NBS TECHNICAL NOTE 1108

U.S. DEPARTMENT OF COMMERCE/National Bureau of Standards

[illegible]

NATIONAL BUREAU OF STANDARDS

SPEED2, A Computer Program for the Reduction of Data from Automatic Data Acquisition Systems

Richard D. Peacock

Center for Fire Research
National Engineering Laboratory
National Bureau of Standards
Washington, DC 20234

and

John M. Smith

Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, DC 20234

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, Secretary

Luther H. Hodges, Jr., Under Secretary

Jordan J. Baruch, Assistant Secretary for Science and Technology

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Issued September 1979

National Bureau of Standards Technical Note 1108
Nat. Bur. Stand. (U.S.), Tech. Note 1108, 153 pages (Sept. 1979)
CODEN: NBTNAE

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
Stock No. 003–003–02112–1 Price $4.75
(Add 25 percent additional for other than U.S. mailing)

TABLE OF CONTENTS

LIST OF TABLES

SPEED2, A COMPUTER PROGRAM FOR THE REDUCTION OF DATA FROM AUTOMATIC DATA ACQUISITION SYSTEMS

Richard D. Peacock and John M. Smith

The voluminous amount of data that can be collected by automatic data acquisition systems requires the use of a digital computer for the reduction of data. A general purpose computer program for the reduction of data collected by automatic data acquisition systems is presented. The program is written with the ability to accept data from a number of different data acquisition systems, with the ability to check the correctness of data included. Through the use of FORTRAN computer programming, the data can be converted to meaningful scientific and engineering units. The data can then be presented in tabular, printer plot or ink pen plot form.

The program is documented, and detailed instructions for its use, with examples, are presented. The use of SPEED2 requires some knowledge of FORTRAN programming language and the executive control language for the computer system in use.

Key words: Computer program; data reduction; data acquisition systems; plotting.

1. INTRODUCTION

In 1968, the Building Research Division of the National Bureau of Standards (NBS) approached the Computer Services Division of NBS with a proposal concerning the design of a series of computer programs to facilitate the analysis of automatically recorded data. During the following two years, a system of computer programs called SPEED (Systematic Plotting and Evaluation of Enumerated Data) was developed and tested. This system was announced at the Ninth Annual Technical Symposium of the Association for Computing Machinery and in an article in Computer Graphics. The following paragraphs, quoted from the Computer Graphics article, which indicated the need for SPEED are still valid:

"During the past ten years the use of data acquisition systems or digital scanning systems has increased many fold. Such systems record automatically the readings of a number of instruments on either punched paper tape or magnetic tape. The individual characteristics of the various commercially available digital scanning systems vary greatly; however as a point of reference it may be noted that: one commercially available system is able to record the readings of up to one thousand instruments; another digital scanning system is able to read a clock and record the time plus the readings of up to twenty-five instruments in 2.3 seconds.

The use of digital scanning systems offers several advantages to the research scientist. First, their rapid recording capabilities allow for more complete data sampling. Second, automatically recorded data is more accurate than data that has been recorded manually.

These advantages are however, to some extent, counterbalanced by several problems which arise. Two problems are caused by the large volume of recorded data. First, it is difficult, if not impossible, to process large volumes of data by hand. Thus, the scientist finds it necessary to make use of the computer. Unfortunately, he is often unfamiliar with the capabilities and limitations of this device. Second, when presented with a large volume of data, it is often difficult for the scientist to rapidly interpret the broad characteristics of general trends that may be present. Two other problems arise in the form in which the data are recorded. The data are generally recorded in millivolts rather than standard units. Thus, some conversion process, usually a linear transformation, is required. Furthermore, the recorded data are not usually directly compatible with computers. In order for a computer to read this data some special computer program must be used to read this data in the recorded form and translate it into the internal computer representation."[1] [1]

[1] Numbers in brackets refer to the literature references listed at the end of this report.

During the nine years since its announcement, SPEED has been widely used at NBS and other computer installations. SPEED has operated successfully on UNIVAC, IBM, and CDC large-scale computers and has proved its usefulness. Unfortunately, during those nine years, many users found it convenient to modify the original system. This has led to the existence of different versions of SPEED, many of which do not include some important features, and few of which correspond directly to the original (and only) documentation.

For these reasons, SPEED has been rewritten with the new system titled SPEED2. The goals of this task have been:

a. To provide a standardized system of programs with appropriate current documentation;

b. To recover features lost through evolution of SPEED, in particular, the handling of input data;

c. To provide a more easily readable program including all old features and several new features; and

d. To take advantage of changes in computer operating systems.

The remainder of this paper provides a description of SPEED2 and instructions for its use.

Although this paper is meant as a detailed guide to the use of the SPEED2 system, some knowledge of FORTRAN programming and executive system control language for the computer in use is assumed.

2. SPEED2, PURPOSE AND STRUCTURE

2.1 Purpose of SPEED2

SPEED2 is designed to aid in the analysis of experimental data composed of a set of readings from a series of instruments. Thus, the basic data for SPEED2 are represented as a matrix

REED(i,j)

where the row index, i, refers to a given scan through the series of instruments, with i=1 corresponding to the first scan, etc. Typically, data acquisition systems scan at a fixed interval of time, so that the row index will also refer to a single time since the start of a test. The column index, j, refers to the readings from a single instrument. The order of the instruments in the columns of the array is controlled by the user through data cards read during the execution of the program.

SPEED2 provides the capability to read the input matrix from a variety of media including magnetic tape, paper tape, mass storage devices, or punched cards (of unconverted or converted data). This includes input checking and appropriate error messages. The ability to modify (correct or alter) individual entries, portions of rows or portions of columns, or entire rows or columns of the matrix is included in SPEED2. SPEED2 also provides for the transformation of analog instrument readings into standard units through linear or nonlinear transformations. This transformed and corrected matrix may be outputted to a printer, card punch, magnetic tape, or mass storage device. The user may generate printer plots or ink pen plots of the readings of any instrument versus a series of readings from other instruments. Finally, the ability to compare entries of several matrices, or to merge several matrices, each representing a different experiment, is included.

2.2 Structure of SPEED2

The system, SPEED2, is composed of a main program, ambiguously titled SPEED2; four major subprograms named PLOT1, PLOT2, PLOT3, and PLOT4; and a number of minor subprograms. Several of these subprograms are proprietary and/or limited in use to a UNIVAC computer. Consistent use of variable names among the various subprograms has been largely achieved.

The major subprogram, PLOT1, provides a means to read, check, and correct the input data recorded by a variety of data acquisition systems as shown in table 1. In addition, the user may specify a unique data acquisition system through card input.

The major subprogram, PLOT2, and its dependent subroutines have the role of reading, correcting, transforming, and printing of the input matrix. This segment of the system also generates printer plots.

The system segment controlled by PLOT3 may be used to generate pen plots on a CalComp plotter.

The segment controlled by PLOT4 is used to make comparisons between several input matrices of different experiments.

3. MAIN PROGRAM, SPEED2

The main program of the SPEED2 system is also titled SPEED2. Although a relatively short program, SPEED2 serves several vital functions.

First, SPEED2 provides dimensioning for many of the arrays used in the system that change in size depending on the size of the input matrix. For the NBS UNIVAC Computer, these dimensions are controlled by PARAMETER cards in the program SPEED2 as shown in appendix A:

```
PARAMETER NROW=30, NCOL=50
PARAMETER MAXPLT=6
PARAMETER MAXCNL=150
```

The entries on the first card, NROW and NCOL, refer to the number of rows (scans) and columns (instruments), respectively, in the input matrix. The second parameter card contains a single parameter, MAXPLT, which specifies the maximum number of curves to be plotted on any one graph in PLOT2 or PLOT3. The parameter MAXCNL defines the actual maximum number of instruments recorded by the data acquisition system in a single scan. It may be larger or smaller than the parameter NCOL. A proper choice of values is important to the efficient operation of the system.

Second, SPEED2 provides direction to the system. Through an input card, the user indicates his intention to use a particular major subprogram, PLOT1, PLOT2, PLOT3, or PLOT4. SPEED2 reads this input card and calls the proper subroutine. The user also has the options, as indicated through an input card, to process several data sets in sequence. SPEED2 provides this capability, and also erases all input from one data set before proceeding to the next.

4. PLOT1, PURPOSE AND STRUCTURE

PLOT1, the first of the major subprograms of SPEED2, provides for the initial input of data from a data acquisition system; checking of this input data; and optional output of the data to card punch, printer, or mass storage. With a minimum of data cards, the user can check the correctness of the data recorded by a data acquisition system.

PLOT1, like SPEED2, reads a data card which indicates the user's choice of a number of options. The parameters are interpreted, and the appropriate subroutines are called to perform the desired tasks.

The first card read by PLOT1 contains a series of parameters which indicate the particular input media and data acquisition system used to record the data, and the action to be taken with the input data matrix. Options include printing all, some, or none of the input records; punching the input data onto punched cards or writing to mass storage or magnetic tape; printing messages when errors are encountered in the input data; and skipping particular records of data.

In addition to this parameter card, PLOT1 requires two cards which indicate the title of the experiment to be printed at the top of each page of output. All other cards are optional, as detailed in section 11.

5. PLOT2, PURPOSE AND STRUCTURE

PLOT2 is the second of the major subprograms of SPEED2. In addition to all the features of PLOT1, PLOT2 provides for the conversion of analog instrument readings into standard units through linear or nonlinear transformations and for analysis of the input matrix.

PLOT2, like SPEED2 and PLOT1 reads data cards which indicate the user's choice of a number of options. These cards are interpreted, and proper subroutines are called to perform the desired tasks.

The first card read by PLOT2 contains a series of parameters which indicate the particular input media of the data matrix (magnetic tape, paper tape, unconverted data cards, converted data cards, etc.), the data acquisition system used to record the data, and the action to be taken with the input data matrix. Options include printing all, some, or none of the records; punching the input data on cards or writing to mass storage; stopping or proceeding after reading the input; printing some, all, or no input error messages; and skipping specified data records.

Next, a series of parameters is read which indicate actions to be taken with the converted data matrix. Options include printing, punching, and writing the converted matrix; generating graphs; making corrections to the matrix; and stopping after the conversion process is completed.

The remainder of the input is composed of the experiment title, cards defining the instruments (or columns), corrections (if any), and directions for the generation of graphs.

5.1 Using PLOT2

In order to use PLOT2 efficiently, the user should first examine the data collected by the data acquisition system to determine the number of rows and columns needed to store the input matrix. The user should also determine at this point whether or not new columns, which augment the input matrix, are to be generated through the use of subprograms CONV and FIND. If necessary, cards should be prepared to change the parameters which define the storage area for the input matrix and cards should be prepared to modify subprogram CONV. Instructions for changing subprogram CONV are given in section 5.2, and examples of both types of changes described above are given in appendix B. These cards should be preceded by the required executive system cards and followed by the appropriate executive system cards necessary to create and run an executable program.

The user then prepares the data cards. The format and content of each data card is given in section 11. It should be noted that the main program, SPEED2, requires two data cards prior to those cards required by PLOT2.

5.2 Data Transformation

As previously noted, instrumentation data readings are normally recorded in millivolts or volts. To convert these readings to standard units of temperature, length, etc., it is often sufficient to transform the reading through a linear equation. Some conversions, however, are considerably more complex and nonlinear.

5.2.1 Linear Transformations

PLOT2 achieves linear transformations as follows: The input data to PLOT2 includes a series of cards which describe each instrument. For each instrument, these cards contain the instrument's number, the instrument's type, and the instrument's name. The name is used only for identification purposes in printing results. The instrument's number is the channel number assigned by the data recording device. The instrument type is a user assigned number. Generally, instruments requiring the same linear transformation are assigned the same type. However, note that there may be only one instrument of type 1 (which usually records time). Following the instrument descriptors, PLOT2 requires a series of coefficient cards. These cards contain the transformation coefficients for each type used, 1 through n (assuming n different instrument types have been assigned).

All of this data is passed from PLOT2 to subprogram CONV. For each instrument, readings have been stored in the columns of the input matrix REED. Thus if instrument number j appears on the i'th descriptor card, its readings are

$$\text{REED}(\ell,i)\ ,\ \ell=1,\text{MAXR}(i)$$

where MAXR(i) is the number of readings of instrument number j. Furthermore, if the type of instrument number j is k, that is ITYPE(i)=k, then the appropriate transformation coefficients are C(k) and ADD(k). Hence, the transformation required is

$$\text{REED}(\ell,i) = \text{REED}(\ell,i) * \text{C}(k) + \text{ADD}(k),\ \ell=1,\text{MAXR}(i)$$

5.2.2 More Complex Transformations

Nonlinear or more complicated transformations require the user to make modifications to subprogram CONV. Such modifications are made by computer programming using ordinary FORTRAN statements. Clearly, the user will wish to modify a given column of the input matrix; however, the user may not know which column of the matrix contains the readings of the instrument in question. This may be answered through the use of subprogram FIND.

Subprogram FIND has four arguments and is invoked through a FORTRAN statement of the form

```
CALL FIND (J,I,KH,NCOL)
```

where KH is the array of instrument numbers in use, NCOL is the maximum dimension of that array, and J is an instrument number. The value of I, is computed by FIND and, is the column of the matrix in which the readings of instrument J have been stored. A convenient convention to follow is the form

```
CALL FIND (72,L72,KH,NCOL)
```

In this example, the readings of instrument number 72 will be found in column L72 (location of instrument 72).

Having located the proper columns of the matrix, nonlinear transformations are carried out through FORTRAN arithmetic statements. Thus the following simple FORTRAN DO loop

```
        CALL FIND (71,L71,KH,NCOL)
        CALL FIND (23,L23,KH,NCOL)
        MAXROW=MAXR(L71)
        DO 100 L=1,MAXROW
        REED(L,L71)=REED(L,L71)+REED(L,L23)
  100   CONTINUE
```

would transform the readings of instrument number 71 by adding to it the readings of instrument number 23. A series of nonlinear transformations might be programmed as shown below:

```
        CALL FIND (71,L71,KH,NCOL)
        CALL FIND (72,L72,KH,NCOL)
        CALL FIND (73,L73,KH,NCOL)
        CALL FIND (74,L74,KH,NCOL)
        CALL FIND (80,L80,KH,NCOL)
        CALL FIND (81,L81,KH,NCOL)
        CALL FIND (82,L82,KH,NCOL)
        MAXROW=MAXR(L82)
        DO 120 I=1,MAXROW
        REED(I,L71)=REED(I,L71)+REED(I,L72)+1.37
        REED(I,L73)=REED(I,L74)*1.231 + 5.6
        REED(I,L82)=SQRT(REED(I,L80)**2 + REED(I,L81)**2)
  120   CONTINUE
```

The first seven cards call subprogram FIND as described above to locate the instruments to be converted. The instruments are then used in simple calculations to serve as an example of the use of subprogram CONV. Obviously, much more complex conversions could be carried out.

The above cards should be inserted in subprogram CONV as noted in the examples shown in appendix B.

5.3 Making Corrections to the Data Matrix

The PLOT2 segment also provides the capability of modifying or correcting entries in the input matrix. Any number of corrections may be made; however, any single correction may apply only to one single entry, consecutive entries in a single column, or consecutive entries in a single row. Corrections are expected if and only if the PLOT2 parameter NCORR is set non-zero by the user.

Assuming NCORR is non-zero, PLOT2 will read a card containing the variables IRL (a low row index), IRH (a high row index), ICL (a low column index), and ICH (a high column index). The following restrictions apply:

a. Either IRL=IRH or ICL=ICH or both. Note, if IRL=IRH, all corrections apply to a single row. If ICL=ICH, all corrections apply to a single column. If IRL=IRH and ICL=ICH, a single entry will be corrected.

b. IRL is greater than zero but less than or equal to IRH or IRL is less than zero. IRL less than zero signifies the end of the corrections.

c. ICL is greater than zero but less than or equal to ICH, or both ICL and ICH are less than zero. Note that both ICL and ICH must be both positive or both negative. If they are negative, they are interpreted to be instrument numbers rather than column numbers. In this case, the column i in which instrument number ICL is stored is found. Similarly, the column j in which instrument number ICH is stored is found. Then the values of ICL and ICH are replaced by i and j respectively. The restriction becomes i is greater than zero but less than or equal to j.

Failure to satisfy any of the above restrictions will result in an error message being printed, and may result in all following data cards being out of order. Thus, particular care must be taken in the preparation of this card.

There are, in effect, only three valid combinations. They are:

a. IRL is equal to IRH and ICL is equal to ICH meaning correct entry REED(IRL,ICL).

b. IRL is less than IRH and ICL is equal to ICH meaning correct entries REED(IRL,ICL) REED(IRL+1,ICL), ..., REED(IRH-1,ICL), and REED(IRH,ICL).

c. IRL is equal to IRH and ICL is less than ICH meaning correct entries REED(IRL,ICL), REED(IRL,ICL+1), ..., REED(IRL,ICH-1), and REED(IRL,ICH).

In any of the above cases, the number of entries to be corrected is

(IRH-IRL) + (ICH-ICL) + 1

These entry corrections are read from a series of data cards punched in the format 8F10.0.

5.4 Generating Graphs

PLOT2 provides the capability to generate printer plots of the readings of either one or more instruments versus those of several other instruments. This is done only if the input parameter NPLOT is set non-zero by the user.

The program PLOT2 reads a card, containing up to 80 characters, which is printed as a heading on the graph. The program also reads a card or cards which contains the word PLOT (in any four consecutive columns, followed by a series of instrument numbers that specify the instruments to be plotted typed in free format separated by at least one space. There are two allowable formats for plot cards. If the card contains a comma separating instruments into pairs such as

```
PLOT    3999,3017    5999,5017    3999,10017
```

then the readings from the first instrument of each pair are taken to be x-values (abscissas) and the second instrument of each pair are taken to be y-values (ordinates). If no comma is found on any of the cards, such as

```
PLOT    999    017    019    021
```

then, the first mentioned instrument's readings are taken to be x-values (abscissas), and readings from the other entered instruments are taken to be y-values (ordinates). From one to n curves may be plotted simultaneously on a single graph, where n equals the SPEED2 parameter MAXPLT.

The user also controls the lower and upper limits of the two axes. A card is read containing the desired lower and upper limits of the x and y axes, XL, XH, YL, and YH, respectively. Care should be taken in choosing the limits to insure that readable values are printed on the axes. One important note is that points which fall outside the ranges determined for the two axes, are neither plotted nor noted.

Finally, the program reads a single card with two messages of 40 characters each. The first is printed vertically as a title for the y-axis, and the second is printed below the graph as a title for the x-axis.

After printing the graph, the program seeks another set of graph defining cards. A blank card signals the end of the plotting phase which is the final portion of PLOT2.

6. PLOT3, CALCOMP PEN PLOTTING OF DATA

The third of the major subprograms of SPEED2, PLOT3, allows the user to prepare high quality ink pen plots. PLOT3 is designed to accept data generated by PLOT2 and prepare a magnetic tape to be plotted off-line on a CalComp plotter. Many of the subroutines used by PLOT3 are proprietary, however, and may be used only by license from California Computer Products, Inc. Like the other major subprograms, PLOT3 reads data cards which indicate the user's choice of a number of options to control the generation of the plots.

The first two cards read in the PLOT3 phase contain parameters which specify certain parameters which control the generation of the graphs produced. The user can specify the type of curve generated, either a smooth, fitted curve, or a curve connecting points on the graph with straight lines; whether to print or not to print the data; the course of action to be taken if points are found that would be out of bounds on the graphs produced; whether to put axes on two or four sides of the plots; the spacing of plotting symbols on the curves produced; and whether or not to read cards to produce legends on the curves. In addition, the user can specify the size of the horizontal and vertical axes of the graphs, and a scaling factor to increase or decrease the overall size of the plots produced.

A number of cards are then read which specify the experiment title, define the instruments, and specify the plots to be produced, in a fashion similar to that used in PLOT2. Note that no conversion of data is made in PLOT3. Only reduced data (data processed by PLOT2) is read by PLOT3.

6.1 Using PLOT3

As with PLOT2, some forethought should be given to the data to allow the most efficient use of the SPEED2 system. The user should examine the data to determine the number of rows and columns needed to store the input matrix and the maximum number of curves on any single graph. If necessary, cards should be prepared to change the parameters which define the storage area for the input matrix exactly as was done with PLOT2. Of course, these cards should be preceded by and followed by the necessary executive system control cards to create an executable program.

The user then prepares the data cards. The format and content of each card is defined in section 11. As with PLOT2, SPEED2 requires data cards prior to those required by PLOT3.

6.2 Generating Pen Plots

PLOT3 is designed to prepare a magnetic tape to be plotted on a CalComp plotter. For each plot prepared, the readings of one or more instruments are plotted versus those of one or more other instruments.

A card is read which contains a figure number and two 39 character title lines which are placed below the plot as the title of the plot. Also read is a card containing two messages of 40 characters each. The first is printed as the title for the y-axis, and the second for the x-axis. All of the titles are centered along their respective axes.

PLOT3 has the capability to produce plots with linear, semi-log (with the log axis along the y-axis), or log-log axes. The program reads a card or cards containing either the word PLOT for a linear plot, SPLOT for a semi-log plot, or LPLOT for a log-log plot in any consecutive columns followed by a series of instrument numbers that specify the instruments to be plotted. As with the PLOT2 printer graphs, there are two allowable formats for the cards. If the card contains a comma separating instruments into pairs, such as

```
PLOT    3999,3017    5999,5017    3999,10017
SPLOT   3999,3017    5999,5017    3999,10017
LPLOT   3999,3017    5999,5017    3999,10017
```

the readings from the first instrument of each pair are taken to be x-values (abscissas), and the second instrument of each pair is taken to be y-values (ordinates). In the above statements, instrument 3999 is plotted versus 3017, 5999 versus 5017, and 3999 versus 10017. If no comma is found on any of the cards, such as

```
PLOT    999    017    019    021
```

then the first mentioned instrument's readings are taken to be x-values (abscissas) and the readings from the other entered instruments are taken to be y-values (ordinates). As with PLOT2, up to MAXPLT curves may be plotted on a single graph. If a card containing instrument numbers contains a semicolon in column 80, then another card is read and interpreted as containing additional instrument numbers for plotting on the same graph. As many cards as are needed may be used to specify instrument numbers.

The user can also control the lower and upper limits of the two axes. A card is read containing the lower and upper limits of each axis. Care should be taken in choosing the limits to insure that readable values are printed on the axes. However, PLOT3 checks these numbers to insure that readable numbers will be printed and may modify the limits to insure readability.

Finally, the program reads a single card containing two messages of 40 characters each. The first is printed as the title for the y-axis, and the second for the x-axis. As with the graph heading, the axis titles should be centered in their 40-character spaces to insure that they will be centered properly on the plot.

For example, the following sequence of cards

```
SPLOT    999,017    36,123    231,84
          93,998               41,86
          78,81
0.0       250.      0.001     100.
```

```
01RATE OF DECOMPOSITION
TIME    (HOURS)                          RATE    (1/SEC)
```

would prepare a single semi-log graph of instrument 999 versus 017, 36 versus 123, 231 versus 84, 93 versus 998, 41 versus 86, and 78 versus 81. The x-axis limits would be from 0.0 to 250. and the y-axis (a log scale) from 0.001 to 100. The title for the plot would be FIGURE 1: RATE OF DECOMPOSITION. The x and y axes would be labeled TIME (HOURS) and RATE (1/SEC), respectively.

PLOT3 also provides a means for the user to place a legend on the graph. The legend can be positioned anywhere on the graph and lines of text, up to 60 characters each, can be optionally prefixed with a symbol to identify legends for individual curves.

After generating the necessary information on the magnetic tape, the program seeks another set of graph defining cards. A blank card signals the end of the plotting, and the program prepares the tape for plotting.

7. PLOT4, COMBINING SEVERAL SETS OF DATA

PLOT4 is the last of the major subprograms of the SPEED2 system. PLOT4 allows the user to perform analysis and plotting on a number of instruments from several different tests, possibly with repeated instrument numbers in the tests. By renumbering the instruments to include a test number for each instrument, PLOT4 creates a new data set consisting of the readings of the renumbered instruments. Further analysis may then be performed on this new data set using PLOT2 or PLOT3 as desired.

As an example, consider an instrument numbered 121 from 11 different tests numbered arbitrarily test 3 through test 13. PLOT4 would perform a renumbering of the 11 instruments as 3121, 4121, 5121, 6121, 7121, 8121, 9121, 10121, 11121, 12121, and 13121, respectively. These instruments may then be punched on cards or written to mass storage and referenced in either PLOT2 or PLOT3 as reduced data using the new instrument numbers 3121 through 13121.

7.1 Using PLOT4

As with the previous major phases of SPEED2, the user should determine the number of rows and columns needed to store the readings of the instruments to be processed. If necessary, cards should be prepared to change the parameters which define the storage area for the input matrix. Of course, these cards should be preceded by the required executive system cards and followed by the appropriate executive system cards necessary to create an executable program.

The user then prepares the data cards. The format and content of each data card is described in detail in section 11 and generally outlined below. It must be noted that as with PLOT1, PLOT2, and PLOT3, the main program, SPEED2 requires two data cards prior to those required by PLOT4.

7.2 Renumbering of Instruments Using PLOT4

The first card read by PLOT4 contains two parameters that specify whether the data is to be printed and/or punched onto cards or mass storage. After reading title cards, cards defining the instruments to be selected from the various tests are included. Each instrument card contains a test number, provided by the user; an instrument number within the test; and a short description of the instrument. The remainder of the cards is comprised of the reduced data from each test preceded by a card of the form

TEST nnn

defining a test number for each test. Specifications for the data cards are given in section 11 along with examples of the use of PLOT4 in appendix B.

8. RESTRICTIONS ON THE USE OF SPEED2

The system, SPEED2, has been tested and continues to be tested by the authors. Users are requested to report any problems which arise through the use of SPEED2. Users should not modify any of the programs in this system with the exception of changing the parameters (see section 3) and inserting nonlinear transformation statements in subprogram CONV (see section 5). Examples of these changes are given in appendix B. Any other substantive changes are strongly discouraged without a detailed knowledge of the internal structure of the SPEED2 system.

While most of SPEED2 is written in a portable subset of FORTRAN IV and thus is compatible with a variety of computers, a number of exceptions were found when the subprograms were verified using the VERIFIER developed by Bell Laboratories [2]. These are described below:

a. Subprograms EREXIT, BYTE and NTRAN are written in UNIVAC assembler language. These would have to be replaced with the equivalent routines to be used on another machine.

b. A number of subprograms make use of Hollerith literals in arithmetic operations. To use SPEED2 on a computer other than a UNIVAC 1100 series computer, the computer must have the ability to arithmetically manipulate character strings.

c. The PARAMETER statement used in the main program, SPEED2, is unique to UNIVAC FORTRAN. Actual dimension limits for the variables NROW, NCOL, MAXPLT, MAXCNL, INDIM, OUTDIM, NPTS, and PLTDIM must be substituted in the main program to use SPEED2 on another computer. An alternate main program that has been successfully used on an INTERDATA 7/32 minicomputer is included in the program listing in appendix A.

d. Finally, PLOT3 makes use of a number of CalComp proprietary subprograms which must be obtained by a given computer installation from California Computer Products, Inc. The authors believe that a detailed description of the many secondary subroutines would be either too brief to be of value or too complex to be comprehended. Thus, such statements have been omitted unless absolutely necessary.

9. NOTES ON THE USE OF SPEED2

9.1 Choice of NROW and NCOL

Choosing NROW and NCOL should be done with care. If values are chosen which are too small, execution will be terminated when the input matrix size exceeds the allocated size. Conversely, computer use charges are based, in part, on the allocated storage. Too large values of NROW and NCOL will cause excess costs. In addition to the input matrix, REED, being dimensioned NROW by NCOL, the arrays KH, ITYPE, C, X. Y, and NAME all depend on NCOL for their size. Thus, the storage allocation for all of these arrays is approximately

(NROW+15)*(NCOL+13).

On the NBS UNIVAC 1108, this product should not exceed 45000.

9.2 Instrument Definition Cards

As noted in section 5, each instrument is defined by a card containing the instrument number, type, and name. The use of these data in the conversion process has already been explained, but other points should be mentioned.

First, subprogram PRINT prints only those instruments defined through an instrument card. Thus, each instrument whose readings are to be printed must be defined, whether the instrument is real or whether it represents some combination of other instruments formed through the use of subprogram CONV. Conversely, if the user wishes to divide his experiment into several smaller sub-experiments, he may omit the processing of data for any instruments by omitting their defining cards. One exception to this rule is the instrument card which defines time. If its defining card is left out, the program will internally reserve space for it in the array REED. However, if its card is not included, the time readings will not be available for plotting or conversion.

Second, subprogram PRINT will print the converted instrument readings, in columns, in the order in which the defining cards appear. Thus, the user may predetermine this printout order in whatever way is most convenient and may easily change the order if the computer run is repeated. As previously mentioned, each column will be headed by the first six characters of the instrument's name.

Finally, if the user indicates a desire to punch the converted matrix on cards or mass storage through setting NPNCH non-zero, the above comments apply to these cards too.

9.3 Unconverted Input

Data acquisition systems record readings from instruments in a variety of different formats. Table 1 presents the formats used in a number of different data systems. Several characteristics common to the different systems are detailed below.

a. A time reading is usually prefixed to the series of instrument readings. Time readings and instrument readings are of different formats.

b. Time readings consist of digits to indicate days, hours, minutes, and/or seconds, perhaps separated by spaces or other special characters.

c. Each instrument reading consists of characters to indicate a channel number assigned by the data acquisition system, the sign of the reading, the value of the reading, and a channel overflow indicator.

d. Within a single scan, the time reading and each instrument reading is separated by spaces and/or other special characters.

e. Scans are separated by a special character sequence on card images or by a record separator on magnetic tape.

f. The end of the test data recording is indicated by a special character sequence or by an end of file mark on magnetic tape.

Table 2 presents sample scans from a number of different data systems. Tables 1 and 2 also provide guidelines to follow if cards are to be prepared by the user.

9.4 Converted Input

Converted instrument readings also require that data be prepared following a particular format. The readings from each instrument are prepared in a format similar to table 2. Each set of readings from an instrument begins with a data card specifying the number of readings of the instrument and the instrument number in a format 2I6, followed by a series of cards containing the readings. The readings, in the format E10.5, are packed eight readings per card. Finally, the end of test is indicated by a card with 999 in columns 78, 79, and 80.

9.5 Creating New Instruments

Complex transformations made through the use of subprogram CONV may include new instruments representing combinations of other instruments.

It should be noted that the user must specify the number of readings in the created instrument by setting MAXR for the created instrument.

10. DIAGNOSTIC AND ERROR MESSAGES

The SPEED2 system is designed to intercept and interpret most user errors and to print a diagnostic or error message indicating the error. While most messages are self-explanatory, they are listed below, along with explanations of the errors and how to correct them.

10.1 Unconverted Data Checking

When the unconverted data are input to the program in any of the available forms, the data are checked for errors in the format of the data or for illegal characters, etc. If errors are found, a message is printed. Five different messages are possible. The first indicates an error in the unconverted data. This can either be an illegal character in the data, or data in which characters are out of sequence, bad, or missing. The message takes the form

CHARACTER SEQUENCE ERROR RECORD: i CHARACTER: j

The record index, i, is the record (scan) number of the particular record in error and the character index, j, is the number of the actual character in error. The normal action taken by SPEED2 is to set the particular reading in error to zero and continue processing. If the user has prepared punched data cards or card images on mass storage of the unconverted data, and the user wishes to correct the error, the data can be edited, corrected, and the program re-run with the corrected set of data. The user could also simply replace the reading in error by using subprogram CORR to correct the reading.

The second type of error message indicates an overflow in a data reading from the data acquisition system. It takes the form

OVERFLOW WARNING RECORD: i CHARACTER: j CHANNEL: k

All comments made for the previous error message also apply for this diagnostic warning. The channel number as recorded by the data acquisition system, k, is also given.

The third type of message indicates simply that the record length of a single scan through the instruments has changed from the last scan. It is simply a warning to the user that something may be wrong with the data. It takes the form:

RECORD LENGTH WARNING RECORD: i OLD LENGTH: j NEW LENGTH: k

The record number, i, the length of the last scan, j, and the length of the current scan, k, is given.

The last two error messages indicate a duplicate time or channel reading in a single scan. The messages take the form:

MULTIPLE TIME ERROR RECORD: i CHARACTER: j

MULTIPLE READING WARNING RECORD: i CHARACTER: j CHANNEL: k

The record number, i, and the character number, j, are given for both messages. The channel number, k, is also given for the channel warning.

10.2 Data Cards and Format Errors

Many user errors result from improper placement of or from incorrect formats on data cards entered by the user. While it is urged that the user carefully check the data cards before running SPEED2, such errors are bound to occur. Each data card read by SPEED2 has a unique identification (SPEED2 Data Card A1, PLOT3 Data Card D3 for instance). If format errors are found on data cards, program execution is halted and an appropriate error message is printed. Two messages are possible for read errors, but both take the same form:

FORMAT ERROR ON iseg DATA CARD icrd

or

END OF FILE ENCOUNTERED ON iseg DATA CARD icrd
PRECEDING DATA CARDS MAY BE IN ERROR

The two variables, iseg and icrd, are the program segment (SPEED2, PLOT1, PLOT2, PLOT3, or PLOT4) and the card identification (A1, D3, etc.).

10.3 Other Messages and Diagnostics

Many other error messages are printed by the SPEED2 system to alert the user of possible errors encountered during the processing of data by SPEED2. A short explanation is provided below for each along with an explanation of how it can be corrected.

ONLY ONE INSTRUMENT OF TYPE 1 IS ALLOWED
SECOND FOUND AT INSTRUMENT NUMBER i

Instrument Type 1 is reserved to indicate the time channel. Obviously, only one channel is allowed for this. If more than one is found, program execution is stopped. The user must correct the instrument cards so there is only one instrument of Type 1, the time channel.

COLUMN STORAGE LIMITS EXCEEDED AT COLUMN i
CHECK PARAMETERS NCOL AND NROW

The user has not allowed enough room for the instruments he wishes to process as defined by the instrument defining cards. The PARAMETER NCOL should be increased to the proper size.

ILLEGAL INPUT FILE TYPE SPECIFIED FOR INTYPE

The user has specified an illegal number for the input file parameter INTYPE, or there is a format error on the data card on which INTYPE is read.

ROW STORAGE LIMITS EXCEEDED AT ROW i
CHECK PARAMETERS NCOL AND NROW

The user has not allowed enough room for the number of scans in the experiment. The PARAMETER NROW should be increased to the proper size.

READING OF INPUT FILE TERMINATED BY NTRAN ERROR, STATUS: i

An unrecoverable error has occurred while reading the magnetic tape with NTRAN. All data read in and processed up to the point of error is processed however. There are ways to read magnetic tapes with such errors; however, none appear to be very reliable. (UNIVAC 1100 only)

INPUT RECORD SIZE OF i EXCEEDS BUFFER SIZE
CHECK PARAMETER NAXCNL

The standard version of SPEED2 is configured for a maximum of 150 channels of data per record (scan). More than the maximum number of channels, MAXCNL, have been processed in one record. To correct this, check the input data and/or increase the PARAMETER MAXCNL.

READING TERMINATED BY READ ERROR OR END OF FILE

Reading of the unconverted input data has been terminated by a read error or by an attempt to read past an end of file. Usually, there is a format error on the input punched cards. The user should check the data to insure it is in the proper format.

NUMBER OF READINGS FOR INSTRUMENT NUMBER i EXCEEDS STORAGE LIMITS.
CHECK PARAMETERS NROW AND NCOL

There are too many readings of reduced data for instrument number i, or there is a format error on the card defining the instrument in the reduced data set. The user should check the data for correctness and/or increase PARAMETER NROW.

ERROR ON FOLLOWING CORRECTION CARD

The user has specified NCORR non-zero and subprogram CORR has found an error on the correction cards. The user should check the format of the data cards. The card in error is printed.

ERROR IN SUBPROGRAM FIND
NO REFERENCE TO INSTRUMENT NUMBER i CAN BE FOUND

Instrument number i was not defined on an instrument card, and the instrument was referenced during transformations using subprogram CONV. The results of the calculations are unpredictable, and the user should check the FORTRAN modifications to subprogram CONV carefully.

11. DATA CARDS FOR THE EXECUTION OF SPEED2

The data cards required for the execution of SPEED2 fall naturally into five major divisions: those cards required by the main program, SPEED2, and cards directing the execution of each of the major phases of SPEED2 -- PLOT1, PLOT2, PLOT3, and PLOT4. In addition, the data cards for each major phase fall into distinct parts. These are described below.

11.1 SPEED2 Data Cards

The main program segment, SPEED2, requires two data cards to control the execution of the other program segments. The content and format of these cards is described in table 3. One data card that specifies the number of sets of data is followed by sets of cards defining the major phase of SPEED2 to be used to process each data set and the cards required by the major phase specified on that card. NSETS sets of these cards are included in the total set of data cards.

11.2 PLOT1 Data Cards

If the user specified PLOT1 on SPEED2 data card A2, then cards are read as defined in table 4 to control the execution of PLOT1. The six distinct parts defined for PLOT1 are:

- A: PLOT1 Control Parameters - required
- B: Experimental Title - required
- C: Data Acquisition System Definition - optional (only if INTYPE=0)
- D: Skipping Records of Data - optional (only if INSKIP is non-zero)
- E: Unconverted Data Cards - optional (only if INTYPE specifies cards)
- F: Converted Data Cards - optional (only if INTYPE=4)

Data cards Part C through F are optional and are only required depending upon the values of certain control parameters defined in Part A.

11.3 PLOT2 Data Cards

The data cards required for the execution of PLOT2 fall naturally into nine distinct parts. Not all parts are required in each run. Whether or not a certain part is required depends on the values of certain input parameters described below and detailed in table 5.

A:	PLOT2 Control Parameters	- required
B:	Experiment Title and Instrument Description	- required
C:	Data Acquisition System Definition (only if INTYPE=0)	- optional
D:	Skipping Records of Data (only if INSKIP is non-zero)	- optional
E:	Unconverted Data Cards (only if INTYPE specifies cards)	- optional
F:	Converted Data Cards (only if INTYPE=4)	- optional
G:	Reading Corrections to Data (only if NCORR is non-zero)	- optional
H:	Cards Read By the User in CONV	- optional
I:	Generating Printer Plots (only if NPLOT is non-zero)	- optional

11.4 PLOT3 Data Cards

The data cards for the execution of PLOT3, described in table 6, fall into four distinct parts. All parts are required for execution.

A:	PLOT3 Control Parameters	- required
B:	Experiment Title and Instrument Definition	- required
C:	Converted Data Cards	- required
D:	Generating Pen Plots	- required

11.5 PLOT4 Data Cards

The data cards required for the execution of PLOT4 are described in table 7. The three major parts are:

A:	PLOT4 Control Parameters	- required
B:	Experiment Title and Instrument Definition	- required
C:	Converted Data Cards for Each Test	- required

12. REFERENCES

[1] Smith, John M., Automatic data evaluation, manipulation, display, and plotting with SPEED, Computer Graphics, Vol. 4, No. 2, 41-53 (Fall 1970).

[2] Ryder, B. G., The FORTRAN verifier: user's guide, Bell Telephone Laboratories, Incorporated, Murray Hill, New Jersey.

Table 1. Recording Format of Several Data Acquisition Systems

Data System	Reading Format	Time Format	End of Record	End of File	Magnetic Tape Format	Card Image Format
VIDAR 5400 Series						
Magnetic Tape	CCCSVVVVVEBB	SSS6SSSSSSBB	EOR	EOF	556 BPI, Even	72A1
Punched Cards	CCCSVVVVVEBB	SSS6SSSSSSBB	BX	FILEND	Parity BCD	
VIDAR Autodata 8 Series						
Punched Cards	NBCCCBSRRRRRRAAB	HH:MM	80*B	FILEND		80A1
VIDAR Autodata 9 Series						
Magnetic Tape	CCCBSVVVVVVAAAEX	DD:HH:MM:SSAAAAX	EOR	EOF		
Punched Cards	CCCBSVVVVVVAAAEX	DD:HH:MM:SSAAAAX	BX	FILEND		80A1
Esterline Angus PD2064						
Magnetic Tape	ACCCBOSRRRRRRBAABB	AAAAAADDBHH:MM:SSBBBBB	EOR	EOF		
Punched Cards	ACCCBOSRRRRRRBAABB	AAAAAADDBHH:MM:SSBBBBB	BX	FILEND		72A1
DORIC						
Punched Cards	OCCCSRRRRRRABB	HHBMMBSS	75*B	FILEND		75A1

EOR - Magnetic Tape End of Record
EOF - Magnetic Tape End of File
C - A Channel Number Digit
S - Sign of Reading
X - The Character "X"

V - Magnitude of Reading
E - Exponent of Reading
B - A Blank
N - A Numeric (0-9)
R - Value of Reading

A - Any Character
O - An Overflow Indicator
D,H,M,S - Days, Hours, Minutes, and Seconds of Time Reading
80* - The Character is Repeated 80 Times

Table 2. Data Acquisition System Formats
VIDAR 5400 SERIES

SPEED2 Data Acquisition System Format Specification

```
INPUT=TAPE
READING=3*(C)(+1++-2--0900)5*(V)(E)2*(K )
TIME=3*(S)(K6)6*(S)2*(K )
EOR=EOR
EOF=EOF
```

```
0006000000  0001000166  0012000016  0021000036  0031000016  0041000016
0051000026  0061000036  0072000016  0081000016  0092000016  0101000046
0112000016  0121000026  0132000016  0141000036  0151000036  0162000016
0172000016  0181000026  0191000066  0201000036  0212000006  0221000026
```

(MAGNETIC TAPE RECORD GAP)

```
0006000003  0002000006  0011000016  0021000036  0031000026  0042000016
0051000026  0061000016  0072000016  0082000006  0092000006  0101000046
0112000016  0121000036  0132000016  0141000026  0151000046  0162000016
0172000016  0181000036  0192000006  0201000046  0212000016  0222000016
```

(MAGNETIC TAPE RECORD GAP)

(MAGNETIC TAPE END OF FILE)

SPEED2 Data Acquisition System Format Specification

```
INPUT=CARD IMAGES
READING=3*(C)(+1++-2--09DD)5*(V)(E)2*(K )
TIME=3*(S)(K6)6*(S)2*(K )
EOR=(K )(KX)
EOF=(KF)(KI)(KL)(KE)(KN)(KD)
```

```
0006000000  0001000166  0012000016  0021000036  0031000016  0041000016
0051000026  0061000036  0072000016  0081000016  0092000016  0101000046
0112000016  0121000026  0132000016  0141000036  0151000036  0162000016
0172000016  0181000026  0191000066  0201000036  0212000006   X
0006000003  0002000006  0011000016  0021000036  0031000026  0042000016
0051000026  0061000016  0072000016  0082000006  0092000006  0101000046
0112000016  0121000036  0132000016  0141000026  0151000046  0162000016
0172000016  0181000036  0192000006  0201000046  0212000016   X
FILEND
```

SPEED2 Data Acquisition System Format Specification

357 3150TC 1

.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.28000+02	.27000+02	.28000+02	.27000+02	.27000+02	.27000+02	.27000+02	.27000+02
.28000+02	.28000+02	.27000+02	.28000+02	.28000+02	.27000+02	.27000+02	.27000+02
.28000+02	.28000+02	.28000+02	.28000+02	.27000+02	.28000+02	.27000+02	.27000+02
.28000+02	.27000+02	.27000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02
.28000+02	.28000+02	.28000+02	.28000+02	.27000+02	.28000+02	.28000+02	.28000+02
.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02
.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02
.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02
.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02	.28000+02
.28000+02	.28000+02	.28000+02	.28000+02	.29000+02	.28000+02	.28000+02	.28000+02
.28000+02	.28000+02	.28000+02	.29000+02	.28000+02	.29000+02	.29000+02	.29000+02
.29000+02	.29000+02	.29000+02	.29000+02	.29000+02	.29000+02	.29000+02	.29000+02
.29000+02	.29000+02	.29000+02	.29000+02	.29000+02	.29000+02	.29000+02	.29000+02
.29000+02	.29000+02	.29000+02	.29000+02	.29000+02	.29000+02	.29000+02	.29000+02
.29000+02	.29000+02	.29000+02	.29000+02	.30000+02	.29000+02	.30000+02	.29000+02
.30000+02	.29000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02
.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02
.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02
.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02
.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02	.30000+02
.31000+02	.30000+02	.30000+02	.31000+02	.31000+02	.31000+02	.31000+02	.31000+02
.31000+02	.31000+02	.31000+02	.31000+02	.31000+02	.31000+02	.31000+02	.31000+02
.31000+02	.31000+02	.31000+02	.31000+02	.31000+02	.31000+02	.31000+02	.31000+02

SPEED2 Data Acquisition System Format Specification

```
INPUT=CARD IMAGES
READING=(N)(K )3*(C)(K )(RO )5*(R)2*(A)(K )
TIME=2*(H)(K:)2*(M)75*(K )
EOR=80*(K )
EOF=(KF)(KI)(KL)(KE)(KN)(KD)


05:24
5 105  0027.6 C 5 106  0030.0 C 5 107  0028.3 C 5 108  0030.2 C
5 109  0441.0 C 1 110  010.52MV 1 111  000.87MV 2 112  0.0984 V

05:25
5 105  0027.3 C 5 106  0030.2 C 5 107  0028.3 C 5 108  0030.5 C
5 109  0443.8 C 1 110  011.18MV 1 111  001.06MV 2 112  0.0994 V

FILEND
```

SPEED2 Data Acquisition System Format Specification

```
INPUT=TAPE
READING=(KC)3*(C)(K )(K 0*N)(+ --)6*(RC )(K )2*(A)2*(K )
TIME=4*(K-N)2*(K )2*(D)(K )2*(H)(K:)2*(M)(K:)2*(S)(K )(KH)(KR)2*(K )
EOR=EOR
EOF=EOF
```

```
0002  01 14:25:00 HR  C000   0073.2 OF  C001   00.012 MV  C002 0 0075.1
OF  C003 0 0078.0 OF  C004 0 0079.3 OF  C005 0 0080.3 OF  C006 0 0080.3
OF  C007 0 0079.4 OF  C008 0 0078.1 OF  C009 0 0075.1 OF  C010 0 0073.6
OF  C011 0 0074.0 OF  C012 0 0072.1 OF  C013 0 0073.1 OF  C014 0 0074.3
OF  C015 0 0074.7 OF  C016 0 0074.4 OF  C017 0 0074.8 OF  C018 0 0074.4
OF  C019 0 0072.6 OF

(MAGNETIC TAPE RECORD GAP)

0002  01 14:26:00 HR  C000   0073.2 OF  C001   00.008 MV  C002 0 0075.0
OF  C003 0 0077.8 OF  C004 0 0079.0 OF  C005 0 0079.7 OF  C006 0 0079.9
OF  C007 0 0079.1 OF  C008 0 0078.5 OF  C009 0 0075.3 OF  C010 0 0073.8
OF  C011 0 0074.4 OF  C012 0 0072.1 OF  C013 0 0072.6 OF  C014 0 0074.8
OF  C015 0 0074.2 OF  C016 0 0073.5 OF  C017 0 0072.1 OF  C018 0 0074.4
OF  C019 0 0072.2 OF

(MAGNETIC TAPE RECORD GAP)

(MAGNETIC TAPE END OF FILE)
```

ESTERLINE ANGUS PD2064

SPEED2 Data Acquisition System Format Specification

```
INPUT=CARD IMAGES
READING=(KC)3*(C)(K )(K D*N)(+ --)6*(RO )(K )2*(A)2*(K )
TIME=4*(K-N)2*(K )2*(D)(K )2*(H)(K:)2*(M)(K:)2*(S)(K )(KH)(KR)2*(K )
EOR=(K )(KX)
EOF=(KF)(KI)(KL)(KE)(KN)(KD)
```

```
0002  01 14:25:00 HR  C000   0073.2 OF  C001   00.012 MV  C002 0 0075.1
OF  C003 0 0078.0 OF  C004 0 0079.3 OF  C005 0 0080.3 OF  C006 0 0080.3
OF  C007 0 0079.4 OF  C008 0 0078.1 OF  C009 0 0075.1 OF  C010 0 0073.6
OF  C011 0 0074.0 OF  C012 0 0072.1 OF  C013 0 0073.1 OF  C014 0 0074.3
OF  C015 0 0074.7 OF  C016 0 0074.4 OF  C017 0 0074.8 OF  C018 0 0074.4
OF  C019 0 0072.6 OF   X
0002  01 14:26:00 HR  C000   0073.2 OF  C001   00.008 MV  C002 0 0075.0
OF  C003 0 0077.8 OF  C004 0 0079.0 OF  C005 0 0079.7 OF  C006 0 0079.9
OF  C007 0 0079.1 OF  C008 0 0078.5 OF  C009 0 0075.3 OF  C010 0 0073.8
OF  C011 0 0074.4 OF  C012 0 0072.1 OF  C013 0 0072.6 OF  C014 0 0074.8
OF  C015 0 0074.2 OF  C016 0 0073.5 OF  C017 0 0072.1 OF  C018 0 0074.4
OF  C019 0 0072.2 OF   X
FILEND
```

SPEED2 Data Acquisition System Format Specification

```
INPUT=TAPE
READING=3*(C)(K )(++--0 )5*(VO )(VK )3*(A)(EO )(KX)
TIME=3*(D)2*(H)(K:)2*(M)(K:)2*(S)4*(K )(KX)
EOR=EOR
EOF=EOF
```

```
22009:30:04    X151 +00029   COX152 +00027   COX153 +00026   COX154 +00028   COX
155 +00028   COX156 +00023   COX157 +00022   COX158 +00022   COX159 +00022   COX
160 +00026   COX161 +00024   COX162 +00024   COX163 +00024   COX164 +00023   COX
165 +00024   COX166 +00023   COX167 +00023   COX168 +00023   COX169 +00024   COX
170 +00023   COX171 +00022   COX172 +00023   COX173 +00026   COX180 +09874  MVOX
181 +10008  MVOX182 +10089  MVOX183 +10041  MVOX184 +10073  MVOX185 +10321  MVOX
186 +00002  MVOX187 +00001  MVOX188 +00001  MVOX189 +00000  MVOX190 +00001  MVOX
191 +00070  MVOX192 +00123  MVOX193 +00022  MVOX194 -00004  MVOX195 -00007  MVOX
196 +00057  MVOX

(MAGNETIC TAPE RECORD GAP)

22009:30:19    X151 +00034   COX152 +00031   COX153 +00033   COX154 +00047   COX
155 +00060   COX156 +00022   COX157 +00022   COX158 +00022   COX159 +00022   COX
160 +00025   COX161 +00024   COX162 +00024   COX163 +00024   COX164 +00023   COX
165 +00024   COX166 +00023   COX167 +00023   COX168 +00023   COX169 +00024   COX
170 +00023   COX171 +00022   COX172 +00022   COX173 +00026   COX180 +09857  MVOX
181 +10001  MVOX182 +10078  MVOX183 +10035  MVOX184 +10068  MVOX185 +10320  MVOX
186 +00002  MVOX187 +00000  MVOX188 +00000  MVOX189 +00001  MVOX190 +00000  MVOX
191 +00082  MVOX192 +00130  MVOX193 +00012  MVOX194 +00039  MVOX195 -00005  MVOX
196 +00036  MVOX

(MAGNETIC TAPE RECORD GAP)

(MAGNETIC TAPE END OF FILE)
```

SPEED2 Data Acquisition System Format Specification

```
INPUT=CARD IMAGES
READING=3*(C)(K )(++--D )5*(VD )(VK )3*(A)(ED )(KX)
TIME=3*(D)2*(H)(K:)2*(M)(K:)2*(S)4*(K )(KX)
EOR=(K )(KX)
EOF=(KF)(KI)(KL)(KE)(KN)(KD)



22009:30:04    X151 +00029   COX152 +00027   COX153 +00026   COX154 +00028   COX
155 +00028   COX156 +00023   COX157 +00022   COX158 +00022   COX159 +00022   COX
160 +00026   COX161 +00024   COX162 +00024   COX163 +00024   COX164 +00023   COX
165 +00024   COX166 +00023   COX167 +00023   COX168 +00023   COX169 +00024   COX
170 +00023   COX171 +00022   COX172 +00023   COX173 +00026   COX180 +09874  MVOX
181 +10008  MVOX182 +10089  MVOX183 +10041  MVOX184 +10073  MVOX185 +10321  MVOX
186 +00002  MVOX187 +00001  MVOX188 +00001  MVOX189 +00000  MVOX190 +00001  MVOX
191 +00070  MVOX192 +00123  MVOX193 +00022  MVOX194 -00004  MVOX195 -00007  MVOX
196 +00057  MVOX X
22009:30:19    X151 +00034   COX152 +00031   COX153 +00033   COX154 +00047   COX
155 +00060   COX156 +00022   COX157 +00022   COX158 +00022   COX159 +00022   COX
160 +00025   COX161 +00024   COX162 +00024   COX163 +00024   COX164 +00023   COX
165 +00024   COX166 +00023   COX167 +00023   COX168 +00023   COX169 +00024   COX
170 +00023   COX171 +00022   COX172 +00022   COX173 +00026   COX180 +09857  MVOX
181 +10001  MVOX182 +10078  MVOX183 +10035  MVOX184 +10068  MVOX185 +10320  MVOX
186 +00002  MVOX187 +00000  MVOX188 +00000  MVOX189 +00001  MVOX190 +00000  MVOX
191 +00082  MVOX192 +00130  MVOX193 +00012  MVOX194 +00039  MVOX195 -00005  MVOX
196 +00036  MVOX X
FILEND
```

SPEED2 Data Acquisition System Format Specification

```
INPUT=CARD IMAGES
READING=(0*K )3*(C)(+ --)6*(R)(A)2*(K )
TIME=(K )15*(K N)2*(H)(K )2*(M)(K )2*(S)(K )(KH)44*(K )
EOR=70*(K )
EOF=(KF)(KI)(KL)(KE)(KN)(KD)
```

```
 60072777        00 00 30 H
 001 0063.2C    002 02.603M    003 0063.9C    004 02.598M    005 0061.2C
 006 0064.6C    007 0023.4C    008 036.71M

 60072777        00 30 30 H
 001 0063.2C    002 02.589M    003 0064.0C    004 02.596M    005 0063.7C
 006 0064.3C    007 0023.5C    008 041.46M

FILEND
```

Table 3. SPEED2 Data Cards

Phase	Part	Card	Variables	Format	Comments
SPEED2	A	A1	NSETS	I5	NSETS is the number of data sets to be processed. NSETS must be greater than or equal to 1.
		A2	IP1,IP2,IP3,IP4	4I5	These variables specify which of the major phases of SPEED2 are to be used for processing of a single data set: If IP1 is non-zero,execute PLOT1. If IP1 is zero, do not execute PLOT1. If IP2 is non-zero,execute PLOT2. If IP2 is zero, do not execute PLOT2. If IP3 is non-zero,execute PLOT3. If IP3 is zero, do not execute PLOT3. If IP4 is non-zero,execute PLOT4. If IP4 is zero, do not execute PLOT4.

Table 4. PLOT1 Data Cards

Phase	Part	Card	Variables	Format	Comments
PLOT1	A	A1	INTYPE,INPRT,INPNCH, INERR,INSKIP	5I5	This card contains parameters which control the input. The various possibilities and their meanings are: INTYPE: 0 -- read PLOT1 data cards Part C to specify a special data acquisition system; 1 -- specifies VIDAR 5400 series magnetic tape data acquisition system; 2 -- not used; 3 -- specifies VIDAR 5400 series format punched card images; 4 -- specifies reduced data format punched card images; 5 -- specifies VIDAR Autodata 8 series format punched card images; 6 -- specifies Esterline Angus PD2064 series magnetic tape data acquisition system; 7 -- specifies Esterline Angus PD2064 series format punched card images; 8 -- specifies VIDAR Autodata 9 series magnetic tape data acquisition system; 9 -- specifies VIDAR Autodata 9 series format punched card images; 10 -- specifies DORIC data acquisition system format punched card images. INPRT: Directs printing of data as recorded by a data acquisition system. If INPRT is equal to zero, no data is printed. If INPRT is greater than zero, INPRT specifies the maximum number of data records to be printed. If INPRT is equal to -1, all data records are printed. If INPRT is equal to -2, only data records that contain errors are printed.

Table 4. PLOT1 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
					INPNCH: Directs the output of the data as recorded by a data acquisition system to card-punch, mass storage, or magnetic tape. If INPNCH is greater than zero, all data records are written to the unit specified by INPNCH. The user may assign this unit to any storage media. (For the NBS UNIVAC 1108, unit 1 is the default card punch and units 9-44 are available to be assigned by the user as mass storage or magnetic tape files.) INERR: Specifies the maximum number of error messages to be printed during processing of data recorded by a data acquisition system. INSKIP: If INSKIP is non-zero, read PLOT1 data cards Part D to specify records of input data to be skipped.
PLOT1	B	B1,B2	TITLE(I),I=1,120	80A1/40A1	These two cards specify the title of the experiment, printed at the top of all pages of output.
PLOT1	C				Part C data cards specify a special data acquisition system. They are read if and only if INTYPE is equal to zero.

Table 4. PLOT1 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
		C1	IN(I),I=1,80	80A1	This card specifies the type of the input media. Two formats are possible: INPUT=CARD IMAGES, CHANNELS PER LINE=N_1 INPUT=TAPE, TAPE FORMAT=C_1 The first format specifies that the input is in card images. Punched cards, data added from mass storage or tape files, and data transferred from remote terminals are considered card images. N_1 is an integer that specifies the number of channels recorded on a single card image. The second format specifies magnetic tape recorded by the data acquisition system. C_1 is a character string that defines the tape format. Currently, only TAPE FORMAT=7TRACK is defined.
		C2,C3	IN(I),I=1,80	80A1	This card specifies the character by character format of an instrument reading and the time reading from the data acquisition system. The format is: READING=$N_1*(C_1)N_2*(C_2)\ldots N_n*(C_n)$ and TIME=$N_1*(C_1)N_2*(C_2)\ldots N_n*(C_n)$ where N_1, N_2, ..., N_n are integers greater than or equal to 1 and indicate the number of times each character C_1, C_2, ..., C_n is repeated. The possibilities for the characters C_1, C_2, ..., C_n are: S -- indicating the seconds portion of the time reading, M -- indicating the minutes portion of the time reading, H -- indicating the hours portion of the time reading, D -- indicating the days portion of the time reading, N -- indicating any numeric digit (0-9),

Table 4. PLOT1 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
					A -- indicating any alphanumeric character, C -- indicating a channel number digit, + -- indicating the character used to identify a positive reading. The + is followed by the actual character used for identifying positive readings. - -- indicating the character used to identify a negative reading. The - is followed by the actual character used for identifying negative readings. V -- indicating a numeric digit of the value of the instrument reading, R -- indicating a numeric digit of the value of the instrument reading, possibly with an embedded decimal point, E -- indicating a numeric digit of the exponent of the instrument reading, O -- indicating the character used to identify an overflow in the instrument reading. The actual character used to identify the overflow follows the O. K -- indicating a special single character. The character follows the K. If several possibilities exist for a single character, then all possibilities are placed within the parenthesis. For instance, if a single character is used to indicate +, -, and overflow, it might be coded as (+1-209) indicating the + character is a 1, the - character is a 2, and an overflow is a 9. Consider a reading as follows: three digits of channel, a single digit as above indicating sign of reading and overflow,

Table 4. PLOT1 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
					five characters indicating the value of the reading, a single character exponent and two spaces. It would be coded as READING=(C)(C)(C)(+1-209)(V)(V)(V)(V)(V)(E)(K)(K) or READING=3*(C)(+1-209)5*(V)(E)2*(K).
		C4	IN(I),I=1,80	80A1	This card specifies the character by character format used to identify the end of a record of data recorded by a data acquisition system. The format is $EOR=N_1*(C_1)N_1*(C_2)\ldots N_n*(C_n)$ or EOR=EOR. The first format is in the same form as the cards for time or readings. The second format is used for magnetic tape and indicates data records are separated by physical record gaps on the tape.
		C5	IN(I),I=1,80	80A1	This card specifies the character by character format used to identify the end of the entire data set recorded by a data acquisition system. The format is $EOF=N_1*(C_1)N_2*(C_2)\ldots N_n*(C_n)$ or EOF=EOF. As before, the first format is similar to the form used for time or instrument readings. The second format is used for magnetic tape to indicate a physical end-of-file mark on the magnetic tape and is used to identify the end of test.

Table 4. PLOT1 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
PLOT1	D	D1			This card specifies the records to be skipped during the processing of data recorded by a data acquisition system. Two formats are possible:
				16I5	The variables specify up to 16 individual records to be skipped during the processing.
				80A1	SKIP=$(C_1N_1)R_1$ $(C_2N_2)R_2$ ... $(C_nN_n)R_n$ where N_1, N_2, ..., N_n are integers greater than or equal to 1 and indicate the number of times each character C_1, C_2, ... C_n is repeated. The possibilities for the characters are: S - indicating the record is to be skipped K - indicating the record is to be kept F - indicating the final record to be processed If several combinations exist for a single C_iN_i, then all possibilities are placed within the parenthesis. R_1, R_2, ..., R_n indicate the record numbers up to which the skipping specifications are effective. For example, if the user wishes to skip two records and keep one record from record 1 to record 200, it would be coded as (S2K1)200. Consider the following requirements: the user wishes to keep records 1 - 5, skip 2 records and keep 1 record for records 6 - 150, and keep every record for records 151 - 599. Record 600 is to be the last record processed. It would be coded as: SKIP=(K1)5 (S2K1)150 (K1)599 (F1)600 Care should be taken to insure that no overlaps or conflicts exist in the parameters.
PLOT1	E				Part E data cards are card images prepared in the format recorded by a data acquisition system. Different formats, as described in Section 9.3 are possible. If the card images were prepared by an earlier run of SPEED2 (by setting INPNCH greater than zero), the set of card images produced should be in the proper format for insertion at this point.

Table 5. PLOT2 Data Cards

Phase	Part	Card	Variables	Format	Comments
PLOT2	A	A1	INTYPE,INPRT,INPNCH, INSTOP,INERR,INSKIP	6I5	This card contains parameters which control the input. The various possibilities and their meanings are: INTYPE: 0 -- read PLOT2 data cards Part C to specify a special data acquisition system; 1 -- specifies VIDAR 5400 series magnetic tape data acquisition system; 2 -- not used; 3 -- specifies VIDAR 5400 series format punched card images; 4 -- specifies reduced data format punched card images; 5 -- specifies VIDAR Autodata 8 series format punched card images; 6 -- specifies Esterline Angus PD2064 series magnetic tape data acquisition system; 7 -- specifies Esterline Angus PD2064 series format punched card images; 8 -- specifies VIDAR Autodata 9 series magnetic tape data acquisition system; 9 -- specifies VIDAR Autodata 9 series format punched card images; 10 -- specifies DORIC data acquisition system format punched card images. INPRT: Directs printing of data as recorded by a data acquisition system. If INPRT is equal to zero, no data is printed. If INPRT is greater than zero, INPRT specifies the maximum number of data records to be printed. If INPRT is equal to -1, all data records are printed. If INPRT is equal to -2, only data records that contain errors are printed.

Table 5. PLOT2 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
					INPNCH: Directs the output of the data as recorded by a data acquisition system to card punch, mass storage, or magnetic tape. If INPNCH is greater than zero, all data records are written to the unit specified by INPNCH. The user may assign this unit to any storage media. (For the NBS UNIVAC 1108, unit 1 is the default card punch and units 9-44 are available to be assigned by the user or mass storage or magnetic tape files.) INSTOP: Directs SPEED2 to stop execution after processing the input data recorded by the data acquisition system. INERR: Specifies the maximum number of error messages to be printed during processing of data recorded by a data acquisition system. INSKIP: If INSKIP is non-zero, read PLOT1 data cards Part D to specify records of input data to be skipped.
		A2	NTEST,NPRT,NPNCH, NPLOT,NCORR	5I5	This card contains certain parameters which control actions concerning the transformed data matrix. The possibilities are: NTEST: A number between 0 and 999 specifying a test number to be prefixed to reduced data instrument numbers output by setting NPNCH greater than zero.

Table 5. PLOT2 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
					NPRT: If NPRT is non-zero,the transformed data matrix is printed. NPNCH: Directs the output of the transformed data matrix to card punch, mass storage, or magnetic tape. If NPNCH is greater than zero, all data records are written to the unit specified by NPNCH. The user may assign this unit to any storage media. (For the NBS UNIVAC 1108, unit 1 is the default card punch, and units 9-44 are available to be assigned by the user as mass storage or magnetic tape files.) NPLOT: If NPLOT is non-zero,PLOT2 data cards Part I are read to generate printer plots of selected instruments. NCORR: If NCORR is non-zero, PLOT2 data cards Part G are read to correct readings of the data matrix.
PLOT2	B	B1,B2	TITLE(I),I=1,120	80A1/40A1	These two cards specify the title of the experiment, printed at the top of all pages of output.
		B3	KH(i),ITYPE(i), NAME(i)	I6,I2,66A1	For each instrument included in the data matrix, there must be a card of this form defining the instrument number, KH(i); the instrument type, ITYPE(i); and the instrument name, NAME(i). The instrument number, KH(i), is either the channel number assigned by the data acquisition system, or, for user created instruments, a unique number assigned by the user. The number of cards is variable; with the end signaled by a card B4, below.

Table 5. PLOT2 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
		B4	IEND	77X,I3	If IEND is equal to 999, this card signals the end of the set of instrument defining cards, B3 above.
		B5	C(i),ADD(i)	1X,2F15.6	Each C(i) and ADD(i) represent the conversion coefficients for all instruments of type i, for the linear conversion defined in section 5.2. There are as many C(i) and ADD(i) as there are different types of instruments as defined in the set of instrument cards, B3 above.
		B6	IEND	77X,I3	If IEND is equal to 999, this card signals the end of the set of conversion coefficient cards.
PLOT2	C				Part C data cards specify a special data acquisition system. They are read if and only if INTYPE is equal to zero.
		C1	IN(I),I=1,80	80A1	This card specifies the type of the input media. Two formats are possible: INPUT=CARD IMAGES, CHANNELS PER LINE=N_1 INPUT=TAPE, TAPE FORMAT=C_1 The first format specifies that the input is in card images. Punched cards, data added from mass storage or tape files, and data transferred from remote terminals are considered card images. N_1 is an integer that specifies the number of channels recorded on a single card image. The second format specifies magnetic tape recorded by the data acquisition system. C_1 is a character string that defines the tape format. Currently, only TAPE FORMAT=7TRACK is defined.

Table 5. PLOT2 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
		C2,C3	IN(I),I=1,80	80A1	This card specifies the character by character format of an instrument reading and the time reading from the data acquisition system. The format is: $READING=N_1*(C_1)N_2*(C_2)...N_n*(C_n)$ and $TIME=N_1*(C_1)N_2*(C_2)...N_n*(C_n)$ where N_1, N_2, ..., N_n are integers greater than or equal to 1 and indicate the number of times each character C_1, C_2, ..., C_n is repeated. The possibilities for the characters C_1, C_2, ..., C_n are: S -- indicating the seconds portion of the time reading, M -- indicating the minutes portion of the time reading, H -- indicating the hours portion of the time reading, D -- indicating the days portion of the time reading, N -- indicating any numeric digit (0-9), A -- indicating any alphanumeric character, C -- indicating a channel number digit, + -- indicating the character used to identify a positive reading. The + is followed by the actual character used for identifying positive readings. - -- indicating the character used to identify a negative reading. The - is followed by the actual character used for identifying negative readings. V -- indicating a numeric digit of the value of the instrument reading, R -- indicating a numeric digit of the value of the instrument reading, possibly with an embedded decimal point, E -- indicating a numeric digit of the exponent of the instrument reading,

Table 5. PLOT2 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
					O -- indicating the character used to identify an overflow in the instrument reading. The actual character used to identify the overflow follows the O. K -- indicating a special single character. The character follows the K. If several possibilities exist for a single character, then all possibilities are placed within the parenthesis. For instance, if a single character is used to indicate +, -, and overflow, it might be coded as (+1-209) indicating the + character is a 1, the - character is a 2, and an overflow is a 9. Consider a reading as follows: three digits of channel, a single digit as above indicating sign of reading and overflow, five characters indicating the value of the reading, a single character exponent and two spaces. It would be coded as READING=(C)(C)(C)(+1-209)(V)(V)(V)(V)(V)(E)(K)(K) or READING=3*(C)(+1-209)5*(V)(E)2*(K).
		C4	IN(I),I=1,80	80A1	This card specifies the character by character format used to identify the end of a record of data recorded by a data acquisition system. The format is EOR=N_1*(C_1)N_2*(C_2)...N_n*(C_n) or EOR=EOR. The first format is in the same form as the cards for time or readings. The second format is used for magnetic tape and indicates data records are separated by physical record gaps on the tape.

Table 5. PLOT2 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
		C5	IN(I),I=1,80	80A1	This card specifies the character by character format used to identify the end of the entire data set recorded by a data acquisition system. The format is $EOF=N_1*(C_1)N_2*(C_2)\ldots N_n*(C_n)$ or EOF=EOF. As before, the first format is similar to the form used for time or instrument readings. The second format is used for magnetic tape to indicate a physical end-of-file mark on the magnetic tape and is used to identify the end of test.
PLOT2	D	D1			This card specifies the records to be skipped during the processing of data recorded by a data acquisition system. Two formats are possible:
				16I5	The variables specify up to 16 individual records to be skipped during the processing.
				80A1	$SKIP=(C_1N_1)R_1\ (C_2N_2)R_2\ \ldots\ (C_nN_n)R_n$ where $N_1, N_2, \ldots, N_n$ are integers greater than or equal to 1 and indicate the number of times each character $C_1, C_2, \ldots C_n$ is repeated. The possibilities for the characters are: S - indicating the record is to be skipped K - indicating the record is to be kept F - indicating the final record to be processed If several combinations exist for a single C_iN_i, then all possibilities are placed within the parenthesis. $R_1, R_2, \ldots, R_n$ indicate the record numbers up to which the skipping specifications are effective. For example, if the user wishes to skip two records and keep one record from record 1 to record 200, it would be coded as (S2K1)200. Consider the following requirements: the user wishes to keep records 1 - 5, skip 2 records and keep 1 record for records 6 - 150, and keep every record for records 151 - 599. Record 600 is to be the last record processed. It would be coded as: SKIP=(K1)5 (S2K1)150 (K1)599 (F1)600 Care should be taken to insure that no overlaps or conflicts exist in the parameters.

Table 5. PLOT2 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
PLOT2	E				Part E data cards are card images prepared in the format recorded by a data acquisition system. Different formats, as described in table 1, are possible. If the card images were prepared by an earlier run of SPEED2 (by setting INPNCH greater than zero), the set of card images produced should be in the proper format for insertion at this point.
PLOT2	F				Part F data cards are data cards prepared from converted data either by an earlier run of SPEED2 (by setting the parameter NPNCH non-zero)or by the user in this format as defined in table 1. These cards are read if and only if the parameter INTYPE is equal to 4.
PLOT2	G	G1	IRL,IRH,ICL,ICH	4I5	The variables define a low row index (IRL), a high row index (IRH), a low column index (ICL), and a high column index (ICH) to define the portion of the data matrix to be corrected. See section 5.3. These cards are read if and only if the parameter NCORR is non-zero.
		G2	REED(i,j),i=IRL,IRH, or j=ICL,ICH	8F10.0	These are the corrections to the matrix. The number of cards required depends on the number of corrections: $\left[\frac{(IRH-IRL)+(ICH-ICL)+1}{8}\right] + 1$ where [ ] represents the greatest integer function. As many additional sets of cards G1 and G2 as are required may be included to perform the necessary corrections. One final card of the form G1 with IRL less than zero must be used to signal the end of Part G data cards.

Table 5. PLOT2 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
PLOT2	H				Nonlinear conversions prepared by the user as changes to subprogram CONV using ordinary FORTRAN statements may contain statements to read in data cards of their own. If the user expects to read in any cards, they should be inserted at this point.
PLOT2	I	I1	PLOT N_1 N_2 ... N_n or PLOT N_1,N_2 N_3,N_4 ... N_{n-1},N_n		This card (or cards) specifies the instruments to be plotted on the graph. The card contains the word PLOT and numbers punched in free format, separated by spaces. In the first format, instrument N_1 is plotted versus N_2, N_3 ... and N_n; in the second, N_1 is plotted versus N_2, N_3 versus N_4, ... and N_{n-1} versus N_n and the numbers are separated into pairs by commas. If the instruments will not fit on one card, they may be continued onto additional cards by placing a semicolon in column 80.
		I2	XL,XH,YL,YH	4F10.0	These variables define the lower and upper limits of the X and Y axes, respectively.
		I3	IHEAD	80A1	IHEAD is the 80 character heading printed above the graph.
		I4	IBUFF,JBUFF	80A1	IBUFF is a 40 character title printed vertically along the Y axis. JBUFF is a 40 character title printed below the X axis. As many sets of Part I cards as desired may be included. A blank card I1 signals the end of Part I data cards.

Table 6. PLOT3 Data Cards

Phase	Part	Card	Variables	Format	Comments
PLOT3	A	A1	NPRT,NSMTH,NOUT, NAXIS,NPNT,NLEG	6I5	These variables control the format of the CalComp plots produced. The possibilities and their meanings are: NPRT: Directs printing of the data matrix. If NPRT is not equal to zero, the transformed data matrix is printed. NSMTH: Defines a smoothing factor for plotting of all data. A linear smoothing using NSMTH points is used to smooth all data in linear plots. If NSMTH is equal to 0 or 1, no smoothing takes place. NOUT: Defines action to be taken if out of bounds points are found. If NOUT is equal to zero, out of bounds points are not plotted. If NOUT is equal to 1, out of bounds points are plotted. If NOUT is equal to 2, execution is terminated if out of bounds points are found. NAXIS: If NAXIS is non-zero,axes are placed on all four sides of every graph produced. NPNT: If NPNT is greater than or equal to zero, a line is drawn through each data point of all curves. If NPNT is not equal to zero, a special plotting symbol is placed at every nth data point, where n=NPNT. If NPNT is less than zero, the line drawn through the data points is suppressed, and a special symbol is placed at every nth data point, where n=NPNT.

Table 6. PLOT3 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
					NLEG: If NLEG is not equal to zero, read PLOT3 data card(s) D5 to produce legends on each graph produced.
		A2	XSIZE,YSIZE,SCALE	3F10.0	These variables define the X axis length in inches (XSIZE), the Y axis length in inches (YSIZE), and a scaling factor (SCALE) to reduce or increase the overall size of all plots. The actual length of the X and Y axes are XSIZE*SCALE and YSIZE*SCALE, respectively.
PLOT3	B	B1,B2	TITLE(I),I=1,120	80A1/40A1	This is the title of the expcriment printed at the top of all pages of output.
		B3	KH(i),ITYPE(i), NAME(i)	I6,I2,66A1	For each instrument included in the data matrix, there must be a card of this form defining the instrument number, KH(i); the instrument type, ITYPE(i); and the instrument name, NAME(i). The instrument number, KH(i), is either the channel number assigned by the data acquisition system, or, for user created instruments, a unique number assigned by the user. The number of cards is variable; with the end signaled by a card B4, below.
		B4	IEND	77X,I3	If IEND is equal to 999, this card signals the end of the set of instrument defining cards, B3 above.
PLOT3	C				Part C data cards are data cards prepared from converted data either by an earlier run of SPEED2 (by setting the parameter NPNCH non-zero)or by the user in this format as defined in table 1.

Table 6. PLOT3 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
PLOT3	D	D1	PLOT N_1 N_2 ... N_n or PLOT N_1, N_2 N_3, N_4 ... N_{n-1}, N_n		This card (or cards) specifies the instruments to be plotted on the graph. The card contains the word PLOT and numbers punched in free format, separated by spaces. In the first format, instrument N_1 is plotted versus N_2, N_3 ... and N_n; in the second, N_1 is plotted versus N_2, N_3 versus N_4, ... and N_{n-1} versus N_n and the numbers are separated into pairs by commas. If the instruments will not fit on one card, they may be continued onto additional cards by placing a semicolon in column 80.
		D2	XL,XH,YL,YH	4F10.0	These variables define the lower and upper limits of the X and Y axes, respectively.
		D3	IFIG,IHEAD	I2,78A1	IFIG: If IFIG is greater than zero, IFIG specifies a figure number that is prefixed to the first line of the title. IHEAD: Two 39 character maximum titles that are placed below the graph. Each 39 character line is centered along the X axis.
		D4	IBUFF,JBUFF	80A1	IBUFF is a 40 character title printed vertically along the Y axis. JBUFF is a 40 character title printed below the X axis.
		D5	IS,ISYM,XTPOS,YTPOS, IHEAD	I1,I2,2F6.0,60A1	This card (or cards) define legends to be placed on the graphs. They are read with each set of plot cards if and only if the parameter NLEG is non-zero. The possibilities and their meanings are:

Table 6. PLOT3 Data Cards (con't)

Phase	Part	Card	Variables	Format	Comments
					IS: If IS is not equal to zero, a special plotting symbol, defined by ISYM is prefixed to the text centered before the first character. ISYM: If IS is not equal to zero and ISYM is greater than or equal to zero, the variable ISYM indicates the special symbol to be placed before the text. This symbol can be used to identify individual curves on a particular plot. Each curve corresponds to a unique plotting symbol. This first curve defined on card D1 above is assigned symbol number 0; the second, symbol number 1; and so on. XTPOS: The distance, in unscaled inches, in the X direction of the lower left-hand corner of the first character of the text IHEAD. YTPOS: The distance, in unscaled inches, in the Y direction of the lower left-hand corner of the first character of the text IHEAD. IHEAD: The text of the legend; up to 60 characters, is placed beginning at the coordinates (XTPOS, YTPOS). If IS is non-zero,the symbol corresponding to symbol number ISYM is prefixed to the text. As many sets of Part D cards as desired may be included. A blank card, D1, signals the end of Part D data cards.

Table 7. PLOT4 Data Cards

Phase	Part	Card	Variables	Format	Comments
PLOT4	A	A1	NPRT,NPNCH	2I5	If NPRT is not equal to zero, the combined data matrix is printed. The variable NPNCH directs the output of the data matrix to card punch, mass storage, or magnetic tape. If NPNCH is greater than zero, all data are written to the unit specified by NPNCH. The user may assign the unit to any storage media. (For the NBS UNIVAC 1108, unit 1 is the default card punch, and units 9-44 are available to be assigned by the user as mass storage or magnetic tape files.)
PLOT4	B	B1,B2	TITLE(I),I=1,120	80A1/40A1	These two cards specify the title of the experiment, printed at the top of all pages of output.
		B3	ITYPE(i),KH(i), NAME(i)	I3,I3,66A1	For each instrument included in the data matrix, there must be a card of this form defining the test number, ITYPE(i); the instrument number, KH(i); and the instrument name, NAME(i). The instrument number, KH(i), is either the channel number assigned by the data acquisition system, or, for user created instruments, a unique number assigned by the user. The number of cards is variable, with the end signaled by a card B4, below.
		B4	IEND	77X,I3	If IEND is equal to 999, this card signals the end of the set of instrument defining cards, B3, above.
PLOT4	C				PLOT4 data cards Part C are comprised of the readings of the instruments from the several tests, with each set identified by a unique test number.

Table 7. PLOT4 Data Cards (con't)

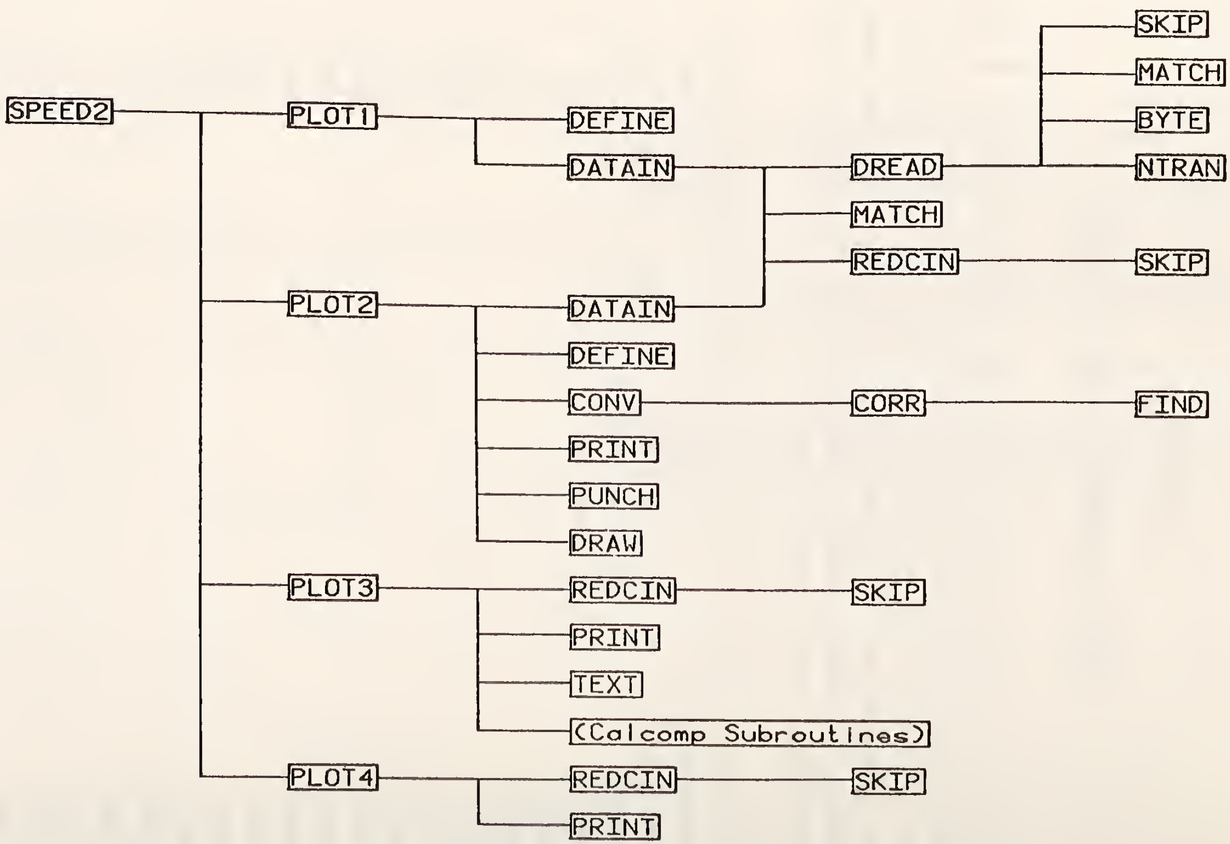
Phase	Part	Card	Variables	Format	Comments
		C1	ITEST	6X,I3	ITEST is the test number assigned by the user to identify the test data that follows. The user may assign any number between 1 and 999. Each test should be assigned a different test number.
		C2			PLOT4 data cards C2 are the set of reduced data cards for the nth test where n=ITEST prepared by an earlier run of SPEED2 (by setting NPNCH non-zero).
		C3	IN	6A1	As many sets of card C1 and cards C2 may be included as necessary to include the data from the several different tests. One final card with the characters FILEND in columns 1 through 6 must be included at the end to signal the end of the data sets.

APPENDIX A

PROGRAM STATEMENT LISTING OF SPEED2

A listing of all subprograms of the SPEED2 system follows. A program flow / program interdependency diagram is shown in figure A-1. While the user need not be concerned with most subprograms, all are included for completeness. Subprograms SPEED2 and CONV are the only two that users would normally change. All other changes should only be attempted by those with a detailed knowledge of the SPEED2 system.

Figure A-1 Program Dependency Diagram



Note: Subprograms DIGIT, EREXIT, RWERR, and RWEOF are called by a number of the subprograms in the SPEED2 system. They are not included in the diagram for simplicity and ease of interpretation.

```
C                             S P E E D 2
C
C     SYSTEMATIC PLOTTING AND EVALUATION OF ENUMERATED DATA
C     (SYSTEM 2)
C
C     WRITTEN BY   JOHN M. SMITH, COMPUTER SERVICES DIVISION, NBS
C                  RICHARD D. PEACOCK, CENTER FOR FIRE RESEARCH, NBS
C
C     SPEED2 IS A COLLECTION OF ROUTINES DESIGNED TO ROUTINELY REDUCE DATA
C     COLLECTED FROM DATA ACQUISITION SYSTEMS.  IT IS DESIGNED TO TRANS-
C     LATE DATA COLLECTED BY A DATA ACQUISITION SYSTEM, PERFORM LINEAR AND
C     NON-LINEAR TRANSFORMATIONS ON THE DATA, AND TO PRODUCE PLOTS OF THE
C     REDUCED DATA.  IN ADDITION, REPORT QUALITY PEN PLOTS CAN BE GENERAT-
C     ED FOR INCLUSION IN REPORTS.
C
C       ******************************************************************
C
C     SPEED2 COMPILE TIME PARAMETERS
C
C     NROW:      MAXIMUM NUMBER OF ROWS (SCANS) IN THE INPUT DATA
C     NCOL:      MAXIMUM NUMBER OF COLUMNS (INSTRUMENTS) TO BE PROCESSED
C     MAXPLT:    MAXIMUM NUMBER OF CURVES TO BE PLOTTED ON A SINGLE CURVE
C     MAXCNL:    MAXIMUM NUMBER OF INSTRUMENTS RECORDED BY THE DATA SYSTEM
C
        PARAMETER NROW=125, NCOL=21
        PARAMETER MAXPLT=6
        PARAMETER MAXCNL=150
C
C       ******************************************************************
C
        PARAMETER OUTDIM=18*MAXCNL+160
        PARAMETER NPTS=MAXPLT*NROW+2
        PARAMETER PLTDIM=2*MAXPLT
        INTEGER VERSN
        COMMON /ERRORS/ ICRD,ISEG
        DIMENSION REED(NROW,NCOL),KH(NCOL),ITYPE(NCOL),C(NCOL),
      2 NAME(NCOL,6),ADD(NCOL),X(NROW,MAXPLT),Y(NROW,MAXPLT),XP3(NPTS),
      3 YP3(NPTS),MAXR(NCOL),IOUT(OUTDIM),MCNL(MAXCNL),IPC(PLTDIM),
      4 JM(PLTDIM),IPN(PLTDIM,6),VERSN(10),ISKIP(320)
        EQUIVALENCE (X,XP3), (Y,YP3)
C
C     VERSN IS USED FOR IDENTIFICATION PURPOSES ONLY, TO IDENTIFY THE
C     VERSION OF SPEED2 IN USE
C
        DATA (VERSN(I),I=1,10) /1H4,1H5,1H.,1H1,1H1,1H ,1H1,1H1,1H0,1H8/
C
        DATA IBL /1H /
        WRITE (6,130) (VERSN(I),I=1,10)
        ISEG=1
        ICRD=11
```

```
C
C     SPEED2 DATA CARD A1:    NSETS
C     NSETS:  NUMBER OF SETS OF DATA TO BE PROCESSED
C
      READ (5,140,ERR=110,END=120) NSETS
C     ZERO ARRAYS TO PREPARE FOR NEXT DATA SET
10    DO 40 I=1,NCOL
      KH(I)=0
      ITYPE(I)=0
      C(I)=0.
      ADD(I)=0.
      MAXR(I)=0
      DO 20 J=1,NROW
20    REED(J,I)=0.
      DO 30 K=1,6
30    NAME(I,K)=IBL
40    CONTINUE
      DO 50 I=1,320
      ISKIP(I)=0
50    CONTINUE
      DO 60 I=1,MAXCNL
      MCNL(I)=0
60    CONTINUE
      ISEG=1
      ICRD=12
C
C     SPEED2 DATA CARD A2:    IP1,IP2,IP3,IP4
C     IP1:     IF IP1 IS NON-ZERO, CALL PLOT1
C     IP2:     IF IP2 IS NON-ZERO, CALL PLOT2
C     IP3:     IF IP3 IS NON-ZERO, CALL PLOT3
C     IP4:     IF IP4 IS NON-ZERO, CALL PLOT4
C
      READ (5,140,ERR=110,END=120) IP1,IP2,IP3,IP4
      IF (IP1.EQ.0) GO TO 70
      CALL PLOT1 (NROW,NCOL,REED,KH,IOUT,OUTDIM,MAXR,MCNL,MAXCNL,ISKIP)
      GO TO 100
70    IF (IP2.EQ.0) GO TO 80
      CALL PLOT2 (NROW,NCOL,REED,KH,ITYPE,C,ADD,NAME,X,Y,MAXPLT,IOUT,OUT
     2DIM,MAXR,IPC,JM,PLTDIM,IPN,MCNL,MAXCNL,ISKIP)
      GO TO 100
80    IF (IP3.EQ.0) GO TO 90
      CALL PLOT3 (NROW,NCOL,REED,KH,ITYPE,NAME,X,Y,MAXPLT,NSETS,NPTS,MAX
     2R,IPC,PLTDIM,IPN,ISKIP)
      GO TO 100
90    IF (IP4.EQ.0) GO TO 100
      CALL PLOT4 (NROW,NCOL,REED,KH,ITYPE,NAME,MAXR,MCNL,MAXCNL,ISKIP)
100   CONTINUE
      NSETS=NSETS-1
      WRITE (6,150) (VERSN(I),I=1,10)
      IF (NSETS.GT.0) GO TO 10
```

```
            STOP
      110   CALL RWERR
      120   CALL RWEOF
      C
      C
      130   FORMAT (46H1  SPEED2 DATA REDUCTION ROUTINES    VERSION: ,10A1)
      140   FORMAT (16I5)
      150   FORMAT (22H1  SPEED2    VERSION: ,10A1,19H    END OF DATA SET)
      C
            END
```

```
C                                S P E E D 2
C
C     SYSTEMATIC PLOTTING AND EVALUATION OF ENUMERATED DATA
C     (SYSTEM 2)
C
C     WRITTEN BY    JOHN M. SMITH, COMPUTER SERVICES DIVISION, NBS
C                   RICHARD D. PEACOCK, CENTER FOR FIRE RESEARCH, NBS
C
C     SPEED2 IS A COLLECTION OF ROUTINES DESIGNED TO ROUTINELY REDUCE DATA
C     COLLECTED FROM DATA ACQUISITION SYSTEMS.  IT IS DESIGNED TO TRANS-
C     LATE DATA COLLECTED BY A DATA ACQUISITION SYSTEM, PERFORM LINEAR AND
C     NON-LINEAR TRANSFORMATIONS ON THE DATA, AND TO PRODUCE PLOTS OF THE
C     REDUCED DATA.  IN ADDITION, REPORT QUALITY PEN PLOTS CAN BE GENERAT-
C     ED FOR INCLUSION IN REPORTS.
C
      INTEGER OUTDIM,PLTDIM,VERSN
      COMMON /ERRORS/ ICRD,ISEG
      DIMENSION REED(100,25),KH(25),ITYPE(25),C(25),NAME(25,6),ADD(25),
     2 X(100,6),Y(100,6),XP3(602),YP3(602),MAXR(25),IOUT(2860),MCNL(150)
     3,IPC(12),JM(12),IPN(12,6),VERSN(10),ISKIP(320)
      EQUIVALENCE (X(1,1),XP3(1)), (Y(1,1),YP3(1))
C
C     VERSN IS USED FOR IDENTIFICATION PURPOSES ONLY, TO IDENTIFY THE
C     VERSION OF SPEED2 IN USE
C
      DATA (VERSN(I),I=1,10) /1H4,1H5,1H.,1H1,1H1,1H ,1HN,1HO,1HN,1H /
C
      DATA IBL /1H /
C
C     ******************************************************************
C
C     SPEED2 COMPILE TIME PARAMETERS
C
C     NROW:     MAXIMUM NUMBER OF ROWS (SCANS) IN THE INPUT DATA
C     NCOL:     MAXIMUM NUMBER OF COLUMNS (INSTRUMENTS) TO BE PROCESSED
C     MAXPLT:   MAXIMUM NUMBER OF CURVES TO BE PLOTTED ON A SINGLE CURVE
C     MAXCNL:   MAXIMUM NUMBER OF INSTRUMENTS RECORDED BY THE DATA SYSTEM
C
      NROW=100
      NCOL=25
      MAXPLT=6
      MAXCNL=150
C
C     ******************************************************************
C
      OUTDIM=18*MAXCNL+160
      NPTS=MAXPLT*NROW+2
      PLTDIM=2*MAXPLT
      WRITE (6,130) (VERSN(I),I=1,10)
      ISEG=1
```

```
          ICRD=11
C
C    SPEED2 DATA CARD A1:   NSETS
C    NSETS:  NUMBER OF SETS OF DATA TO BE PROCESSED
C
          READ (5,140,ERR=110,END=120) NSETS
C    ZERO ARRAYS TO PREPARE FOR NEXT DATA SET
10        DO 40 I=1,NCOL
          KH(I)=0
          ITYPE(I)=0
          C(I)=0.
          ADD(I)=0.
          MAXR(I)=0
          DO 20 J=1,NROW
20        REED(J,I)=0.
          DO 30 K=1,6
30        NAME(I,K)=IBL
40        CONTINUE
          DO 50 I=1,320
          ISKIP(I)=0
50        CONTINUE
          DO 60 I=1,MAXCNL
          MCNL(I)=0
60        CONTINUE
          ISEG=1
          ICRD=12
C
C    SPEED2 DATA CARD A2:   IP1,IP2,IP3,IP4
C    IP1:     IF IP1 IS NON-ZERO, CALL PLOT1
C    IP2:     IF IP2 IS NON-ZERO, CALL PLOT2
C    IP3:     IF IP3 IS NON-ZERO, CALL PLOT3
C    IP4:     IF IP4 IS NON-ZERO, CALL PLOT4
C
          READ (5,140,ERR=110,END=120) IP1,IP2,IP3,IP4
          IF (IP1.EQ.0) GO TO 70
          CALL PLOT1 (NROW,NCOL,REED,KH,IOUT,OUTDIM,MAXR,MCNL,MAXCNL,ISKIP)
          GO TO 100
70        IF (IP2.EQ.0) GO TO 80
          CALL PLOT2 (NROW,NCOL,REED,KH,ITYPE,C,ADD,NAME,X,Y,MAXPLT,IOUT,OUT
         2DIM,MAXR,IPC,JM,PLTDIM,IPN,MCNL,MAXCNL,ISKIP)
          GO TO 100
80        IF (IP3,EQ.0) GO TO 90
          CALL PLOT3 (NROW,NCOL,REED,KH,ITYPE,NAME,X,Y,MAXPLT,NSETS,NPTS,MAX
         2R,IPC,PLTDIM,IPN,ISKIP)
          GO TO 100
90        IF (IP4.EQ.0) GO TO 100
          CALL PLOT4 (NROW,NCOL,REED,KH,ITYPE,NAME,MAXR,MCNL,MAXCNL,ISKIP)
100       CONTINUE
          NSETS=NSETS-1
          WRITE (6,150) (VERSN(I),I=1,10)
```

```
          IF (NSETS.GT.0) GO TO 10
          STOP
    110   CALL RWERR
    120   CALL RWEOF
    C
    C
    130   FORMAT (46H1  SPEED2 DATA REDUCTION ROUTINES     VERSION: ,10A1)
    140   FORMAT (16I5)
    150   FORMAT (22H1  SPEED2    VERSION: ,10A1,19H    END OF DATA SET)
    C
          END
```

```
      SUBROUTINE PLOT1 (NROW,NCOL,REED,KH,IOUT,OUTDIM,MAXR,MCNL,MAXCNL,
     2 ISKIP)
C
C     ****************************************************************
C
      COMMON /ERRORS/ ICRD,ISEG
      INTEGER OUTDIM
      DIMENSION REED(NROW,NCOL),KH(NCOL),IOUT(OUTDIM),TITLE(120),
     2 MAXR(NCOL),INDAS(22),ICHRS(320),MCNL(MAXCNL),ISKIP(320)
      ISEG=2
      ICRD=11
C
C    PLOT1 DATA CARD A1:    INTYPE,INPRT,INPNCH,INERR,INSKIP
C    INTYPE: UNCONVERTED INPUT DATA FILE TYPE
C    INPRT:  DIRECTS PRINTING OF UNCONVERTED INPUT DATA
C    INPNCH: DIRECTS PUNCHING OF UNCONVERTED INPUT DATA
C    INERR:  NUMBER OF INPUT ERROR MESSAGES TO BE PRINTED
C    INSKIP: DIRECTS SKIPPING OF INPUT RECORDS
C
      READ (5,30,ERR=10,END=20) INTYPE,INPRT,INPNCH,INERR,INSKIP
      ICRD=21
C
C    PLOT1 DATA CARDS B1&B2:    TITLE
C    TITLE:  SPECIFIES TITLE OF EXPERIMENT PRINTED AT TOP OF EACH PAGE
C
      READ (5,40,ERR=10,END=20) (TITLE(I),I=1,120)
      WRITE (6,50)
      WRITE (6,60) INTYPE,INPRT,INPNCH,INERR,INSKIP
      NR=NROW*NCOL
      NC=1
      KH(1)=-1
      ITIME=1
      CALL DEFINE (INTYPE,ICHRS,INDAS,IREAD,ITIM,IEOR,IEOF,MEDIA,TITLE,I
     2NSKIP,ISKIP,NCNL,NTFORM)
      CALL DATAIN (NR,NC,REED,KH,INTYPE,INPRT,INPNCH,MAXR,ITIME,INERR,IN
     2SKIP,IOUT,OUTDIM,ICHRS,INDAS,IREAD,ITIM,IEOR,IEOF,MEDIA,TITLE,MCNL
     3,MAXCNL,ISKIP,NCNL,NTFORM)
      RETURN
   10 CALL RWERR
   20 CALL RWEOF
C
C
   30  FORMAT (16I5)
   40  FORMAT (80A1/40A1)
   50  FORMAT (27H0  PLOT1 CONTROL PARAMETERS,/)
   60  FORMAT (8H INTYPE:,I5,2X,7H INPRT:,I5,2X,7HINPNCH:,I5,2X,7H INERR:
     2,I5,2X,7HINSKIP:,I5)
C
      END
```

```
      SUBROUTINE PLOT2 (NROW,NCOL,REED,KH,ITYPE,C,ADD,NAME,X,Y,MAXPLT,
     2 IOUT,OUTDIM,MAXR,IPC,JM,PLTDIM,IPN,MCNL,MAXCNL,ISKIP)
C
C     *****************************************************************
C
      COMMON /ERRORS/ ICRD,ISEG
      INTEGER OUTDIM,PLTDIM,CHAR
      DIMENSION REED(NROW,NCOL),KH(NCOL),ITYPE(NCOL),C(NCOL),ADD(NCOL),
     2 NAME(NCOL,6),X(NROW,MAXPLT),Y(NROW,MAXPLT),MAXR(NROW),
     3 IOUT(OUTDIM),TITLE(120),IN(80),CHAR(10),IC(6),IPC(PLTDIM),
     4 JM(PLTDIM),IHEAD(80),ISYM(6),ICHRS(320),INDAS(22),IPN(PLTDIM,6),
     5 IBUFF(40),JBUFF(40),MCNL(MAXCNL),ISKIP(320)
      DATA (CHAR(J),J=1,10) /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
      DATA (ISYM(J),J=1,6) /1H*,1HX,1H0,1H+,1H-,1H./
      DATA NSEMI,NCOM /1H;,1H,/
      ISEG=3
      ICRD=11
C
C    PLOT2 DATA CARD A1:    INTYPE,INPRT,INPNCH,INSTOP,INERR,INSKIP
C    INTYPE: UNCONVERTED INPUT DATA FILE TYPE
C    INPRT:  DIRECTS PRINTING OF UNCONVERTED INPUT DATA
C    INPNCH: DIRECTS PUNCHING OF UNCONVERTED INPUT DATA
C    INSTOP: IF INSTOP IS NON-ZERO, STOP PROCESSING
C    INERR:  NUMBER OF INPUT ERROR MESSAGES TO BE PRINTED
C    INSKIP: DIRECTS SKIPPING OF INPUT RECORDS
C
      READ (5,270,ERR=220,END=230) INTYPE,INPRT,INPNCH,INSTOP,INERR,INSK
     2IP
      ICRD=12
C
C    PLOT2 DATA CARD A2:    NTEST,NPRT,NPNCH,NPLOT,NCORR
C    NTEST:  TEST NUMBER
C    NPRT:   DIRECTS PRINTING OF CONVERTED DATA
C    NPNCH:  DIRECTS PUNCHING OF CONVERTED DATA
C    NPLOT:  IF NPLOT IS NON-ZERO, PRINTER PLOTS ARE TO BE GENERATED
C    NCORR:  IF NCORR IS NON-ZERO, READ CORRECTIONS TO DATA
C
      READ (5,270,ERR=220,END=230) NTEST,NPRT,NPNCH,NPLOT,NCORR
      ICRD=21
C
C    PLOT2 DATA CARDS B1&B2:   TITLE
C    TITLE:  SPECIFIES TITLE OF EXPERIMENT PRINTED AT TOP OF EACH PAGE
C
      READ (5,280,ERR=220,END=230) (TITLE(I),I=1,120)
      WRITE (6,290)
      WRITE (6,390) INTYPE,INPRT,INPNCH,INSTOP,INERR,INSKIP
      WRITE (6,400) NTEST,NPRT,NPNCH,NPLOT,NCORR
      K=1
      ITIME=0
      WRITE (6,300)
```

```
   10    ICRD=23
   C
   C  PLOT2 DATA CARDS B3&B4:   KH,ITYPE,NAME,IEND
   C  KH:      INSTRUMENT NUMBER
   C  ITYPE:   INSTRUMENT TYPE
   C  NAME:    DESCRIPTION OF INSTRUMENT
   C  IEND:    DIRECTS END OF INSTRUMENT CARDS
   C
         READ (5,310,ERR=220,END=230) KH(K),ITYPE(K),(NAME(K,J),J=1,6),(IN(
        2I),I=1,60),IEND
         IF (IEND.EQ.999) GO TO 30
         IF (ITYPE(K).EQ.1.AND.ITIME.NE.0) GO TO 20
         IF (ITYPE(K).EQ.1) ITIME=K
         WRITE (6,320) KH(K),ITYPE(K),(NAME(K,J),J=1,6),(IN(I),I=1,60),K
         IF (MOD(K,5).EQ.0) WRITE (6,250)
         IF (MOD(K-35,40).EQ.0) WRITE (6,410) (TITLE(I),I=1,120)
         IF (MOD(K-35,40).EQ.0) WRITE (6,300)
         K=K+1
         IF (K.LE.NCOL) GO TO 10
   C  TOO MANY INSTRUMENTS, KILL THE RUN
         WRITE (6,370) K
         CALL EREXIT
         STOP
   20    WRITE (6,260) KH(K)
         CALL EREXIT
         STOP
   30    WRITE (6,330)
         MAXC=K-1
         K=1
   40    ICRD=25
   C
   C  PLOT2 DATA CARDS B5&B6:   C,ADD,IEND
   C  C:       CONVERSION CONSTANT
   C  ADD:   ADDITION CONSTANT
   C  IEND:    DIRECTS END OF CONVERSION CARDS
   C
         READ (5,340,ERR=220,END=230) C(K),ADD(K),IEND
         IF (IEND.EQ.999) GO TO 50
         WRITE (6,350) K,C(K),K,ADD(K)
         K=K+1
         GO TO 40
   C  READ INPUT FILE.
   50    CALL DEFINE (INTYPE,ICHRS,INDAS,IREAD,ITIM,IEOR,IEOF,MEDIA,TITLE,I
        2NSKIP,ISKIP,NCNL,NTFORM)
         CALL DATAIN (NROW,NCOL,REED,KH,INTYPE,INPRT,INPNCH,MAXR,ITIME,INER
        2R,INSKIP,ICUT,CUTDIM,ICHRS,INDAS,IREAD,ITIM,IEOR,IEOF,MEDIA,TITLE,
        3MCNL,MAXCNL,ISKIP,NCNL,NTFORM)
         IF (INSTOP.NE.0) RETURN
   C  CONVERT INPUT FILE.
         CALL CONV (NROW,NCOL,REED,KH,ITYPE,C,ADD,NCORR,MAXR,MAXC)
```

```
C   PRINT OUT CONVERTED DATA (IF NPRT .NE. 0)
      IF (NPRT.NE.0) CALL PRINT (NROW,NCOL,REED,KH,ITYPE,NAME,ITIME,MAXR
     2,MAXC,TITLE)
C   PUNCH OUT REDUCED DATA (IF NPNCH IS NON-ZERO)
      IF (NPNCH.NE.0) CALL PUNCH (NROW,NCOL,REED,KH,NAME,MAXR,MAXC,NTEST
     2,NPNCH)
      IF (NPLOT.EQ.0) RETURN
C   GENERATE PLOTS ( IF NPLOT .NE. 0 )
60    ICRD=91
C
C   PLOT2 DATA CARD I1:   IN
C   IN:      CARD SPECIFIES INSTRUMENTS TO BE PLOTTED
C
      READ (5,380,ERR=220,END=230) IN
      IP=0
      K=0
      DO 120 I=1,80
      DO 70 J=1,10
      IF (IN(I).EQ.CHAR(J)) GO TO 110
70    CONTINUE
      IF (K.EQ.0) GO TO 120
      IP=IP+1
C   CALCULATES INSTRUMENT NUMBER.
      L=0
      DO 80 J=1,K
      M=IABS(K-J)
80    L=L+IC(J)*10**M
C   LOCATE COLUMN INSTRUMENT IS STORED IN.
      DO 90 J=1,MAXC
      IF (KH(J).EQ.L) IPC(IP)=J
90    CONTINUE
      IF (IPC(IP).NE.0) GO TO 100
      WRITE (6,240) L,IN
      IP=IP-1
100   K=0
      GO TO 120
110   K=K+1
      IC(K)=J-1
120   CONTINUE
      IF (IN(80).EQ.NSEMI) GO TO 60
      IF (IP.EQ.0) RETURN
      DO 130 I=1,80
      IF (IN(I).EQ.NCOM) GO TO 170
130   CONTINUE
      L=IPC(1)
      IN(1)=NAME(L,1)
      DO 160 I=2,IP
      K=IPC(I)
      MAXROW=MAXR(K)
      DO 140 J=1,MAXROW
```

```
            X(J,I-1)=REED(J,L)
      140   Y(J,I-1)=REED(J,K)
            JM(I-1)=MAXROW
            DO 150 M=1,6
      150   IPN(I-1,M)=NAME(K,M)
      160   CONTINUE
            IM=IP-1
            GO TO 210
      170   DO 200 I=1,IP,2
            L=IPC(I)
            K=IPC(I+1)
            MAXROW=MAXR(K)
            ICRV=I/2+1
            DO 180 J=1,MAXROW
            X(J,ICRV)=REED(J,L)
      180   Y(J,ICRV)=REED(J,K)
            JM(ICRV)=MAXROW
            DO 190 M=1,6
      190   IPN(ICRV,M)=NAME(K,M)
      200   CONTINUE
            IM=IP/2
      210   ICRD=92
      C
      C   PLOT2 DATA CARD I2:   XL,XH,YL,YH
      C   XL:      X-AXIS LOWER LIMIT
      C   XH:      X-AXIS UPPER LIMIT
      C   YL:      Y-AXIS LOWER LIMIT
      C   YH:      Y-AXIS UPPER LIMIT
      C
            READ (5,360,ERR=220,END=230) XL,XH,YL,YH
            ICRD=93
      C
      C   PLOT2 DATA CARD I3:   IHEAD
      C   IHEAD:  GRAPH HEADING
      C
            READ (5,380,ERR=220,END=230) IHEAD
            ICRD=94
      C
      C   PLOT2 DATA CARD I4:   IBUFF,JBUFF
      C   IBUFF, JBUFF:  Y-AXIS, X-AXIS TITLES
      C
            READ (5,380,ERR=220,END=230) IBUFF,JBUFF
            CALL DRAW (JM,IM,X,Y,XL,XH,YL,YH,IBUFF,JBUFF,IHEAD,TITLE,IPN,NROW,
           2PLTDIM)
            GO TO 60
      220   CALL RWERR
      230   CALL RWEOF
      C
      C
      240   FORMAT (24H0NO FIND FOR INSTRUMENT ,I6,23H ERROR ON THE FOLLOWING,
```

```
         2 10H PLOT CARD,/,1X,80A1)
   250    FORMAT (1H )
   260    FORMAT (41H0ONLY ONE INSTRUMENT OF TYPE 1 IS ALLOWED,/,
         2 35H SECOND FOUND AT INSTRUMENT NUMBER ,I3)
   270    FORMAT (16I5)
   280    FORMAT (80A1/40A1)
   290    FORMAT (27H0  PLOT2 CONTROL PARAMETERS/)
   300    FORMAT (22H0 TABLE OF INSTRUMENTS/,6X,
         2 41HKH ITYPE I.D.      INSTRUMENT DESCRIPTION,T87,6HCOLUMN,/,1H ,
         3 92(1H-),/)
   310    FORMAT (I6,I2,66A1,3X,I3)
   320    FORMAT (1X,I6,I3,3X,6A1,2X,60A1,T87,I6)
   330    FORMAT (//,22H0TABLE OF MULTIPLIERS ,10X,15HTABLE OF ADDERS)
   340    FORMAT (1X,F15.6,F15.6,I49)
   350    FORMAT (2H C,I2,1H=,F15.6,12X,3HADD,I2,1H=,F15.6)
   360    FORMAT (4F10.0)
   370    FORMAT (42H0COLUMN STORAGE LIMITS EXCEEDED AT COLUMN ,I4,/,
         2 31H CHECK PARAMETERS NCOL AND NROW)
   380    FORMAT (80A1)
   390    FORMAT (8H INTYPE:,I5,2X,7H INPRT:,I5,2X,7HINPNCH:,I5,2X,7HINSTOP:
         2,I5,2X,7H INERR:,I5,2X,7HINSKIP:,I5)
   400    FORMAT (8H  NTEST:,I5,2X,7H  NPRT:,I5,2X,7H NPNCH:,I5,2X,7H NPLOT:
         2,I5,2X,7H NCORR:,I5)
   410    FORMAT (1H1,120A1)
C
          END
```

```
      SUBROUTINE PLOT3 (NROW,NCOL,REED,KH,ITYPE,NAME,X,Y,MAXPLT,NSETS,
     2 NPTS,MAXR,IPC,PLTDIM,IPN,ISKIP)
C
C     ******************************************************************
C
      COMMON /ERRORS/ ICRD,ISEG
      INTEGER CHAR,PLTDIM,XPOW,XDIG,YPOW,YDIG,FTIT
      REAL IY,IX
      DIMENSION REED(NROW,NCOL),KH(NCOL),ITYPE(NCOL),NAME(NCOL,6),
     2 TITLE(120),IN(80),CHAR(10),IC(6),IPC(PLTDIM),AREA(1024),IBUFF(40)
     3,JBUFF(40),X(NPTS),Y(NPTS),IHEAD(78),MAXR(NCOL),IPN(PLTDIM,6),
     4 FTIT(10),ISKIP(320)
      DATA (CHAR(J),J=1,10) /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
      DATA (FTIT(J),J=1,10) /1HF,1HI,1HG,1HU,1HR,1HE,1H ,1H ,1H ,1H:/
      DATA NCOM,NSEMI,NS,NP,NL /1H,,1H;,1HS,1HP,1HL/
C   INITIALIZE THE CALCOMP TAPE FOR THE PLOTTING
      CALL PLOTS (AREA,1024,7)
      CALL PLOT (0.,-11.,-3)
      CALL PLOT (0.,1.5,-3)
      WRITE (6,710)
C
      ISEG=4
      ICRD=11
C
C   PLOT3 DATA CARD A1:   NPRT,NSMTH,NOUT,NAXIS,NPNT,NLEG
C   NPRT:   IF NPRT IS NON-ZERO, PRINT DATA
C   NSMTH:  SMOOTHING FACTOR
C   NOUT:   DETERMINES FATE OF OUT-OF-BOUNDS POINTS
C   NAXIS:  TWO OR FOUR SIDED AXES
C   NPNT:   SPACING OF POINTS ON CURVES
C   NLEG:   READ CARDS FOR LEGENDS ON CURVES
C
      READ (5,720,ERR=630,END=640) NPRT,NSMTH,NOUT,NAXIS,NPNT,NLEG
      ICRD=12
C
C   PLOT3 DATA CARD A2:   XSIZE, YSIZE, SCALE
C   XSIZE:  LENGTH OF THE X AXIS OF THE PLOT IN INCHES
C   YSIZE:  LENGTH OF THE Y AXIS OF THE PLOT IN INCHES
C   SCALE:  SCALE FACTOR FOR PLOTTING
C
      READ (5,880,ERR=630,END=640) XSIZE,YSIZE,SCALE
      IF (XSIZE.LT.1.0) XSIZE=10.0
      IF (YSIZE.LT.1.0) YSIZE=8.0
      IF (SCALE.EQ.0.) SCALE=1.0
      IF (NSMTH.LT.0) NSMTH=0
      CALL FACTOR (SCALE)
      WRITE (6,730) NPRT,NSMTH,NOUT,NAXIS,NPNT,NLEG
      WRITE (6,740) XSIZE,YSIZE,SCALE
      ICRD=21
C
```

```
C   PLOT3 DATA CARDS B1 & B2:    TITLE
C   TITLE:  SPECIFIES TITLE OF EXPERIMENT PRINTED AT TOP OF EACH PAGE
C
      READ (5,820,ERR=630,END=640) (TITLE(I),I=1,120)
      K=1
      ITIME=0
      WRITE (6,830)
 10   ICRD=23
C
C   PLOT3 DATA CARDS B3&B4:   KH,ITYPE,NAME,IEND
C   KH:      INSTRUMENT NUMBER
C   ITYPE:   INSTRUMENT TYPE
C   NAME:    DESCRIPTION OF INSTRUMENT
C   IEND:    DIRECTS END OF INSTRUMENT CARDS
C
      READ (5,840,ERR=630,END=640) KH(K),ITYPE(K),(NAME(K,J),J=1,6),(IN(
     2I),I=1,60),IEND
      IF (IEND.EQ.999) GO TO 30
      IF (ITYPE(K).EQ.1.AND.ITIME.NE.0) GO TO 20
      IF (ITYPE(K).EQ.1) ITIME=K
      WRITE (6,850) KH(K),ITYPE(K),(NAME(K,J),J=1,6),(IN(I),I=1,60),K
      IF (MOD(K,5).EQ.0) WRITE (6,800)
      IF (MOD(K-35,40).EQ.0) WRITE (6,660) (TITLE(I),I=1,120)
      IF (MOD(K-35,40).EQ.0) WRITE (6,830)
      K=K+1
      IF (K.LE.NCOL) GO TO 10
C   TOO MANY INSTRUMENTS, KILL THE RUN
      WRITE (6,860) K
      CALL EREXIT
      STOP
 20   WRITE (6,810) KH(K)
      CALL EREXIT
      STOP
 30   MAXC=K-1
C   READ IN DATA
      CALL REDCIN (NROW,NCOL,REED,KH,0,0,MAXR,ITIME,0,0,ISKIP)
C   PRINT OUT DATA
      IF (NPRT.NE.0) CALL PRINT (NROW,NCOL,REED,KH,ITYPE,NAME,ITIME,MAXR
     2,MAXC,TITLE)
      WRITE (6,750) (TITLE(I),I=1,120)
      IPPLOT=0
      IP=0
      IPTYP=0
      NGRAPH=0
 40   ICRD=41
C
C   PLOT3 DATA CARD D1:   IN
C   IN:      SPECIFIES INSTRUMENTS TO BE PLOTTED
C
      READ (5,870,ERR=630,END=640) (IN(I),I=1,80)
```

```
      K=0
      DO 100 I=1,80
      IF (IN(I).EQ.NCOM) IPPLOT=1
      IF (IN(I).EQ.NS.AND.IN(I+1).EQ.NP) IPTYP=1
      IF (IN(I).EQ.NL.AND.IN(I+1).EQ.NP) IPTYP=-1
      DO 50 J=1,10
      IF (IN(I).EQ.CHAR(J)) GO TO 90
50    CONTINUE
      IF (K.EQ.0) GO TO 100
      IP=IP+1
C   CALCULATES INSTRUMENT NUMBER.
      L=0
      DO 60 J=1,K
      M=IABS(K-J)
60    L=L+IC(J)*10**M
C   LOCATE COLUMN INSTRUMENT IS STORED IN.
      DO 70 J=1,MAXC
      IF (KH(J).EQ.L) IPC(IP)=J
70    CONTINUE
      IF (IPC(IP).NE.0) GO TO 80
      WRITE (6,670) L,IN
80    K=0
      GO TO 100
90    K=K+1
      IC(K)=J-1
100   CONTINUE
      IF (IN(80).EQ.NSEMI) GO TO 40
      IF (IP.EQ.0) GO TO 620
C   DETERMINE AXIS LIMITS SO THAT READABLE VALUES ARE PRINTED ON AXES
      IOUT=0
      ICRD=42
C
C   PLOT3 DATA CARD D2:   XL,XH,YL,YH
C   XL:      X-AXIS LOWER LIMIT
C   XH:      X-AXIS UPPER LIMIT
C   YL:      Y-AXIS LOWER LIMIT
C   YH:      Y-AXIS UPPER LIMIT
C
      READ (5,880,ERR=630,END=640) XL,XH,YL,YH
      IF (IPTYP.NE.0) GO TO 140
      DELY=(YH-YL)/YSIZE
      N=0
      K=0
110   IF (DELY.GT..8) GO TO 120
      DELY=DELY*10.
      K=K+1
      GO TO 110
120   N=N-K-1
      IY=DELY*10.
130   IY=IY/10.
```

```
      N=N+1
      IF (IY.GT.8.) GO TO 130
      DELY=1.
      IF (IY.GT.1.) DELY=2.
      IF (IY.GT.2.) DELY=4.
      IF (IY.GT.4.) DELY=5.
      IF (IY.GT.5.) DELY=8.
      DELY=DELY*10.**N
      YH=YL+DELY*YSIZE
      YPOW=N
      GO TO 150
140   IF (YL.EQ.0.0.OR.YH.EQ.0.0) GO TO 630
      DELY=FLOAT(INT(ALOG10(YH/YL)+0.99))/YSIZE
      YH=YL*10.**(DELY*YSIZE)
150   IF (IPTYP.EQ.-1) GO TO 190
      DELX=(XH-XL)/XSIZE
      N=0
      K=0
160   IF (DELX.GT..8) GO TO 170
      DELX=DELX*10.
      K=K+1
      GO TO 160
170   N=N-K-1
      IX=DELX*10.
180   IX=IX/10.
      N=N+1
      IF (IX.GT.8.) GO TO 180
      DELX=1.
      IF (IX.GT.1.) DELX=2.
      IF (IX.GT.2.) DELX=4.
      IF (IX.GT.4.) DELX=5.
      IF (IX.GT.5.) DELX=8.
      DELX=DELX*10.**N
      XH=XL+DELX*XSIZE
      XPOW=N
      GO TO 200
190   IF (XL.EQ.0.0.OR.XH.EQ.0.0) GO TO 630
      DELX=FLOAT(INT(ALOG10(XH/XL)+0.99))/XSIZE
      XH=XL*10.**(DELX*XSIZE)
200   ICRD=43
C
C   PLOT3 DATA CARD D3:   IFIG,IHEAD
C   IFIG:   FIGURE NUMBER (IF ANY)
C   IHEAD:  TITLE OF PLOT (2  39 CHARACTER TITLE LINES MAXIMUM)
C
      READ (5,890,ERR=630,END=640) IFIG,IHEAD
      YTPOS=-0.95
      IF (IFIG.EQ.0) GO TO 210
C   PUT TITLE WITH FIGURE NUMBER ON GRAPH
      XTPOS=XSIZE/2.-3.57
```

```
            K=IFIG/10
            FTIT(8)=1H
            IF (K.NE.0) FTIT(8)=CHAR(K+1)
            K=IFIG-K
            FTIT(9)=CHAR(K+1)
            CALL TEXT (XTPOS,YTPOS,.14,FTIT,0.,10)
            XTPOS=XSIZE/2.-1.89
            CALL TEXT (XTPOS,YTPOS,.14,IHEAD,0.,39)
            YTPOS=-1.15
            CALL TEXT (XTPOS,YTPOS,.14,IHEAD(40),0.,39)
            GO TO 220
      C   PUT TITLE WITHOUT FIGURE NUMBER ON GRAPH
      210   XTPOS=XSIZE/2.-2.73
            CALL TEXT (XTPOS,YTPOS,.14,IHEAD,0.,39)
            YTPOS=-1.1
            CALL TEXT (XTPOS,YTPOS,.14,IHEAD(40),0.,39)
      220   ICRD=44
      C
      C   PLOT3 DATA CARD D4:   IBUFF, JBUFF
      C   IBUFF:  VERTICAL AXIS TITLE
      C   JBUFF:  HORIZONTAL AXIS TITLE
      C
            READ (5,900,ERR=630,END=640) (IBUFF(J),J=1,40),(JBUFF(J),J=1,40)
      C   CREATE THE AXES OF THE GRAPH USING THE TITLES SUPPLIED
      C   DRAW Y AXIS
            IF (IPTYP.NE.0) GO TO 290
            YTPOS=0.
            CALL PLOT (0.,0.,3)
      230   YTPOS=MIN(YTPOS+1.,YSIZE)
            CALL PLOT (0.,YTPOS,2)
            IF (AMOD(YTPOS,1.).NE.0.) GO TO 240
            CALL PLOT (.125,YTPOS,2)
            CALL PLOT (0.,YTPOS,3)
      240   IF (YTPOS.LT.YSIZE) GO TO 230
      C   FIND NUMBER OF CHARACTERS IN AXIS LABEL NUMBERS
            YDIG=0
            YP=10.*AMAX1(ABS(YL),ABS(YH))
            IF (YP.LT.10.) GO TO 260
      250   YP=YP/10.
      260   YDIG=YDIG+1
            IF (YP.GE.10.) GO TO 250
            IF (YPOW.GT.0) YPOW=0
            NDEC=IABS(YPOW)
            IF (NDEC.EQ.0) NDEC=-1
            IF (NDEC.NE.-1) YDIG=YDIG+1
            XTPOS=FLOAT(-(IABS(YPOW)+YDIG+1))*0.14
      C   PUT NUMBERS ON
            YTPOS=-1.0
            YNUM=YL
      270   YTPOS=AMIN1(YTPOS+1.,YSIZE)
```

```
           IF (AMOD(YTPOS,1.).NE.0..AND.YTPOS.EQ.YSIZE) GO TO 280
           CALL NUMBER (XTPOS,YTPOS-0.07,.14,YNUM,0.,NDEC)
 280       YNUM=YNUM+DELY
           IF (YTPOS.LT.YSIZE) GO TO 270
 C    AXIS TITLE
           YTPOS=YSIZE/2.-2.8
           XTPOS=XTPOS-.20
           CALL TEXT (XTPOS,YTPOS,.14,IBUFF,90.,40)
           GO TO 300
 290       CALL LGAXIS (0.,0.,1H ,0,YSIZE,90.,YL,DELY)
           YTPOS=YSIZE/2.-2.8
           CALL TEXT (-0.5,YTPOS,.14,IBUFF,90.,40)
 C    DRAW X AXIS
 300       IF (IPTYP.EQ.-1) GO TO 370
           XTPOS=0.
           CALL PLOT (0.,0.,3)
 310       XTPOS=AMIN1(XTPOS+1.,XSIZE)
           CALL PLOT (XTPOS,0.,2)
           IF (AMOD(XTPOS,1.).NE.0.) GO TO 320
           CALL PLOT (XTPOS,.125,2)
           CALL PLOT (XTPOS,0.,3)
 320       IF (XTPOS.LT.XSIZE) GO TO 310
 C    PUT NUMBERS BELOW TIC MARKS ON X AXIS
           XTPOS=-1.
           XNUM=XL
 330       XTPOS=AMIN1(XTPOS+1.,XSIZE)
           IF (AMOD(XTPOS,1.).NE.0..AND.XTPOS.EQ.XSIZE) GO TO 360
 C    FIND NUMBER OF CHARACTERS IN AXIS LABEL NUMBER
           XDIG=0
           XP=10.*XNUM
           IF (XP.LT.10.) GO TO 350
 340       XP=XP/10.
 350       XDIG=XDIG+1
           IF (XP.GE.10.) GO TO 340
           IF (XPOW.GT.0) XPOW=0
           NDEC=IABS(XPOW)
           IF (NDEC.EQ.0) NDEC=-1
           IF (NDEC.EQ.-1) XDIG=XDIG-1
           XTPOS=XTPOS-FLOAT((IABS(XPOW)+XDIG+1))*0.07
           CALL NUMBER (XTPOS,-.19,.14,XNUM,0.,NDEC)
           XTPOS=XTPOS+FLOAT((IABS(XPOW)+XDIG+1))*0.07
 360       XNUM=XNUM+DELX
           IF (XTPOS.LT.XSIZE) GO TO 330
 C    ADD AXIS TITLE
           XTPOS=XSIZE/2.-2.8
           CALL TEXT (XTPOS,-.53,.14,JBUFF,0.,40)
           GO TO 380
 370       CALL LGAXIS (0.,0.,1H ,-1,XSIZE,0.,XL,DELX)
           XTPOS=XSIZE/2.-2.8
           CALL TEXT (XTPOS,-0.53,.14,JBUFF,0.,40)
```

```
  380 IF (NAXIS.EQ.0) GO TO 430
C   PUT AXES ON THE OTHER TWO SIDES OF THE GRAPH, Y AXIS FIRST
      YTPOS=0.
      CALL PLOT (XSIZE,0.,3)
  390 YTPOS=AMIN1(YTPOS+1.,YSIZE)
      CALL PLOT (XSIZE,YTPOS,2)
      IF (AMOD(YTPOS,1.).NE.0.) GO TO 400
      CALL PLOT (XSIZE-.125,YTPOS,2)
      CALL PLOT (XSIZE,YTPOS,3)
  400 IF (YTPOS.LT.YSIZE) GO TO 390
C  DRAW X AXIS
      XTPOS=0.
      CALL PLOT (0.,YSIZE,3)
  410 XTPOS=AMIN1(XTPOS+1.,XSIZE)
      CALL PLOT (XTPOS,YSIZE,2)
      IF (AMOD(XTPOS,1.).NE.0.) GO TO 420
      CALL PLOT (XTPOS,YSIZE-.125,2)
      CALL PLOT (XTPOS,YSIZE,3)
  420 IF (XTPOS.LT.XSIZE) GO TO 410
  430 NCRV=IP-1
      IF (IPPLOT.EQ.1) NCRV=IP/2
      DO 540 I=1,NCRV
      IF (IPPLOT.EQ.1) GO TO 440
      IXX=IPC(1)
      IYY=IPC(I+1)
      GO TO 450
  440 IXX=IPC(2*I-1)
      IYY=IPC(2*I)
  450 NPT=0
      MAXROW=MAXR(IYY)
      DO 470 L=1,MAXROW
      IF (REED(L,IXX).GE.XL.AND.REED(L,IXX).LE.XH.AND.REED(L,IYY).GE.YL.
     2AND.REED(L,IYY).LE.YH) GO TO 460
      IOUT=IOUT+1
      IF (NOUT.EQ.0) GO TO 470
  460 NPT=NPT+1
      X(NPT)=REED(L,IXX)
      Y(NPT)=REED(L,IYY)
  470 CONTINUE
      ISYM=I-1
      X(NPT+1)=XL
      X(NPT+2)=DELX
      Y(NPT+1)=YL
      Y(NPT+2)=DELY
      IF (NPT.EQ.0) GO TO 520
      IF (IPTYP.EQ.0) GO TO 480
      CALL LGLINE (X,Y,NPT,1,NPNT,ISYM,IPTYP)
      GO TO 520
C   SMOOTH DATA IF NSMTH IS NOT EQUAL TO ZERO
  480 IF (NSMTH.EQ.0) GO TO 510
```

```
      DO 500 L=1,NPT
      IPTH=NPT-L+1
      IPTL=MAX0(NPT-L-NSMTH-1,1)
      Y(IPTH)=Y(IPTH)/FLOAT((IPTH-IPTL+1))
      IF (IPTH.EQ.1) GO TO 500
      IPTHM1=IPTH-1
      DO 490 LL=IPTL,IPTHM1
      Y(IPTH)=Y(IPTH)+Y(LL)/FLOAT((IPTH-IPTL+1))
  490 CONTINUE
  500 CONTINUE
  510 CALL LINE (X,Y,NPT,1,NPNT,ISYM)
  520 DO 530 J=1,6
  530 IPN(I,J)=NAME(IYY,J)
  540 CONTINUE
      IF (NLEG.EQ.0) GO TO 550
C
C   PLOT3 DATA CARD D5:   IS,ISYM,XTPOS,YTPOS,IHEAD
C   IS:      PLOTTING SYMBOL FLAG
C   ISYM:    PLOTTING SYMBOL PREFIXED TO LEGEND IF IS IS NON-ZERO
C   XTPOS:   X POSITION OF LEGEND
C   YTPOS:   Y POSITION OF LEGEND
C   IHEAD:   LEGEND
C
      READ (5,650,END=630,ERR=640) IS,ISYM,XTPOS,YTPOS,(IHEAD(I),I=1,60)
      IF (IS.EQ.0.AND.ISYM.EQ.0.AND.XTPOS.EQ.0.0.AND.YTPOS.EQ.0.0.AND.IH
     2EAD(1).EQ.1H ) GO TO 550
      IF (IS.NE.0) CALL SYMBOL (XTPOS-.105,YTPOS+.035,.07,ISYM,0.,-1)
      CALL TEXT (XTPOS,YTPOS,.07,IHEAD,0.,60)
C   PUT A LEGEND BLOCK IN BOTTOM RIGHT HAND CORNER
  550 CALL FACTOR (1.0)
      XTPOS=XSIZE*SCALE+0.5
      YTPOS=0.1*FLOAT(NCRV/2+MOD(NCRV,2))+0.03
      DO 580 I=1,NCRV,2
      YTPOS=YTPOS-0.10
      CALL SYMBOL (XTPOS,YTPOS+.035,.07,I-1,0.,-1)
      DO 560 J=1,6
  560 IN(J)=IPN(I,J)
      CALL TEXT (XTPOS+.14,YTPOS,.07,IN,0.,6)
      IF (I.EQ.NCRV) GO TO 590
      CALL SYMBOL (XTPOS+0.7,YTPOS+.035,.07,I,0.,-1)
      DO 570 J=1,6
  570 IN(J)=IPN(I+1,J)
      CALL TEXT (XTPOS+0.84,YTPOS,.07,IN,0.,6)
  580 CONTINUE
  590 XTPOS=XSIZE*SCALE+0.4
      YTPOS=0.1*FLOAT(NCRV/2+MOD(NCRV,2))+0.03
      YLEN=0.1*FLOAT(NCRV/2+MOD(NCRV,2))+0.03
      CALL PLOT (XTPOS,YTPOS,3)
      CALL PLOT (XTPOS+1.425,YTPOS,2)
      CALL PLOT (XTPOS+1.425,YTPOS-YLEN,2)
```

```
          CALL PLOT (XTPOS,YTPOS-YLEN,2)
          CALL PLOT (XTPOS,YTPOS,2)
          CALL FACTOR (SCALE)
C     WRITE OUT INFORMATION ABOUT THE GRAPH WE JUST FINISHED
          DO 610 I=1,IP
          J=IPC(I)
          IPC(I)=KH(J)
          DO 600 K=1,6
  600     IPN(I,K)=NAME(J,K)
  610     CONTINUE
          IF (IPPLOT.EQ.0) WRITE (6,760) ((IPN(I,J),J=1,6),IPC(I),I=1,IP)
          IF (IPPLOT.EQ.1) WRITE (6,680) ((IPN(I,J),J=1,6),IPC(I),I=1,IP)
          WRITE (6,770) XL,DELX,XH,YL,DELY,YH
          WRITE (6,780) JBUFF,IBUFF
          IF (IFIG.EQ.0) WRITE (6,790) IHEAD
          IF (IFIG.NE.0) WRITE (6,910) IFIG,IHEAD
          IF (IOUT.NE.0) WRITE (6,690)
          IF (YSIZE*SCALE+1.5.GT.10.75) WRITE (6,700)
          XMOVE=XSIZE+7.0
          CALL PLOT (XMOVE,0.,-3)
          IPPLOT=0
          IP=0
          IPTYP=0
          NGRAPH=NGRAPH+1
          IF (MOD(NGRAPH,5).EQ.0) WRITE (6,750) (TITLE(I),I=1,120)
          GO TO 40
  620     IF (NSETS.LE.1) CALL PLOT (0.,0.,999)
          RETURN
  630     CALL RWERR
  640     CALL RWEOF
C
C
  650     FORMAT (I1,I2,2F6.0,60A1)
  660     FORMAT (1H1,120A1)
  670     FORMAT (31H0NO FIND FOR INSTRUMENT NUMBER ,I6,
         2 14H ERROR ON THE 19HFOLLOWING PLOT CARD,/,1X,80A1)
  680     FORMAT (17H0CALCOMP PLOT OF ,2(6A1,2H (,I6,4H) , ,6A1,2H (,I6,
         2 5H)    ),20(/,T18,2(6A1,2H (,I6,4H) , ,6A1,2H (,I6,5H)    )))
  690     FORMAT (29H0POINTS OUT OF BOUNDS ON PLOT)
  700     FORMAT (48H0PAPER WIDER THAN 12 INCHES MUST BE USED ON PLOT)
  710     FORMAT (25H0PLOT3 CONTROL PARAMETERS)
  720     FORMAT (16I5)
  730     FORMAT (8H0  NPRT:,I5,2X,7H NSMTH:,I5,2X,7H  NOUT:,I5,2X,7H NAXIS:
         2,I5,2X,7H  NPNT:,I5,2X,7H  NLEG:,I5)
  740     FORMAT (8H  XSIZE:,F5.2,2X,7H YSIZE:,F5.2,2X,7H SCALE:,F5.2)
  750     FORMAT (1H1,120A1,/,23H0CALCOMP PLOTS PREPARED)
  760     FORMAT (17H0CALCOMP PLOT OF ,6A1,2H (,I6,9H)  VERSUS,3(2X,6A1,2H (
         2,I6,1H)),7(/,T37,3(2X,6A1,2H (,I6,1H))))
  770     FORMAT (21H X-AXIS LOWER LIMIT: ,G8.3,21H    DELX (PER INCH): ,
         2 G8.3,14H UPPER LIMIT: ,G8.3,/,21H Y-AXIS LOWER LIMIT: ,G8.3,
```

```
          3 21H    DELY (PER INCH): ,G8.3,14H UPPER LIMIT: ,G8.3)
    780    FORMAT (15H X-AXIS TITLE: ,40A1,/,15H Y-AXIS TITLE: ,40A1)
    790    FORMAT (14H GRAPH TITLE: ,39A1,/,14X,39A1)
    800    FORMAT (1H )
    810    FORMAT (41H0ONLY ONE INSTRUMENT OF TYPE 1 IS ALLOWED,/,
          2 35H SECOND FOUND AT INSTRUMENT NUMBER ,I3)
    820    FORMAT (80A1/40A1)
    830    FORMAT (22H0 TABLE OF INSTRUMENTS/,3X,
          2 41HKH ITYPE I.D.,      INSTRUMENT DESCRIPTION,T87,6HCOLUMN,/,1H ,
          3 92(1H-),/)
    840    FORMAT (I6,I2,66A1,3X,I3)
    850    FORMAT (1X,I6,I3,3X,6A1,2X,60A1,T87,I6)
    860    FORMAT (42H0COLUMN STORAGE LIMITS EXCEEDED AT COLUMN ,I4,/,
          2 31H CHECK PARAMETERS NCOL AND NROW)
    870    FORMAT (80A1)
    880    FORMAT (4F10.0)
    890    FORMAT (I2,78A1)
    900    FORMAT (80A1)
    910    FORMAT (21H GRAPH TITLE: FIGURE ,I2,3H:  ,39A1,/,26X,39A1)
    C
           END
```

```
      SUBROUTINE PLOT4 (NROW,NCOL,REED,KH,ITYPE,NAME,MAXR,MCNL,MAXCNL,
     2 ISKIP)
C
C     *****************************************************************
C
      COMMON /ERRORS/ ICRD,ISEG
      DIMENSION REED(NROW,NCOL),KH(NCOL),ITYPE(NCOL),NAME(NCOL,6),
     2 MAXR(NCOL),TITLE(120),IN(80),MCNL(MAXCNL),ISKIP(320)
      DATA NF /1HF/
      ISEG=5
      ICRD=11
C
C   PLOT4 DATA CARD A1:    NPRT,NPNCH
C   NPRT:   DIRECTS PRINTING OF CONVERTED DATA
C   NPNCH:  DIRECTS PUNCHING OF CONVERTED DATA
C
      READ (5,110,ERR=80,END=90) NPRT,NPNCH
      ICRD=21
C   PLOT4 DATA CARDS B1 & B2:    TITLE
C   TITLE:  SPECIFIES TITLE OF EXPERIMENT PRINTED AT TOP OF EACH PAGE
C
      READ (5,120,ERR=80,END=90) (TITLE(I),I=1,120)
      WRITE (6,130)
      WRITE (6,140) NPRT,NPNCH
      K=1
      WRITE (6,160)
   10 ICRD=23
C
C   PLOT4 DATA CARDS B3:    ITYPE,KH,NAME,IEND
C   ITYPE:  TEST NUMBER FOR THIS INSTRUMENT
C   KH:     INSTRUMENT (CHANNEL) NUMBER FOR THIS INSTRUMENT
C   NAME:   DESCRIPTION OF INSTRUMENT
C   IEND:   DIRECTS END OF INSTRUMENT CARDS
C
      READ (5,150,ERR=80,END=90) ITYPE(K),KH(K),(NAME(K,J),J=1,6),(IN(I)
     2,I=1,60),IEND
      IF (IEND.EQ.999) GO TO 20
      WRITE (6,170) KH(K),ITYPE(K),(NAME(K,J),J=1,6),(IN(J),J=1,60),K
      IF (MOD(K,5).EQ.0) WRITE (6,180)
      IF (MOD(K-35,40).EQ.0) WRITE (6,100) (TITLE(I),I=1,120)
      IF (MOD(K-35,40).EQ.0) WRITE (6,160)
      K=K+1
      IF (K.LE.NCOL) GO TO 10
C   TOO MANY INSTRUMENTS, KILL THE RUN
      WRITE (6,250) K
      CALL EREXIT
      STOP
   20 MAXC=K-1
   30 ICRD=31
C
```

```
      C   PLOT4 DATA CARD C1:    ITEST
      C   ITEST:  TEST NUMBER OF THIS DATA SET
      C
            READ (5,190,ERR=80,END=90) (IN(I),I=1,6),ITEST
            IF (IN(1).EQ.NF) GO TO 70
            DO 40 I=1,MAXC
            IF (ITYPE(I).NE.ITEST) KH(I)=KH(I)+1000*ITYPE(I)
      40    CONTINUE
            CALL REDCIN (NROW,NCOL,REED,KH,0,0,MAXR,0,0,0,ISKIP)
            WRITE (6,200) ITEST
            INSTS=0
            DO 50 I=1,MAXC
            IF (ITYPE(I).NE.ITEST) GO TO 50
            IF (MAXR(I).EQ.0) GO TO 50
            INSTS=INSTS+1
            MCNL(INSTS)=ITYPE(I)*1000+KH(I)
            IF (NPNCH.EQ.0) GO TO 50
            MAXROW=MAXR(I)
            WRITE (NPNCH,220) MAXROW,MCNL(INSTS),(NAME(I,J),J=1,6)
            WRITE (NPNCH,230) (REED(J,I),J=1,MAXROW)
      50    CONTINUE
            WRITE (6,210) (MCNL(I),I=1,INSTS)
            DO 60 I=1,MAXC
            IF (ITYPE(I).NE.ITEST) KH(I)=KH(I)-1000*ITYPE(I)
      60    CONTINUE
            GO TO 30
      70    IF (NPNCH.EQ.0) RETURN
            WRITE (NPNCH,240)
            END FILE NPNCH
            REWIND NPNCH
            IF (NPRT.NE.0) CALL PRINT (NROW,NCOL,REED,KH,ITYPE,NAME,0,MAXR,MAX
           2C,TITLE)
            RETURN
      80    CALL RWERR
            STOP
      90    CALL RWEOF
            STOP
      C
      C
      100   FORMAT (1H1,120A1)
      110   FORMAT (16I5)
      120   FORMAT (80A1/40A1)
      130   FORMAT (27H0  PLOT4 CONTROL PAPAMETERS,/)
      140   FORMAT (8H   NPRT:,I5,2X,7H NPNCH:,I5)
      150   FORMAT (I3,I3,2X,66A1,3X,I3)
      160   FORMAT (22H0 TABLE OF INSTRUMENTS/,3X,
           2 44HKH    TEST    I.D.    INSTRUMENT DESCRIPTION,T87,6HCOLUMN,/,
           3 1H ,92(1H-),/)
      170   FORMAT (2X,I3,3X,I3,5X,6A1,2X,60A1,T87,I6)
      180   FORMAT (1H )
```

```
      190   FORMAT (6A1,I3)
      200   FORMAT (/,7HOTEST: ,I3,10X,15HNEW INSTRUMENTS,/)
      210   FORMAT (10(I6,6X))
      220   FORMAT (I6,I6,6A1)
      230   FORMAT (8E10.5)
      240   FORMAT (77X,3H999)
      250   FORMAT (36HOROW STORAGE LIMITS EXCEEDED AT ROW ,I6,/,1X,
           2 30HCHECK PARAMETERS NROW AND NCOL)
      C
            END
```

```
      SUBROUTINE DEFINE (INTYPE,ICHRS,INDAS,IREAD,ITIME,IEOR,IEOF,MEDIA,
     2 TITLE,INSKIP,ISKIP,NCNL,NTFORM)
C
C     *****************************************************************
C
      COMMON /ERRORS/ ICRD,ISEG
      INTEGER READ,TIME,EOR,EOF,DAS,TFORM
      DIMENSION ICHRS(320),INDAS(22),IN(80),INPUT(10,38),READ(10,56),
     2 TIME(10,68),EOR(10,16),EOF(10,28),DAS(10,22),ICHARS(26),
     3 TITLE(120),ISKIP(320),TFORM(1,6)
      DATA (INPUT(1,J),J=1,38) /1HI,1HN,1HP,1HU,1HT,1H=,1HT,1HA,1HP,1HE,
     2 1H,,1H ,1HT,1HA,1HP,1HE,1H ,1HF,1HO,1HR,1HM,1HA,1HT,1H=,1H7,1HT,
     3 1HR,1HA,1HC,1HK,8*1H /
      DATA (INPUT(2,J),J=1,38) /38*1H /
      DATA (INPUT(3,J),J=1,38) /1HI,1HN,1HP,1HU,1HT,1H=,1HC,1HA,1HR,1HD,
     2 1H ,1HI,1HM,1HA,1HG,1HE,1HS,1H,,1H ,1HC,1HH,1HA,1HN,1HN,1HE,1HL,
     3 1HS,1H ,1HP,1HE,1HR,1H ,1HL,1HI,1HN,1HE,1H=,1H6/
      DATA (INPUT(4,J),J=1,38) /38*1H /
      DATA (INPUT(5,J),J=1,38) /1HI,1HN,1HP,1HU,1HT,1H=,1HC,1HA,1HR,1HD,
     2 1H ,1HI,1HM,1HA,1HG,1HE,1HS,1H,,1H ,1HC,1HH,1HA,1HN,1HN,1HE,1HL,
     3 1HS,1H ,1HP,1HE,1HR,1H ,1HL,1HI,1HN,1HE,1H=,1H4/
      DATA (INPUT(6,J),J=1,38) /1HI,1HN,1HP,1HU,1HT,1H=,1HT,1HA,1HP,1HE,
     2 1H,,1H ,1HT,1HA,1HP,1HE,1H ,1HF,1HO,1HR,1HM,1HA,1HT,1H=,1H7,1HT,
     3 1HR,1HA,1HC,1HK,8*1H /
      DATA (INPUT(7,J),J=1,38) /1HI,1HN,1HP,1HU,1HT,1H=,1HC,1HA,1HR,1HD,
     2 1H ,1HI,1HM,1HA,1HG,1HE,1HS,1H,,1H ,1HC,1HH,1HA,1HN,1HN,1HE,1HL,
     3 1HS,1H ,1HP,1HE,1HR,1H ,1HL,1HI,1HN,1HE,1H=,1H4/
      DATA (INPUT(8,J),J=1,38) /1HI,1HN,1HP,1HU,1HT,1H=,1HT,1HA,1HP,1HE,
     2 1H,,1H ,1HT,1HA,1HP,1HE,1H ,1HF,1HO,1HR,1HM,1HA,1HT,1H=,1H7,1HT,
     3 1HR,1HA,1HC,1HK,8*1H /
      DATA (INPUT(9,J),J=1,38) /1HI,1HN,1HP,1HU,1HT,1H=,1HC,1HA,1HR,1HD,
     2 1H ,1HI,1HM,1HA,1HG,1HE,1HS,1H,,1H ,1HC,1HH,1HA,1HN,1HN,1HE,1HL,
     3 1HS,1H ,1HP,1HE,1HR,1H ,1HL,1HI,1HN,1HE,1H=,1H5/
      DATA (INPUT(10,J),J=1,38) /1HI,1HN,1HP,1HU,1HT,1H=,1HC,1HA,1HR,1HD
     2,1H ,1HI,1HM,1HA,1HG,1HE,1HS,1H,,1H ,1HC,1HH,1HA,1HN,1HN,1HE,1HL,
     3 1HS,1H ,1HP,1HE,1HR,1H ,1HL,1HI,1HN,1HE,1H=,1H5/
      DATA (READ(1,J),J=1,56) /1HP,1HE,1HA,1HD,1HI,1HN,1HG,1H=,1H3,1H*,
     2 1H(,1HC,1H),1H(,1H+,1H1,1H+,1H+,1H-,1H2,1H-,1H-,1HO,1H9,1HO,1HO,
     3 1H),1H5,1H*,1H(,1HV,1H),1H(,1HE,1H),1H2,1H*,1H(,1HK,1H ,1H),1H ,
     4 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (READ(2,J),J=1,56) /1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     3 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     4 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (READ(3,J),J=1,56) /1HR,1HE,1HA,1HD,1HI,1HN,1HG,1H=,1H3,1H*,
     2 1H(,1HC,1H),1H(,1H+,1H1,1H+,1H+,1H-,1H2,1H-,1H-,1HO,1H9,1HO,1HO,
     3 1H),1H5,1H*,1H(,1HV,1H),1H(,1HE,1H),1H2,1H*,1H(,1HK,1H ,1H),1H ,
     4 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (READ(4,J),J=1,56) /1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
```

```
     3 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     4 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (READ(5,J),J=1,56) /1HR,1HE,1HA,1HD,1HI,1HN,1HG,1H=,1H(,1HN,
     2 1H),1H(,1HK,1H ,1H),1H3,1H*,1H(,1HC,1H),1H(,1HK,1H ,1H),1H(,1HR,
     3 1HO,1H ,1H),1H5,1H*,1H(,1HR,1H),1H2,1H*,1H(,1HA,1H),1H(,1HK,1H ,
     4 1H),1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (READ(6,J),J=1,56) /1HR,1HE,1HA,1HD,1HI,1HN,1HG,1H=,1H(,1HK,
     2 1HC,1H),1H3,1H*,1H(,1HC,1H),1H(,1HK,1H ,1H),1H(,1HK,1H ,1HO,1H*,
     3 1HN,1H),1H(,1H+,1H ,1H-,1H-,1H),1H6,1H*,1H(,1HR,1HO,1H ,1H),1H(,
     4 1HK,1H ,1H),1H2,1H*,1H(,1HA,1H),1H2,1H*,1H(,1HK,1H ,1H)/
      DATA (READ(7,J),J=1,56) /1HR,1HE,1HA,1HD,1HI,1HN,1HG,1H=,1H(,1HK,
     2 1HC,1H),1H3,1H*,1H(,1HC,1H),1H(,1HK,1H ,1H),1H(,1HK,1H ,1HO,1H*,
     3 1HN,1H),1H(,1H+,1H ,1H-,1H-,1H),1H6,1H*,1H(,1HR,1HO,1H ,1H),1H(,
     4 1HK,1H ,1H),1H2,1H*,1H(,1HA,1H),1H2,1H*,1H(,1HK,1H ,1H)/
      DATA (READ(8,J),J=1,56) /1HR,1HE,1HA,1HD,1HI,1HN,1HG,1H=,1H3,1H*,
     2 1H(,1HC,1H),1H(,1HK,1H ,1H),1H(,1H+,1H+,1H-,1H-,1HO,1H ,1H),1H5,
     3 1H*,1H(,1HV,1HO,1H ,1H),1H(,1HV,1HK,1H ,1H),1H3,1H*,1H(,1HA,1H),
     4 1H(,1HE,1HO,1H ,1H),1H(,1HK,1HX,1H),1H ,1H ,1H ,1H ,1H /
      DATA (READ(9,J),J=1,56) /1HR,1HE,1HA,1HD,1HI,1HN,1HG,1H=,1H3,1H*,
     2 1H(,1HC,1H),1H(,1HK,1H ,1H),1H(,1H+,1H+,1H-,1H-,1HO,1H ,1H),1H5,
     3 1H*,1H(,1HV,1HO,1H ,1H),1H(,1HV,1HK,1H ,1H),1H3,1H*,1H(,1HA,1H),
     4 1H(,1HE,1HO,1H ,1H),1H(,1HK,1HX,1H),1H ,1H ,1H ,1H ,1H /
      DATA (READ(10,J),J=1,56) /1HR,1HE,1HA,1HD,1HI,1HN,1HG,1H=,1H(,1HO,
     2 1H*,1HK,1H ,1H),1H3,1H*,1H(,1HC,1H),1H(,1H+,1H ,1H-,1H-,1H),1H6,
     3 1H*,1H(,1HR,1H),1H(,1HA,1H),1H2,1H*,1H(,1HK,1H ,1H),1H ,1H ,1H ,
     4 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (TIME(1,J),J=1,68) /1HT,1HI,1HM,1HE,1H=,1H3,1H*,1H(,1HS,1H),
     2 1H(,1HK,1H6,1H),1H6,1H*,1H(,1HS,1H),1H2,1H*,1H(,1HK,1H ,1H),1H ,
     3 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     4 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     5 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (TIME(2,J),J=1,68) /1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     3 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     4 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     5 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (TIME(3,J),J=1,68) /1HT,1HI,1HM,1HE,1H=,1H3,1H*,1H(,1HS,1H),
     2 1H(,1HK,1H6,1H),1H6,1H*,1H(,1HS,1H),1H2,1H*,1H(,1HK,1H ,1H),1H ,
     3 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     4 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     5 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (TIME(4,J),J=1,68) /1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     3 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     4 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     5 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (TIME(5,J),J=1,68) /1HT,1HI,1HM,1HE,1H=,1H2,1H*,1H(,1HH,1H),
     2 1H(,1HK,1H:,1H),1H2,1H*,1H(,1HM,1H),1H7,1H5,1H*,1H(,1HK,1H ,1H),
     3 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     4 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
```

```
     5 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (TIME(6,J),J=1,68) /1HT,1HI,1HM,1HE,1H=,1H4,1H*,1H(,1HK,1H-,
     2 1HN,1H),1H2,1H*,1H(,1HK,1H ,1H),1H2,1H*,1H(,1HD,1H),1H(,1HK,1H ,
     3 1H),1H2,1H*,1H(,1HH,1H),1H(,1HK,1H:,1H),1H2,1H*,1H(,1HM,1H),1H(,
     4 1HK,1H:,1H),1H2,1H*,1H(,1HS,1H),1H(,1HK,1H ,1H),1H(,1HK,1HH,1H),
     5 1H(,1HK,1HR,1H),1H2,1H*,1H(,1HK,1H ,1H)/
      DATA (TIME(7,J),J=1,68) /1HT,1HI,1HM,1HE,1H=,1H4,1H*,1H(,1HK,1H-,
     2 1HN,1H),1H2,1H*,1H(,1HK,1H ,1H),1H2,1H*,1H(,1HD,1H),1H(,1HK,1H ,
     3 1H),1H2,1H*,1H(,1HH,1H),1H(,1HK,1H:,1H),1H2,1H*,1H(,1HM,1H),1H(,
     4 1HK,1H:,1H),1H2,1H*,1H(,1HS,1H),1H(,1HK,1H ,1H),1H(,1HK,1HH,1H),
     5 1H(,1HK,1HR,1H),1H2,1H*,1H(,1HK,1H ,1H)/
      DATA (TIME(8,J),J=1,68) /1HT,1HI,1HM,1HE,1H=,1H3,1H*,1H(,1HD,1H),
     2 1H2,1H*,1H(,1HH,1H),1H(,1HK,1H:,1H),1H2,1H*,1H(,1HM,1H),1H(,1HK,
     3 1H:,1H),1H2,1H*,1H(,1HS,1H),1H4,1H*,1H(,1HK,1H ,1H),1H(,1HK,1HX,
     4 1H),1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     5 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (TIME(9,J),J=1,68) /1HT,1HI,1HM,1HE,1H=,1H3,1H*,1H(,1HD,1H),
     2 1H2,1H*,1H(,1HH,1H),1H(,1HK,1H:,1H),1H2,1H*,1H(,1HM,1H),1H(,1HK,
     3 1H:,1H),1H2,1H*,1H(,1HS,1H),1H4,1H*,1H(,1HK,1H ,1H),1H(,1HK,1HX,
     4 1H),1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     5 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (TIME(10,J),J=1,68) /1HT,1HI,1HM,1HE,1H=,1H(,1HK,1H ,1H),1H1,
     2 1H5,1H*,1H(,1HK,1H ,1HN,1H),1H2,1H*,1H(,1HH,1H),1H(,1HK,1H ,1H),
     3 1H2,1H*,1H(,1HM,1H),1H(,1HK,1H ,1H),1H2,1H*,1H(,1HS,1H),1H(,1HK,
     4 1H ,1H),1H(,1HK,1HH,1H),1H4,1H4,1H*,1H(,1HK,1H ,1H),1H ,1H ,1H ,
     5 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (EOR(1,J),J=1,16) /1HE,1HO,1HR,1H=,1HE,1HO,1HR,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H /
      DATA (EOR(2,J),J=1,16) /1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H /
      DATA (EOR(3,J),J=1,16) /1HE,1HO,1HR,1H=,1H(,1HK,1H ,1H),1H(,1HK,
     2 1HX,1H),1H ,1H ,1H ,1H /
      DATA (EOR(4,J),J=1,16) /1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H /
      DATA (EOR(5,J),J=1,16) /1HE,1HO,1HR,1H=,1H8,1H0,1H*,1H(,1HK,1H ,
     2 1H),1H ,1H ,1H ,1H ,1H /
      DATA (EOR(6,J),J=1,16) /1HE,1HO,1HR,1H=,1HE,1HO,1HR,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H /
      DATA (EOR(7,J),J=1,16) /1HE,1HO,1HR,1H=,1H(,1HK,1H ,1H),1H(,1HK,
     2 1HX,1H),1H ,1H ,1H ,1H /
      DATA (EOR(8,J),J=1,16) /1HE,1HO,1HR,1H=,1HE,1HO,1HR,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H /
      DATA (EOR(9,J),J=1,16) /1HE,1HO,1HR,1H=,1H(,1HK,1H ,1H),1H(,1HK,
     2 1HX,1H),1H(,1HK,1H ,1H)/
      DATA (EOR(10,J),J=1,16) /1HE,1HO,1HR,1H=,1H7,1H0,1H*,1H(,1HK,1H ,
     2 1H),1H ,1H ,1H ,1H ,1H /
      DATA (EOF(1,J),J=1,28) /1HE,1HO,1HF,1H=,1HE,1HO,1HF,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     3 1H ,1H /
      DATA (EOF(2,J),J=1,28) /1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
```

```
     2 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     3 1H ,1H /
      DATA (EOF(3,J),J=1,28) /1HE,1HO,1HF,1H=,1H(,1HK,1HF,1H),1H(,1HK,
     2 1HI,1H),1H(,1HK,1HL,1H),1H(,1HK,1HE,1H),1H(,1HK,1HN,1H),1H(,1HK,
     3 1HD,1H)/
      DATA (EOF(4,J),J=1,28) /1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     3 1H ,1H /
      DATA (EOF(5,J),J=1,28) /1HE,1HO,1HF,1H=,1H(,1HK,1HF,1H),1H(,1HK,
     2 1HI,1H),1H(,1HK,1HL,1H),1H(,1HK,1HE,1H),1H(,1HK,1HN,1H),1H(,1HK,
     3 1HD,1H)/
      DATA (EOF(6,J),J=1,28) /1HE,1HO,1HF,1H=,1HE,1HO,1HF,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     3 1H ,1H /
      DATA (EOF(7,J),J=1,28) /1HE,1HO,1HF,1H=,1H(,1HK,1HF,1H),1H(,1HK,
     2 1HI,1H),1H(,1HK,1HL,1H),1H(,1HK,1HE,1H),1H(,1HK,1HN,1H),1H(,1HK,
     3 1HD,1H)/
      DATA (EOF(8,J),J=1,28) /1HE,1HO,1HF,1H=,1HE,1HO,1HF,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     3 1H ,1H /
      DATA (EOF(9,J),J=1,28) /1HE,1HO,1HF,1H=,1H(,1HK,1HF,1H),1H(,1HK,
     2 1HI,1H),1H(,1HK,1HL,1H),1H(,1HK,1HE,1H),1H(,1HK,1HN,1H),1H(,1HK,
     3 1HD,1H)/
      DATA (EOF(10,J),J=1,28) /1HE,1HO,1HF,1H=,1H(,1HK,1HF,1H),1H(,1HK,
     2 1HI,1H),1H(,1HK,1HL,1H),1H(,1HK,1HE,1H),1H(,1HK,1HN,1H),1H(,1HK,
     3 1HD,1H)/
      DATA (DAS(1,J),J=1,22) /1HV,1HI,1HD,1HA,1HR,1H ,1H5,1H4,1H0,1H0,
     2 1H ,1HS,1HE,1HR,1HI,1HE,1HS,1H ,1H ,1H ,1H ,1H /
      DATA (DAS(2,J),J=1,22) /1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     2 1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
      DATA (DAS(3,J),J=1,22) /1HV,1HI,1HD,1HA,1HR,1H ,1H5,1H4,1H0,1H0,
     2 1H ,1HS,1HE,1HR,1HI,1HE,1HS,1H ,1H ,1H ,1H ,1H /
      DATA (DAS(4,J),J=1,22) /1HS,1HP,1HE,1HE,1HD,1H2,1H ,1HR,1HE,1HD,
     2 1HU,1HC,1HE,1HD,1H ,1HD,1HA,1HT,1HA,1H ,1H ,1H /
      DATA (DAS(5,J),J=1,22) /1HV,1HI,1HD,1HA,1HR,1H ,1HA,1HU,1HT,1HO,
     2 1HD,1HA,1HT,1HA,1H ,1HE,1HI,1HG,1HH,1HT,1H ,1H /
      DATA (DAS(6,J),J=1,22) /1HE,1HS,1HT,1HE,1HR,1HL,1HI,1HN,1HE,1H ,
     2 1HA,1HN,1HG,1HU,1HS,1H ,1HP,1HD,1H2,1H0,1H6,1H4/
      DATA (DAS(7,J),J=1,22) /1HE,1HS,1HT,1HE,1HR,1HL,1HI,1HN,1HE,1H ,
     2 1HA,1HN,1HG,1HU,1HS,1H ,1HP,1HD,1H2,1H0,1H6,1H4/
      DATA (DAS(8,J),J=1,22) /1HV,1HI,1HD,1HA,1HR,1H ,1HA,1HU,1HT,1HO,
     2 1HD,1HA,1HT,1HA,1H ,1HN,1HI,1HN,1HE,1H ,1H ,1H /
      DATA (DAS(9,J),J=1,22) /1HV,1HI,1HD,1HA,1HR,1H ,1HA,1HU,1HT,1HO,
     2 1HD,1HA,1HT,1HA,1H ,1HN,1HI,1HN,1HE,1H ,1H ,1H /
      DATA (DAS(10,J),J=1,22) /1HD,1HO,1HR,1HI,1HC,1H ,1HD,1HI,1HG,1HI,
     2 1HT,1HR,1HE,1HN,1HD,1H ,1H2,1H2,1H0,1H ,1H ,1H /
      DATA (ICHARS(I),I=1,26),NCHARS /1HN,1HA,1HC,1H+,1H-,1HV,1HR,1HE,
     2 1HO,1HK,1HD,1HH,1HM,1HS,1H(,1H),1HS,1HK,1HI,1HP,1H=,1HK,1HS,1HF,
     3 1H(,1H),16/
      DATA (TFORM(1,J),J=1,6),NFORM /1H7,1HT,1HR,1HA,1HC,1HK,1/
```

```
            DATA NI,NC,NT,NR,NE,NF,NBL,NEQU,NSTAR /1HI,1HC,1HT,1HR,1HE,1HF,1H
           2,1H=,1H*/
      C   DECODE THE DATA ACQUISITION SYSTEM SPECIFICATIONS
            IC=1
            WRITE (6,750) (TITLE(I),I=1,120)
            IF (INTYPE.EQ.0) WRITE (6,710)
            IF (INTYPE.GE.1.AND.INTYPE.LE.10) WRITE (6,720) (DAS(INTYPE,J),J=1
           2,22)
            DO 440 NCRD=1,5
            ICRD=30+NCRD
            IF (INTYPE.NE.0) GO TO 10
      C
      C   PLOT1 OR PLOT2 DATA CARDS C1 - C5:  IN
      C   IN:     SPECIFIES DATA ACQUISITION SYSTEM FORMAT
      C
            READ (5,700,END=660,ERR=670) (IN(I),I=1,80)
            GO TO 110
      C   IT'S ONE OF THE PREASSIGNED TYPES, GET IT
      10    IF (INTYPE.GE.1.AND.INTYPE.LE.10.AND.INTYPE.NE.2) GO TO 20
            WRITE (6,690)
            CALL EREXIT
            STOP
      20    DO 30 I=1,80
      30    IN(I)=1H
            DO 80 I=1,68
            IF (NCRD.NE.1) GO TO 40
            IF (I.GT.38) GO TO 90
            IN(I)=INPUT(INTYPE,I)
      40    IF (NCRD.NE.2) GO TO 50
            IF (I.GT.56) GO TO 90
            IN(I)=READ(INTYPE,I)
      50    IF (NCRD.NE.3) GO TO 60
            IN(I)=TIME(INTYPE,I)
      60    IF (NCRD.NE.4) GO TO 70
            IF (I.GT.16) GO TO 90
            IN(I)=EOR(INTYPE,I)
      70    IF (NCRD.NE.5) GO TO 80
            IF (I.GT.28) GO TO 90
            IN(I)=EOF(INTYPE,I)
      80    CONTINUE
      90    DO 100 J=1,22
      100   INDAS(J)=DAS(INTYPE,J)
            IF (INTYPE.EQ.4) GO TO 450
      110   WRITE (6,730) (IN(I),I=1,80)
      C   IS CARD AN INPUT CARD
            IF (IN(1).NE.NI) GO TO 160
            IF (IN(7).NE.NT) GO TO 140
      C   CARD IS INPUT TAPE, DECODE IT
            MEDIA=2
            IF (IN(24).NE.NEQU) GO TO 670
```

```
      DO 130 NTFORM=1,NFORM
      DO 120 I=25,30
      J=I-24
      IF (IN(I).NE.TFORM(NTFORM,J)) GO TO 130
  120 CONTINUE
      GO TO 440
  130 CONTINUE
      GO TO 670
  140 MEDIA=1
      IF (IN(37).NE.NEQU) GO TO 670
      NCNL=0
      DO 150 I=38,80
      CALL DIGIT (IN(I),IDIG,ISTAT)
      IF (ISTAT.NE.0) GO TO 440
      NCNL=NCNL*10+IDIG
  150 CONTINUE
      GO TO 440
C   IS CARD A READING CARD
  160 IF (IN(1).EQ.NR) IREAD=IC
C   IS CARD A TIME CARD
      IF (IN(1).EQ.NT) ITIME=IC
C   IS CARD AN END OF RECORD CARD
      IF (IN(1).NE.NE.OR.IN(3).NE.NR) GO TO 170
      IEOR=IC
      IF (MEDIA.EQ.1) GO TO 180
      IF (IN(5).NE.NE.OR.IN(7).NE.NR) GO TO 670
      ISUB=IC
      IC=IC+1
      IDIGIT=1
      GO TO 430
C   IS CARD AN END OF FILE CARD
  170 IF (IN(1).NE.NE.OR.IN(3).NE.NF) GO TO 180
      IEOF=IC
      IF (MEDIA.EQ.1) GO TO 180
      IF (IN(5).NE.NE.OR.IN(7).NE.NF) GO TO 670
      ISUB=IC
      IC=IC+1
      IDIGIT=1
      GO TO 430
  180 ISUB=IC
      IC=IC+1
      IL=0
      IS=0
      DO 190 I=1,80
      II=81-I
      IF (IN(II).NE.NBL.AND.IL.EQ.0) IL=II
      IF (IN(I).EQ.NEQU) IS=I+1
  190 CONTINUE
      IDIGIT=1
      NREP=0
```

```
            IPAREN=0
            ICHR=IS
      200   DO 210 I=1,NCHARS
            IF (ICHARS(I).EQ.IN(ICHR)) GO TO 240
      210   CONTINUE
      C   CHARACTER NOT RECOGNIZED, IF IT'S WITHIN (), ASSUME SPECIAL CHAR
            IF (IPAREN.EQ.0) GO TO 220
            ICHR=ICHR-1
            GO TO 340
      220   CALL DIGIT (IN(ICHR),IDIG,ISTAT)
            IF (ISTAT.NE.0) GO TO 230
            NREP=NREP*10+IDIG
            GO TO 420
      230   IF (IN(ICHR).NE.NSTAR) GO TO 670
            GO TO 420
      240   IF (IPAREN.NE.1.AND.I.LE.14) GO TO 670
            GO TO (250,260,270,280,290,  300,310,320,330,340,  350,360,370,380
           2,390,  400), I
      C   ANY NUMERIC CHARACTER (DIGIT 0-9)
      250   ICHRS(IC)=-1
            IC=IC+1
            GO TO 410
      C   ANY ALPHABETIC CHARACTER (ANY CHARACTER BUT A NUMERIC)
      260   ICHRS(IC)=-2
            IC=IC+1
            GO TO 410
      C   A CHANNEL NUMERAL  (DIGIT 0-9  PART OF A CHANNEL NUMBER)
      270   ICHRS(IC)=-3
            IC=IC+1
            GO TO 410
      C   POSITIVE SIGN CHARACTER
      280   ICHRS(IC)=-4
            ICHRS(IC+1)=IN(ICHR+1)
            IC=IC+2
            ICHR=ICHR+1
            GO TO 410
      C   NEGATIVE SIGN CHARACTER
      290   ICHRS(IC)=-5
            ICHRS(IC+1)=IN(ICHR+1)
            IC=IC+2
            ICHR=ICHR+1
            GO TO 410
      C   MANTISSA OF READING (INTEGER VALUE) CHARACTER
      300   ICHRS(IC)=-6
            IC=IC+1
            GO TO 410
      C   MANTISSA OF READING (REAL NUMBER WITH EMBEDDED DECIMAL) CHARACTER
      310   ICHRS(IC)=-7
            IC=IC+1
            GO TO 410
```

```
C   EXPONENT OF READING (INTEGER VALUE) CHARACTER
320   ICHRS(IC)=-8
      IC=IC+1
      GO TO 410
C   OVERFLOW INDICATOR (ANY CHARACTER)
330   ICHRS(IC)=-9
      ICHRS(IC+1)=IN(ICHR+1)
      IC=IC+2
      ICHR=ICHR+1
      GO TO 410
C   A SPECIAL CHARACTER
340   ICHRS(IC)=-10
      ICHRS(IC+1)=IN(ICHR+1)
      IC=IC+2
      ICHR=ICHR+1
      GO TO 410
C   DAYS COMPONENT OF TIME READING (NUMERAL 0-9)
350   ICHRS(IC)=-11
      IC=IC+1
      GO TO 410
C   HOURS COMPONENT OF TIME READING (NUMERAL 0-9)
360   ICHRS(IC)=-12
      IC=IC+1
      GO TO 410
C   MINUTES COMPONENT OF TIME READING (NUMERAL 0-9)
370   ICHRS(IC)=-13
      IC=IC+1
      GO TO 410
C   SECONDS COMPONENT OF TIME READING (NUMERAL 0-9)
380   ICHRS(IC)=-14
      IC=IC+1
      GO TO 410
C   A LEFT PARENTHESIS, THE START OF NEXT CHARACTER SPECIFICATION
390   IF (IPAREN.NE.0) GO TO 670
      IPAREN=1
      ICHRS(IC)=IDIGIT
      ICHRS(IC+1)=0
      ICHRS(IC+2)=0
      ICHRS(IC+3)=0
      IPOSS=IC+1
      IC=IC+4
      NPOSS=0
      GO TO 420
C   A RIGHT PARENTHESIS, THE END OF THIS CHARACTER SPECIFICATION
400   ICHRS(IPOSS)=IC
      IF (NREP.EQ.0) NREP=1
      ICHRS(IPOSS+1)=NREP
      ICHRS(IPOSS+2)=NPOSS
      IDIGIT=IDIGIT+NREP
      IPOSS=0
```

```
          NPOSS=0
          NREP=0
          IPAREN=0
          GO TO 420
      410 NPOSS=NPOSS+1
      420 ICHR=ICHR+1
          IF (ICHR.LE.IL) GO TO 200
      430 ICHRS(IC)=-15
          IC=IC+1
          ICHRS(ISUB)=IDIGIT-1
      440 CONTINUE
      450 IF (INSKIP.EQ.0) RETURN
          ICRD=41
      C
      C  PLOT1 OR PLOT2 DATA CARD D1:   IN
      C  IN:     SPECIFIES CHANNELS TO BE SKIPPED
      C
          READ (5,700,END=660,ERR=670) (IN(I),I=1,80)
      C  CHECK TO MAKE SURE IT IS A SKIP CARD
          DO 460 I=1,5
          IF (IN(I).NE.ICHARS(I+16)) GO TO 620
      460 CONTINUE
      C  IT IS, DECODE THE CARD INTO ISKIP
          WRITE (6,730) (IN(I),I=1,80)
          ISKIP(1)=1
          IC=2
          DO 470 I=1,80
          II=81-I
          IF (IN(II).NE.NBL) GO TO 480
      470 CONTINUE
          GO TO 670
      480 IL=II
          IPAREN=0
          IDIGIT=1
          NFINAL=0
          ICHR=6
      490 DO 500 I=22,26
          IF (IN(ICHR).EQ.ICHARS(I)) GO TO 510
      500 CONTINUE
          IF (IN(ICHR).NE.NBL) GO TO 670
          ICHR=ICHR+1
          GO TO 610
      510 I=I-21
          GO TO (520,530,550,560,570), I
      C  RECORD IS TO BE KEPT, CHARACTER SPECIFICATION = K
      520 ISKIP(IC)=0
          ISKIP(IC+1)=0
          GO TO 540
      C  RECORD IS TO BE SKIPPED, CHARACTER SPECIFICATION = S
      530 ISKIP(IC)=-1
```

```
            ISKIP(IC+1)=0
      540   ICHR=ICHR+1
            CALL DIGIT (IN(ICHR),IDIG,ISTAT)
            IF (ISTAT.EQ.1) GO TO 600
            ISKIP(IC+1)=ISKIP(IC+1)*10+IDIG
            GO TO 540
      C   LAST RECORD SPECIFICATION, CHARACTER SPECIFICATION = F
      550   ISKIP(IC)=-2
            NFINAL=1
            ISKIP(IC+1)=0
            GO TO 540
      C   LEFT PARENTHESIS, START OF A NEW SPCIFICATION
      560   IF (IPAREN.NE.0) GO TO 670
            IPAREN=1
            ISKIP(IC)=IDIGIT
            ISKIP(IC+1)=0
            ISKIP(IC+2)=0
            ISKIP(IC+3)=0
            IPOSS=IC+1
            IC=IC+4
            NPOSS=0
            ICHR=ICHR+1
            GO TO 610
      C   RIGHT PARENTHESIS, THE END OF THIS SPECIFICATION
      570   IF (IPAREN.NE.1) GO TO 670
            IPAREN=0
            IF (NPOSS.EQ.0) GO TO 670
            ISKIP(IPOSS)=IC
            ISKIP(IPOSS+1)=NPOSS
            IDIGIT=IDIGIT+1
            ISKIP(IPOSS+2)=0
      580   ICHR=ICHR+1
            CALL DIGIT (IN(ICHR),IDIG,ISTAT)
            IF (ISTAT.EQ.1) GO TO 590
            ISKIP(IPOSS+2)=ISKIP(IPOSS+2)*10+IDIG
            GO TO 580
      590   IF (ISKIP(IPOSS+2).EQ.0) GO TO 670
            IF (NFINAL.EQ.1) ISKIP(IPOSS+2)=ISKIP(IPOSS+2)+1
            NFINAL=0
            GO TO 610
      600   NPOSS=NPOSS+1
            IC=IC+2
      610   IF (ICHR.LT.IL) GO TO 490
            ISKIP(IPOSS)=0
            WRITE (6,740)
            RETURN
      C   SIMULATE CARD IN THE OLD SINGLE CHANNEL FORMAT
      620   IVAL=0
            ISKIP(1)=0
            IC=3
```

```
            DO 650 I=1,80
      630   CALL DIGIT (IN(I),IDIG,ISTAT)
            IF (ISTAT.EQ.1) GO TO 640
            IVAL=IVAL*10+IDIG
            GO TO 630
      640   IF (IN(I).NE.NBL) GO TO 660
            IF (IN(I-1).EQ.NBL) GO TO 650
            ISKIP(IC)=IVAL
            IVAL=0
            IC=IC+1
      650   CONTINUE
            ISKIP(2)=IC-3
            WRITE (6,680) (IN(I),I=1,80)
            RETURN
      660   CALL RWEOF
      670   CALL RWERR
C
C
      680   FORMAT (6H SKIP=,80A1,/1H0)
      690   FORMAT (43H0ILLEGAL FILE TYPE SPECIFICATION FOR INTYPE)
      700   FORMAT (80A1)
      710   FORMAT (50H0INPUT DATA AS RECORDED BY DATA ACQUISITION SYSTEM,/,
          2 26H0DATA FORMAT SPECIFICATION,/)
      720   FORMAT (27H0INPUT DATA AS RECORDED BY ,22A1,/,
          2 26H0DATA FORMAT SPECIFICATION,/)
      730   FORMAT (1X,80A1)
      740   FORMAT (1H0)
      750   FORMAT (1H1,120A1)
C
            END
```

```
      SUBROUTINE DATAIN (NROW,NCOL,REED,KH,INTYPE,INPRT,INPNCH,MAXR,
     2 ITIME,INERR,INSKIP,IOUT,OUTDIM,ICHRS,INDAS,IREAD,ITIM,IEOR,IEOF,
     3 MEDIA,TITLE,MCNL,MAXCNL,ISKIP,NCNL,NTFORM)
C
C     *******************************************************************
C
      COMMON /ERRORS/ ICRD,ISEG
      INTEGER OUTDIM,FORMAT,RECHRS
      DIMENSION REED(NROW,NCOL),KH(NCOL),IOUT(OUTDIM),MAXR(NCOL),
     2 ICHRS(320),INDAS(22),FORMAT(3),TITLE(120),MCNL(MAXCNL),
     3 ISKIP(320)
      DATA NBL,NCOM,NDASH /1H ,1H,,1H-/
      IF (INTYPE.EQ.4) GO TO 250
      IPRT=0
      IDIAG=0
      IERR=0
      IROW=0
      IREC=0
      RECHRS=0
   10 IRERR=0
      IRDIAG=0
      I=1
      CALL DREAD (IOUT,OUTDIM,INSKIP,L,LCHRS,IREC,IROW,INTYPE,ICHRS,IREA
     2D,IEOR,IEOF,MEDIA,ISKIP,NCNL,NTFORM)
      IF (L.LT.0) GO TO 190
      DO 20 J=1,MAXCNL
      MCNL(J)=-1
   20 CONTINUE
      ICNL=0
      IR=0
      IB=-ICHRS(IREAD)
      IC=0
      IT=0
      IROW=IROW+1
      IF (IROW.LE.NROW) GO TO 30
      WRITE (6,290) IROW
      CALL EREXIT
C   HAS RECORD LENGTH CHANGED  IF SO, PRINT A WARNING
   30 IF (IROW.EQ.1) RECHRS=LCHRS
      IF (RECHRS.EQ.LCHRS) GO TO 40
      IDIAG=IDIAG+1
      IRDIAG=IRDIAG+1
      IF (IERR+IDIAG.GT.INERR.AND.INPRT.NE.-2) GO TO 40
      IF (IRERR+IRDIAG.EQ.1) WRITE (6,300)
      WRITE (6,390) IREC,RECHRS,LCHRS
      RECHRS=LCHRS
C   CHECK FOR TIME OR CHANNEL READING
   40 CALL MATCH (IOUT,OUTDIM,ICHRS,ITIM,I,ISIGN,ICHAN,IDEC,IVALUE,IEXP,
     2IDAY,IHR,IMIN,ISEC,IMAT,NCHRS,NOVFL)
      IF (IMAT.EQ.1) GO TO 170
```

```
            CALL MATCH (IOUT,OUTDIM,ICHRS,IREAD,I,ISIGN,ICHAN,IDEC,IVALUE,IEXP
           2,IDAY,IHR,IMIN,ISEC,IMAT,NCHRS,NOVFL)
            IF (IMAT.EQ.1) GO TO 70
      C   NO MATCH, PERHAPS WE NEED AN ERROR MESSAGE
            NERR=I+NCHRS
            IF (IOUT(NERR).EQ.NBL) IB=IB+1
            IF (IOUT(NERR).EQ.NBL.AND.IB.GE.2) GO TO 60
            IF (NERR.GE.LCHRS-ICHRS(IEOR)) GO TO 190
            IF (IABS(NERR-IC).LT.ICHRS(IREAD)-1.AND.IR.EQ.IROW) GO TO 50
            IERR=IERR+1
            IRERR=IRERR+1
            IF (IERR+IDIAG.GT.INERR.AND.INPRT.NE.-2) GO TO 50
            IF (IRERR+IRDIAG.EQ.1) WRITE (6,300)
            WRITE (6,310) IREC,NERR
            IC=NERR
      50    IB=0
      60    IR=IROW
            IF (NERR.GE.LCHRS-ICHRS(IEOR)) GO TO 190
            I=I+1
            GO TO 30
      70    IF (NOVFL.EQ.0) GO TO 80
      C   OVERFLOW OF READING, PRINT AN ERROR MESSAGE
            IDIAG=IDIAG+1
            IRDIAG=IRDIAG+1
            IF (IERR+IDIAG.GT.INERR.AND.INPRT.NE.-2) GO TO 130
            IF (IRERR+IRDIAG.EQ.1) WRITE (6,300)
            WRITE (6,280) IREC,NOVFL,ICHAN
            GO TO 130
      C   A MATCH WAS FOUND ON A READING, CALCULATE IT
      80    IF (ICNL.EQ.0) GO TO 100
            DO 90 K=1,ICNL
            IF (MCNL(ICNL).EQ.ICHAN) GO TO 120
      90    CONTINUE
      100   ICNL=ICNL+1
            IF (ICNL.LE.MAXCNL) GO TO 110
            WRITE (6,260) ICNL
            CALL EREXIT
      110   MCNL(ICNL)=ICHAN
            GO TO 140
      120   IDIAG=IDIAG+1
            IRDIAG=IRDIAG+1
            IF (IERR+IDIAG.GT.INERR.AND.INPRT.NE.-2) GO TO 130
            IF (IRERR+IRDIAG.EQ.1) WRITE (6,300)
            WRITE (6,410) IREC,I,ICHAN
      130   I=I+NCHRS+1
            IF (I.GE.LCHRS-ICHRS(IEOR)) GO TO 190
            GO TO 30
      140   DO 150 J=1,NCOL
            IF (ICHAN.EQ.KH(J)) GO TO 160
      150   CONTINUE
```

```
          I=I+NCHRS+1
          IF (I.GE.LCHRS-ICHRS(IEOR)) GO TO 190
          GO TO 30
  160     REED(IROW,J)=ISIGN*IVALUE*10.**(-IDEC)*10.**(-IEXP)
          IF (IDEC.NE.0) REED(IROW,J)=REED(IROW,J)*10.
          I=I+NCHRS+1
          IF (I.GE.LCHRS-ICHRS(IEOR)) GO TO 190
          GO TO 30
C   A MATCH WAS FOUND ON TIME, CALCULATE IT
  170     IF (IT.EQ.0) GO TO 180
          IERR=IERR+1
          IRERR=IRERR+1
          IF (IERR+IDIAG.GT.INERR.AND.INPRT.NE.-2) GO TO 180
          IF (IRERR+IRDIAG.EQ.1) WRITE (6,300)
          WRITE (6,400) IREC,I
  180     IT=1
          IVALUE=ISEC+60*IMIN+3600*IHR+86400*IDAY
          IF (IROW.EQ.1) INT=IVALUE
          IF (ITIME.NE.0) REED(IROW,ITIME)=IVALUE-INT
          I=I+NCHRS+1
          IF (I.GE.LCHRS-ICHRS(IEOR)) GO TO 190
          GO TO 30
C   END OF RECORD, PRINT INPUT RECORDS IF PRINT FLAGS ARE ON
  190     IF (INPRT.EQ.0) GO TO 220
          IF (INPRT.EQ.-2.AND.IRERR.EQ.0) GO TO 220
          IF (IPRT.EQ.INPRT) GO TO 220
          IPRT=IPRT+1
          IF (MOD(IPRT,10).EQ.0) WRITE (6,380) (TITLE(I),I=1,120)
          WRITE (6,320) IREC,IROW,LCHRS,IRERR,IRDIAG
          NREAD=(130-ICHRS(ITIM))/ICHRS(IREAD)
          NUMPL=NREAD*ICHRS(IREAD)+ICHRS(ITIM)
          IF (NUMPL.GT.LCHRS) NUMPL=LCHRS
          WRITE (6,330) (IOUT(I),I=1,NUMPL)
          K=NUMPL+1
          NUMPL=NREAD*ICHRS(IREAD)
          NX=ICHRS(ITIM)+1
  200     IF (NX.LE.ICHRS(IREAD)) GO TO 210
          NX=NX-ICHRS(IREAD)
          NUMPL=NUMPL+ICHRS(IREAD)
          GO TO 200
  210     ENCODE (340,FORMAT) NX,NUMPL
          IF (K.LE.LCHRS) WRITE (6,FORMAT) (IOUT(I),I=K,LCHRS)
C   PUNCH INPUT RECORD IF PUNCH FLAG IS ON
  220     IF (INPNCH.EQ.0) GO TO 230
          NUMPL=(80/ICHRS(IREAD))*ICHRS(IREAD)
          ENCODE (270,FORMAT) NUMPL
          WRITE (INPNCH,FORMAT) (IOUT(I),I=1,LCHRS)
  230     IF (L.GE.0) GO TO 10
          NERR=IERR+IDIAG
          IF (INTYPE.EQ.0) WRITE (6,370)
```

```
        IF (INTYPE.NE.0) WRITE (6,350) (INDAS(I),I=1,22)
        WRITE (6,360) NERR,IERR,IDIAG,IREC,IROW
        DO 240 I=1,NCOL
        MAXR(I)=IROW
  240   CONTINUE
        RETURN
C   INPUT IS FROM REDUCED DATA CARD IMAGES, GO READ THEM
  250   CALL REDCIN (NROW,NCOL,REED,KH,INPRT,INPNCH,MAXR,ITIME,INERR,INSKI
       2P,ISKIP)
        RETURN
C
C
  260   FORMAT (22H0INPUT RECORD SIZE OF ,I5,20H EXCEEDS BUFFER SIZE,/,
       2 23H CHECK PARAMETER MAXCNL)
  270   FORMAT (1H(,I2,3HA1))
  280   FORMAT (34H OVERFLOW WARNING          RECORD: ,I5,13H  CHARACTER: ,
       2 I6,13H    CHANNEL: ,I6)
  290   FORMAT (36H0ROW STORAGE LIMITS EXCEEDED AT ROW ,I4,
       2 31H CHECK PARAMETERS NROW AND NCOL)
  300   FORMAT (/1H0)
  310   FORMAT (34H CHARACTER SEQUENCE ERROR RECORD: ,I5,13H  CHARACTER: ,
       2 I6)
  320   FORMAT (9H0RECORD: ,I4,8H   ROW: ,I4,20H   CHARACTER COUNT: ,I4,
       2 11H   ERRORS: ,I4,13H   WARNINGS: ,I4)
  330   FORMAT (1X,130A1)
  340   FORMAT (1H(,I2,1HX,I3,3HA1))
  350   FORMAT (22H0READING OF DATA FROM ,22A1,10H COMPLETED)
  360   FORMAT (17H TOTAL MESSAGES: ,I6,3X,8HERRORS: ,I6,3X,10HWARNINGS: ,
       2 I6,3X,9HRECORDS: ,I6,3X,6HMAXR: ,I6)
  370   FORMAT (55H0READING OF DATA FROM DATA ACQUISITION SYSTEM COMPLETED
       2)
  380   FORMAT (1H1,120A1)
  390   FORMAT (34H RECORD LENGTH WARNING     RECORD: ,I5,13H OLD LENGTH: ,
       2 I6,13H NEW LENGTH: ,I6)
  400   FORMAT (34H MULTIPLE TIME ERROR       RECORD: ,I5,13H  CHARACTER: ,
       2 I6)
  410   FORMAT (34H MULTIPLE READING WARNING RECORD: ,I5,13H  CHARACTER: ,
       2 I6,13H    CHANNEL: ,I6)
C
        END
```

```
      SUBROUTINE DREAD (IOUT,OUTDIM,INSKIP,L,LCHRS,IREC,IROW,INTYPE,
     2 ICHRS,IREAD,IEOR,IEOF,MEDIA,ISKIP,NCNL,NTFORM)
C
C     ******************************************************************
C
      INTEGER OUTDIM
      COMMON /ERRORS/ ICRD,ISEG
      DIMENSION IOUT(OUTDIM),ISKIP(320),ICHRS(320)
      DATA NE,NAT /1HE,1H@/
10    II=0
      IREC=IREC+1
      IF (MEDIA.EQ.1) GO TO 80
C   INPUT IS TO BE FROM MAGNETIC TAPE, READ A BLOCK OF DATA
      IF (NTFORM.NE.1) GO TO 70
C   7 TRACK TAPE INPUT
      CALL NTRAN (7,26,24,22)
      IOUT(1)=NE
      CALL NTRAN (7,2,OUTDIM,IOUT,LSTAT,20,L)
      IF (LSTAT.EQ.-2) GO TO 60
      IF (IOUT(1).EQ.NE) LSTAT=-4
      IF (IOUT(1).EQ.NE) GO TO 160
      IF (LSTAT.EQ.-3) CALL NTRAN (7,22)
      IF (LSTAT.EQ.-3) LSTAT=L
      IF (LSTAT.LT.0) GO TO 160
      IF (L.GT.OUTDIM/6) GO TO 180
      DO 30 I=1,L
      II=L+1-I
      IC=IOUT(II)
      DO 20 J=1,6
      IK=6*(II-1)+7-J
      CALL BYTE (IC,7-J,IOUT(IK))
20    CONTINUE
30    CONTINUE
      LCHRS=6*L
40    IF (IOUT(LCHRS).NE.NAT) GO TO 50
      LCHRS=LCHRS-1
      GO TO 40
50    IOUT(LCHRS+1)=1H
      IOUT(LCHRS+2)=1HX
      LCHRS=LCHRS+2
      ICHRS(IEOR)=2
      GO TO 150
60    IOUT(1)=1HF
      IOUT(2)=1HI
      IOUT(3)=1HL
      IOUT(4)=1HE
      IOUT(5)=1HN
      IOUT(6)=1HD
      ICHRS(IEOF)=6
      GO TO 170
```

```
      C   OTHER FORMATS AREN'T DEFINED, FORMAT ERROR
      70    ICRD=31
            GO TO 190
      C   INPUT IS TO BE FROM CARD IMAGES, READ A SCAN OF DATA
      80    L=1
            NEOF=1
            NEOR=1
      90    ICRD=41
      C
      C   PLOT1 OR PLOT2 DATA CARDS D1
      C   THESE CARDS ARE PREVIOUSLY PREPARED DATA CARDS OF THE UNCONVERTED
      C   DATA
      C
            K=NCNL*ICHRS(IREAD)+L-1
            IF (K.LE.OUTDIM) GO TO 100
            WRITE (6,230) K
            CALL EREXIT
      100   READ (5,210,ERR=190,END=200) (IOUT(I),I=L,K)
            MEOF=K-ICHRS(IEOF)
            DO 110 I=NEOF,MEOF
            CALL MATCH (IOUT,OUTDIM,ICHRS,IEOF,I,0,0,0,0,0,0,0,0,0,IMAT,NCHRS,
           20)
            IF (IMAT.EQ.1) GO TO 170
      110   CONTINUE
            NEOF=MEOF+1
            MEOR=K-ICHRS(IEOR)
            DO 120 I=NEOR,MEOR
            CALL MATCH (IOUT,OUTDIM,ICHRS,IEOR,I,0,0,0,0,0,0,0,0,0,IMAT,NCHRS,
           20)
            IF (IMAT.EQ.1) GO TO 130
      120   CONTINUE
            GO TO 140
      130   LCHRS=I+ICHRS(IEOR)
            GO TO 150
      140   NEOR=MEOR+1
            L=K+1
            GO TO 90
      C   IS THIS SCAN TO BE SKIPPED
      150   IF (INSKIP.EQ.0) RETURN
            CALL SKIP (IREC,ISKIP,NSKIP)
            IF (NSKIP.EQ.0) RETURN
            IF (NSKIP.EQ.-2) GO TO 60
            WRITE (6,240) IREC
            GO TO 10
      160   WRITE (6,220) LSTAT
      170   L=-2
            LCHRS=ICHRS(IEOF)
            RETURN
      180   WRITE (6,230) L
            CALL EREXIT
```

```
      190   CALL RWERR
      200   CALL RWEOF
     C
     C
      210   FORMAT (80A1)
      220   FORMAT (59H0READING OF INPUT FILE TERMINATED BY NTRAN ERROR, STATU
           2S : ,I5)
      230   FORMAT (22H0INPUT RECORD SIZE OF ,I5,20H EXCCEDS BUFFER SIZE,/,
           2 23H CHECK PARAMETER MAXCNL)
      240   FORMAT (9H0RECORD: ,I4,12H     SKIPPED)
     C
            END
```

```
      SUBROUTINE MATCH (IOUT,OUTDIM,ICHRS,ISUB,I,ISIGN,ICHAN,IDEC,IVALUE
     2,IEXP,IDAY,IHR,IMIN,ISEC,IMAT,NCHRS,NOVFL)
C
C     ******************************************************************
C
      INTEGER OUTDIM
      DIMENSION IOUT(OUTDIM),ICHRS(320)
      DATA NPER /1H./
      ICHAN=0
      ISIGN=0
      NOVFL=0
      IDEC=0
      IVALUE=0
      IEXP=0
      IDAY=0
      IHR=0
      IMIN=0
      ISEC=0
      IMAT=0
      NCHRS=ICHRS(ISUB)-1
      IS=ISUB+1
      NREP=ICHRS(IS+2)
      NCHRS1=NCHRS+1
      DO 190 K=1,NCHRS1
      IK=I+K-1
      NPOSS=ICHRS(IS+3)
      IC=IS+4
      DO 170 IPOSS=1,NPOSS
      INDEX=IABS(ICHRS(IC))
      GO TO (10,20,30,40,50,  60,70,100,110,120,  130,140,150,160), INDE
     2X
C   CHECK FOR ANY NUMERIC DIGIT
10    IC=IC+1
      CALL DIGIT (IOUT(IK),IDIG,ISTAT)
      IF (ISTAT.NE.0) GO TO 170
      GO TO 180
C   CHECK FOR ANY ALPHANUMERIC CHARACTER
20    IC=IC+1
      GO TO 180
C   CHECK FOR A CHANNEL NUMBER DIGIT
30    IC=IC+1
      CALL DIGIT (IOUT(IK),IDIG,ISTAT)
      IF (ISTAT.NE.0) GO TO 170
      ICHAN=ICHAN*10+IDIG
      GO TO 180
C   CHECK FOR SIGN POSITIVE CHARACTER SPECIFICATION
40    IC=IC+2
      IF (IOUT(IK).NE.ICHRS(IC-1)) GO TO 170
      ISIGN=1
      GO TO 180
```

```
      C   CHECK FOR SIGN NEGATIVE CHARACTER SPECIFICATION
      50    IC=IC+2
            IF (IOUT(IK).NE.ICHRS(IC-1)) GO TO 170
            ISIGN=-1
            GO TO 180
      C   CHECK FOR READING VALUE DIGIT
      60    IC=IC+1
            CALL DIGIT (IOUT(IK),IDIG,ISTAT)
            IF (ISTAT.NE.0) GO TO 170
            IVALUE=IVALUE*10+IDIG
            GO TO 180
      C   CHECK FOR A REAL VALUE DIGIT POSSIBLY WITH DECIMAL POINT
      70    IC=IC+1
            CALL DIGIT (IOUT(IK),IDIG,ISTAT)
            IF (ISTAT.NE.0.AND.IOUT(IK).NE.NPER) GO TO 170
            IF (IOUT(IK).EQ.NPER.AND.IDEC.NE.0) GO TO 170
            IF (IDEC.EQ.0) GO TO 80
            IDEC=IDEC+1
      80    IF (IOUT(IK).NE.NPER) GO TO 90
            IDEC=1
            GO TO 180
      90    IVALUE=IVALUE*10+IDIG
            GO TO 180
      C   CHECK FOR AN EXPONENT DIGIT
      100   IC=IC+1
            CALL DIGIT (IOUT(IK),IDIG,ISTAT)
            IF (ISTAT.NE.0) GO TO 170
            IEXP=IEXP*10+IDIG
            GO TO 180
      C   CHECK FOR OVERFLOW OF READING
      110   IC=IC+2
            IF (IOUT(IK).NE.ICHRS(IC-1)) GO TO 170
            ISIGN=0
            NOVFL=IK
            GO TO 180
      C   CHECK FOR ANY SINGLE SPECIAL CHARACTER
      120   IC=IC+2
            IF (IOUT(IK).NE.ICHRS(IC-1)) GO TO 170
            GO TO 180
      C   CHECK FOR TIME READING DAYS DIGIT
      130   IC=IC+1
            CALL DIGIT (IOUT(IK),IDIG,ISTAT)
            IF (ISTAT.NE.0) GO TO 170
            IDAY=IDAY*10+IDIG
            GO TO 180
      C   CHECK FOR TIME READING HOURS DIGIT
      140   IC=IC+1
            CALL DIGIT (IOUT(IK),IDIG,ISTAT)
            IF (ISTAT.NE.0) GO TO 170
            IHR=IHR*10+IDIG
```

```
      GO TO 180
C   CHECK FOR TIME READING MINUTES DIGIT
150   IC=IC+1
      CALL DIGIT (IOUT(IK),IDIG,ISTAT)
      IF (ISTAT.NE.0) GO TO 170
      IMIN=IMIN*10+IDIG
      GO TO 180
C   CHECK FOR TIME READING SECONDS DIGIT
160   IC=IC+1
      CALL DIGIT (IOUT(IK),IDIG,ISTAT)
      IF (ISTAT.NE.0) GO TO 170
      ISEC=ISEC*10+IDIG
      GO TO 180
C   NO MATCH -- TRY OTHER POSSIBILITIES
170   CONTINUE
      NCHRS=K-1
      IMAT=0
      NOVFL=0
      RETURN
C   MATCH ON THIS CHARACTER -- GO TO NEXT ONE
180   NREP=NREP-1
      IF (NREP.EQ.0) IS=ICHRS(IS+1)
      IF (NREP.EQ.0) NREP=ICHRS(IS+2)
190   CONTINUE
      IMAT=1
      RETURN
      END
```

```
      SUBROUTINE DIGIT (ICHAR,IDIG,ISTAT)
C
C     ******************************************************************
C
      INTEGER CHAR
      DIMENSION CHAR(10)
      DATA (CHAR(I),I=1,10) /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
      DO 10 I=1,10
      IF (CHAR(I).EQ.ICHAR) GO TO 20
10    CONTINUE
C   CHARACTER IS NOT A NUMERIC
      ISTAT=1
      RETURN
C   CHARACTER IS A NUMERIC CHARACTER
20    IDIG=I-1
      ISTAT=0
      RETURN
      END
```

```
          SUBROUTINE SKIP (IREC,ISKIP,NSKIP)
C
C         ***********************************************************************
C
          DIMENSION ISKIP(320)
          IF (ISKIP(1).EQ.1) GO TO 30
C   FORMAT IS OLD SINGLE CHANNEL FORMAT
          NSPEC=ISKIP(2)
          DO 10 I=1,NSPEC
          IF (ISKIP(I+2).EQ.IREC) GO TO 20
10        CONTINUE
          NSKIP=0
          RETURN
20        NSKIP=-1
          RETURN
C   FORMAT IS BLOCKED SKIP, CHECK TO SEE IF RECORD IS SKIPPED
30        IF (IREC.NE.1) GO TO 40
          NSPEC=2
          NPOSS=1
          NREC=1
C   ARE WE PAST THE LAST RECORD FOR THIS SPECIFICATION
40        IF (IREC.LE.ISKIP(NSPEC+3)) GO TO 60
          IF (ISKIP(NSPEC+1).NE.0) GO TO 50
          NSKIP=0
          RETURN
50        NSPEC=ISKIP(NSPEC+1)
          NPOSS=1
          NREC=1
C   ARE WE PAST THE LAST POSSIBILITY FOR THIS SPECIFICATION
60        IF (NPOSS.LE.ISKIP(NSPEC+2)) GO TO 70
          NPOSS=1
          NREC=1
C   ARE WE PAST THE LAST RECORD FOR THIS POSSIBILITY
70        IPOSS=2*(NPOSS-1)+4+NSPEC
          IF (NREC.LE.ISKIP(IPOSS+1)) GO TO 80
          NPOSS=NPOSS+1
          NREC=1
          GO TO 60
C   IF WE MADE IT THIS FAR, LET'S USE THE PARAMETER
80        NREC=NREC+1
          IF (ISKIP(IPOSS).NE.-2) GO TO 100
          IF (IREC.GE.ISKIP(NSPEC+3)) GO TO 90
          NSKIP=0
          RETURN
90        NSKIP=-2
          RETURN
100       NSKIP=ISKIP(IPOSS)
          RETURN
          END
```

```
      SUBROUTINE REDCIN (NROW,NCOL,REED,KH,INPRT,INPNCH,MAXR,ITIME,INERR
     2,INSKIP,ISKIP,LASREC)
C
C     ******************************************************************
C
      COMMON /ERRORS/ ICRD,ISEG
      DIMENSION REED(NROW,NCOL),KH(NCOL),MAXR(NCOL),R(8),IEXP(8),
     2 ISKIP(320)
      ICSKIP=0
      NCHAN=0
C   READ CHANNEL NUMBER AND NUMBER OF READINGS
      IF (ISEG.EQ.2.OR.ISEG.EQ.3) ICRD=60
      IF (ISEG.EQ.4) ICRD=30
      IF (ISEG.EQ.5) ICRD=32
10    READ (5,190,ERR=120,END=130) NREAD,ICHAN,IEND
      IF (IEND.EQ.999) GO TO 110
C   LOCATE COLUMN INSTRUMENT IS STORED IN (IF ANY)
      DO 20 J=1,NCOL
      IF (ICHAN.EQ.KH(J)) GO TO 30
20    CONTINUE
C   THIS CHANNEL IS NOT NEEDED, SKIP OVER THE READINGS
      NCARDS=(NREAD+7)/8
      READ (5,140,ERR=120,END=130) (IN,I=1,NCARDS)
      ICSKIP=ICSKIP+1
      IF (INPRT.EQ.0) GO TO 10
      IF (IPRT.EQ.INPRT) GO TO 10
      WRITE (6,150) ICHAN
      IPRT=IPRT+1
      GO TO 10
C   READ IN THIS CHANNEL'S READINGS INTO ARRAY REED
30    IREC=0
      IROW=0
40    READ (5,210,ERR=120,END=130) (R(I),IEXP(I),I=1,8)
      DO 80 I=1,8
      IREC=IREC+1
      IF (INSKIP.EQ.0) GO TO 50
      CALL SKIP (IREC,ISKIP,NSKIP)
      IF (NSKIP.EQ.-1) GO TO 70
      IF (NSKIP.EQ.-2) GO TO 70
50    CONTINUE
      IF (IROW.LT.NROW) GO TO 60
      WRITE (6,200) ICHAN
      CALL EREXIT
      STOP
60    IROW=IROW+1
      REED(IROW,J)=R(I)*10.**IEXP(I)
70    IF (IREC.GE.NREAD) GO TO 90
80    CONTINUE
      GO TO 40
90    MAXR(J)=IROW
```

```
      C   PRINT INSTRUMENTS READINGS (IF PRINT FLAGS ARE ON)
             IF (INPRT.EQ.0) GO TO 100
             IF (IPRT.EQ.INPRT) GO TO 100
             IPRT=IPRT+1
             NRSKIP=NREAD-IROW
             WRITE (6,170) ICHAN,NREAD,IROW,NRSKIP,MAXR(J)
             WRITE (6,180) (REED(L,J),L=1,IROW)
             IPRT=IPRT+1
      100    NCHAN=NCHAN+1
             GO TO 10
      C   END OF READINGS SET, RETURN FROM WHENCE WE CAME
      110    ICHAN=NCHAN+ICSKIP
             WRITE (6,160) ICHAN,NCHAN,ICSKIP
             RETURN
      120    CALL RWERR
      130    CALL RWEOF
      C
      C
      140    FORMAT (A1)
      150    FORMAT (10H0CHANNEL: ,I6,10H   SKIPPED)
      160    FORMAT (56H0READING OF DATA FROM SPEED2 REDUCED DATA      COMPLETE
            2D,/,22H TOTAL CHANNELS READ: ,I4,3X,6HUSED: ,I4,3X,9HSKIPPED: ,
            3 I4)
      170    FORMAT (10H0CHANNEL: ,I6,3X,9HRECORDS: ,I4,3X,6HUSED: ,I4,3X,
            2 9HSKIPPED: ,I4,3X,6HMAXR: ,I4)
      180    FORMAT (1X,10E12.5)
      190    FORMAT (I6,I6,65X,I3)
      200    FORMAT (42H0NUMBER OF READINGS FOR INSTRUMENT NUMBER ,I3,
            2 57H EXCEEDS STORAGE LIMITS    CHECK PARAMETERS NROW AND NCOL)
      210    FORMAT (8(F7.0,I3))
      C
             END
```

```
      SUBROUTINE PRINT (NROW,NCOL,REED,KH,ITYPE,NAME,ITIME,MAXR,MAXC,
     2 TITLE)
C
C     ******************************************************************
C
      DIMENSION REED(NROW,NCOL),KH(NCOL),NAME(NCOL,6),MAXR(NCOL),
     2 ITYPE(NCOL),IBUF(120),JBUF(120),TITLE(120)
      DATA IBL /1H /
      KHL=1
   10 KHH=KHL+9
      KHH=MIN0(KHH,MAXC)
      IL=0
      DO 60 L=1,NROW
      IP=0
      DO 20 I=1,120
      IBUF(I)=IBL
      JBUF(I)=IBL
   20 CONTINUE
      ENCODE (70,JBUF) (REED(L,I),I=KHL,KHH)
      DECODE (80,JBUF) (IBUF(I),I=1,120)
      DO 50 I=KHL,KHH
      IF (L.GT.MAXR(I)) GO TO 30
      IP=1
      GO TO 50
   30 IWL=(I-KHL)*12+1
      IWH=IWL+11
      DO 40 J=IWL,IWH
   40 IBUF(J)=IBL
   50 CONTINUE
      IF (IP.EQ.0) GO TO 60
      IL=IL+1
      IF (MOD(IL,40).EQ.1) WRITE (6,140) (TITLE(I),I=1,120)
      IF (MOD(IL,40).EQ.1) WRITE (6,100) ((NAME(I,II),II=1,6),I=KHL,KHH)
      IF (MOD(IL,40).EQ.1) WRITE (6,110) (IBL,KH(I),I=KHL,KHH)
      IF (MOD(IL,10).EQ.1) WRITE (6,120)
      IF (ITIME.NE.0) WRITE (6,130) REED(L,ITIME),(IBUF(I),I=1,120)
      IF (ITIME.EQ.0) WRITE (6,90) (IBUF(I),I=1,120)
   60 CONTINUE
      KHL=KHL+10
      IF (KHL.LE.MAXC) GO TO 10
      RETURN
C
C
   70 FORMAT (10(1X,G10.4,1X))
   80 FORMAT (120A1)
   90 FORMAT (1X,T12,120A1)
  100 FORMAT (1H0,T12,10(4X,6A1,2X))
  110 FORMAT (11X,10(A1,2X,1H(,I6,1H),1X))
  120 FORMAT (1H )
  130 FORMAT (1X,G10.4,T12,120A1)
```

```
      140   FORMAT (1H1,120A1)
      C
            END
```

```
      SUBROUTINE PUNCH (NROW,NCOL,REED,KH,NAME,MAXR,MAXC,NTEST,NPNCH)
C
C     ****************************************************************
C
      DIMENSION REED(NROW,NCOL),KH(NCOL),NAME(NCOL,6),MAXR(NCOL),
     2 IBUF(80),IZERO(10),ICHR(17)
      DATA (IZERO(I),I=1,10),(ICHR(I),I=1,17),IBL /1H+,1H.,5*1H0,1H+,
     2 2*1H0,1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H-,1H ,1H+,1H.,1H-
     3,1H ,1H ,1H /
      DO 80 I=1,MAXC
      KHN=MOD(KH(I),1000)+1000*NTEST
      WRITE (NPNCH,90) MAXR(I),KHN,(NAME(I,J),J=1,6)
      MAXRL=1
   10 MAXRH=MAXRL+7
      MAXRH=MIN0(MAXRH,MAXR(I))
      DO 20 J=1,80
      IBUF(J)=IBL
   20 CONTINUE
      DO 70 L=MAXRL,MAXRH
      IWL=(L-MAXRL)*10+1
      IF (REED(L,I).EQ.0.0) GO TO 30
      IEXP=INT(ALOG10(ABS(REED(L,I))))+1
      IF (ABS(REED(L,I)).LT.1.0) IEXP=IEXP-1
      IVAL=INT(ABS(100000.*REED(L,I)/10.**IEXP)+0.5)
      IREED=1000*IVAL+ABS(IEXP)
      IBUF(IWL+1)=ICHR(14)
      IC=ISIGN(1,IEXP)
      IBUF(IWL+7)=ICHR(12+IC)
      IC=IFIX(SIGN(1.,REED(L,I)))
      IBUF(IWL)=ICHR(16+IC)
   30 DO 60 J=1,10
      IF (REED(L,I).EQ.0.0) GO TO 50
      IF (J.GT.8.OR.J.EQ.3) GO TO 40
      IC=MOD(IREED,10)
      K=IWL+10-J
      IBUF(K)=ICHR(IC+1)
   40 IREED=IREED/10
      GO TO 60
   50 K=IWL+J-1
      IBUF(K)=IZERO(J)
   60 CONTINUE
   70 CONTINUE
      WRITE (NPNCH,100) (IBUF(J),J=1,80)
      MAXRL=MAXRL+8
      IF (MAXRL.LE.MAXR(I)) GO TO 10
   80 CONTINUE
      WRITE (NPNCH,110)
      END FILE NPNCH
      REWIND NPNCH
      RETURN
```

```
C
C
90    FORMAT (2I6,6A1)
100   FORMAT (80A1)
110   FORMAT (T78,3H999)
C
      END
```

```
      SUBROUTINE CONV (NROW,NCOL,REED,KH,ITYPE,C,ADD,NCORR,MAXR,MAXC)
C
C     ******************************************************************
C
      DIMENSION REED(NROW,NCOL),KH(NCOL),ITYPE(NCOL),C(NCOL),ADD(NCOL),
     2 MAXR(NCOL)
      DO 20 I=1,MAXC
      K=ITYPE(I)
      IF (K.EQ.0) GO TO 20
      IF (C(K).EQ.0.) GO TO 20
      MAXROW=MAXR(I)
      DO 10 L=1,MAXROW
   10 REED(L,I)=REED(L,I)*C(K)+ADD(K)
   20 CONTINUE
      IF (NCORR.EQ.0) GO TO 30
      CALL CORR (NROW,NCOL,REED,KH,MAXC)
   30 CONTINUE
      RETURN
      END
```

SUBPROGRAM: CORR READ IN CORRECTIONS TO DATA

```
      SUBROUTINE CORR (NROW,NCOL,REED,KH,MAXC)
C
C     ******************************************************************
C
      COMMON /ERRORS/ ICRD,ISEG
      DIMENSION REED(NROW,NCOL),KH(NCOL)
      WRITE (6,90)
   10 ICRD=71
C
C   PLOT2 DATA CARD G1:    IRL,IRH,ICL,ICH
C   IRL:    LOW ROW INDEX FOR CORRECTIONS TO BE READ IN
C   IRH:    HIGH ROW INDEX FOR CORRECTIONS TO BE READ IN
C   ICL:    LOW COLUMN (OR INSTRUMENT) INDEX FOR CORRECTIONS
C   ICH:    HIGH COLUMN (OR INSTRUMENT) INDEX FOR CORRECTIONS
C
      READ (5,80,ERR=60,END=70) IRL,IRH,ICL,ICH
      WRITE (6,80) IRL,IRH,ICL,ICH
      IF (IRL.LE.0) RETURN
      IF (IRH.LT.IRL) GO TO 50
      IF (ICH.LE.0) GO TO 20
      IF (IRH.EQ.IRL) GO TO 20
      IF (ICH.NE.ICL) GO TO 50
   20 IF (ICL.GT.0) GO TO 40
      DO 30 I=1,MAXC
      IF (KH(I)+ICL.EQ.0) ICL=I
      IF (KH(I)+ICH.EQ.0) ICH=I
   30 CONTINUE
   40 IF (ICL.LE.0.OR.ICH.LE.0) GO TO 50
      IF (ICH.LT.ICL) GO TO 50
      ICRD=72
C
C   PLOT2 DATA CARD(S) G2:    REED
C   REED:   CORRECTIONS TO THE MATRIX
C
      IF (IRL.EQ.IRH) READ (5,110,ERR=60,END=70) (REED(IRL,I),I=ICL,ICH)
      IF (ICL.EQ.ICH) READ (5,110,ERR=60,END=70) (REED(I,ICL),I=IRL,IRH)
      IF (IRL.EQ.IRH) WRITE (5,100,ERR=60,END=70) (REED(IRL,I),I=ICL,ICH
     2)
      IF (ICL.EQ.ICH) WRITE (5,100,ERR=60,END=70) (REED(I,ICL),I=IRL,IRH
     2)
      GO TO 10
   50 WRITE (6,120)
      WRITE (6,80) IRL,IRH,ICL,ICH
   60 CALL RWERR
   70 CALL RWEOF
C
C
   80 FORMAT (10I5)
   90 FORMAT (54H1 CORRECTION TO DATA MATRIX ADDED AT TIME OF EXECUTION/
     2)
```

```
      100   FORMAT (8F15.6)
      110   FORMAT (8F10.0)
      120   FORMAT (48H0***** ERROR ON FOLLOWING CORRECTION CARD ***** )
      C
            END
```

```
      SUBROUTINE FIND (I,J,KH,NCOL)
C
C     ***********************************************************************
C
      DIMENSION KH(NCOL)
      DO 10 K=1,NCOL
      IF (KH(K).EQ.I) GO TO 20
10    CONTINUE
      WRITE (6,30) I
      RETURN
20    J=K
      RETURN
C
C
30    FORMAT (62H0 ERROR IN SUBROUTINE FIND, NO REFERENCE TO INSTRUMENT
     2NUMBER ,I4,13H CAN BE FOUND)
C
      END
```

```
      SUBROUTINE DRAW (JM,IM,X,Y,X0,XM,Y0,YM,IBUFF,JBUFF,IHEAD,TITLE,IPN
     2,NROW,PLTDIM)
C
C     *****************************************************************
C
      INTEGER PLTDIM
      DIMENSION X(NROW,PLTDIM),Y(NROW,PLTDIM),JM(PLTDIM),P(101),YSC(6),
     2 XSC(11),ISYM(6),IPN(PLTDIM,6),IHEAD(80),IBUFF(40),JBUFF(40),
     3 TITLE(120)
      INTEGER RM1,R,C,P,DASH,PLUS,BLANK
      DATA BLANK,DASH,PLUS,(ISYM(I),I=1,6) /1H ,1H-,1H+,1H*,1HX,1H0,1H+,
     2 1H-,1H./
      WRITE (6,250) (TITLE(I),I=1,120),(IHEAD(I),I=1,80)
      WRITE (6,280)
      DYM=YM-Y0
      DXM=XM-X0
      YSC(1)=YM
      XSC(1)=X0
      DO 10 L=2,6
   10 YSC(L)=YSC(L-1)-DYM/5.
      DO 20 L=2,11
   20 XSC(L)=XSC(L-1)+DXM/10.
      DO 30 J=1,IM
      IM1=JM(J)-1
      DO 30 I=1,IM1
      IP1=I+1
      IMM=JM(J)
      DO 30 II=IP1,IMM
      IF (Y(II,J).LT.Y(I,J)) GO TO 30
      TEMP=Y(I,J)
      Y(I,J)=Y(II,J)
      Y(II,J)=TEMP
      TEMP=X(I,J)
      X(I,J)=X(II,J)
      X(II,J)=TEMP
   30 CONTINUE
      R=0
      L=0
   40 R=R+1
      IF (R.LE.6.OR.R.GE.47) GO TO 50
      ICHAR=IBUFF(R-6)
      GO TO 60
   50 ICHAR=BLANK
   60 CONTINUE
      RM1=R-1
      AR1=(RM1-.5)/50.
      AR2=(RM1+.5)/50.
      DO 70 C=1,101
   70 P(C)=BLANK
      J=IM
```

```
   80  I=0
   90  I=I+1
       IF (I-JM(J)) 100,100,110
  100  IF (Y(I,J).GE.YM-AR1*DYM) GO TO 90
       IF (Y(I,J).LT.YM-AR2*DYM) GO TO 110
       C=100.0*((X(I,J)-X0)/DXM)+1.5
       IF (C.GT.101.OR.C.LT.1) GO TO 90
       K=MOD(J,6)+1
       P(C)=ISYM(K)
       GO TO 90
  110  J=J-1
       IF (J) 120,120,80
  120  K=MOD(R-6,6)+1
       IF (MOD(RM1,5)) 130,150,130
  130  IF (R.LE.6) GO TO 140
       IF (R-6.LE.IM) WRITE (6,260) ICHAR,PLUS,(P(I),I=1,101),PLUS,(IPN(R
      2-6,I),I=1,6),ISYM(K)
  140  IF (R-6.GT.IM.OR.R.LE.6) WRITE (6,210) ICHAR,PLUS,(P(I),I=1,101),P
      2LUS
       GO TO 200
  150  IF (MOD(RM1,10)) 160,180,160
  160  IF (R.LE.6) GO TO 170
       IF (R-6.LE.IM) WRITE (6,260) ICHAR,DASH,(P(I),I=1,101),DASH,(IPN(R
      2-6,I),I=1,6),ISYM(K)
  170  IF (R-6.GT.IM.OR.R.LE.6) WRITE (6,210) ICHAR,DASH,(P(I),I=1,101),D
      2ASH
       GO TO 200
  180  L=L+1
       IF (R.LE.6) GO TO 190
       IF (R-6.LE.IM) WRITE (6,270) ICHAR,YSC(L),DASH,(P(I),I=1,101),DASH
      2,(IPN(R-6,I),I=1,6),ISYM(K)
  190  IF (R-6.GT.IM.OR.R.LE.6) WRITE (6,220) ICHAR,YSC(L),DASH,(P(I),I=1
      2,101),DASH
       GO TO 200
  200  IF (R.LT.51) GO TO 40
       WRITE (6,280)
       WRITE (6,230) XSC
       WRITE (6,240) (JBUFF(I),I=1,40)
       RETURN
C
C
  210  FORMAT (1H ,A1,14X,103A1)
  220  FORMAT (1H ,A1,F13.2,1X,103A1)
  230  FORMAT (9X,11F10.2)
  240  FORMAT (1H0,46X,40A1)
  250  FORMAT (1H1,120A1,/,26X,80A1)
  260  FORMAT (1H ,A1,14X,103A1,3X,6A1,2X,A1)
  270  FORMAT (1H ,A1,F13.2,1X,103A1,3X,6A1,2X,A1)
  280  FORMAT (1H ,16X,10(10H+---------),1H+)
C
```

```
101:          END
```

SUBPROGRAM: TEXT PUT TEXT ONTO CALCOMP PLOTS

```
      SUBROUTINE TEXT (XPOS,YPOS,HEIGHT,ICHARS,ANGLE,NCHRS)
C
C     ******************************************************************
C
      DIMENSION ICHARS(NCHRS)
      DO 10 I=1,NCHRS
      XTPOS=(I-1)*HEIGHT*COS(ANGLE*0.01745329)+XPOS
      YTPOS=(I-1)*HEIGHT*SIN(ANGLE*0.01745329)+YPOS
      CALL SYMBOL (XTPOS,YTPOS,HEIGHT,ICHARS(I),ANGLE,1)
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RWERR
C
C     ******************************************************************
C
      COMMON /ERRORS/ ICRD,ISEG
      DIMENSION IN(80),NSEG(5,6),NPART(9)
      DATA ((NSEG(I,J),J=1,6),I=1,5) /1HS,1HP,1HE,1HE,1HD,1H2,1HP,1HL,
     2 1HO,1HT,1H1,1H ,1HP,1HL,1HO,1HT,1H2,1H ,1HP,1HL,1HO,1HT,1H3,1H ,
     3 1HP,1HL,1HO,1HT,1H4,1H /
      DATA (NPART(I),I=1,9) /1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI/
      IPART=ICRD/10
      ICRD=ICRD-IPART*10
      IF (ICRD.EQ.0) WRITE (6,20) (NSEG(ISEG,J),J=1,6),NPART(IPART)
      IF (ICRD.NE.0) WRITE (6,30) (NSEG(ISEG,J),J=1,6),NPART(IPART),ICRD
      READ (5,40,ERR=10,END=10) (IN(I),I=1,80)
      WRITE (6,50) (IN(I),I=1,80)
   10 CALL EREXIT
      STOP
C
C     *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
      ENTRY RWEOF
C
C     *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
      IPART=ICRD/10
      ICRD=ICRD-IPART*10
      IF (ICRD.EQ.0) WRITE (6,60) (NSEG(ISEG,J),J=1,6),NPART(IPART)
      IF (ICRD.NE.0) WRITE (6,70) (NSEG(ISEG,J),J=1,6),NPART(IPART),ICRD
      CALL EREXIT
      STOP
C
C
   20 FORMAT (17H0FORMAT ERROR ON ,6A1,17H  DATA CARD PART ,A1,
     2 31H     CHECK FORMAT SPECIFICATIONS)
   30 FORMAT (17H0FORMAT ERROR ON ,6A1,12H  DATA CARD ,A1,I1,
     2 31H     CHECK FORMAT SPECIFICATIONS)
   40 FORMAT (80A1)
   50 FORMAT (45H0THE FIRST CARD NOT READ DURING EXECUTION WAS,/,1X,80A1
     2,/,38H PRECEEDING DATA CARDS MAY BE IN ERROR)
   60 FORMAT (28H0END OF FILE ENCOUNTERED ON ,6A1,17H  DATA CARD PART ,
     2 A1,31H     CHECK FORMAT SPECIFICATIONS,/,
     3 38H PRECEEDING DATA CARDS MAY BE IN ERROR)
   70 FORMAT (28H0END OF FILE ENCOUNTERED ON ,6A1,12H  DATA CARD ,A1,I1,
     2 31H     CHECK FORMAT SPECIFICATIONS,/,
     3 38H PRECEEDING DATA CARDS MAY BE IN ERROR)
C
      END
```

SUBPROGRAM: BYTE UNPACK CHARACTERS PACKED SIX PER WORD

```
.
. BYTE IS A SUBROUTINE TO OBTAIN A SINGLE 6 BIT BYTE LEFT JUSTIFIED
. SPACE FILLED FROM A WORD PACKED 6 BYTES PER WORD.
. BYTE IS CALLED WITH THREE ARGUMENTS:
.
.          CALL BYTE (PACKEDWORD,BYTENUMBER,UNPACKEDCHARACTER)
.
                AXR$.
$(1).
BYTE*           L          A0,SPACE.          BLANK FILL FIRST
                S          A0,*2,X11.
                L          A1,*1,X11.         BYTE NUMBER
                EX         LOAD,A1.           GET THE CHARACTER
                S,S1       A0,*2,X11.         STORE IT
                J          4,X11.             AND RETURN
$(0).
LOAD            +          0.                 SINGLE BYTE LOAD INSTRUCTION TABLE
                L,S1       A0,*0,X11.
                L,S2       A0,*0,X11.
                L,S3       A0,*0,X11.
                L,S4       A0,*0,X11.
                L,S5       A0,*0,X11.
                L,S6       A0,*0,X11.
SPACE           '      '.
                END.
```

SUBPROGRAM: EREXIT ERROR TERMINATION

```
      .
      . EREXIT  IS A SUBROUTINE FOR ERROR TERMINATION OF THE EXECUTION OF SPEED2
      .         TO BE USED ONLY ON A UNIVAC 1100 SERIES COMPUTER.  A SIMILAR ROUTINE
      .         MUST BE WRITTEN FOR OTHER MACHINES TO PRINT AN ERROR MESSAGE AND
      .         BOMB OUT EXECUTION.
      .
                AXR$.
     $(1).
     EREXIT*    L          A0,(0777705,ERRMSG). PRINT ERROR MESSAGE AND STOP
                ER         PRINT$.
                L,U        A0,LOGMSG.           PUT ERROR MESSAGE ON TEAR SHEET
                ER         CSF$.
                ER         ERR$.                AND BOMB OUT PROGRAM EXECUTION
     $(0).
     ERRMSG     ' END OF SPEED2    ERROR EXIT'
     LOGMSG     '@LOG SPEED2 ERROR EXIT -- RESULTS MAY BE INCORRECT . '.
                END.
```

SUBPROGRAM: MAP MAP TO CREATE SPEED2

```
LIB NBS*CALCOM.
LIB NBS*JNTRAN.
SEG SPEED-2                          . MAIN PROGRAM
   IN SPEED2
SEG PLOT-1-2*,(SPEED-2)              . PLOT1, PLOT2
   IN PLOT1,PLOT2,ERROR
SEG PLOT-3-4*,(SPEED-2)              . PLOT3
   IN PLOT3,PLOT4
END
```

APPENDIX B

SAMPLE RUNSTREAMS AND EXAMPLES OF THE USE OF SPEED2

This appendix provides a number of examples of runstreams, data cards and FORTRAN code used to reduce data collected by data acquisition systems using SPEED2. Examples of the use of PLOT1, PLOT2, PLOT3, and PLOT4 are presented and described in some detail. Several examples are included describing the use of PLOT2, the most complicated phase of SPEED2.

EXAMPLE 1. DATA PRINTOUT AND ERROR CHECKING USING PLOT1

As a first step in the reduction of data using SPEED2, PLOT1 provides the ability to quickly list the data recorded by a data acquisition system, check the correctness of the data and, if desired, output the data to punched cards, mass storage, or magnetic tape. This short example illustrates the use of PLOT1.

Explanation (numbers refer to lines within the example):

1. Standard UNIVAC 1108 EXEC 8 run initiation card specifying run ID (FIRE01), account number (00000-AAAAAA), project ID (PROJECT), priority (M), and maximum run time and page estimate.

2. Assign a temporary mass storage file called 11.

3. Assign a magnetic tape (FIRE01) containing data recorded by a data acquisition system to logical unit 7. The assignment options (MTJI), and tape unit type (8C) are dependent upon the data acquisition system tape unit recording format (here, 7 track, 500 BPI, odd parity BCD).

4. Execute the program SPEED2 contained in the mass storage file SPEED2*SPEED2.

5. SPEED2 data card A1 -- specifies one data set is to be processed.

6. SPEED2 data card A2 -- specifies PLOT1 is to be called.

7. PLOT1 data card A1 -- entries specify: Esterline Angus PD2064 magnetic tape input (INTYPE=6); a printout of all input data records is to be produced from the data recorded on the magnetic tape (INPRT=-1); a copy of the input data is to be output to logical unit 11 (INPNCH=11); up to 100 error messages are to be printed (INERR=100); and no input data records are to be skipped -- do not read PLOT1 data cards D1 (INSKIP=0).

8-9. PLOT1 data cards B1 and B2 -- title of the experiment.

10. Copy the raw data written into the mass storage file called 11 into a permanent mass storage file called FRANKLIN. The data is identified by the element name FIRE01/RAW within the file.

11. UNIVAC 1108 EXEC 8 run termination card.

EXAMPLE 1: DATA PRINTOUT AND ERROR CHECKING USING PLOT1

```
@RUN,M FIRE01,00000-AAAAAA,PROJECT,1,200
@ASG,T 11.
@ASG,MTJI 70.,8C,FIRE01
@XQT SPEED2*SPEED2.SPEED2
     1
     1    0    0    0
     6   -1   11  100    0
'FRANKLIN HEATER -- FIREPLACE' FIRE TEST                    01:31:78
TEST: FIRE01
@COPY,I 11.,FRANKLIN.FIRE01/RAW
@FIN
```

EXAMPLE 2. THE REDUCTION OF DATA USING PLOT2

After the data have been processed with PLOT1, the data must be reduced and converted into scientific units for analysis. PLOT2 is used for this purpose. In the first example, data recorded on magnetic tape were processed by PLOT1 and output to a mass storage file for permanent storage. This example illustrates the use of PLOT2 to reduce data generated by PLOT1 and produce printer plots of the data. The data are to be converted with only linear equations. Hence, there is no need to use FORTRAN coding to convert the data.

<u>Explanation (numbers refer to lines within the example):</u>

1. Standard UNIVAC 1108 EXEC 8 run initiation card.

2. Copy the mass storage file SPEED2*SPEED2 containing all the subprograms of SPEED2 into the temporary workspace file TPF$.

3-5. Call the FORTRAN language processor to compile the main program SPEED2, changing the size of the data arrays to NROW=240 and NCOL=25 to allow up to 24 columns and 240 rows of data (note that one extra column is needed by SPEED2 over and above the number columns needed by the user).

6. Call the @MAP processor (the collector) to collect together the relocatable elements of the subprograms of the SPEED2 system to produce an absolute element for execution.

7. Assign a temporary mass storage file called 10.

8. Assign an existing mass storage file called FRANKLIN.

9. Execute the absolute element SPEED2 created earlier with the @MAP processor.

10-11. SPEED2 data cards A1 and A2 -- specifies one data set (NSETS=1) is to be processed by PLOT2 (IP1=0, IP2=1, IP3=0, IP4=0).

12. PLOT2 data card A1 -- entries specify: Esterline Angus punched card image format (INTYPE=7); only the first two input data records are to printed (INPRT=2); do not output the raw data (INPNCH=0); do not stop processing after reading the input data (INSTOP=0); do not print any input error messages (INERR=0); and do not skip any input data records -- do not read PLOT2 data card D1 (INSKIP=0).

13. PLOT2 data card A2 -- entries specify: a test number of zero (NTEST=0); print the transformed data matrix (NPRT=1); output the transformed data matrix to unit 10 -- in this case, a previously assigned mass storage file (NPNCH=10); read PLOT2 data cards part I to generate printer plots (NPLOT=1); and do not read PLOT2 data cards part G to correct readings of the data matrix (NCORR=0).

14-15. PLOT2 data cards B1 and B2 -- title of the experiment.

16-35. PLOT2 data cards B3 -- instrument definition cards for instruments numbered 0 - 18 and a time channel numbered 999.

36. PLOT2 data card B4 signals the end of the set of instrument defining cards.

37-39. PLOT2 data cards B5 specifying conversion coefficients for instrument types 1, 2, and 3.

40. PLOT2 data card B6 signals the end of the set of coefficient cards.

41. Element FIRE01/RAW in the mass storage program file FRANKLIN contains raw data images in the format of the data acquisition system prepared by using PLOT1 as in example 1. The @ADD control statement is used to add into the runstream at this point the entire contents of the element. These are PLOT2 data cards part E.

42-45. PLOT2 data cards I1, I2, I3, and I4, the first of several sets of four cards to generate printer plots from the reduced data. Here, channel 999 is plotted as the x-axis versus channel 1 as the y-axis. The axis limits are 0 - 5000 for the x-axis and 0 - 0.25 for the y-axis. The graph title is INCIDENT RADIANT FLUX ON BACK WALL SURFACE. Axis titles are HEAT FLUX WATTS PER SQUARE CM (y-axis) and ELAPSED TIME IN SECONDS (x-axis).

46-69. Six additional sets of four cards to generate six separate printer plots.

70. A blank card signals the end of the plot generating cards.

71. Copy the temporary mass storage file named 10 containing the reduced data output by SPEED2 (NPNCH=10) into permanent storage in the element named FIRE01/REDUCED within the mass storage program file named FRANKLIN.

72. Standard UNIVAC 1108 EXEC 8 run termination card.

```
@RUN,M FIRE01,00000-AAAAAA,PROJECT,2,100
@COPY SPEED2*SPEED2.,TPF$.
@FOR,W SPEED2
-25,25
     PARAMETER NROW=240, NCOL=25
@MAP,N MAP,SPEED2
@ASG,T 10.
@ASG,A FRANKLIN.
@XQT SPEED2
    1
    0    1    0    0
    7    2    0    0    0    0
    0    1   10    1    0
'FRANKLIN HEATER -- FIREPLACE' FIRE TEST                          01:31:78
TEST: FIRE01
  000103FLUX   HEAT FLUX RADIOMETER, REAR WALL           W/CM2
  000002TC REF REFERENCE THERMOCOUPLE                    DEG C
  000202RW 46  TEMPERATURE, REAR WALL, 0.46 M UP         DEG C
  000302RW 107 TEMPERATURE, REAR WALL, 1.07 M UP         DEG C
  000402RW 168 TEMPERATURE, REAR WALL, 1.68 M UP         DEG C
  000502RW 229 TEMPERATURE, REAR WALL, 2.29 M UP         DEG C
  000602SW 46  TEMPERATURE, SIDE WALL, 0.46 M UP         DEG C
  000702SW 107 TEMPERATURE, SIDE WALL, 1.07 M UP         DEG C
  000802SW 168 TEMPERATURE, SIDE WALL, 1.68 M UP         DEG C
  000902SW 229 TEMPERATURE, SIDE WALL, 2.29 M UP         DEG C
  001002HEARTH TEMPERATURE, HEARTH CENTER                DEG C
  001102RT LEG TEMPERATURE, BENEATH RIGHT LEG            DEG C
  001202LT LEG TEMPERATURE, BENEATH LEFT  LEG            DEG C
  001302R  LEG TEMPERATURE, BENEATH REAR  LEG            DEG C
  001402R SIDE TEMPERATURE, RIGHT SIDE OF FIREPLACE      DEG C
  001502R REAR TEMPERATURE, RIGHT REAR OF FIREPLACE      DEG C
  001602C REAR TEMPERATURE, CENTER REAR OF FIREPLACE     DEG C
  001702L REAR TEMPERATURE, LEFT REAR OF FIREPLACE       DEG C
  001802L SIDE TEMPERATURE, LEFT SIDE OF FIREPLACE       DEG C
  099901 TIME  ELAPSED TIME SINCE IGNITION               SEC
                                                                              999
C 1.0
C 0.55556        32.0
C 0.2335
                                                                              999
@ADD FRANKLIN.FIRE01/RAW
PLOT 999    1
0.0         5000.      0.0         0.25
INCIDENT RADIANT FLUX ON BACK WALL SURFACE
     HEAT FLUX  WATTS PER SQUARE CM       ELAPSED TIME IN SECONDS
PLOT 999    2 3 4 5
0.0         5000.      0.0         100.
TEMPERATURE ON REAR WALL BEHIND FIREPLACE
    TEMPERATURE  DEG C                      ELAPSED TIME IN SECONDS
PLOT 999    6 7 8 9
```

```
0.0         5000.      0.0        100.
TEMPERATURE ON SIDE WALL RIGHT OF FIREPLACE
    TEMPERATURE  DEG C                           ELAPSED TIME IN SECONDS
PLOT 999 10
0.0         5000.      0.0        100.
TEMPERATURE AT FRONT EDGE OF HEARTH
    TEMPERATURE  DEG C                           ELAPSED TIME IN SECONDS
PLOT 999    11 12 13
0.0         5000.      0.0        100.
TEMPERATURE ON FLOOR BENEATH LEGS OF FIREPLACE
    TEMPERATURE  DEG C                           ELAPSED TIME IN SECONDS
PLOT 999    14 18
0.0         5000.      0.0        500.
TEMPERATURE ON SIDES OF FIREPLACE
    TEMPERATURE  DEG C                           ELAPSED TIME IN SECONDS
PLOT 999    15 16 17
0.0         5000.      0.0        500.
TEMPERATURE ON REAR SURFACE OF FIREPLACE
    TEMPERATURE  DEG C                           ELAPSED TIME IN SECONDS

@COPY,I 10.,FRANKLIN.FIRE01/REDUCED
@FIN
```

EXAMPLE 3. USING PLOT3 TO GENERATE PEN PLOTS FROM REDUCED DATA

For reports and other presentations of data, the user may desire better quality graphs of the data reduced and stored using PLOT1 and PLOT2. In this example, PLOT3 is used to generate a pen plot from data reduced as shown in example 2.

Explanation (numbers refer to lines within the example):

1. Standard UNIVAC 1108 EXEC 8 run initiation card.

2. Copy the mass storage file SPEED2*SPEED2 containing all the subprograms of SPEED2 into the temporary workspace file TPF$.

3-5. Call the FORTRAN language processor to compile the main program SPEED2, changing the size of the data arrays to NROW=240 and NCOL=3.

6. Call the @MAP processor (the collector) to collect together the relocatable elements of the subprograms of the SPEED2 system to produce an absolute element for execution.

7. Assign an existing mass storage file (A-option) called FRANKLIN.

8-9. Assign a magnetic tape CAL to logical unit 7 to be used for the output of the CALCOMP plotting commands.

10. Execute the absolute element SPEED2 created earlier with the @MAP processor.

11-12. SPEED2 data cards A1 and A2 -- specifies one data set (NSET=1) is to be processed using PLOT3 (IP1, IP2, IP4 equal zero and IP3 equals one).

13. PLOT3 data card A1 -- entries specify: data matrix is not to be printed (NPRT=0); smooth the data before plotting using a running five point linear average (NSMTH=5); ignore all out of bounds points (NOUT=0); place axes on all four sides of each graph produced (NAXIS=1); draw a line through each data point of all curves, placing a special plotting symbol every 10 points (NPNT=10); and do not produce any legends on the graphs (NLEG=0).

14. PLOT3 data card A2 -- entries specify the size of the graphs produced. The x-axis length is 10.0 scaled inches (XSIZE=10.0); the y-axis length is 5.0 scaled inches (YSIZE=5.0); and the scaling factor is 0.8 (SCALE=0.8).

15-16. PLOT3 data cards B1 and B2 -- the title of the experiment.

17-18. PLOT3 data cards B3 -- instrument definition cards for channels 002 and 999.

19. PLOT3 data card B4 signals the end of the instrument definition cards.

20. Element FIRE01/REDUCED in the mass storage program file FRANKLIN contains the set of reduced data output by PLOT2. The @ADD control statement is used to add into the runstream at this point the entire contents of the element. These are PLOT3 data cards part C.

21-24. PLOT3 data cards D1, D2, D3, and D4 describing the pen plot to be produced. Channel 999 is plotted on the x-axis versus channel 002 on the y-axis. The axis limits are 0 - 5000 for the x-axis and 0 - 100 for the y-axis. The graph title is FIGURE 4: SURFACE TEMPERATURE ON REAR WALL BEHIND FIREPLACE. Axis titles are TEMPERATURE (DEG C) and TIME SINCE IGNITION (SEC) for the y and x axes respectively.

25. A blank card signals the end of the plot generating cards.

26. Standard UNIVAC 1108 EXEC8 run termination card.

EXAMPLE 3: USING PLOT3 TO GENERATE PEN PLOTS FROM REDUCED DATA

```
@RUN,M FIRE01,00000-AAAAAA,PROJECT,2,100
@COPY SPEED2*SPEED2.,TPF$.
@FOR,W SPEED2
-25,25
      PARAMETER NROW=240, NCOL=3
@MAP,N MAP,SPEED2
@ASG,A FRANKLIN.
@ASG,TJ 7.,8C,CALW
@MSG,W REEL CAL BELONGS TO 00000-AAAAAA
@XQT SPEED2
    1
    0    0    1    0
    0    5    0    1   10    0
10.0        5.0       0.8
'FRANKLIN HEATER -- FIREPLACE' FIRE TEST
PLOTS
0000020324 IN  TEMPERATURE ON REAR WALL BEHIND FIREPLACE
00099903TIME   TIME IN SECONDS SINCE IGNITION
                                                                              999
@ADD FRANKLIN.FIRE01/REDUCED
PLOT 999,002
0.0        5000.      0.0        100.
04SURFACE TEMPERATURE ON REAR WALL       BEHIND FIREPLACE
      TEMPERATURE  (DEG C)                  TIME SINCE IGNITION (SEC)

@FIN
```

EXAMPLE 4. COMBINING SEVERAL SETS OF DATA USING PLOT4

If a number of related tests have been processed using SPEED2, with the reduced data stored on mass storage, magnetic tape, punched cards, or the like, the user may wish to compare instruments from several different tests using PLOT2 or PLOT3. This example illustrates the use of PLOT4 to create a new combined set of reduced data that may then be further processed with PLOT2 or PLOT3.

Explanation (numbers refer to lines within the example):

1. Standard UNIVAC 1108 EXEC 8 run initiation card.

2. Copy the mass storage file SPEED2*SPEED2 containing all the subprograms of SPEED2 into the temporary workspace file TPF$.

3-5. Call the FORTRAN language processor to compile the main program to change the size of the data arrays.

6. Call the @MAP processor (the collector) to collect together the relocatable elements of the subprograms of the SPEED2 system to produce an absolute element for execution.

7. Assign a temporary mass storage file called 10.

8. Assign an existing mass storage file called FRANKLIN.

9. Execute the absolute element SPEED2 created earlier with the @MAP processor.

10-11. SPEED2 data cards A1 and A2 -- specifies one data set is to be processed using PLOT4.

12. PLOT4 data card A1 -- entries specify: do not print the combined data matrix (NPRT=0); and output the combined data matrix to unit 10 -- in this case, a previously assigned mass storage file (NPNCH=10).

13-14. PLOT4 data cards B1 and B2 -- the title of the experiment.

15-102. PLOT4 data cards B3 -- instrument definition cards for tests 1 - 4.

103. PLOT4 data card B4 signals the end of the set of instrument defining cards.

104. PLOT4 data card C1 specifies that the data which follows belongs to test number 1.

105. Element FIRE01/REDUCED in the mass storage program file FRANKLIN contains the set of reduced data for test number 1 output by PLOT2. The @ADD control statement is used to add into the run-stream at this point the entire contents of the element. These are PLOT4 data cards C2.

106-111. Three similar sets of PLOT4 data cards part C for tests 2, 3 and 4.

112. PLOT4 data card C3 specifying the end of the data sets.

113. Standard UNIVAC 1108 EXEC8 run termination card.

EXAMPLE 4: COMBINING SEVERAL SETS OF DATA USING PLOT4

```
@RUN,M FIRE,00000-AAAAA,PROJECT,2,100
@COPY SPEED2*SPEED2.,TPF$.
@FOR,W SPEED2
-25,25
      PARAMETER NROW=200, NCOL=90
@MAP,N MAP,SPEED2
@ASG,T 10.
@ASG,A FRANKLIN.
@XQT SPEED2
    1
    0    0    0    1
    0   10
'FRANKLIN HEATER -- FIREPLACE' FIRE TEST
ALL TESTS
00100103FLUX   HEAT FLUX RADIOMETER, REAR WALL          W/CM2
00100002TC REF REFERENCE THERMOCOUPLE                   DEG C
00100202RW 46  TEMPERATURE, REAR WALL, 0.46 M UP        DEG C
00100302RW 107 TEMPERATURE, REAR WALL, 1.07 M UP        DEG C
00100402RW 168 TEMPERATURE, REAR WALL, 1.68 M UP        DEG C
00100502RW 229 TEMPERATURE, REAR WALL, 2.29 M UP        DEG C
00100602SW 46  TEMPERATURE, SIDE WALL, 0.46 M UP        DEG C
00100702SW 107 TEMPERATURE, SIDE WALL, 1.07 M UP        DEG C
00100802SW 168 TEMPERATURE, SIDE WALL, 1.68 M UP        DEG C
00100902SW 229 TEMPERATURE, SIDE WALL, 2.29 M UP        DEG C
00101002HEARTH TEMPERATURE, HEARTH CENTER               DEG C
00101102RT LEG TEMPERATURE, BENEATH RIGHT LEG           DEG C
00101202LT LEG TEMPERATURE, BENEATH LEFT  LEG           DEG C
00101302R  LEG TEMPERATURE, BENEATH REAR  LEG           DEG C
00101402R SIDE TEMPERATURE, RIGHT SIDE OF FIREPLACE     DEG C
00101502R REAR TEMPERATURE, RIGHT REAR OF FIREPLACE     DEG C
00101602C REAR TEMPERATURE, CENTER REAR OF FIREPLACE    DEG C
00101702L REAR TEMPERATURE, LEFT REAR OF FIREPLACE      DEG C
00101802L SIDE TEMPERATURE, LEFT SIDE OF FIREPLACE      DEG C
00101902FLOOR  TEMPERATURE, FLOOR SURFACE CENTER        DEG C
00102002S STAK TEMPERATURE ON SIDE WALL AT STACK        DEG C
00199901 TIME  ELAPSED TIME SINCE IGNITION              SEC
00200103FLUX   HEAT FLUX RADIOMETER, REAR WALL          W/CM2
00200002TC REF REFERENCE THERMOCOUPLE                   DEG C
00200202RW 46  TEMPERATURE, REAR WALL, 0.46 M UP        DEG C
00200302RW 107 TEMPERATURE, REAR WALL, 1.07 M UP        DEG C
00200402RW 168 TEMPERATURE, REAR WALL, 1.68 M UP        DEG C
00200502RW 229 TEMPERATURE, REAR WALL, 2.29 M UP        DEG C
00200602SW 46  TEMPERATURE, SIDE WALL, 0.46 M UP        DEG C
00200702SW 107 TEMPERATURE, SIDE WALL, 1.07 M UP        DEG C
00200802SW 168 TEMPERATURE, SIDE WALL, 1.68 M UP        DEG C
00200902SW 229 TEMPERATURE, SIDE WALL, 2.29 M UP        DEG C
00201002HEARTH TEMPERATURE, HEARTH CENTER               DEG C
00201102RT LEG TEMPERATURE, BENEATH RIGHT LEG           DEG C
00201202LT LEG TEMPERATURE, BENEATH LEFT  LEG           DEG C
00201302R  LEG TEMPERATURE, BENEATH REAR  LEG           DEG C
```

51:	00201402R SIDE	TEMPERATURE, RIGHT SIDE OF FIREPLACE	DEG C
52:	00201502R REAR	TEMPERATURE, RIGHT REAR OF FIREPLACE	DEG C
53:	00201602C REAR	TEMPERATURE, CENTER REAR OF FIREPLACE	DEG C
54:	00201702L REAR	TEMPERATURE, LEFT REAR OF FIREPLACE	DEG C
55:	00201802L SIDE	TEMPERATURE, LEFT SIDE OF FIREPLACE	DEG C
56:	00201902FLOOR	TEMPERATURE, FLOOR SURFACE CENTER	DEG C
57:	00202002S STAK	TEMPERATURE ON SIDE WALL AT STACK	DEG C
58:	00299901 TIME	ELAPSED TIME SINCE IGNITION	SEC
59:	00300103FLUX	HEAT FLUX RADIOMETER, REAR WALL	W/CM2
60:	00300002TC REF	REFERENCE THERMOCOUPLE	DEG C
61:	00300202RW 46	TEMPERATURE, REAR WALL, 0.46 M UP	DEG C
62:	00300302RW 107	TEMPERATURE, REAR WALL, 1.07 M UP	DEG C
63:	00300402RW 168	TEMPERATURE, REAR WALL, 1.68 M UP	DEG C
64:	00300502RW 229	TEMPERATURE, REAR WALL, 2.29 M UP	DEG C
65:	00300602SW 46	TEMPERATURE, SIDE WALL, 0.46 M UP	DEG C
66:	00300702SW 107	TEMPERATURE, SIDE WALL, 1.07 M UP	DEG C
67:	00300802SW 168	TEMPERATURE, SIDE WALL, 1.68 M UP	DEG C
68:	00300902SW 229	TEMPERATURE, SIDE WALL, 2.29 M UP	DEG C
69:	00301002HEARTH	TEMPERATURE, HEARTH CENTER	DEG C
70:	00301102RT LEG	TEMPERATURE, BENEATH RIGHT LEG	DEG C
71:	00301202LT LEG	TEMPERATURE, BENEATH LEFT LEG	DEG C
72:	00301302R LEG	TEMPERATURE, BENEATH REAR LEG	DEG C
73:	00301402R SIDE	TEMPERATURE, RIGHT SIDE OF FIREPLACE	DEG C
74:	00301502R REAR	TEMPERATURE, RIGHT REAR OF FIREPLACE	DEG C
75:	00301602C REAR	TEMPERATURE, CENTER REAR OF FIREPLACE	DEG C
76:	00301702L REAR	TEMPERATURE, LEFT REAR OF FIREPLACE	DEG C
77:	00301802L SIDE	TEMPERATURE, LEFT SIDE OF FIREPLACE	DEG C
78:	00301902FLOOR	TEMPERATURE, FLOOR SURFACE CENTER	DEG C
79:	00302002S STAK	TEMPERATURE ON SIDE WALL AT STACK	DEG C
80:	00399901 TIME	ELAPSED TIME SINCE IGNITION	SEC
81:	00400103FLUX	HEAT FLUX RADIOMETER, REAR WALL	W/CM2
82:	00400002TC REF	REFERENCE THERMOCOUPLE	DEG C
83:	00400202RW 46	TEMPERATURE, REAR WALL, 0.46 M UP	DEG C
84:	00400302RW 107	TEMPERATURE, REAR WALL, 1.07 M UP	DEG C
85:	00400402RW 168	TEMPERATURE, REAR WALL, 1.68 M UP	DEG C
86:	00400502RW 229	TEMPERATURE, REAR WALL, 2.29 M UP	DEG C
87:	00400602SW 46	TEMPERATURE, SIDE WALL, 0.46 M UP	DEG C
88:	00400702SW 107	TEMPERATURE, SIDE WALL, 1.07 M UP	DEG C
89:	00400802SW 168	TEMPERATURE, SIDE WALL, 1.68 M UP	DEG C
90:	00400902SW 229	TEMPERATURE, SIDE WALL, 2.29 M UP	DEG C
91:	00401002HEARTH	TEMPERATURE, HEARTH CENTER	DEG C
92:	00401102RT LEG	TEMPERATURE, BENEATH RIGHT LEG	DEG C
93:	00401202LT LEG	TEMPERATURE, BENEATH LEFT LEG	DEG C
94:	00401302R LEG	TEMPERATURE, BENEATH REAR LEG	DEG C
95:	00401402R SIDE	TEMPERATURE, RIGHT SIDE OF FIREPLACE	DEG C
96:	00401502R REAR	TEMPERATURE, RIGHT REAR OF FIREPLACE	DEG C
97:	00401602C REAR	TEMPERATURE, CENTER REAR OF FIREPLACE	DEG C
98:	00401702L REAR	TEMPERATURE, LEFT REAR OF FIREPLACE	DEG C
99:	00401802L SIDE	TEMPERATURE, LEFT SIDE OF FIREPLACE	DEG C
100:	00401902FLOOR	TEMPERATURE, FLOOR SURFACE CENTER	DEG C

EXAMPLE 4: COMBINING SEVERAL SETS OF DATA USING PLOT4

```
00402002S STAK TEMPERATURE ON SIDE WALL AT STACK      DEG C
00499901 TIME  ELAPSED TIME SINCE IGNITION            SEC
                                                                          999
TEST  001
@ADD FRANKLIN.FIRE01/REDUCED
TEST  002
@ADD FRANKLIN.FIRE02/REDUCED
TEST  003
@ADD FRANKLIN.FIRE03/REDUCED
TEST  004
@ADD FRANKLIN.FIRE04/REDUCED
FILEND
@FIN
```

EXAMPLE 5. USING PLOT2 AND SUBPROGRAM CONV FOR COMPLICATED DATA TRANSFORMATION

The reduction of data using SPEED2 can, of course, be considerably more complex than presented in example 2. In this example, subprogram CONV is used with FORTRAN code to provide complicated data reduction and analysis.

Explanation (numbers refer to lines within the example):

1. Standard UNIVAC 1108 EXEC 8 run initiation card.

2. Copy the mass storage file SPEED2*SPEED2 containing all the subprograms of SPEED2 into the temporary workspace file TPF$.

3-5. Call the FORTRAN language processor to compile the main program SPEED2, changing the size of the data arrays to NROW=51 and NCOL=36 to allow up to 36 columns and 51 rows of data (note that one extra column is needed by SPEED2 over and above the number columns needed by the user).

6-7. Call the FORTRAN language processor to compile the subprogram CONV to provide FORTRAN changes for data reduction. Insert the FORTRAN code that follows in the runstream after line 17.

8-9. Call subprogram FIND to determine the location of instrument number 126. The subscript (column) of the array REED containing the readings from instrument number 126 is returned in the variable L126.

10. Establish the variable MAXROW as the maximum number of readings from instrument number 126. This variable will be used throughout as the maximum number of readings for all instruments.

11-14. Convert the readings of instrument number 126 with a simple FORTRAN do loop.

15-34. Similar conversions for other instruments.

35-36. A more complicated conversion on several instruments by using an additional FORTRAN DO loop with instruments numbered 129 through 134.

37-43. Convert the millivolt output for these instruments into meaningful units.

44-46. Create new instruments (which must be defined in instrument cards) numbered 329 through 334 and set the maximum number of readings for these instruments by letting MAXR(i) equal MAXROW for these instruments.

47-54. Define the readings for these user created instruments as conversions of readings from instruments scanned by the data acquisition system.

55-97. Similar conversions for other instruments.

98-121. Create four new instruments as different conversions from a single instrument.

122. Call the @MAP processor (the collector) to collect together the relocatable elements of the subprograms of the SPEED2 system to produce an absolute element for execution.

123. Assign a temporary mass storage file called 10.

124. Execute the absolute element SPEED2 created earlier with the @MAP processor.

125-130. SPEED2 data cards A1 and A2, PLOT2 data cards A1, A2, B1 and B2.

131-165. PLOT2 data cards B3 -- instrument definition cards for instruments to be processed.

166. PLOT2 data card B4 signals the end of the set of instrument defining cards.

167-170. PLOT2 data cards B5 specifying conversion coefficients for instrument types 1, 2, 3, and 4.

171. PLOT2 data card B6 signals the end of the set of coefficient cards.

172. Element TEST17 in the mass storage program file RAW-DATA contains raw data images in the format of the data acquisition system prepared by using PLOT1 or PLOT2. The @ADD control statement is used to add into the runstream at this point the entire contents of the element. These are PLOT2 data cards part E.

173-216. Several sets of four cards to generate separate printer plots.

217. A blank card signals the end of the plot generating cards.

218. Copy the temporary mass storage file named 10 containing the reduced data output by SPEED2 (NPNCH=10) into permanent storage in the element named TEST17 within the mass storage program file named REDUCED-DATA.

219. Standard UNIVAC 1108 EXEC 8 run termination card.

```
@RUN,M TEST17,00000-AAAAAA,PROJECT,4,200
@COPY SPEED2*SPEED2.,TPF$.
@FOR,S SPEED2
-25,25
      PARAMETER NROW=51, NCOL=36
@FOR,S CONV
-17
C *** 2% CO *** LIRA 32062 ***
      CALL FIND(126,L126,KH,NCOL)
      MAXROW=MAXR(L126) @ ESTABLISH MAXROW FOR ALL INSTRUMENTS
      DO 50 L=1,MAXROW
      Y=REED(L,L126)
      REED(L,L126)=-.000435+.014073*Y+.000033*(Y**2)
   50 CONTINUE
C *** 10% CO *** LIRA 30759 ***
      CALL FIND(123,L123,KH,NCOL)
      DO 60 L=1,MAXROW
      Y=REED(L,L123)
      IF(Y.LE.1.)REED(L,L123)=.0769231*Y
      IF(Y.GT.1.)
     1 REED(L,L123)=-.042538+.077776*Y+.000130*(Y**2)+.000001*(Y**3)
   60 CONTINUE
C *** 20% CO2 *** LIRA 32371 ***
      CALL FIND(121,L121,KH,NCOL)
      DO 70 L=1,MAXROW
      Y=REED(L,L121)
      REED(L,L121)=.1207022+.1406010*Y+.0004877*(Y**2)+.0000010*(Y**3)
   70 CONTINUE
C *** 5% CO *** BECKMAN 1521-E ***
      CALL FIND(128,L128,KH,NCOL)
      DO 100 L=1,MAXROW
      Y=REED(L,L128)
      REED(L,L128)=0.05*Y
  100 CONTINUE
C *** SMOKE METERS ***
      DO 90 I=129,134
      CALL FIND(I,LINST,KH,NCOL)
      ORIGNL=REED(1,LINST)
      DO 80 L=1,MAXROW  @ DETERMINE % TRANSMISSION
      REED(L,LINST)=REED(L,LINST)/ORIGNL*100.
      IF (REED(L,LINST).GT.100.) REED(L,LINST)=100.
      IF (REED(L,LINST).LT.0.) REED(L,LINST)=0.
   80 CONTINUE
      J=I+200
      CALL FIND(J,JINST,KH,NCOL)
      MAXR(JINST)=MAXROW
      PATH=0.7684
      IF(J.GT.331) PATH=0.7430
      DO 85 L=1,MAXROW  @ DETERMINE OD/M
      TRANS=REED(L,LINST)
```

```
      IF(TRANS.LT.0.001) TRANS=0.001
      REED(L,JINST)=(ALOG10(100./TRANS))/PATH
   85 CONTINUE
   90 CONTINUE
C *** O2 READINGS ***
      DO 130 K=125,127
      I=K
      IF(K.EQ.126) I=124  @ DO CHANNELS 124, 125, 127
      CALL FIND(I,LINST,KH,NCOL)
      ORIGNL=REED(1,LINST)
      DO 120 L=1,MAXROW   @ COMPUTE % CONCENTRATION
      REED(L,LINST)=REED(L,LINST)*21./ORIGNL
  120 CONTINUE
      J=I+200
      CALL FIND(J,JINST,KH,NCOL)
      MAXR(JINST)=MAXROW
      DO 125 L=1,MAXROW   @ COMPUTE (CHANGE IN % CONCENTRATION) = (% DEPLETION)
      REED(L,JINST)=21.0-REED(L,LINST)
  125 CONTINUE
  130 CONTINUE
C
C *** BURN RATE OF CRIB (PERCENT LOSS) ***
C
C                  ORIG WT - CURRENT WT
C    PCT LOSS = -------------------- X 100.
C                        ORIG WT
C
      CALL FIND(135,L135,KH,NCOL)
      CALL FIND(305,L305,KH,NCOL)
      MAXR(L305)=MAXROW
      ORIGNL=REED(1,L135)
      DO 190 L=1,MAXROW
  190 REED(L,L305)=(ORIGNL-REED(L,L135))/ORIGNL*100.
C
C *** BURN RATE OF CRIB (PERCENT LOSS PER SECOND) ***
C
C    PCT LOSS   CHANGE IN PCT LOSS OVER EACH 10 SEC INTERVAL
C    -------- = --------------------------------------------
C      SEC                          10.
C
      CALL FIND(306,L306,KH,NCOL)
      MAXR(L306)=MAXROW
      ORIGNL=REED(1,L135)
      REED(1,L306)=0.
      DO 200 L=2,MAXROW
      LM1=L-1
  200 REED(L,L306)=(REED(L,L305)-REED(LM1,L305))/10.
C
C *** BURN RATE OF CRIB (GRAMS/SEC) ***
C
```

```
            CALL FIND(307,L307,KH,NCOL)  @ LOAD CELL READINGS IN GRAMS
            CALL FIND(308,L308,KH,NCOL)  @ 10 SECOND INTERVAL
            CALL FIND(309,L309,KH,NCOL)  @ 30 SECOND INTERVAL
            CALL FIND(310,L310,KH,NCOL)  @ 60 SECOND INTERVAL
            MAXR(L307)=MAXROW
            MAXR(L308)=MAXROW
            MAXR(L309)=MAXROW
            MAXR(L310)=MAXROW
      C   CONVERT LOAD CELL READINGS FROM POUNDS TO GRAMS
            DO 210 L=1,MAXROW
      210   REED(L,L307)=REED(L,L135)*453.6
      C   DETERMINE BURN RATE (G/S) OVER A 10-SEC, 30-SEC, 60-SEC INTERVAL
            DO 220 L=2,MAXROW
            LM1=L-1
      220   REED(L,L308)=(REED(LM1,L307)-REED(L,L307))/10.  @ 10-SEC INTERVAL
            DO 230 L=4,MAXROW
            LM3=L-3
      230   REED(L,L309)=(REED(LM3,L307)-REED(L,L307))/30.  @ 30-SEC INTERVAL
            DO 240 L=7,MAXROW
            LM6=L-6
      240   REED(L,L310)=(REED(LM6,L307)-REED(L,L307))/60.  @ 60-SEC INTERVAL
      @MAP,S MAP,SPEED2
      @ASG,T 10.
      @XQT SPEED2
          1
          0    1    0    0
          3   -1    0    0    0    0
          0    1   10    1    0
          ***    MOBILE HOME FIRE TEST 17    ***    MAY 15, 1978    ***

      00007102TC    1     TC- LIVINGRM,CENTR    1 INCH FROM CEILING
      00007202TC    2     TC- LIVINGRM,CENTR   10 INCH FROM CEILING
      00007302TC   23     TC- CORRIDOR,NORTH    1 INCH FROM CEILING
      00007402TC   37     TC- CORRIDOR,SOUTH   32 INCH FROM CEILING
      00012004RAD   1     RADIOMETER- LIVINGRM,CENTR  FLOOR LEVEL    #54511
      00012902HSM   4     HORIZ SMOKE METER- CORRIDOR,NORTH  2 FT ABV FLR (% TRANS)
      00013002HSM   5     HORIZ SMOKE METER- CORRIDOR,NORTH  4 FT ABV FLR (% TRANS)
      00013102HSM   6     HORIZ SMOKE METER- CORRIDOR,NORTH  6 FT ABV FLR (% TRANS)
      00013202HSM   8     HORIZ SMOKE METER- CORRIDOR,SOUTH  2 FT ABV FLR (% TRANS)
      00013302HSM   9     HORIZ SMOKE METER- CORRIDOR,SOUTH  4 FT ABV FLR (% TRANS)
      00013402HSM  10     HORIZ SMOKE METER- CORRIDOR,SOUTH  6 FT ABV FLR (% TRANS)
      00032902HSM   4     HORIZ SMOKE METER- CORRIDOR,NORTH  2 FT ABV FLR (OD/M)
      00033002HSM   5     HORIZ SMOKE METER- CORRIDOR,NORTH  4 FT ABV FLR (OD/M)
      00033102HSM   6     HORIZ SMOKE METER- CORRIDOR,NORTH  6 FT ABV FLR (OD/M)
      00033202HSM   8     HORIZ SMOKE METER- CORRIDOR,SOUTH  2 FT ABV FLR (OD/M)
      00033302HSM   9     HORIZ SMOKE METER- CORRIDOR,SOUTH  4 FT ABV FLR (OD/M)
      00033402HSM  10     HORIZ SMOKE METER- CORRIDOR,SOUTH  6 FT ABV FLR (OD/M)
      00012302CO   13     CO  - CENTR OF LIVINGRM AT 5 FT ABOVE FLOOR  LIRA 30759
      00012102CO2  14     CO2 - CENTR OF LIVINGRM AT 5 FT ABOVE FLOOR  LIRA 32371
      0001240202   15     O2  - CENTR OF LIVINGRM AT 5 FT ABOVE FLOOR
```

```
0001250202  24     02  - DOOR  OF LIVINGRM AT 1 IN ABOVE FLOOR
00012602CO  16     CO  - SOUTH CORRIDOR    AT 5 FT ABOVE FLOOR  LIRA 32062
0001270202  17     02  - SOUTH CORRIDOR    AT 5 FT ABOVE FLOOR
00012802CO  25     CO  - SOUTH CORRIDOR    AT 3 FT ABOVE FLOOR
0003240202  15     02  - CENTER OF LIVINGRM, 5 FT ABV FLR (CHG IN CONC)
0003250202  24     02  - DOOR OF LIVINGRM,   1 IN ABV FLR (CHG IN CONC)
0003270202  17     02  - SOUTH CORRIDOR,     5 FT ABV FLR (CHG IN CONC)
00013503L CELL     LOAD CELL (POUNDS)
00030502% LOSS     BURN RATE IN PERCENT LOSS
00030602%/SEC      BURN RATE IN PERCENT LOSS PER SECOND
00030702GRAMS      LOAD CELL (GRAMS)
00030802G/S 10     BURN RATE IN GRAMS PER SECOND (10-SEC INTERVAL)
00030902G/S 30     BURN RATE IN GRAMS PER SECOND (30-SEC INTERVAL)
00031002G/S 60     BURN RATE IN GRAMS PER SECOND (60-SEC INTERVAL)
00000001TIME       TIME IN SECONDS AFTER IGNITION
                                                                              999
 1.0
 1.0
 .0940               660.0
 2.69
                                                                              999
@ADD RAW-DATA.TEST17
PLOT 0000 0071 0072 0073 0074
0.0        1250.      0.0        1000.
THERMOCOUPLES- LIV RM CENTER (1,10 IN), CORR N (1 IN), CORR S (32 IN)
                DEGREES C                          SECONDS AFTER IGNITION
PLOT 0000 0120
0.         1250.      0.         5.
RADIOMETER- CENTER OF LIVING ROOM, FLOOR LEVEL
           WATTS PER SQUARE CM                     SECONDS AFTER IGNITION
PLOT 0000 0329 0330 0331
0.         1250.      0.         7.5
SMOKE METERS- CORRIDOR, NORTH (2,4,6 FT ABOVE FLOOR)
         OPTICAL DENSITY PER METER                 SECONDS AFTER IGNITION
PLOT 0000 0332 0333 0334
0.         1250.      0.         7.5
SMOKE METERS- CORRIDOR, SOUTH (2,4,6 FT ABOVE FLOOR)
         OPTICAL DENSITY PER METER                 SECONDS AFTER IGNITION
PLOT 0000 0124 0125 0127
0.         1250.      0.         25.
02- LIV RM CENTER (5 FT), LIV RM DOOR (1 IN), CORR SOUTH (5 FT)
       PERCENT GAS IN THE ATMOSPHERE               SECONDS AFTER IGNITION
PLOT 0000 0123 0121
0.         1250.      0.         20.
 10% CO, 20% CO2- LIVING ROOM CENTER (5 FT ABOVE FLOOR)
       PERCENT GAS IN THE ATMOSPHERE               SECONDS AFTER IGNITION
PLOT 0000 0126 0128
0.         1250.      0.         5.
 2% CO (5 FT UP), 5% CO (3 FT UP) - CORRIDOR SOUTH
       PERCENT GAS IN THE ATMOSPHERE               SECONDS AFTER IGNITION
```

```
PLOT 0000 0135
0.          1250.       0.        40.
LOAD CELL
          CRIB WEIGHT IN POUNDS                        SECONDS AFTER IGNITION
PLOT 0000 0308
0.         1250.      0.        50.
BURN RATE IN GRAMS PER SECOND (10-SECOND INTERVAL)
            GRAMS PER SECOND                           SECONDS AFTER IGNITION
PLOT 0000 0309
0.          1250.       0.        50.
BURN RATE IN GRAMS PER SECOND (30-SECOND INTERVAL)
            GRAMS PER SECOND                           SECONDS AFTER IGNITION
PLOT 0000 0310
0.          1250.       0.        50.
BURN RATE IN GRAMS PER SECOND (60-SECOND INTERVAL)
            GRAMS PER SECOND                           SECONDS AFTER IGNITION

@COPY,I 10.,REDUCED-DATA.TEST17
@FIN
```

NBS-114A (REV. 9-78)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS Tech Note 1108	2. Gov't. Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

SPEED2, A Computer Program for the Reduction of Data from Automatic Data Acquisition Systems

5. Publication Date

September 1979

6. Performing Organization Code

7. AUTHOR(S)

Richard D. Peacock and John M. Smith

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, DC 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

13. Type of Report & Period Covered

Final

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

[X] Document describes a computer program; SF-185, FIPS Software Summary, is attached.

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

The voluminous amount of data that can be collected by automatic data acquisition systems requires the use of a digital computer for the reduction of data. A general purpose computer program for the reduction of data collected by automatic data acquisition systems is presented. The program is written with the ability to accept data from a number of different data acquisition systems, with the ability to check the correctness of data included. Through the use of FORTRAN computer programming, the data can be converted to meaningful scientific and engineering units. The data can then be presented in tabular, printer plot or ink pen plot form.

The program is documented, and detailed instructions for its use, with examples, are presented. The use of SPEED2 requires some knowledge of FORTRAN programming language and the executive control language for the computer system in use.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Computer program; data reduction; data acquisition systems; plotting.

18. AVAILABILITY	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PRINTED PAGES
[X] Unlimited	UNCLASSIFIED	153
[ ] For Official Distribution. Do Not Release to NTIS	20. SECURITY CLASS (THIS PAGE)	22. Price
[X] Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402, SD Stock No. SN003-003-02112-1	UNCLASSIFIED	$4.75
[ ] Order From National Technical Information Service (NTIS), Springfield, VA. 22161		

USCOMM-DC

Waste Heat Management Guidebook

A typical plant can save about 20 percent of its fuel—just by installing waste heat recovery equipment. But with so much equipment on the market, how do you decide what's right for you?

Find the answers to your problems in the *Waste Heat Management Guidebook*, a new handbook from the Commerce Department's National Bureau of Standards and the Federal Energy Administration.

The *Waste Heat Management Guidebook* is designed to help you, the cost-conscious engineer or manager, learn how to capture and recycle heat that is normally lost to the environment during industrial and commercial processes.

The heart of the guidebook is 14 case studies of companies that have recently installed waste heat recovery systems and profited. One of these applications may be right for you, but even if it doesn't fit exactly, you'll find helpful approaches to solving many waste heat recovery problems.

In addition to case studies, the guidebook contains information on:

- sources and uses of waste heat
- determining waste heat requirements
- economics of waste heat recovery
- commercial options in waste heat recovery equipment
- instrumentation
- engineering data for waste heat recovery
- assistance for designing and installing waste heat systems

To order your copy of the *Waste Heat Management Guidebook*, send $2.75 per copy (check or money order) to Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. A discount of 25 percent is given on orders of 100 copies or more mailed to one address.

The *Waste Heat Management Guidebook* is part of the EPIC industrial energy management program aimed at helping industry and commerce adjust to the increased cost and shortage of energy.

U.S. DEPARTMENT OF COMMERCE/National Bureau of Standards
FEDERAL ENERGY ADMINISTRATION/Energy Conservation and Environment

☆U.S. GOVERNMENT PRINTING OFFICE: 1979-281-067/226

NBS TECHNICAL PUBLICATIONS

PERIODICALS

NONPERIODICALS

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards