code flexibility while simplifying execution of the numerous Octopus utility routines required to interface file manipulation and plot generation. For example, adding new format options to PSEPLOT is simply a matter of adding new plotting controllees. PREPLOT or POSTPLOT require no generic changes since neither has any direct plotting function.

PSEPLOT processes all data for any PSE test in under two central-processing-unit (CPU) minutes. It can be executed for different tests with minimal (one card) changes in input, yet retains many user options for file transport and plot output. Although the large number of system calls and file operations in PSEPLOT makes it somewhat susceptible to perturbations in Octopus, an extensive execution record simplifies error recovery.

INTRODUCTION

PSEPLOT is a computer code developed for plotting physical data recorded during the 25 tests of the Mark I Boiling Water Reactor (BWR) Pressure Suppression Experiment (PSE). The primary purpose of PSE was to determine the vertical forces on a 1/5-scale BWR pressure suppression system during a hypothetical loss-of-coolant accident. Physical data were recorded at some 200 active transducers connected to the experimental facility during each test.[1]

The large quantity of data recorded during the PSE gave rise to two significant tasks:

- To present the data in a clear, concise manner that would allow quick reference to a given physical observable as well as straightforward comparisons among tests;
- To expedite actual data processing.

In response to the first task, it was decided to generate time plots of the data using a four-per-page format.

Responding to the second task, we first had to consider the quantity of data recorded. The instrumentation for each test included four multiplex amplifiers or low frequency acquisition units (LOFAUs), each connected to as many as 60 transducers. Because complete presentation of the PSE data required generation of several thousand plots, a comprehensive computer code that would generate the PSE plots and also perform all required data file manipulations became quite attractive.

The specific operations that such a code should perform were outlined:

- Retrieve PSE data from the filing system within which it is stored.
- Generate plots of data recorded for a specified experimental time period.

- Obtain hard copy of the plots.
- Store the plots for future reference in a manner consistent with that used to store the PSE data.

We built in features to make the code attractive to users:

- User input structured so that PSE data could be processed from test to test with minimal changes (one-card changes were considered desirable).
- Sufficient options provided to allow the user flexibility with regard to data retrieval and plot disposition.
- Code designed to be as resistant as possible to perturbations in the computer system.
- Record of execution provided to the user so that code errors (whether user- or system-related) could be quickly isolated and resolved.

PSEPLOT was developed in response to these specifications. Actually, PSEPLOT is an ORDER[2] control routine that links three independent subsidiary controllees:

PREPLOT - a preprocessor that retrieves engineering data from long-term storage (tape or Elephant photostore);

FOURPLOT - a routine that performs four-per-page plotting of the engineering data;

POSTPLOT - a postprocessor that controls disposition of the plots generated by FOURPLOT.

The multiple-controllee approach arises from the original concept of PSEPLOT as a general PSE plotting controller and not as simply a four-per-page routine. Note the code flexibility gained by assigning each controllee a specific function--PREPLOT and POSTPLOT only manipulate files while FOURPLOT is strictly a plotting routine. Thus, adding format options is a matter of providing additional plotting controllees within PSEPLOT. No generic changes are required in either PSEPLOT or POSTPLOT. In fact, we plan to incorporate additional format capabilities into PSEPLOT.

PSEPLOT was developed to be executed with the following minimum data supplied by the user: PSE test number, LOFAU numbers, and the upper and lower bounds of the desired time window. The price paid for such simplified input is that PSEPLOT is only compatible with the PSE data filing system and is not generally applicable beyond PSE. However, the generic concepts behind PSEPLOT are not limited to the PSE alone, and with minor modifications PSEPLOT could be adapted for use with any filing system similar to that used for the PSE.

ORGANIZATION OF PSE DATA

Data recorded from a given transducer during a test is stored in an "engineering file" whose name follows the general format

WLnnnlttt

where *nnn* is the PSE test number without decimal point, *l* is the number of the LOFAU to which the transducer is connected, and *ttt* is the number designator of the transducer. File WL312109, for example, contains data recorded from strain-gauge transducer 109, LOFAU 2, during Test 3.1. A partial listing of this engineering file (Fig. 1) illustrates the basic data storage format.

For plotting purposes, the quantity *WLnnnl* is referred to as the "basename" of the engineering file.

Engineering files corresponding to all transducers connected to each LOFAU are collected in "engineering libraries," one library for each of the four LOFAUs used during a given test. These libraries are master files that efficiently organize many individual files under one heading without sacrificing individual file identity.[3] PSE library names follow the general format.

LXENGlnnn

where *nnn* is the test number and *l* is the LOFAU number. Fig. 2 shows a typical library listing for library LXENG23.1, which contains the sample engineering file WL312109 previously discussed.

NRC TEST NO. 3.1
MPX CHANNEL: 9
TIME PER POINT: 2.86350E-03
NUMBER OF POINTS: 3492
STRAIN, MICRO IN/IN (SG-7) RNGHDR, 4-0 DEG.

1.9147E-01	-1.7233E+00	1.1488E+00	1.9147E-01	1.9147E-01	1.9147E-01
1.9147E-01	-1.7233E+00	1.1488E+00	1.9147E-01	1.9147E-01	-7.6589E-01
1.1486E+00	-2.6806E+00	1.1486E+00	1.9147E-01	1.9147E-01	1.9147E-01
1.9147E-01	-7.6589E-01	1.1488E+00	1.9147E-01	1.1488E+00	1.9147E-01
1.9147E-01	-7.6589E-01	1.9147E-01	-7.6589E-01	1.9147E-01	1.9147E-01
1.9147E-01	1.9147E-01	1.9147E-01	-7.6589E-01	1.9147E-01	1.9147E-01
1.9147E-01	1.9147E-01	1.1488E+00	-7.6589E-01	1.1488E+00	1.9147E-01
1.9147E-01	1.9147E-01	1.1488E+00	-7.6589E-01	1.9147E-01	-7.6589E-01
-7.6589E-01	-7.6589E-01	1.9147E-01	-7.6589E-01	1.1488E+00	-7.6589E-01
-7.6589E-01	-7.6589E-01	1.9147E-01	-1.7233E+00	1.1488E+00	1.9147E-01
1.9147E-01	-7.6589E-01	1.9147E-01	-7.6589E-01	1.9147E-01	-1.7233E+00
1.9147E-01	-7.6589E-01	-7.6589E-01	-7.6589E-01	1.1488E+00	-1.7233E+00
-7.6589E-01	-7.6589E-01	-7.8589E-01	1.9147E-01	1.9147E-01	-1.7233E+00
1.9147E-01	-1.7233E+00	-7.6589E-01	1.9147E-01	1.9147E-01	-1.7233E+00
1.9147E-01	-7.6589E-01	1.9147E-01	1.9147E-01	1.9147E-01	-2.6806E+00
-7.6589E-01	-7.6589E-01	-7.6589E-01	1.9147E-01	1.9147E-01	-1.7233E+00
-7.8589E-01	-2.6806E+00	-7.6589E-01	-7.6589E-01	1.9147E-01	-7.6589E-01
1.9147E-01	-1.7233E+00	1.9147E-01	-1.7233E+00	-7.6589E-01	-1.7233E+00
1.9147E-01	-2.6806E+00	1.9147E-01	-1.7233E+00	-7.8589E-01	-1.7233E+00
-7.6589E-01	-1.7233E+00	1.9147E-01	-7.6589E-01	-7.6589E-01	1.9147E-01
1.9147E-01	-2.6806E+00	-7.6589E-01	-2.6806E+00	-7.8589E-01	-7.6589E-01
1.9147E-01	-7.6589E-01	1.9147E-01	-2.6806E+00	-7.6589E-01	-1.7233E+00
1.9147E-01	-7.8589E-01	1.9147E-01	-2.6808E+00	-7.6589E-01	-1.7233E+00
-7.6589E-01	-1.7233E+00	1.9147E-01	-2.6808E+00	-7.6589E-01	-1.7233E+00
-7.8589E-01	-1.7233E+00	-7.6589E-01	-7.8589E-01	-7.6589E-01	-1.7233E+00
-1.7233E+00	-1.7233E+00	-7.6589E-01	-2.6808E+00	1.9147E-01	-1.7233E+00
-7.6589E-01	-1.7233E+00	-7.6589E-01	-3.6380E+00	-1.7233E+00	-1.7233E+00
-1.7233E+00	-7.6589E-01	-7.6589E-01	-1.7233E+00	-7.6589E-01	-3.6380E+00

FIG. 1. Partial listing of sample engineering file WL312109.

ADDRESS	LENGTH	NAME	ADDRESS	LENGTH	NAME
0	11123	WL312103	367301	11123	WL312303
11123	11123	WL312104	400424	11123	WL312304
22246	11123	WL312105	411547	11123	WL312305
33371	11123	WL312106	422672	11123	WL312306
44514	11123	WL312107	434015	11123	WL312307
55637	11123	WL312109	445140	11123	WL312308
66762	11123	WL312110	456263	11123	WL312309
100105	11123	WL312111	467406	11123	WL312310
111230	11123	WL312113	500531	11123	WL312311
122353	11123	WL312114	511654	11123	WL312312
133476	11123	WL312115	522777	11123	WL312314
144621	11123	WL312201	534122	11123	WL312315
155744	11123	WL312202	545245	11123	WL312401
167067	11123	WL312204	556370	11123	WL312404
200212	11123	WL312205	567513	11123	WL312405
211335	11123	WL312206	600636	11123	WL312406
222460	11123	WL312207	611761	11123	WL312407
233603	11123	WL312208	623104	11123	WL312408
244726	11123	WL312209	634227	11123	WL312409
256051	11123	WL312210	645352	11123	WL312410
267174	11123	WL312211	656475	11123	WL312411
300317	11123	WL312212	667620	11123	WL312412
311442	11123	WL312213	700743	11123	WL312413
322565	11123	WL312214	712066	11123	WL312414
333710	11123	WL312215	723211	11123	WL312415
345033	11123	WL312301	734334	11123	WL312501
356156	11123	WL312302	745457	11123	WL312502
			SPACE IS	20044	
			BEGINS AT	756602	
			INDEX SPACE	244	DECIMAL

FIG. 2. Engineering file list, library LXENG23.1.

FORMAT OF FOUR-PER-PAGE PLOTS

A dd80 file containing a set of four-per-page plots of data in the sample engineering library LXENG23.1 is available as follows:

ELF RDS .406362:PSEAIR:PLOTS:SAMPLE

View the plots by using the utility routine DDTV.[4] Hard copy may be obtained either by using DDTV or by giving the dd80 file to User-1.*

The following information is included on each individual plot as shown in Fig. 3:

- PSE air test number
- Engineering file name
- Location of transducer
- Parameter measured.

In addition, each page of plots lists the air test number, LOFAU number, and page number out of the total for that LOFAU. This information ensures that the page will stand alone should it become separated from its set.

If a plot is not generated for a particular LOFAU multiplex channel, the message "FILE NAME NOT USED: *filename*" is printed in place of a plot. Information regarding the status of the corresponding transducer can then be obtained from the transducer status chart in Ref. 1.

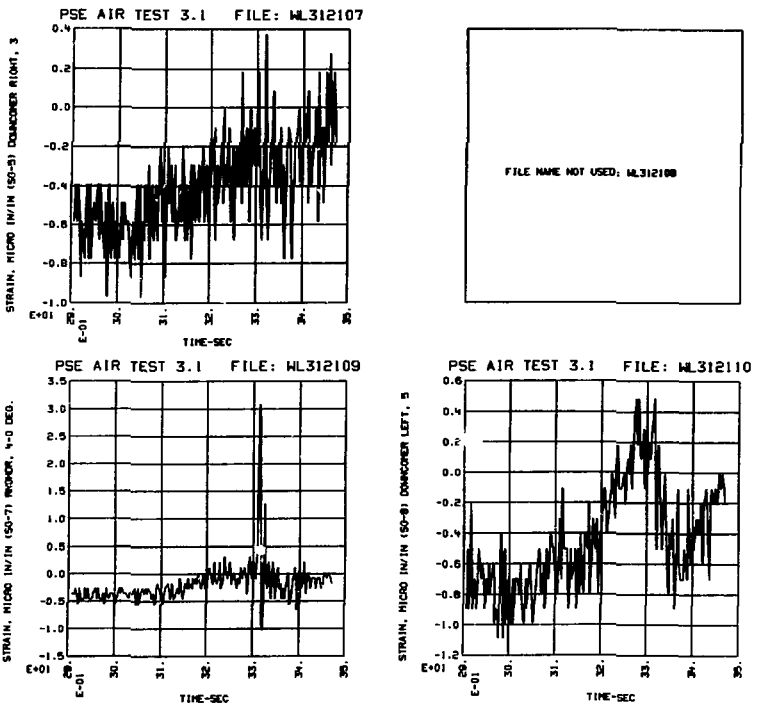
PSEPLOT OPERATION

Users can vary the operations performed by PSEPLOT, but, in general, plots are generated by PSEPLOT as follows:

- Execution of PSEPLOT begins.
- Execution of PREPLOT begins.
- User-supplied input data are read and engineering library names are generated accordingly.

---

*The term "User-1" is jargon for a system of problem programs recognized by the FLOE operating system as privileged extensions of itself. See Ref. 7 for details.



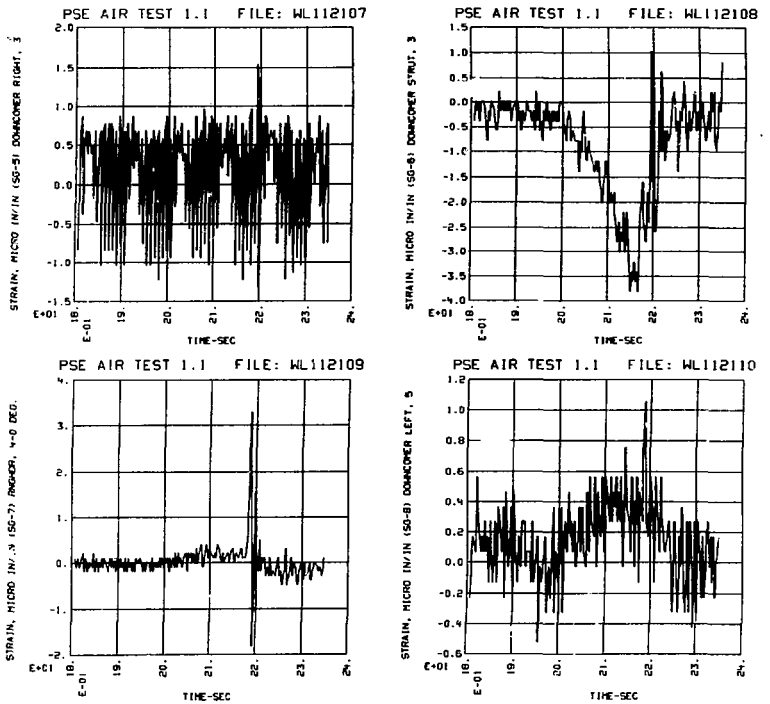
NOTE: THESE DATA ARE ORIGINAL AND UNFILTERED

FIG. 3. Sample page of PSE data plotted using PSEPLOT.

- Engineering libraries for the specified test are copied from long-term storage to disk.
- Engineering libraries are assigned dummy names for use in PSEPLOT.
- Control returns to PSEPLOT.
- An engineering library is selected for processing.
- The engineering files stored within the library are copied to disk.
- Execution of FOURPLOT begins.

FOURPLOT systematically searches (by transducer number, in ascending numerical order) the disk file index for file names corresponding to each of the transducers that may have been connected to the LOFAU (i.e., engineering library) in question. If such a file is found, it is opened and the data within the specified time window plotted. If no file is found, a message to this effect is printed in lieu of a plot. This plotting scheme ensures that plots for a given LOFAU will follow an identical page format for all tests and thus simplifies test-to-test comparisons (Fig. 4). No attempt is made by FOURPLOT to clarify the absence of a plot for a particular file because detailed information pertaining to transducer status is supplied in Ref. 1.

- Execution of FOURPLOT ends upon completion of the engineering file search.
- Control returns to PSEPLOT.
- The engineering files on disk are destroyed and a new library selected.
- The search and plotting sequence is repeated until all of the engineering libraries have been processed.
- Execution of POSTPLOT begins.
- The dd80 files containing the actual plots are copied into a dd80 library and are then sent to user-specified output devices. The dd80 library is then copied into long-term storage for future reference.
- Execution of POSTPLOT ends.
- Control returns to PSEPLOT.
- Working files are erased from disk. A recording of execution, record of ORDER, and all dd80 generated by FOURPLOT remain on disk.
- Execution of PSELOT ends.



NOTE: THESE DATA ARE ORIGINAL AND UNFILTERED

FIG. 4. Sample page of data plotted for PSE Test 1.1.

The dd80 files that remain on disk are collected in file families, one family per LOFAU, whose names follow the general format

DXWL*nnn1*A

Note that the family name is simply the basename of the engineering files with prefix "DX." The plots for our sample library LXENG23.1, for example, require one dd80 file which is named DXWL312A. Should it be necessary to use more than one dd80 file for a set of plots, the subsequent files are named DXWL*nnn1*B, DXWL*nnn1*C, etc.

INPUT DATA

Input data for PSEPLOT is read from a disk file named INPLOT. The user is not required to create this file directly; instead, data cards are included in the ORDER control routine and creation of INPLOT proceeds automatically. Cards 6 through 11 of PSEPLOT are designated as data cards (Appendix A). A discussion of each card's structure and examples of use follow (Note that the English system of units is to be used except where otherwise noted):

CARD 6 (A10,2E10.3,4I5), Control data card

TESTNO	-	PSE test number
TMIN	-	lower bound of plotting time window
TMAX	-	upper bound of plotting time window
LOFAU(I),I=1,4	-	LOFAU number(s) to be plotted

Card 6 supplies the core information required for the processing of data from any PSE test, i.e., the test number, the LOFAUs desired, and the time window within which the plots will be generated. From one to four LOFAUs may be specified for a single execution of PSEPLOT.

Example 1

Plot data from all four LOFAUs for PSE test 3.1 between 2.905 and 3.465 sec.

```
2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)2 4 6 8(5)
      3.1     2.905     3.465    1    2    3    4
```

Example 2

Plot data from LOFAU 2 for PSE test 3.1 between 2.905 and 3.465 sec.

```
2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)2 4 6 8(5)
      3.1     2.905     3.465    2
```

CARD 7 (4A10), Engineering library ELF directory

ISTOR(I),I=I,3 - ELF directory in which PSE engineering libraries are stored (30 characters maximum)

TAPENO - PSE data tape (see below)

In the default mode, PREPLOT copies the engineering libraries from the Elephant photostore system using the utility routine MCT[5] without a tape backup. If TAPENO is included on Card 7, then PREPLOT (i.e., MCT) will attempt to read the PSE data from the specified tape should photostore be unavailable. If the tape number is prefixed with an asterisk (*), then PREPLOT will read the PSE data directly from the tape specified, disregarding any ELF directory information on the card.

If directory information on Card 7 is omitted and TAPENO is not prefixed with an asterisk, PREPLOT will copy the PSE engineering libraries from the ELF take directory .581225:PSEAIR:TEST*nnn*, where *nnn* is the test number on Card 6.

Example 3

Copy engineering libraries from the private ELF directory .R:PSEAIR:DATA. Tape AA203 is to be the MCT backup.

```
2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)
.R:PSEAIR:DATA                    AA203
```

Example 4

Copy engineering libraries directly from tape AA203.

```
2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)
                                  *AA203
```

CARD 8 (4A10), Engineering library names

LIB(I),I=1,4 - Engineering library names (ten characters or less)

If Card 8 is left blank, library names will be generated internally for each LOFAU following the format LXENG*lnnn*, where *l* is the LOFAU number and *nnn* is the test number including decimal points. (See Example 6 for one exception to this rule.) If the library names are put on Card 8, the order in which they appear must correspond to the order in which the LOFAU numbers are specified on Card 6, i.e., LIB(I) corresponds with LOFAU(I).

Example 5

Input engineering library names for Test 3.1 directly. Note that the names given here are those that would be generated internally by PREPLOT and are therefore not required for an actual run. Note also how the order in which the libraries are specified corresponds with the LOFAU numbers specified in Example 1.

```
2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)
LXENG13.1 LXENG23.1 LXENG33.1 LXENG43.1
```

Example 6

Input engineering library names for Test 1.3.1 directly.

```
2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)
LXENG1131 LXENG2131 LXENG3131 LXENG4131
```

Note that the engineering libraries for Test 1.3.1 are the only PSE air test libraries that do not follow the standard naming scheme. Because decimal points are not included in the library names, the library names must be input directly for this test.

CARD 9 (15,3A10), dd80 library options

IRPLC = 0 - produce an error message if a name duplication occurs in library (default)

= 1 - replace existing files in dd80 library if name duplication occurs

DD80LIB - dd80 library name

LIBLNGTH - library length (octal)

LIBTAPE - (see below)

Card 9 allows the user to select both the name and octal length of the library file into which the four-per-page dd80 files will be written. If no library name is specified, dd80 files will be copied into a default library named LXDX*nnn*, where *nnn* is the PSE test number with decimal points included. The default library length for a new library is 500000 octal words (whether DD80LIB is specified or not), or it is the current length of a library on disk or in long-term storage.

LIBTAPE is equivalent to the parameter TAPENO on Card 7, in this case specifying the backup tape that is to be searched for DD80LIB in the event that photostore is unavailable. As with TAPENO, if LIBTAPE is prefixed with an asterisk, the dd80 library (if it exists) is copied directly from the specified tape and ELF directory information on Card 10 is disregarded. If LIBTAPE is left blank, TAPENO will be the MCT backup tape used.

Example 7

Plots are to be stored in a library called LXENG4PLT with a length of 1000000 octal words. Replace existing files in the library if any name duplications occur. Tape AA203 is the dd80 library backup tape.

```
2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2  5
   1 LXENG4PLT   1000000     AA203
```

CARD 10 (3A10), dd80 library directory

LSTOR(I),I=1,3 - ELF directory in which DD80 library is stored (30 characters maximum)

If Card 10 is left blank and the dd80 library does not already exist on disk, the ELF take directory .581225:PSEAIR:TEST*nnn* is searched for the dd80 library specified on Card 9 (or for the default library LXDX*nnn* if DD80LIB has been left blank).

Card format is identical to that of Card 7 with the tape number omitted.

CARD 11 (15,6A10), Plot output options

IPLOT = 0 - no fiche/dd80 hardcopy output (default)
= 1 - dd80 hardcopy only
= 2 - fiche output through utility routine FROG
IFROG(I),I=1,6 - FROG options (only used if IPLOT = 2)

Card 11 controls disposition of the plots after the dd80 files have been copied into the dd80 library. For IPLOT = 1, copies of the dd80 files are given to User-1 for printing as 14 in. x 14 in. dd80 hardcopy. If IPLOT = 2, POSTPLOT acts as a controller for the utility routine FROG[6] and microfiche output is generated in accordance with user-supplied FROG options (default is 105mm, cinema format). If both fiche and dd80 hardcopy are desired, use the FROG option dd80.

Example 8

Generate 35mm film of plots in cinema format together with dd80 hardcopy. Destroy the dd80 files after use.

```
2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)2 4 6 8(5)
   2      35MM.      CINE.      DD80.      DEST.
```

CARD 12 (3A10), dd80 library long term storage

LWRIT(1),I=1,3 - ELF directory into which dd80 library is to be written (30 characters maximum)

Card 12 controls long-term storage of the dd80 library. The card is left blank if no storage of the dd80 library is desired. Otherwise, the library is written into the ELF photostore directory specified with the right-adjusted word TAPE as tape backup. If the right-adjusted word TAPE is entered for IWRIT(1), the library will be copied directly to LIBTAPE.

Example 9

Write the dd80 library directly to LIBTAPE.

```
2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)
     TAPE
```

Note that this option is valid only if LIBTAPE has been specified on Card 9.

EXECUTION

To generate plots of PSE data it is first necessary to have on disk the controllees PREPLOT, FOURPLOT, and POSTPLOT in addition to the ORDER controller PSEPLOT. These may be obtained by copying the library file LXPSEPLOT from the ELF take directory .406362:PSEAIR and then using the LIX execute line

```
LIX LXPSEPLOT!GR. PSEPLOT PREPLOT FOURPLOT POSTPLOT!END /1 1
```

to copy the above mentioned files to disk.

PSEPLOT is next modified as discussed in the preceding section to supply input data and to control disposition of the program execution log.

PSEPLOT may be run in either a batch or remote mode. Most users will probably find it more convenient to run PSEPLOT from a tel ype, especially when processing large amounts of data.

SAMPLE PROBLEM

Plot all data recorded between 2.905 and 3.465 sec for PSE air test 3.1, store the plots in a dd80 library named LXENG4PLT, and output the dd80 files to microfiche and dd80 hardcopy. The dd80 library will not be written into long-term storage. The PSEPLOT input data could take the following form:

```
Column:     2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)2 4 6 8(5)
Card 6:           3.1     2.905     3.465    4    3    2    1
Card 7:   [.581225:PSEAIR:TEST3.1]
Card 8:   [ LXENG43.1 LXENG33.1 LXENG23.1 LXENG13.1 ]
Card 9:       1 LXENG4PLT     500000
Card 10:  [.581225:PSEAIR:TEST3.1]
Card 11:      2     35MM.      CINE.      DD80.      DEST.
```

Bracketed terms represent the defaults generated by PSEPLOT if the indicated cards are left blank.

To begin execution of PSEPLOT, the user types the execute line as follows:

```
ORDER PSEPLOT / t v
```

For any PSE test the amount of machine time required to process one set of plots (i.e., one LOFAU) varies with the number of active transducers connected to the specified LOFAU. A good rule of thumb is to allow at least one CPU minute for each LOFAU that is to be processed.

FILE PREPROCESSING

After the execute line is entered, PSEPLOT responds with the following:

```
*ID   752NUS        0.00//001  1  09/14 12.512U G. HOLMAN   BOX RO4
EXECUTION
     RECORD OF EXECUTION ----- PROGRAM PREPLOT
BEGIN EXECUTION OF PREPLOT:  U 09/14/77  12:30:48
     PSE TEST NUMBER:          3.1
     LOFAU NUMBER(S)      4    3    2    1
     EXPANDED PLOT TIMES (SEC):  TMIN = 2.905E+00
                                 TMAX = 3.465E+00
     LIBRARY FILES:    LXENG43.1   LXENG33.1   LXENG23.1   LXENG13.1
BEGIN EXECUTION OF UTILITY ROUTINE MCT:  12:30:52
```

PREPLOT next reads the engineering libraries from the photostore. Routine/user interaction with MCT is allowed during this phase of execution. The routine will accept all messages normally accepted by MCT (END, RPT, etc.) and will send MCT error messages directly to the teletype.

If any of the engineering libraries exist on disk prior to the execution of MCT, a message to this effect is sent to the teletype and the file name is deleted from the MCT execute line.

Upon termination of MCT, the routine will respond as follows:

```
END EXECUTION OF UTILITY ROUTINE MCT:          12:43:42
SET DUMMY LIBRARY FILE NAMES
     FILE NAME   LXENG43.1  SWITCHED TO   DUMY 1
     FILE NAME   LXENG33.1  SWITCHED TO   DUMY 2
     FILE NAME   LXENG23.1  SWITCHED TO   DUMY 3
     FILE NAME   LXENG13.1  SWITCHED TO   DUMY 4
END EXECUTION OF PREPLOT:   12:43:46
```

This message indicates that PREPLOT has switched the names of the engineering libraries to dummy names to be used by PSEPLOT. Had PREPLOT been unable to perform a name change because a particular library had not been found on disk (most likely because of an MCT error), a message would have been printed to this effect.

PREPLOT then terminates, modifying the ORDER branch buffer in accordance with the number of files retrieved from long-term storage. Because four libraries have been retrieved from storage, four sets of plots will be generated. Control within the ORDER routine will, therefore, be transferred to the card with the branch name PLOT4. In general, control is transferred to the card with the branch name PLOT*n* where *n* is the number of engineering libraries found on disk after the execution of MCT. Thus, if only two LOFAUs were originally specified, or if PREPLOT successively retrieved only two libraries out of four originally specified, the ORDER job stream would branch to PLOT2 accordingly. If no libraries are successfully retrieved from storage, control is transferred to the branch name LAST and PSEPLOT terminates.

PLOT GENERATION

Four new files exist on disk in addition to the libraries DUMY1 through DUMY4. These files are named INPUT1 through INPUT4, respectively, and contain the following FOURPLOT input data for each of the four engineering libraries: engineering file base name, time window for plotting, and engineering library name.

PSEPLOT responds with

```
*NXT   752NUS          0.04 PLOT 4  1   09/14  12.730U  G. HOLMAN  BOX R04
 EXECUTION
*NXT   752NUS          0.13 //003  1   09/14  12.739U  G. HOLMAN  BOX R04
```

indicating execution of utility routine LIX to copy all engineering files in the engineering library DUMY4 (i.e., LXENG13.1) to disk and execution of the utility routine COPY to copy the file INPUT4 to the file INPUT required by FOURPLOT.

The actual plotting of the data stored in library LXENG13.1 is recorded as follows:

```
*NXT   752NUS        0.13//004          12.740U  G. HOLMAN        BOX R04
 EXECUTION
          RECORD OF EXECUTION-----PROGRAM FOURPLOT
BEGIN EXECUTION OF FOURPLOT:  U 09/14/77  12:44:26
     DATA FILE LIBRARY:   LXENG13.1
     BASE NAME:     WL311
     DD80 FAMILY NAME:    DXWL311A
     EXPANDED PLOT TIMES (SEC):   TMIN = 2.905E+00
                                  TMAX = 3.465E+00
PLOTTING  . . . . . . . . .
     CONSULT EXECUTION LOG FOR LIST OF FILES PLOTTED
END EXECUTION OF FOURPLOT:  12:46:15
MACHINE ITEM USED (MIN):    0.745
```

A detailed listing of files actually plotted is included in the execution log RUNLOG left on disk after termination of PSEPLOT.

Following the generation of this particular set of plots, PSEPLOT responds

```
*NXT   752NUS        0.67    END4   1   09/14   12.771U  G. HOLMAN BOX R04
EXECUTION
```

erasing from disk the engineering library and all of the engineering files (using the utility routine DESTROY) before proceeding to next set of plots. The process is then repeated for cards PLOT3 through END3, PLOT2 through END2, and PLOT1 through END1, i.e., for each engineering library DUMY3 through DUMY1 (LXENG23.1 through LXENG43.1) remaining on disk.

During the execution of FOURPLOT, the user may request a short edit message as follows:

```
      User:  EDIT
   Routine:          FILE filename
```

where *filename* is the name of the last engineering file processed by FOURPLOT. The user may also manually terminate FOURPLOT at any time by typing "END" as follows:

```
      User:  END
   Routine:  MSG END RECEIVED FROM USER
             CODE TERMINATING MANUALLY BY USER OPTION
                LAST FILE PLOTTED WAS filename
```

Upon manual termination of FOURPLOT, job control will pass to END4 (or in gneral, END*n*) as previously noted.

FILE POSTPROCESSING

At the successful completion of plot generation, four new files exist on disk, DXWL311A, DXWL312A, DXWL313A, and DXWL314A. These are the dd80 files that contain the four-per-page plots generated for LOFAUs 1 through 4, respectively. It now remains for POSTPLOT to write these files into the dd80 library LXENG4PLT.

Initiation of the controllee POSTPLOT is indicated by

```
*NXT   752NUS        2.10 NX2LA  1     09/14  12.854U  G. HOLMAN   BOX R04
                     RECORD OF EXECUTION ----- PROGRAM POSTPLOT
BEGIN EXECUTION OF POSTPLOT:  U        09/14/77  12:51:19
                     PSE TEST NUMBER:  3.1
                     DD80 FILES:  DXWL311A  DXWL312A  DXWL313A  DXWL314A
```

Having searched the user's private file index for all dd80 ("DX") files, POSTPLOT now begins writing these files into the specified dd80 library.

```
BEGIN EXECUTION OF UTILITY ROUTINE LIX:  12:51:19
     DD80 LIBRARY NAME:  LXENG4PLT
                     U1021 WED AUG 31, 77  LXENG4PLT  OK  RW
     WRITE FILE      DXWL311A INTO LIBRARY
             OK
     WRITE FILE      DXWL312A INTO LIBRARY
             OK
     WRITE FILE      DXWL313A INTO LIBRARY
             OK
     WRITE FILE      DXWL314A INTO LIBRARY
              NAME DUPLICATION         DXWL314A
             OK
     REPLACE FILE    DXWL314A IN LIBRARY
             OK
ALL DONE
END EXECUTION OF UTILITY ROUTINE LIX
```

Note here that the dd80 library already existed in the user's private file index. Had this not been the case, the library would have first been sought in long-term storage using the utility routine MCT (as with the controlleee PREPLOT, user/routine interaction with MCT is allowed). If the file had still not been found, a new file LXENG4PLT would have been created by POSTPLOT. In this case, the user would have been informed by the message

```
DD80 LIBRARY NAME:   LEXENG4PLT
CREATE NEW LIBRARY FILE   LXENG4PLT
```

As indicated, one name duplication was encountered in LXENG4PLT. For a name duplication in the library, the suitable LIX error message is sent to the teletype and the offending file replaced or not replaced in accordance with the parameter IRPLC defined by the user.

Once library operations are completed, disposition of the dd80 library and the individual dd80 files is undertaken as specified by the user. Since in our current example no long-term storage of the dd80 library was requested, POSTPLOT responds with

```
    NO LONG TERM STORAGE OF DD80 LIBRARY REQUESTED
```

followed immediately by

```
BEGIN OUTPUT OF DD80 PLOTS:  12:51:56
BEGIN EXECUTION OF UTILITY ROUTINE FROG:  12:51:59
     DD80 FILE FAMILY NAMES:  DXWL311A  DXWL312A  DXWL313A  DXWL314A
     FROG OPTIONS SELECTED:  DD80.  35MM.  CINE.  DEST.
END EXECUTION OF UTILITY ROUTINE FROG:  12:53:05
END EXECUTION OF POSTPLOT:  12:53:06
MACHINE TIME USED (MIN):  0.026
```

If dd80 hardcopy alone is specified (i.e., by setting IPLOT = 1 on Card 8), each dd80 file is copied and sent to User-1. Had storage of the dd80 library been requested, MCT would have been executed as before, this time in a write mode.

TERMINATION

Following successful completion of POSTPLOT, all working files are destroyed. The record of execution, stored in the disk file RUNLOG, is then output as specified by the user. PSEPLOT now terminates, writing the record of ORDER into a disk file named ORDERROOO.

The following files remain on disk after normal job completion:

- PSEPLOT
- The three original controllees
- All dd80 files generated by FOURPLOT
- RUNLOG
- ORDERROOO

PSEPLOT also may be terminated manually at any time during execution either by setting sense switch 1 (SW1) or by typing (CTRL-O) at the teletype.

RECORD OF EXECUTION--RUNLOG

The comprehensive record of execution left on disk after completion of the sample problem is given in Appendix B. This record is essentially identical in format to the teletype record of execution except that a detailed listing of files plotted during each execution of FOURPLOT is provided.

RESTRICTIONS AND RECOMMENDATIONS

The following restrictions must be observed when running PSEPLOT:

- PSEPLOT may not be run simultaneously under more than one suffix at a time.
- The utility routine LIX may not be run if PSEPLOT is active under a different suffix. Failure to observe this restriction can disrupt library operations in PSEPLOT as a result of the single suffix restriction on LIX.
- Any files in the user's private file index beginning with the characters $, I, or WL will be destroyed during PSEPLOT file cleanup operations whether they are connected with PSEPLOT or not.

The following may be regarded as recommendations only:

- All file names beginning with DX should be removed from disk prior to execution of PSEPLOT. While failure to do this will not adversely affect job execution, it can allow superfluous files to be added to the dd80 library or unnecessary file replacements to be made within the library.
- Following each execution of PSEPLOT, create a new record file from the contents of RUNLOG and ORDERROOO. This file can then be stored along with the dd80 files generated by PSEPLOT as a permanent record of plot generation. Otherwise, the contents of RUNLOG and ORDERROO will be lost through execution of PSEPLOT.

These last two recommendations may be disregarded if the system is not to be used.

- PSEPLOT may be run simultaneously on more than one machine, but this is not recommended due to restrictions on the Elephant filing system. If PSEPLOT is to be simultaneously run on more than one machine, it is recommended that the engineering libraries first be copied from photostore manually using either ELF or MCT.
- Because of the single suffix restriction on the Elephant system, neither ELF nor MCT should be run if either PREPLOT or POSTPLOT is active under a different suffix.

ERRORS

During early development, it was recognized that the large number of file operations performed during any execution of PSEPLOT left the code somewhat susceptible to perturbations in the computer system. Therefore, considerable effort went into insulating the code against system errors and into providing the user with the means to effectively trace code errors.

Whenever possible, error flags are provided in PSEPLOT's subsidiary controllees as protection against file create errors, file access errors, or FROST call errors that may result from errors in system calls. If such an error should occur, a suitable message is sent to the user and recorded in the execution log. In certain cases, the message includes an error number, which corresponds to information in the CDC 7600 System Calls Manual.[7] The offending controllee then terminates and execution of PSEPLOT continues.

Errors in user input, other than format errors, are treated similarly where possible. Improper input format usually results in an ORDERLIB error causing immediate abortion of the controllee.

After a controllee error, transfer of job control within PSEPLOT varies with the particular controllee and the type of error detected. Following a normal error termination of PREPLOT, control will branch to LAST and PSEPLOT will terminate after sending the execution log RUNLOG to a user-selected output device. Detection of a FLOE or ORDERLIB error, however, causes PSEPLOT to terminate immediately. In either case, the controllee drop file +PREPLOT remains on disk following termination of PSEPLOT.

POSTPLOT differs only slightly in that all errors send job control to LAST, with the drop file +POSTPLOT left on disk following termination of PSEPLOT. Note that errors in POSTPLOT will not directly affect dd80 files generated by PSEPLOT.

Transfer of job control is handled differently for FOURPLOT. Since the processing of each LOFAU is an independent operation within PSEPLOT, the detection of an error during one execution of FOURPLOT will not affect a subsequent execution. Therefore, instead of terminating PSEPLOT, the name of the FOURPLOT drop file +FOURPLOT is switched to +FOURPLOT*n* where *n* corresponds with the branch name PLOT*n* under which FOURPLOT is currently executing. This file remains on disk after termination of PSEPLOT.

Job control then proceeds as if no error had been detected in FOURPLOT. Thus, only the errant LOFAU, and not the entire test, need be reprocessed after PSEPLOT is completed.

DEBUGGING

Should problems arise during execution of PSEPLOT, the following debugging steps are recommended:

- Check the execution log.
  Most problems can be resolved immediately using the execution log. If the problem was caused by a system call error, repeat execution of PSEPLOT. Most likely, the job will execute properly the second time around, unless OCTOPUS is having major problems.

  Always check the execution log after any execution of PSEPLOT, even if PSEPLOT executed normally, to be sure that all of the engineering files for the specified LOFAUs were plotted. During times of heavy system usage, PSEPLOT may not be able to copy all of the files in an engineering library to disk because the system file index is full. Should this occur, repeat execution of PSEPLOT for the errant LOFAU(s).

- Check Input Data.
  Improper input data are the source of most code problems. If PSEPLOT terminated normally, refer to the execution log. If an ORDERLIB error was detected, check the input data directly.
- Use the Utility Routine DBUG.
  The utility routine DBUG[8] may be used to debug any of the PSEPLOT controllee drop files. To use DBUG, type the execute line

DBUG *dropfile ctle/* t v

where *dropfile* and *ctle* are the names of the drop file and controllee respectively. Source listings of PREPLOT, FOURPLOT, and POSTPLOT may be obtained by copying the files SPREPLOT, SFOURPLOT, and SPOSTPLOT respectively from the library LXPSEPLOT.

REFERENCES

1. E. K. Collins and W. Lai (ed.), Final Air Test Results for the 1/5-Scale Mark I Boiling Water Reactor Pressure Suppression Experiment, Lawrence Livermore Laboratory, Livermore, CA, Rept. UCRL-52371 (1977).
2. R. Potter et al. The ORDER System, Lawrence Livermore Laboratory, Livermore, CA, Rept. LTSS-202 (1974).
3. J. Ramus, Utility Routine LIX, Lawrence Livermore Laboratory, Livermore, CA, Rept. UR-321 (1975).
4. P. Keller, Utility Routine DDTV, Lawrence Livermore Laboratory, Livermore, CA, Rept. UR-412 (1975).
5. J. Wade, Utility Routine MCT, Lawrence Livermore Laboratory, Livermore, CA, Rept. UR-408 (1973).
6. C. Hunt, Utility Routine FROG, Lawrence Livermore Laboratory, Livermore, CA, Rept. UR-418 (1975).
7. P. Du Bois et al. CDC 7600 System Calls and I/O Requests, Lawrence Livermore Laboratory, Livermore, CA, Rept. LTSS-10 (1975).
8. D. Thompson et al. Utility Routine DBUG, Lawrence Livermore Laboratory, Livermore, CA, Rept. UR-108 (1973).

Appendix A

ORDER Control Routine PSEPLOT

```
 2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)2 4 6 8(5)2 4 6 8(6)2 4 6 8(7)2

*ID   account              job identification          your name  BOX ann
* BRANCH LAST,EXIT,EXIT,EXIT,EXIT
* FILES P-MCT
* XEQ PREPLOT


   Note:  The following eight cards generate the data
          file INPLOT


* DATA INPLOT A
      3.1     2.905     3.465    4    3    2    1
 .R:PSEAIR:PLOTS
 LXENG43.1 LXENG33.1 LXENG23.1 LXENG13.1
    1 LXENG4PLT
 .R:PSEAIR:PLOTS
    2     35MM.     CINE.     DD80.     DEST.
 .R:PSEAIR:PLOTS

*NXT  752NUS   PLOT4                                 your name  BOX ann
* BRANCH ,PLOT3,PLOT3,PLOT3,LAST
* XEQ LIX
* XEQMES DUMY4\GR. ALL.\END
*NXT  752NUS                                         your name  BOX ann
* XEQ COPY
* XEQMES INPUT4 INPUT
*NXT  752NUS                                         your name  BOX ann
* BRANCH END4,,,,LAST
* FILES +FOURPLOT INPUT
* XEQ FOURPLOT
*NXT  752NUS                                         your name  BOX ann
* XEQ SWITCH
* XEQMES +FOURPLOT +FOURPLOT4
*NXT  752NUS   END4                                  your name  BOX ann
* XEQ DESTROY
* XEQMES DUMY4 ALWITH. WL ALWITH. $

*NXT  752NUS   PLOT3                                 your name  BOX ann
* BRANCH ,PLOT2,PLOT2,PLOT2,LAST
* XEQ LIX
* XEQMES DUMY3^GR. ALL.\END
*NXT  752NUS                                         your name  BOX ann
* XEQ COPY
* XEQMES INPUT3 INPUT
```

```
*NXT  752NUS                                          your name  BOX ann
* BRANCH END3,,,,LAST
* FILES +FOURPLOT INPUT
* XEQ FOURPLOT
*NXT  752NUS                                          your name  BOX ann
* XEQ SWITCH
* XEQMES +FOURPLOT +FOURPLOT3
*NXT  752NUS   END3                                   your name  BOX ann
* XEQ DESTROY
* XEQMES DUMY3 ALWITH. WL ALWITH. $

*NXT  752NUS   PLOT2                                  your name  BOX ann
* BRANCH ,PLOT1,PLOT1,PLOT1,LAST
* XEQ LIX
* XEQMES DUMY2\GR. ALL.\END
*NXT  752NUS                                          your name  BOX ann
* XEQ COPY
* XEQMES INPUT2 INPUT
*NXT  752NUS                                          your name  BOX ann
* BRANCH END2,,,,LAST
* FILES +FOURPLOT INPUT
* XEQ FOURPLOT
*NXT  752NUS                                          your name  BOX ann
* XEQ SWITCH
* XEQMES +FOURPLOT +FOURPLOT2
*NXT  752NUS   END2                                   your name  BOX ann
* XEQ DESTROY
* XEQMES DUMY2 ALWITH. WL ALWITH. $

*NXT  752NUS   PLOT1                                  your name  BOX ann
* BRANCH ,LAST,LAST,LAST,LAST
* XEQ LIX
* XEQMES DUMY1\GR. ALL.\END
*NXT  732NUS                                          your name  BOX ann
* XEQ COPY
* XEQMES INPUT1 INPUT
*NXT  752NUS                                          your name  BOX ann
* BRANCH END1,,,,LAST
* FILES +FOURPLOT INPUT
* XEQ FOURPLOT
*NXT  752NUS                                          your name  BOX ann
* XEQ SWITCH
* XEQMES +FOURPLOT +FOURPLOT1
*NXT  752NUS   END1                                   your name  BOX ann
* XEQ DESTROY
* XEQMES DUMY1 ALWITH. WL ALWITH. $

*NXT  752NUS   NX2LAST                                your name  BOX ann
* BRANCH ,LAST,LAST,LAST,LAST
* XEQ POSTPLOT
```

```
*NXT  752NUS                                          your name  BOX ann
* XEQ DESTROY
* XEQMES +PREPLOT +POSTPLOT ALWITH. 1

*NXT  752NUS   LAST                                   your name  BOX ann
* PRINT ORDE
```

Note: The following cards are suggestions only. These may be modifed by the user to control disposition of the execution log and/or record of ORDER.

```
* XEQ ALLOUT
* XEQMES RJET nn RUNLOG\BOX ann & id\END
```

Appendix B

Sample Record of Execution --- PSE Air Test 3.1

RECORD OF EXECUTION ----- PROGRAM PREPLOT

```
BEGIN EXECUTION OF PREPLOT: U 09/14/77  12:30:48

     PSE TEST NUMBER:          3.1

     LOFAU NUMBER(S):     4    3    2    1

     EXPANDED PLOT TIMES (SEC):  TMIN = 2.905E+00
                                 TMAX = 3.465E+00

     LIBRARY FILES:    LXENG43.1  LXENG33.1  LXENG23.1  LXENG13.1

BEGIN EXECUTION OF UTILITY ROUTINE MCT

  VERSION ..  02/18U  08.10.56

  DATE .. U 09/14/77  TIME ..   12:31:36  USER NO ..  406362

  CONTROLLER .....    +PREPLOT

  CONTROLLEE .....    +MCT/ELF

  ACCESS LEVEL .... 2

  EXECUTE LINE FOLLOWS ....

ELF RDS .581225 : PSEAIR : TEST3.1 FILES LXENG13.1 LXENG23.1 LXENG33.1

 ERROR:   (LXENG13.1): (H) PHOTO STORE TRANSFER ERROR. RDS
 WILL RETRY IN 2 MIN.
 ERROR:   (LXENG13.1): (H) PHOTO STORE TRANSFER ERROR. RDS
 WILL RETRY IN 2 MIN.
 ERROR:   (LXENG13.1): (H) PHOTO STORE TRANSFER ERROR. RDS
 WILL RETRY IN 2 MIN.
 LXENG13.1, SIZE =      294912
 LXENG23.1, SIZE =      262144
 LXENG33.1, SIZE =      196608
 LXENG43.1, SIZE =      196608
```

```
  TOTAL COUNT =  1531

  ALL DONE.              12:43:42

END EXECUTION OF UTILITY ROUTINE MCT

SET DUMMY LIBRARY FILE NAMES

     FILE NAME  LXENG43.1 SWITCHED TO     DUMY1

     FILE NAME  LXENG33.1 SWITCHED TO     DUMY2

     FILE NAME  LXENG23.1 SWITCHED TO     DUMY3

     FILE NAME  LXENG13.1 SWITCHED TO     DUMY4

END EXECUTION OF PREPLOT:    12:43:46

MACHINE TIME USED (MIN):     0.025
```

```
          RECORD OF EXECUTION ------ PROGRAM FOURPLOT

BEGIN EXECUTION OF FOURPLOT: U 09/14/77  12:44:26

     DATA FILE LIBRARY:  LXENG13.1

     BASE NAME:      WL311

     DD80 FAMILY NAME:   DXWL311A

     EXPANDED PLOT TIMES (SEC):  TMIN = 2.905E+00
                                 TMAX = 3.465E+00


  PLOTTING . . . . . .

     FILE NAMES USED FOR DATA STORAGE

          WL311103  WL311104  WL311105  WL311106  WL311107  WL311109

          WL311110  WL311111  WL311113  WL311114  WL311115  WL311201

          WL311202  WL311203  WL311204  WL311205  WL311206  WL311207

          WL311208  WL311209  WL311210  WL311211  WL311212  WL311213

          WL311214  WL311301  WL311302  WL311303  WL311304  WL311305

          WL311307  WL311308  WL311309  WL311311  WL311312  WL311313

          WL311314  WL311315  WL311401  WL311402  WL311403  WL311404

          WL311405  WL311406  WL311407  WL311408  WL311409  WL311410

          WL311411  WL311412  WL311413  WL311414  WL311415  WL311501

          WL311502

     FILE NAMES NOT USED FOR DATA STORAGE

          WL311108  WL311112

END EXECUTION OF FOURPLOT:  12:46:15

MACHINE TIME USED (MIN):    0.745
```

```
          RECORD OF EXECUTION ----- PROGRAM FOURPLOT

BEGIN EXECUTION OF FOURPLOT: U 09/14/77  12:46:57

     DATA FILE LIBRARY:  LXENG23.1

     BASE NAME:      WL312

     DD80 FAMILY NAME:   DXWL312A

     EXPANDED PLOT TIMES (SEC):  TMIN = 2.905E+00
                                 TMAX = 3.465E+00


  PLOTTING . . . . . .

     FILE NAMES USED FOR DATA STORAGE

          WL312103  WL312104  WL312105  WL312106  WL312107  WL312109

          WL312110  WL312111  WL312113  WL312114  WL312115  WL312201

          WL312202  WL312204  WL312205  WL312206  WL312207  WL312208

          WL312209  WL312210  WL312211  WL312212  WL312213  WL312214

          WL312215  WL312301  WL312302  WL312303  WL312304  WL312305

          WL312306  WL312307  WL312308  WL312309  WL312310  WL312311

          WL312312  WL312314  WL312315  WL312401  WL312404  WL312405

          WL312406  WL312407  WL312408  WL312409  WL312410  WL312411

          WL312412  WL312413  WL312414  WL312415  WL312501  WL312502

     FILE NAMES NOT USED FOR DATA STORAGE

          WL312108

END EXECUTION OF FOURPLOT:  12:48:02

MACHINE TIME USED (MIN):   0.721
```

```
          RECORD OF EXECUTION ----- PROGRAM FOURPLOT

BEGIN EXECUTION OF FOURPLOT: U 09/14/77  12:48:50

     DATA FILE LIBRARY:  LXENG33.1

     BASE NAME:      WL313

     DD80 FAMILY NAME:   DXWL313A

     EXPANDED PLOT TIMES (SEC):  TMIN = 2.905E+00
                                 TMAX = 3.465E+00


  PLOTTING . . . . . .

     FILE NAMES USED FOR DATA STORAGE

           WL313103  WL313104  WL313105  WL313106  WL313107  WL313108

           WL313109  WL313110  WL313111  WL313112  WL313113  WL313114

           WL313115  WL313201  WL313202  WL313203  WL313204  WL313205

           WL313206  WL313207  WL313208  WL313209  WL313210  WL313211

           WL313303  WL313304  WL313305  WL313306  WL313307  WL313308

           WL313309  WL313310  WL313311  WL313312  WL313313

     FILE NAMES NOT USED FOR DATA STORAGE

              NONE

END EXECUTION OF FOURPLOT:  12:49:44

MACHINE TIME USED (MIN):    0.470
```

```
          RECORD OF EXECUTION ----- PROGRAM FOURPLOT

BEGIN EXECUTION OF FOURPLOT: U 09/14/77  12:50:18

     DATA FILE LIBRARY:  LXENG43.1

     BASE NAME:      WL314

     DD80 FAMILY NAME:   DXWL314A

     EXPANDED PLOT TIMES (SEC):  TMIN = 2.905E+00
                                 TMAX = 3.465E+00


  PLOTTING . . . . . .

     FILE NAMES USED FOR DATA STORAGE

          WL314102  WL314103  WL314104  WL314105  WL314106  WL314107

          WL314108  WL314109  WL314110  WL314111  WL314112  WL314113

          WL314114  WL314115  WL314201  WL314302  WL314304  WL314305

          WL314306  WL314308  WL314309  WL314311  WL314312  WL314402

          WL314403  WL314404  WL314405  WL314406  WL314407  WL314408

          WL314409  WL314410  WL314512  WL314513  WL314514

     FILE NAMES NOT USED FOR DATA STORAGE

          WL314303  WL314310

END EXECUTION OF FOURPLOT:  12:51:15

MACHINE TIME USED (MIN):   0.475
```

```
          RECORD OF EXECUTION ----- PROGRAM POSTPLOT

BEGIN EXECUTION OF POSTPLOT: U 09/14/77  12:51:19

     PSE TEST NUMBER:        3.1

     DD80 FILES:   DXWL311A  DXWL312A  DXWL313A  DXWL314A


BEGIN EXECUTION OF UTILITY ROUTINE LIX:  12:51:19

     DD80 LIBRARY NAME:  LXENG4PLT

   WRITE FILE   DXWL311A INTO LIBRARY

   WRITE FILE   DXWL312A INTO LIBRARY

   WRITE FILE   DXWL313A INTO LIBRARY

   WRITE FILE   DXWL314A INTO LIBRARY

   REPLACE FILE   DXWL314A IN LIBRARY

     ALL DONE

END EXECUTION OF UTILITY ROUTINE LIX

 NO LONG TERM STORAGE OF DD80 LIBRARY REQUESTED


BEGIN EXECUTION OF UTILITY ROUTINE FROG:  12:51:59

     DD80 FILE FAMILY NAMES:   DXWL311A  DXWL312A  DXWL313A  DXWL314A

     FROG OPTIONS SELECTED:       DD80.      35MM.      CINE.
```