N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE

"Made available under NASA sponsorship
In the interest of early and wide dissemination of Earth Resources Survey
Program information and without liability
for any use made thereof."

JSC-12602
8.0-10181
NASA CR-
160673

PROGRAM DOCUMENTATION FOR MODIFICATIONS TO THE CLASY PROGRAM

Job Order 81-127
CPD 713

(E80-10181) MODIFICATIONS TO THE CLASY PROGR M (Lockheed Electronics Co.) 102 p
HC A06/MF A01 CSCL 05B

N80-28782

Unclas
G3/43 00181

Prepared By

Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15200
For
EARTH OBSERVATIONS DIVISION

JUN 1980
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

National Aeronautics and Space Administration
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas
April 1977

LEC-10481

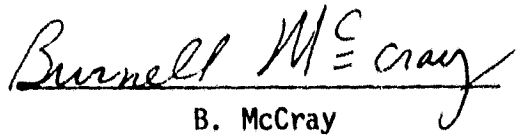
JSC-12602

MODIFICATIONS TO THE CLASY PROGRAM

Job Order 81-127

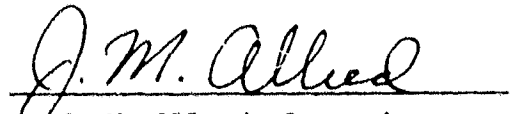
(TIRF 76-0076)

PREPARED BY



B. McCray

APPROVED BY



J. M. Allred, Supervisor
Physical Sciences Section

W. J. Reicks, Manager
Applied Mechanics Department

Prepared By
Lockheed Electronics Company, Inc.
For
EARTH OBSERVATIONS DIVISION
National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

April 1977

LEC-10481

CONTENTS

PRECEDING PAGE BLANK NOT FILMED

CONTENTS (CONT'D)

1. SCOPE

This specification establishes the modifications to the CLASY program as specified in IDSD Category 1 Job Order 81-127 Task Agreement, titled CLASY Program Modification.

2. APPLICABLE DOCUMENTS

The following documents form a part of this specification:

J. O. 81-127 Task Agreement titled: CLASY Program Modification

TIRF 76-0076

Memorandum dated March 17, 1976, from Rice University, Institute of Computer Sciences (D. L. Van Rooy) to Ken Baker/TF3, NASA-JSC; Reference: documentation of SUPER-SCRAM

3. SYSTEM DESCRIPTION

3.1 HARDWARE DESCRIPTION

Not applicable--see section 4 for the operational configuration required by the modified CLASY program.

3.2 SOFTWARE DESCRIPTION

The CLASY system of subprograms was originated by Dr. Michael Rassbach, a post-doctoral contractor for NASA-JSC, Earth Observations Division (TF).

The purpose for his development of CLASY was the implementation of an iterative statistical clustering algorithm which had theoretical promise for application to classification of earth resources (image) data acquired from the LANDSAT (formerly ERTS) satellite.

These modifications to CLASY are (1) to improve the program execution time, especially input/output overhead, in order to make it feasible to test and evaluate the CLASY program, and (2) to implement a randomized input data scrambling technique which was obtained from Rice University by the Earth Observations Division's Research, Test and Evaluation (RT&E) Branch (TF3). The scrambling of input data vectors is a necessity in CLASY due to the algorithm's sensitivity to correlated data values.

The CLASY system of subprograms consists of the main driver program, CLASY, and 56 subprograms, not including the Univac system routines utilized by the program. Four of the subprograms (LOCK, GET, FREE, and BYTRAN) were originally programmed in Univac assembly language (SLEUTH II), the remaining subprograms and the main program, CLASY, are converted to the Univac Fortran V language from the former (original) Univac reentrant Fortran language, RFOR.

The driver program for the clustering system is CLASY. The data handling subprogram for the system is CLASY1 (with an alternate entry, CLASY2). CLASY1 is reprogrammed to implement the required data input improvement and to implement the randomized data scrambling technique. CLASY1 provides the data setup for acquisition on demand by the iterative statistical subprogram, STATIS. STATIS obtains the data from CLASY1 via the alternate entry into the CLASY1 subprogram, which is CLASY2. STATIS initiates the clustering procedure, operating on one pixel (data vector) at a time in setting up clusters and making the cluster split/combine decisions. Each pixel is examined 10 times by STATIS during the clustering procedure.

To implement the required modifications and the modifications suggested by the program originator to improve the program's reliability, the following routines in the CLASY clustering system received changes: CLASY, CLASY1, ADJUST, LOCK (formerly LOC), CLPR, CLDUMP, MISH, SETUP9, CMBK10, ELIM, MULTI, SEPER, TR, STATIS, CLUST.

The modified subprograms are discussed in the order of their use by the CLASY system.

The overall CLASY system is flowcharted in Appendix A. Listings of the modified routines are shown in Appendix B. Sample output from the CLASY system is shown in Appendix C.

3.2.1 SOFTWARE COMPONENT NO. 1 (CLASY)

3.2.1.1 Linkages

CLASY is the driver program of the CLASY clustering system. CLASY calls SETUP9, READTP, CLASY1, MULTI, and CLUMSMP.

3.2.1.2 Interfaces

The common blocks INFORM, CLUSTR, CLUS, MISC, and STPAR and calling arguments are used in the program CLASY as interfaces with other routines in the clustering system.

3.2.1.3 Inputs

CLASY calls SETUP9, which reads the input supervisor (control) cards. The supervisor cards and their functions are described in the discussion of SETUP9 (section 3.2.2), and on the description of CLASY input (section 4.2.1).

3.2.1.4 Outputs

None.

3.2.1.5 Storage Requirement

Storage used: Code = 146_8 Data = 47047_8

3.2.1.6 Description

CLASY is the driver program for the clustering routines. It was rewritten to (1) enable CLASY1 to retrieve large blocks of data (dimensioned ARRAY (20 000)) from drum (2) to pre-calculate the amount of data RREAD (drum read subroutine) will read from drum, and (3) to make DATAB (the array containing the scrambled data) and PV (the array passed to the clustering routine, STATIS) to reside in the same locations as the large data array, "ARRAY". All other logic in the program remains the same as in the pre-modified version of CLASY.

3.2.1.7 Flowchart

See Appendix A.

3.2.1.8 Listings

See Appendix B for program.

3.2.2 SOFTWARE COMPONENT NO. 2 (SETUP9)

3.2.2.1 Linkage

SETUP9 is called from CLASY. SETUP9 calls NXTCHR and NUMBER, which are entry points in subroutine FIND.

3.2.2.2 Interface

Interface is accomplished through calling arguments and the following common blocks: INFORM, SUPCUM, and CLUSTR.

3.2.2.3 Inputs

See Appendix C.

3.2.2.4 Output

SETUP9 prints out a summary of the input to CLASY, and also prints an error message on the line printer if an invalid input card is detected. If an error is detected, SETUP9 prints the following message "INVALID INPUT CARD--IGNORED", processing continues.

3.2.2.5 Storage Requirement

Storage used: Code = 275_8 Data = 157_8

3.2.2.6 Description

SETUP9 reads and analyzes all cards input to the CLASY program. SETUP 9 was modified to add "0" (zero) as a new symbol to the symbol array "SMBLS". The following control cards are input to the modified CLASY program, to be analysed by SETUP9. In all cards, the "keyword" begins in card column 1, and any parameters on the card are placed from card columns 11 through 72, inclusive.

1. "CHANNEL" CARD (i.e., "CHANNEL 1,5,9,13")

The "CHANNEL" card specifies the channel numbers to be used in clustering the multi-channel data vectors. At present the maximum number of channels allowed to specify is four. The identification "CHANNEL" starts in column 1, and the actual channel numbers, separated by commas, start in card column 11, and must be terminated by column 72.

2. "PRINT" CARD (i.e., "PRINT 1,3,3")

The "PRINT" card specifies how to print the cluster map. The identification "PRINT" starts in column 1, the actual print parameters start in card column 11, separated by commas, and ending by column 72.

3. "HED1" card

4. "HED2" card

These two cards may be used to specify any arbitrary heading for the printer output, including the cluster map. Any alphanumeric characters put into card columns 11-72 of these two cards will be output as a page heading.

5. "NPOS" card

This card, used previously in the unmodified CLASY program, specified the number of positions to skip to read a pixel point (i.e. to "scramble" the data). This card is not used in the modified CLASY program.

6. "NPTS" card

This card, used previously in the unmodified CLASY program, specified the number of pixels to retrieve from the data set at each point. This card is not used in the modified CLASY program. (NPTS was used in the original programs data scrambling technique).

7. "DATE" card

This card is used to specify the date or any eight characters. Will be printed at the upper right hand corner of each page of printer output.

8. "COMMENT" card

The "COMMENT" card is equivalent in use and format with the "HED1" and "HED2" cards, described above.

9. "*END*" card

This card specifies the end of all supervisor (control) card (described above) input to CLASY. This card is a mandatory input to CLASY, to initiate the clustering process.

3.2.2.7 Flowchart

See Appendix A.

3.2.2.8 Listings

See Appendix B for the modified program listing.

3.2.3 SOFTWARE COMPONENT NO. 3 (CLASY1)

3.2.3.1 Linkage

CLASY1 is the primary entry point, and is called from CLASY. CLASY2 (an alternate entry point in CLASY1) is called from STATIS. CLASY1 calls RREAD, CLDUMP, CMERR, and ZOR (Function ZOR is the random number generator used in the data scrambling technique).

3.2.3.2 Interface

Interface is accomplished through calling arguments and the following common blocks: INFORM, CLUSTR, CLUS, MISC, and STPAR.

3.2.3.3 Inputs

None.

3.2.3.4 Output

CLASY1 outputs an error message if RREAD attempts to read data from an illegal drum address. The following message is printed "***ERROR***RREAD TRY TO READ THE ILLEGAL ADDRES',____

3.2.3.5 Storage Requirement

Storage used: Code = 537_8 Data = 66_8

3.2.3.6 Description

CLASY1 performs the input image data-handling function for the CLASY clustering system and makes the image data available to the iterative statistical subprogram, STATIS. The original image data from the area on the input tape (file) which has been designated by the input field-definition card(s), is prestored on drum as one continuous block of data. The Univac random file access routines -- RINIT, RWRITE -- are utilized to place the input data on the drum as the data is read from the input tape (file) by the TAPERD subprogram.

CLASY1 precomputes the base addresses for three data-buffering arrays in core-storage. One data-buffering array, O, is used to retrieve a block of original image data from the prestored drum.

Another array, A, is used to contain a set of integers -- 1,2,3,..., N where N=the number of data vectors in the original image array, O. The third array, D_S, will contain a set of data vectors obtained from the original image set, O, but stored in D_S, such that each data vector's original spatial location is randomly rearranged. The data scrambling technique utilized in placing the rearranged data vectors into D_S, was obtained from Rice University via RT&E and is the required modification implemented by this modification to the CLASY clustering system. The implementation of the randomized data scrambling technique is performed as follows:

(1) Given an array of original image data vectors, O, an array, A, of integers -- 1,2,3,.... , N with N = the number of data vectors in O, and an array, D_S, for storage of scrambled data vectors obtained from O.

(2) Scramble the elements of A

a. Obtain a random number, Z_i, from the uniform random number generator;

$$Z_i = ZOR(0), \quad 0. \leq Z_i \leq 1.0$$

b. Multiply the random number Z_i, by N, the largest integer in A;

$$IX_i = N \times Z_i + 1$$

c. Using IX_i as an index, scramble the integers in A as follows:

$$TEMP = A_i$$

$$A_i = IX_i$$

$$A_{IX_i} = TEMP$$

d. Execute the above procedure (a-c) N times, with

$$i = N, N-1, N-2, \ldots, 1$$

e. Create the new (scrambled) set of data vectors, D_S, as follows;

$$\text{for } i = 1, 2, 3, \ldots, N$$

$$D_{S_i} = O(j), \text{ where } j = A_i$$

The scrambled data set, D_S, is made available in large blocks for retrieval on demand from the statistical clustering subprogram, STATIS, via the alternate entry in CLASY1, which is CLASY2. The scrambling of data for STATIS is necessitated by the clustering algorithm's sensitivity to correlated data.

The buffering of blocks of data from the drum to the original data array, O, is done in parallel with the use of scrambled data from D_S by STATIS. The Univac random file access routine, RREAD, is used to transfer the data from drum to original image array, O.

The data buffering technique as described accomplished the second objective of the modifications to the CLASY clustering system--namely, the improvement of program execution time.

3.2.3.7 Flowchart

See Appendix A.

3.2.3.8 Listings

See Appendix B for program listing.

3.2.4 SOFTWARE COMPONENT NO. 4 (MULTI)

3.2.4.1 Linkage

MULTI is called from CLASY. MULTI calls DATFIX, ALFREE, CLINIT, STATIS and CLDUMP.

3.2.4.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, STPAR, INFORM and CLUSTR.

3.2.4.3 Inputs

None.

3.2.4.4 Output

Upon leaving this subroutine, a message is printed.

"EXIT FROM MULTI"

3.2.4.5 Storage Requirements

Storage used: Code = 111_8 Data = 15_8

3.2.4.6 Description

MULTI calls the routines to initialize the clustering algorithm. MULTI was modified to make use of unused areas in core. ARRAY (EXARAY) was previously dimensioned but never used. A method was devised to make use of this unused core such that the PV array and DATAB array (array utilized in STATIS and the array containing scrambled data vectors) be made to utilize the same area of core (i.e., the PV and DATAB arrays were made equivalent storage areas, in ARRAY (EXARAY)).

3.2.4.7 Flowchart

See Appendix A.

3.2.4.8 Listings

See Appendix B for program listing.

3.2.5 SOFTWARE COMPONENT NO. 5 (STATIS)

3.2.5.1 Linkages

STATIS is called by MULTI. STATIS calls DISC, CLASY2, CORECT, DOTSQ, VPV, VMTV, MPVS, ADJUST, CLDUMP, and EXP.

3.2.5.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, STPAR, CLUSTR, and RAND.

3.2.5.3 Inputs

None.

3.2.5.4 Outputs

STATIS outputs two warning messages. They are: "***WARNING ON THE__ INDEX(KL)=", "**SUSPECTED BAD DATA POINT --STATIS**IDO=__,ROOT__, VECTOR__"

3.2.5.5 Storage Requirements

Storage used: Code = 1205_8 Data = 260_8

3.2.5.6 Description

STATIS takes each input data vector and classifies it on a fractional, probabilistic basis. It then updates the various statistical parameters associated with the classes (clusters) indicated and checks to see if any of these classes is potentially two. Those which are will be referred to the routine "SPLIT". The one modification made to STATIS was the re-dimensioning of the PV array, which contains the data vectors to be clustered.

3.2.5.7 Flowchart

See Appendix A.

3.2.5.8 Listings

See Appendix B for program listing.

3.2.6 SOFTWARE COMPONENT NO. 6 (ADJUST)

3.2.6.1 Linkage

Adjust is called from STATIS. ADJUST calls GET, TR, DOTSQ, SQMTX, MINV, UNIF, CLPR, TRIMTX, DENCAL, SPLIT, FREE, CLDUMP, SEPER, SUBLIM, ELIM, CORECT, JOIN, APRIOR, SQRT, ALDG, EXP, and XPRI.

3.2.6.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, STPAR, CLUSTR, and JOINPR.

3.2.6.3 Inputs

None.

3.2.6.4 Outputs

ADJUST prints out three brief messages concerning statistical information and three error messages. They are; "ADJUST__ WEIGHT__ WAS__ SPFAC__ CHANGE__ __ __", "STATISTICS: TRACE__ SKEW__ KURT__ TESTS (SPLIT >0): __ __ __", "###HAVE SPLIT__ WEIGHT__ SUBS__ __", "W/OVOL ERROR IN ADJUST: KL,W,NEW W,VOL __ __ __ __", "***EXTRAPOLATION PROBLEM IN ADJUST: ITER, INDEX(KL), VOLIN, OVOL, CVOL__ __ __ __ __", "LOG ERROR IN ADJUST: I, IM, KL, K/VRIN= __ __ __ __ __ __"

3.2.6.5 Storage Requirements

Storage used: Code = 2305_8 Data = 332_8

3.2.6.6 Description

One modificatio made to ADJUST was to eliminate the use of subscripted subscripts for those arrays which previously presumed the RFOR (reentrant Fortran) compiler. The change was made to enable the routine to compile under the Univac 1110-EXEC 8 Fortran V compiler. The second change made was that input to the "ALOG" routine is forced to be positive by use of the absolute value of the input being sent to "ALOG". The reason for the change was that an occasional negative value was being sent to "ALOG", causing an error termination. The routine previously referenced a dimensioned variable in several calling statements causing an error during program execution. These errors were corrected. Other modifications which were suggested by the CLASY program originator, Dr. Mike Rassbach, are as follows (where "-N" is Fortran compiler change card format, "-" indicates that it is a change card "N" or "M,N" are the line numbers, in the Fortran symbolic listing, to be changed):

```
-9
      MQP = MQ+1
-19
      C  Properly, Kurt should be adjusted for the
      C  discrete point effect.  This has not yet
      C  been done, but should not have any
      C  major effect, since Kurt is used
      C  only in the crude scan.
-70, 70
      EXF = WINFC*VACCEL(KADTY)
-73
      WADJ(KL) = W(KL)*(1.+DWFAC)
      IF(W(KL).LT.WSIM) WADJ(KL) = 2.*W(KL)+WDELSM
      DCORR=(DW+WADJ[KL])/24
      IF(KADTY.EQ.2) DCORR=DW/12.
      DO 118 I = 1, MQS,MQP
118   ALINK(LB+I-1)=ALINK(LB+I-1) + DCORR
```

```
-74, 74
     EXF = WINFC * MACCEL(KADTY)
-86, 86
     IF(ITX.GE.10)PRINT 772,ITX,INDEX(KL),VOLIN(KL),OVOL,CVOL
-88
     IF(ITX.GE.20)CALL CLPR(KL,NADJ,SUM,SKEW,KURT)
-99
     CALL CLPR(KL,NADJ,SUM,SKEW,KURT)
```

3.2.6.7 Flowchart

See Appendix A.

3.2.6.8 Listings

See Appendix B for program listings.

3.2.7 SOFTWARE COMPONENT NO. 7 (TR)

3.2.7.1 Linkage

TR is called from ADJUST.

3.2.7.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, and STPAR.

3.2.7.3 Inputs

None.

3.2.7.4 Outputs

None.

3.2.7.5 Storage Requirements

Storage used: Code = 124_8 Data = 35_8

3.2.7.6 Description

The modification made to TR was the elimination of subscripted subscripts, to enable this routine to be compiled under the Fortran V compiler on the Univac 1110 (EXEC 8 system). The subscripted subscript notations was a feature of the reentrant Fortran compiler RFOR, utilized by the program originator in development of CLASY.

3.2.7.7 Flowchart

See Appendix A.

3.2.7.8 Listings

See Appendix B for program listing.

3.2.8 SOFTWARE COMPONENT NO. 8 (CLPR)

3.2.8.1 Linkage

CLPR is called from CLDUMP, ADJUST, SEPER and JOIN. CLPR calls GET, LOCK, SQMTX, MINV, and FREE.

3.2.8.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, and STPAR.

3.2.8.3 Inputs

None.

3.2.8.4 Outputs

CLPR prints out seven messages concerning statistical information. These messages are: "CLUSTER__ INDEX__ PROPORTION__ W*__ SPLIT__ WEIGHT__ WAS__ ADJUST__ TO__ PROPORTION: PROP__ CIN__ CTUT__ OLD PROP__ CIN__ ODEN__ DIFFER__ VOLUME__ ROUT__ DCON__", "LOCATION__ LINK__ SUBS__ SUPER__ __ SYMBOL__", "NET PROB__ DIRECT__ CUMS__ __", "CUMS__ __", "MEAN__ __ __ __", "KURT(*W)__ __ __ __", "OLD COVARIANCE__ __ __ __ __"

3.2.8.5 Storage Requirements

Storage used: Code = 740_8, Data = 353_8

3.2.8.6 Description

The modifications made to CLPR were (1) the elimination of subscripted subscripts (2) a reference to one of CLASY's clustering routines, named "LOC", caused ambiguity, because there existed a Univac system routine with the same name. Therefore this reference and all other references to "LOC" were changed to "LOCK". The originator of CLASY , Dr. Michael Rassbach, provided changes to be made in CLPR. One modification suggested by Dr. Rassbach was the changing of a format statement so that the printer will skip a line before writing the statistical mean. Additionally, the following changes were suggested by Dr. Rassbach, and implemented in CLPR:

```
-53, 54
     LA=GET(MQS)
     LB=GET(MQS)
```

Where "-53, 54" indicates that line No.'s 53, 54 were replaced by the two Fortran statements following "-53, 54".

3.2.8.7 Flowchart

See Appendix A.

3.2.8.8 Listings

See Appendix B for program listing

3.2.9 SOFTWARE COMPONENT NO. 9 (CLDUMP)

3.2.9.1 Linkage

CLDUMP is called from CLASY1, MULTI, CLASY2 and ADJUST. CLDUMP calls ISPLIT and CLPR.

3.2.9.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, STPAR, and CLUSTR.

3.2.9.3 Inputs

None.

3.2.9.4 Outputs

A header concerning the dump of the clusters and a debug statement giving the value of the number of the split cluster and the value of the print parameter.

3.2.9.5 Storage Requirements

Storage used: Code = 153_8, Data = 33_8

3.2.9.6 Description

The modifications made to CLDUMP were the elimination of the logical "IF" statement that checked the value of ISPLIT and the print variable, PROUT, before the call is made to CLPR. A debug printout statement was added to allow printout of these variables for checkout purposes.

3.2.9.7 Flowchart

See Appendix A.

3.2.9.8 Listings

See Appendix B for program listing.

3.2.10 SOFTWARE COMPONENT NO. 10 (ELIM)

3.2.10.1 Linkage

ELIM is called from ADJUST. ELIM calls SUBLIM and TRFREE.

3.2.10.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, and STPAR.

3.2.10.3 Inputs

None.

3.2.10.4 Outputs

ELIM prints out two messages, one statement when a cluster has been eliminated and the other is an error message when a structural error has occurred. These messages are: "###ELIMINATE__ LINK, LSUBS, LSUPER = __ __ __", "**STRUCTURAL ERROR AT ELIM: KEL, KFAITH, KOLD, INIT__ __ __ __ __".

3.2.10.5 Storage Requirements

Storage used: Code = 176_8, Data = 46_8

3.2.10.6 Description

The modifications made to ELIM were the eliminations of subscripted subscripts in arrays to allow the routine to be compiled under the Fortran V compiler of the Univac 1110-EXEC 8 system.

3.2.10.7 Flowchart

See Appendix A.

3.2.10.8 Listings

See Appendix B for program listings.

3.2.11 SOFTWARE COMPONENT NO. 11 (SEPER)

3.2.11.1 Linkages

SEPER is called from ADJUST. SEPER calls CLPR, DGNCAL, and FREE.

3.2.11.2 Interface

Interface is accomplished through the following common blocks: CLUS, MISC, and STPAR.

3.2.11.3 Inputs

None.

3.2.11.4 Outputs

SEPER prints one statement, which is printed whenever a cluster is split. This statement is : "###SEPARATE__ SUPER, SUBS__ __ SPFAC__"

3.2.11.5 Storage Requirements

Storage used: Code = 242_8, Data = 46_8

3.2.11.6 Descriptions

The modifications made to SEPER were the elimination of subscripted subscripts in arrays, to enable the routine to be compiled under the Fortran V compiler of the Univac 1110-EXEC 8 system.

3.2.11.7 Flowchart

See Appendix A.

3.2.11.8 Listings

See Appendix B on program listing.

3.2.12 SOFTWARE COMPONENT NO. 12 (LOCK, formerly "LOC)

3.2.12.1 Linkages

LOCK is a function subprogram and is called from CLPR, and CLPRM.

3.2.12.2 Interface

Interface is accomplished through the function arguments.

3.2.12.3 Inputs

None.

3.2.12.4 Output

None.

3.2.12.5 Storage Requirements

Storage used: Code $\simeq 16_{10}$

3.2.12.6 Descriptions

The modification made to LOCK (formerly "LOC") was the changing of the external reference name from 'LOC' to 'LOCK'. The name 'LOC' is also the name of a Fortran V system routine which caused ambiguity between the two routines during execution of CLASY. Changes were made to all other routines that referenced this routine, to change the reference to be "LOCK" rather than "LOC".

3.2.12.7 Flowchart

See Appendix A.

3.2.12.8 Listings

See Appendix B for program listing.

3.2.13 SOFTWARE COMPONENT NO. 13 (MISH)

3.2.13.1 Linkages

The Fortran V procedure, MISH, is included in the following routines: ACOM, ADJUST, ALFREE, AMSQ, APRIOR, CBLO, CLASY, CLASY1, CLDUMP, CLINIT, CLPR, CLPRM, CLUSMP, CLUST, CORECT, DATFIX, DOTSQ, E1GROT, ELIM, ISPLIT, JOIN, MINV, MLT, MPVS, MTVEC, MULTI, MVEC, SEPER, SPLIT, SQMTX, STATIS, STOFLO, SUBLIM, TR, TRFREE, TR1MTX, VMTV, and VPV.

3.2.13.2 Interface

The Fortran V procedure, MISH, is used as the interface for the following common blocks: CLUS, MISC, and STPAR.

3.2.13.3 Inputs

None.

3.2.13.4 Output

None.

3.2.13.5 Storage Requirements

Storage used: Code ≈ 50

3.2.13.6 Descriptions

The modification made to the proc, MISH, was the changing of the symbol array 'NSYMB' dimension from 11 to "12" (i.e., NSYMB(12)). This was done because another symbol was added to the large symbol array, 'SYM'.

3.2.13.7 Flowchart

See Appendix A.

3.2.13.8 Listings

See Appendix B for program listing.

3.2.14 SOFTWARE COMPONENT NO. 14 (CMBK10)

3.2.14.1 Linkages

The Fortran V procedure, CMBK10, is included in the following routines: ADJUST, CLASY, CLASY1, CLDUMP, CLINIT, CLUSMP, JOIN, MULTI, READTP, SETUP9, and STATIS.

3.2.14.2 Interface

The Fortran V procedure, CMBK10, is used as the interface for the common block , CLUSTR.

3.2.14.3 Inputs

None.

3.2.14.4 Output

None.

3.2.14.5 Storage Requirement

Storage used: Code ≈ 10

3.2.14.6 Description

The modification made to the Fortran V procedure CMBK10 was the changing of the parameter, MAXPOP, value from 60 to 61. This was done to increase the dimension of the symbol array to account for the addition of another symbol to the array. The symbol added to the array was the character '0' (zero).

3.2.14.7 Flowchart

See Appendix A.

3.2.14.8 Listings

See Appendix B for program listing.

3.2.15 SOFTWARE COMPONENT NO. 15 (CLUST)

3.2.15.1 Linkage

CLUST is called from CLUSMP. CLUST calls ISPLIT, CORECT, DOTSQ, and EXP.

3.2.15.2 Interface

Interface is accomplished through calling arguments and the following common blocks: CLUS, MISC, STPAR, BIGCOM.

3.2.15.3 Inputs

None.

3.2.15.4 Outputs

None.

3.2.15.5 Storage Requirements

Storage used: Code = 225_8, Data = 56_8

3.2.15.6 Descriptions

The originator of the CLASY clustering system, Dr. Michael Rassbach, provided the following changes to incorporate into the routine 'CLUST'. The changes suggested by Dr. Rassbach, and implemented in CLUST are as follows:

```
-15
    C    USES PCOND FOR PPASS, DIST FOR DISS (KL)
-18, 18
    PCOND (KROT) = 1.
-25, 25
    130 PCOND (KL) = PROP (KL)/PR IRCM (KFAITH)*PCOND(KFAITH)
-34,34
    DIST = DOTSQ (REL,VRIN(KL+1)*W(KL))
-35,35
    IF(ABS(DIST+DCON(KL)).LE.160.) GO TO 531
-38,38
    P = XP(DIST+DCON(KL))/VOLRT(KL))*PCOND(KL)
-46
    149 PCOND(KL) = 0
-47,47
    KL =  KFATH
```

3.2.15.7 Flowchart

See Appendix A.

3.2.15.8 Listing

See Appendix B for program listing.

4. OPERATION

4.1 TEST PROCEDURE

The test procedure for verification of the modifications to the CLASY clustering system is described below.

For the identical image data set and CLASY input run parameters (area of image clustered, and channels selected):

(1) Provide a baseline run on the original (pre-modification) CLASY clustering program,

(2) Provide a second run on the CLASY clustering program with all modifications documented herein,

  (2a) Compare program execution times for the two runs, for an indication of improvement.

(3) Provide a run on the modified CLASY program which will list the array of input original image data, and the array containing the input data to be passed to STATIS after application of the randomized data scrambling technique,

  (3a) Compare the listing of the two arrays for any evidence of data value changes or extraneous data values introduced.

(4) Compare the cluster map output from the modified CLASY program to the cluster map output by the baseline (original) program, with reference to the known statistics and class definitions of the simulated image data used to make the runs.

4.2 CLASY PROGRAM OPERATION

The CLASY clustering program, as modified, is operational on the Univac 1110 (or 1108) under the EXEC8 operating system. The program utilizes the Univac Fortran V compiler (the original program utilizes the Univac

reentrant Fortran compiler, RFOR), the Univac assembler (for three assembly language routines--FREE, GET, and LOCK), and the Univac system random file access routines, RINIT, RREAD, and RWRITE.

4.2.1 CLASY Program Input

The required input to the CLASY program consists of one tape (or file) containing the multichannel image data, and the special-format card input.

The image data tape (file) is presumed to be in either of two specific formats--either "LARSYS II" format or "UNIVERSAL" format. The tape (file) reading program in CLASY, TAPERD, accepts either of these formats and self-determines the correct method of reading the data.

The card input to CLASY is of a specialized "free-field" format. Two types of card input are accepted--"control" cards (not to be confused with Univac system control cards) and "field definition" cards. The control cards are used to implement the execution options available in the program. The keywords for each "control" card are shown below. The field definition cards are used to delineate the specific areas of the input image data which are to be clustered by CLASY. The format of the field definition cards is shown below.

4.2.1.1 Control Card Input

The format and use of the "control" cards is as follows:

(1) The keyword begins in card column 1, and must be spelled exactly as shown.

(2) The parameter(s) on the control card begin in or after card column 11 and end in or before card column 72.

(3) The comma "," is the recognized separator for integer parameters, and blanks are ignored.

(4) If a default is shown, the control card is optional input to the program (default value is provided if the card is not input).

(5) If the default is specified as "none", the card is mandatory input to the program.

(6) The order of the input control cards is immaterial, except for the "*END*" card and the "$END*" card. The "*END*" card must be the last card of the set of control cards, and is to be followed immediately by one or more field definition cards. The "$END*" card must follow the last field definition card. The "$END*" card is the last card of the input.

(7) The "control" cards recognized by CLASY are as follows:

Keyword (CC 1-8)	Paramater(s) (CC 11-72)	Function
HED1	Any alphanumeric characters (default: standard page heading, line 1)	The characters supplied in CC 11-72 will replace the first line of output page heading
HED2	Any alphanumeric characters (default: standard page heading, line 2)	The characters supplied in CC 11-72 will replace the second line of the standard output page heading.
DATE	Any eight alphanumeric characters (default: the current calendar date, in upper right-hand corner of each output page)	The (up to) eight characters beginning in CC 11 will be placed in the upper right-hand corner of each page of printer output.
CHANNELS	N1, N2, N3, N4 with N an integer (default: none)	The channel numbers to be used in selecting the data vectors to be clustered. The channel numbers must be a subset of the actual channels available in the input image data.
PRINT	1,3,3 (default: limited interim printout during program execution)	The parameters "1", "3", "3" are used as printout control to generate diagnostic data during the iterative clustering process.

Keyword (Cont'd)	Parameter(s) (Cont'd)	Function (Cont'd)
END	None (default: none)	Mandatory card--signifies the end of the control card set, and the beginning of the field definition card input.
$END*	None (default: none)	Mandatory card--must follow the last field definition card--initiates CLASY program execution.

4.2.1.2 Field Definition Card

The "field definition" card(s) delineate the area on the image data tape (file) which is to be clustered by CLASY, in terms of pixel coordinates (sample, line) for each vertex of the "field", or area, up to a maximum of 10 vertices for a given field. An alphanumeric field identification may be supplied in card columns 1-6, but is not required. Beginning in card column 11, through card column 72, coordinate pairs are given (always in sample, line order) enclosed in parentheses, separated by a comma ",". The first pair given for a field must be the incrementation desired in the lines and pixels to be read from the input image tape (file). I.E., "(2,3)" would indicate every second pixel on each line, and every third line to be read. The second and succeeding coordinate pairs are the (sample, line) coordinates of the vertices of the field. A continuation of coordinate pairs on the next card is indicated by an asterisk "*". Up to 10 coordinate pairs (vertices) will be accepted for one field. An example of the field definition card(s) is as follows:

	CC 1	CC 11
CARD 1	FLDNAM	(2,3) , (2,1) , (196,3) , (100,50) *
CARD 2		(196,100) , (1,100) , (20,30)

The result of the above card will be that a rectangular area will be read from the image data which bounds the given irregularly shaped field defined above. The coordinates of the rectangular area will be (1,1) , (196,1) , (196,100) , (1,100).

The coordinates for the rectangular area are self-determined by the tape (file) reading program. Internally, the actual field coordinates which were input are used to extract for clustering only the specific pixels that are within the actual field defined.

The input field vertices must be defined on the card(s) in clockwise order.

4.2.2 CLASY PROGRAM OUTPUT

The output by CLASY is all line-printer output. Interim printout of statistical parameters and diagnostic data is provided during the iterative cluster-forming process.

The final output is a "map", with a symbolic representation of the area clustered, with each pixel of the area classified using the statistics (mean and covariance) from final cluster set determined by CLASY. The symbols on the "map" represent the cluster (= class) which is the most likely parent distribution for the given pixel. The "map" is output by subprogram CLUSMP.

Sample output is shown in Appendix C.

4.2.3 CLASY PROGRAM RESTRICTIONS

The known restrictions inherent in the program are (1) the program is restricted to a maximum of four channels, due to program array dimensioning which presumes this maximum, (2) the program will not successfully execute with only one channel, (3) a data vector containing a zero value in the channel of interest will cause an error termination of the program's execution, (4) the size of the original image data set read from the input tape (or file) and placed on drum must be containable in 1,310,717 locations of drum storage available to the random access routines (RINIT, RREAD, and RWRITE).

TEST VERIFICATION

For CLASY Program Modification

This verification is being conducted to insure that the delivered program products satisfy the requirements as originally stated by the requesting organization.

M C Trivelpiece
NASA Monitor

J. C. Martin
Requestor

Burnell McCray
Developer

Barbara Hills
Cognizant System Manager

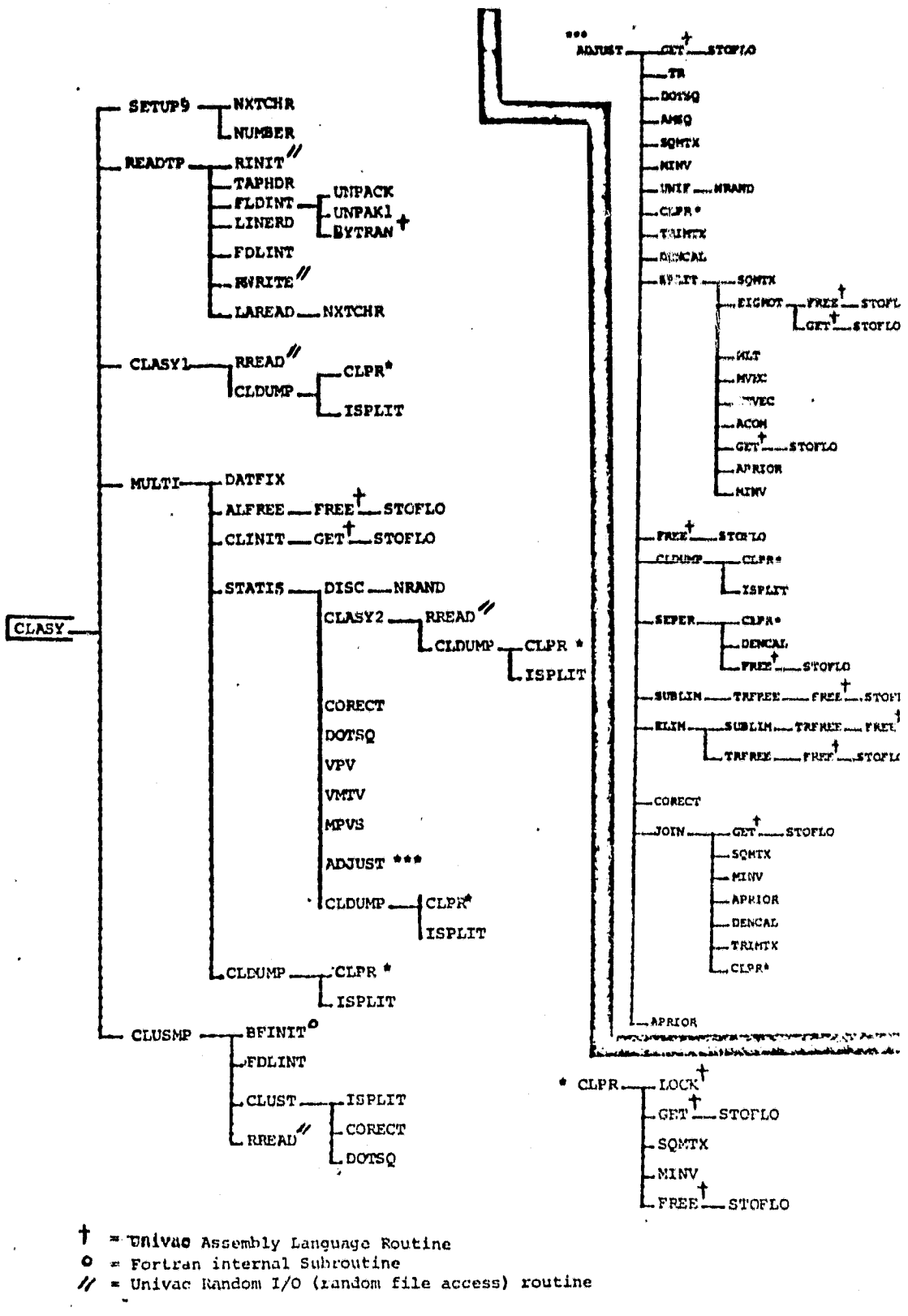
Quality Assurance

C. D. Gardner
Test Conductor

Verification Date: 4/25/77

APPENDIX A

CLASY SYSTEM FLOWCHART



† = Univac Assembly Language Routine
° = Fortran internal Subroutine
// = Univac Random I/O (random file access) routine

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX B

LISTINGS OF MODIFIED CLASY SUBPROGRAMS

```
@FOR,SUW ADJUST,ADJUST
FOR S7E3-24/07/77-18:33:14 (4,3)
-13
      MOP=MO+1
-19
C PROPERLY,KURT SHOULD BE ADJUSTED FOR THE DISCRETE POINT EFFECT.THIS
C HAS NOT YET BEEN DONE ,BUT SHOULD NOT HAVE ANY MAJOR EFFECT,SINCE
C KURT IS USED ONLY IN THE CRUDE SCAN.
-32,33
      CALL MINV(ALINK(LD),ALINK(LVA),ALINK(LD),RVOL)
      OVOL=-ALOG(ABS(RVOL))/APL-ALOG(ABS(W(KL)))
-39,39
      LVA=LD
-43,43
      LAX=LD
-56,56
      IF(W(KL).GT.UNIF(4570.)*PROP(KL).OR.W(KL).LE.0.OR.DW.LE.0.OR.RVOL
     *.LE.0)CALL CLPR(KL,NADJ,SUM,SKEW,KURT)
      IF(W(KL).LE.0.OR.DW.LE.0.OR.RVOL.LE.0)PRINT 771,KL,W(KL),DW,RVOL
 771  FORMAT(1H ,'W/OVOL ERROR IN ADJUST:KL,W,NEW W,VOL',I6,3E15.7)
-67,67
      EXF=ANTC*VACCEL(KARTY)
-73
      WADJ(KL)=W(KL)*(1.+D*FAC)
      IF(W(KL).LT.-SIM) WADJ(KL)=2.*W(KL)+WDELSM
      DCORR=(D.*WADJ(KL))/24
      IF(KARTY.EQ.3)DCORR=D*/12.
      DO 118 I=1,MOS,MOP
 118  ALINK(LD+I-1)=ALINK(LD+I-1)+DCORR
-71,71
      EXF=WINFL*MACCEL(KARTY)
-74,74
 114  ALINK(LD+I-1)=AR*(ALINK(LR+I-1)*EXF*ALINK(LVA+I-1))
      CALL TRIMIX(CVAR(KL+1),ALINK(LD))
      CALL MINV(ALINK(LVA),ALINK(LD),VOLIN(KL))
-79,79
      IF(ITX.EQ.25)EXF=0.
      IF(VOLIN(KL).LT.0..AND.ITX.LT.26)GO TO 117
-80,31
      CVOL=ALOG(ABS(VOLIN(KL)))/APL-ALOG(ABS(W(KL)))
      IF(ABS(OVOL-CVOL).GT.VOLLIM.AND.ITX.LT.26)GO TO 117
      IF(ITX.GE.17)PRINT 772,ITX,INDEX(KL),VOLIN(KL),OVOL,CVOL
 772  FORMAT(1H ,'***EXTRAPOLATION PROBLEM IN ADJUST:ITER,INDEX(KL),VOLI
     *N,OVOL,CVOL',I4,I6,3E15.7)
      IF(ITX.GE.26)CALL CLPR(KL,NADJ,SUM,SKEW,KURT)
-122,122
      KC=LSUBS(KL)
      KDC=LINK(KC)
      PRINT 354,INDEX(KL),W(KL),INDEX(KC),INDEX(KDC)
-148,148
 255  KCC=LSUPLR(KL)
      KDC=LINK(KL)
      IF(W(KL).LT.UNIF(WOJOIN).OR.NOJO.NE.0.OR.LINK(KL).EQ.0
     1 .OR.LSUPS(KCC).LE.KL.AND.LINK(KCC).EQ.0)GO TO 253

   SUBROUTINE ADJUST     ENTRY POINT 002326
```

ORIGINAL PAGE IS
OF POOR QUALITY

STORAGE USED: CODE(1) [illegible]; DATA(0) [illegible]; BLANK COMMON(2) [illegible]

COMMON BLOCKS:

```
[illegible]   CLUS     [illegible]
[illegible]   MISC     [illegible]
[illegible]   STRAH    [illegible]
[illegible]   CLUSTR   [illegible]
[illegible]   JOINFR   [illegible]
```

EXTERNAL REFERENCES (BLOCK, NAME)

```
[illegible]   GET
[illegible]   PE
[illegible]   DOTSZ
[illegible]   ABS
[illegible]   SCRTX
[illegible]   MINV
[illegible]   UNIF
[illegible]   CLFS
[illegible]   TRIMTX
[illegible]   [illegible]CAL
[illegible]   SPLIT
[illegible]   FREL
[illegible]   CLDUMP
[illegible]   SEPL
[illegible]   SUBLIM
[illegible]   FLIP
[illegible]   CURECT
[illegible]   JOIN
[illegible]   AF[illegible]
[illegible]   SQRT
[illegible]   ALOG
[illegible]   NERT[illegible]
[illegible]   NIO2[illegible]
[illegible]   NIO1[illegible]
[illegible]   EXP
[illegible]   XPRI
[illegible]   NERR[illegible]
```

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

[illegible]

B-3 40

```
*DIAGNOSTIC* THE VARIABLE, IMOT, IS REFERENCED IN THIS PROGRAM, BUT IS NOWHERE ASSIGNED A VALUE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  113 CSUM(KL+I)=SUM(KL+I)
      WADJ(KL)=W(KL)*(1.+DWFAC)
      IF(W(KL).LT.WSIM) WADJ(KL)=2.*W(KL)+WDLLSM
      DCORR=(DN*WADJ(KL))/24
      IF(KARTY.[illegible])DCORR=DN/12.
      DO 113 I=1,MES,M[illegible]
  113 ALINK(LP+I-1)=ALINK(L3+I-1)+DCORR
      EXF=AINF*NACCEL(KARTY)
      ITX=1
  117 DO 114 I=1,NOS
  114 ALINK(LD+I-1)=WP*(ALINK(LP+I-1))*EXF*ALINK(LVA+I-1))
      CALL TRIPTX(OVAR(KL+I),ALINK(LD))
      CALL MINV(ALINK(LVA),ALINK(LP),VOLIN(KL))
      EXF=EXF*.[illegible]
      ITX=ITX+1
      IF(ITX.[illegible].25)EXF=[illegible].
      IF(VOLIN(KL).LT.[illegible].AND.ITX.LT.25)GO TO 117
      CVOL=ALOG(ABS(VOLIN(KL)))/AN[illegible]-ALOG(ABS(W(KL)))
      IF(ABS(OVOL-CVOL).GT.VOLLIM.AND.ITX.LT.75)GO TO 117
      IF(ITX.GE.[illegible])PRINT 772,ITX,INDEX(KL),VOLIN(KL),OVOL,CVOL
  772 FORMAT(1H ,'***EXTRAPOLATION PROBLEM IN ADJUST:ITER,INDEX(KL),VOLI
     *N,OVOL,CVOL',I4,I6,3E15.7)
      IF(ITX.[illegible])CALL (LPRIKL,NA,J,SUM,SKEW,KURT)
      CALL TRIPTX(V[illegible](KL+I),ALINK(LVA))
      EXF=AINF*NACCEL(KARTY)
      PRK=PROP(KL)
      ODEN=W(KF)-CTOT(KL)
      CINV=CIN(KL)-OCIN(KL)
      PROP(KL)=(CINV)/(DEN-ODEN(KL))
      IF(PROP(KL).GT.[illegible].AND.PROP(KL).LE.1.) GO TO 139
      PRINT 639,INDEX(KL),PRK,PROP(KL),PRIRCM(KF),W(KL),CIN(KL),
     1    OCIN(KL),CINV,W(KF),CTOT(KL),DEN,ODEN(KL)
  639 FORMAT('[illegible] PROP:PRK,P,CM,W',I3,4E9.4/'    (ERROR CONT) CIN',
     1  3E9.4,'  W(KF),CTOT,DEN,ODEN',4E9.4)
      PROP(KL)=PRK
  139 ALP=ALOG(PROP(KL))
      ALPO=ALOG(OPROP(KL))
      DALP=ALP-ALPO
      PFAC=EXP(T[illegible]F*DALP)
      PROP(KL)=PROP(KL)*PFAC
      OPROP(KL)=PROP(KL)
      PRIRCM(KF)=PRIRCM(KF)+PROP(KL)-PRK
      CIN(KL)=CINV*PFAC
      OCIN(KL)=CIN(KL)
      ODEN(KL)=(CIN(KL)/PROP(KL)*PRIRCM(KF))
      CTOT(KL)=W(KF)-ODEN(KL)
      KK=LSUBS(KF)
  141 CALL [illegible](KK,1./PRIRCM(KF),W(KF))
      KK=LINK(KK)
      IF(KK.NE.0) GO TO 141
C   SPLIT THE CLUSTER
      IF(LSUBS(KL).NE.0) GO TO 202
      IF(TW[illegible]S.LT.[illegible].AND.SPTES.LT.[illegible].AND.URATES.LT.[illegible]) GO TO 202
      NSGS=[illegible]T(MOS)
      NTAUS=[illegible]T(MOS)
      N[illegible]=[illegible]T(MOS)
      NPS[illegible]=[illegible]T(MOS)
```

```
      NDTAU=GET(MCS)
C     D       PRINT 613,LA,LB,LD,LVA,NSGSC,NTAUSG,NDUM,NDSG,NDTAU
C     D  613 FORMAT(' GET IN ADJ:',11I6)
      CALL SPLIT(KL,SUM,SKEW,KURT,OSUM,OVAR,
     1 LINK(LA),LINK(LB),LINK(LD),LINK(LVA),LINK(NSGSQ),
     2 LINK(NTAUSQ),LINK(NDUM),LINK(NDSG),LINK(NDTAU))
      KC=LSUBS(KL)
      KDC=LINK(KC)
      PRINT 354,INDEX(KL),W(KL),INDEX(KC),INDEX(KDC)
  354 FORMAT(' FARHIVE SPLIT',I7,' WEIGHT',E11.5,' SUBS',2I3)
      CALL FREE(NSGSC,MCS)
      CALL FREE(NTAUSC,MCS)
      CALL FREE(NDUM,MCS)
      CALL FREE(NDSG,MCS)
      CALL FREE(NDTAU,MCS)
      CALL CLDUMP(KL)
      GO TO 204
C   ELIMINATE THIS CLUSTER IN FAVOR OF ITS SUBCLUSTERS, IF IT IS
C       SPLIT WITH ODDS GREATER THAN SEPTH.
  200 IF(SPFAC(KL).LE.SEPTH*SPCOR) GO TO 30
      CALL SEPER(KL)
      GO TO 299
   30 CONTINUE
C   ELIMINATE THE SUBCLUSTERS IF THEY ARE DOMINATED BY THE MAIN
C       CLUSTER.
      SPBND=(SPFAC(KL)-OPRIOR(KL))/SPCOR
      IF((SPBND.LE.SILTH.OR.PORAT(KL)/DN.LT.PORATH.AND.SPBND
     1   .LT.SPMVTH).AND.LSUBS(KL).NE.0) CALL SUBLIM(KL)
C   ELIMINATE THIS CLUSTER (AND PERHAPS ITS COCLUSTER) IF ITS
C       PROPORTION BECOMES TOO SMALL.
  204 IF(PROP(KL).GE.ELIMTH.OR.NOELIM.NE.0)  GO TO 205
      CALL ELIM(KL)
      GO TO 299
C   FIND LIKELY OVERLAPS OF THIS CLUSTER WITH THOSE FURTHER DOWN THE LIST.
  205 KCC=LSUPER(KL)
      KDC=LINK(KL)
      IF(W(KL).LT.UNIF(ADJOIN).OR.NOJC.NE.0.OR.LINK(KL).EQ.0
     1 .OR.LSUBS(KCC).EQ.KL.AND.LINK(KDC).EQ.0)GO TO 250
      RMIN=1.E26
      K=LINK(KL)
  211 IF(FJOIN.LT.UNIF(1.)) GO TO 213
      WW=W(K)/W(KL)
      CALL CORECT(FV,SUM(KL+1),WW,SUM(K+1))
      RR=(DOTSQ(FV,VRIN(KL+1))+DOTSQ(FV,VRIN(K+1))*WW)*.5/W(KL)
      DO 212 I=1,MM
      IM=MXAR(I+1)
      IF(VRIN(KL+IM)*VRIN(K+IM).LE.0.) PRINT 612,I,IM,INDEX(KL),INDEX
     1   (K),KL,K,(J,VRIN(KL+J),VRIN(K+J),J=1,MM)
  612 FORMAT(' LOG ERROR IN ADJUST:   I,IM,KL,K/VRIN=',2I5,2I3,2I7/(I5
     1      ,2E15.6))
  212 RR=RR+VRJOIN*(ALOG(ABS(VRIN(KL+IM)/VRIN(K+IM))+1.E-25)**2)
      RP=RR/(.1E*(WW-1./WW)**2+1.)
      IF(RR.GT.RMIN) GO TO 213
      KMAX=K
      RMIN=RR
  213 K=LINK(K)
      IF(K.GT.0) GO TO 211
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      IF(-[illegible].GT.ALIM*AMD) GO TO 25[illegible]
      NJO=JCIN(KL,KMAX,SUM,SKEW,KURT,CSUM,OVAR,LINK(LA),LINK(LB),
     1    LINK(LD),LINK(LVA))
  25[illegible] CONTINUE
      DO 101 I=1,MO
  101 SKEW(KL+I)=0.
      DO 102 I=1,MM
  102 KURT(KL+I)=0.
C     ADJUST DW(KL)
      SPFAC(KF)=AMAX1(SPFAC(KF),AMIN1(SPFAC(KF)+DW*BETTER,OPRIOR(KF)))
      PCRAT(KL)=0.
      SPFAC(KL)=-9999.9
      IF(LSUM[illegible](KL).[illegible]) SPFAC(KL)=APRIOR(KL)
      OPRIOR(KL)=SPFAC(KL)
      DCON(KL)=DLCON
      VOLIN(KL)=ABS(VOLIN(KL))*.87565137 63E-25*(6.28318530 7/W(KL))**M2
      VOLRT(KL)=SQRT(VOLIN(KL))
      DW(KL)=W(KL)
      WADJ(KL)=W(KL)*(1.+DWFAC)
      IF(W(KL).LT.WSIM) WADJ(KL)=2.*W(KL)+WD[illegible]LSM
      IF(ISADJ(KL).EQ.NPTSO) INDEX(KL)=-IABS(INDEX(KL))
      I[illegible](KL)=NPTSU+TOTFIX
  399 CONTINUE
      CALL FREE(LA,MCS)
      CALL FREE(LB,MCS)
      CALL FREE(LD,MCS)
      CALL FREE(LVA,MCS)
      RETURN
      END
```

END OF COMPILATION: 1 DIAGNOSTICS.

.FIN

```
#FOR,S  CLASY,CLASY
FOR S*E3-04/21/77-21:10:10 (1,)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000146; DATA(0) 047047; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    INFORM 000407
    0004    CLUSTR 000123
    0005    CLUS   002017
    0006    MISC   000113
    0007    STPAR  003016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0012    SETUP9
    0011    READTP
    0010    CLASY1
    0013    MULTI
    0014    CLUSMP
    0015    NINTR$
    0016    NSTOP$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000004 ICL       0001    000051 ZCL       0001    000074 SCL       0005 R  000020 ALINK     0006    000064 AMM
    0006    000375 AMOFAC    0006    000377 AMOMAX    0006    000376 AMOMIN    0006    000130 AMORAT    0006    000054 AMQ
    0004 I  000007 AREA      0000 I  000002 ARRAY     0003    000351 BESTVC    0006    000127 BETTER    0006    000102 BIAS
    0005 R  000627 CIN       0004 I  000002 CLSNAM    0007    000401 CONLV     0006    000111 COPLEN    0005 R  000010 CIOI
    0003    000053 DATAPE    0005 R  000041 DCON      0006    000353 DELT      0003 R  000043 DISS      0003    000112 DIVSIZ
    0003    000062 DUPSYM    0006    000073 DWFAC     0006    000061 ELIMTH    0006    000052 EPS       0003    000217 FETVC2
    0003    000161 FETVEC    0003    000071 FILESV    0005 R  000150 GEN       0006    000074 GRACTM    0005 R  000047 GREF
    0003    000000 HEAD      0003    000255 HISVEC    0004 I  000000 IBEGIN    0003    000123 IPLOCM    0005 I  000023 IDADJ
    0006 I  000020 IDATA     0005 I  000024 INDEX     0005    000166 INDXVL    0003    000313 INVERT    0004 I  000004 IPT
    0004 I  000122 IO1       0005 I  000000 JUNK      0003    000113 KEEPLV    0004 I  000112 KKT       0004 I  000113 KL&C
    0004 I  047044 KPLACE    0006    000051 KROOT     0000 I  047042 LAST      0004 I  000121 LBUF      0005 I  000020 LINK
    0006    000003 LKUPT     0004 I  000105 LNCAT     0006    000007 LOSUM     0006    000074 LOWAR     0000 I  047043 LPLACE
    0006 I  000002 LR        0006    000006 LSKEW     0005 I  000021 LSUBS     0006    000075 LSUM      0005 I  000022 LSUPER
    0006 I  000005 LV        0006    000002 LVRIN     0007 R  000012 MACCEL    0003    000052 MAPTAP    0004 I  000003 MAXBF
    0003    000172 MAXCLS    0003    000064 MAXDIV    0003    000155 MAXFET    0003    000074 MAXFLD    0003    000165 MINDIV
    0006    000001 MM        0006    000110 MODE      0006    000000 MQ        0006 I  000011 MXAF      0005    000014 NARL
    0006    000210 NINCLS    0003    000173 NOCLSZ    0003    000121 NOCLS3    0003 I  000120 NOFEAT    0003    000101 NOFET2
    0003    000102 NOFET0    0004 I  000006 NOFLD     0003    000175 NOFLD2    0003    000176 NOFLD3    0003    000111 NOGRP
    0003    000110 NOHIST    0003    000107 NOSPEC    0003    000177 NOTREO    0004 I  000120 NPTS      0006    000070 NPTSO
    0005 I  000043 NSYMB     0005 I  000220 NTB       0005    000216 NTBSZM    0005    000215 NTOP      0005    000017 NWANT
    0004 I  000117 NWORS     0004 I  000005 NWDS      0005 R  000045 OCIN      0006    000055 ODCON     0005 R  000047 ODEN
    0005 R  000046 OPRIOR    0005 R  000035 OPROP     0005 R  000036 OW        0007 R  000010 PACCEL    0003    000056 PAGSIZ
    0005 R  000050 PCOND     0003    000122 PCISZ     0005 R  000025 PCUM      0006    000103 PJOIN     0005 R  000043 PPASS
    0005 R  000042 PGRAY     0005    000071 PGRATH    0005 R  000026 PRIRCM    0004 I  000106 PRNT      0005 R  000031 PROP
    0004 I  000115 PROUT     0003    000114 PRTLEV    0004 I  000114 PRTME     0005 R  000044 PST       0003    000054 SAVTAP
    0006    000065 SPLTH     0006    000062 SEPTH     0003    000067 SERIAL    0007    000002 SMBND     0007    000003 SKCHI
    0003    000120 SPCBAS    0006    000112 SPCOM     0005 R  000032 SPFAC     0003    000066 SPLMAX    0006    000072 SPMVTH
    0004 I  000010 SYM       0003    000057 TAPCHK    0003    000070 TAPESV    0003    000063 THRSYM    0000 I  000000 TOP
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0005 I 000001 TOPIT      0004 I 000116 TOTPIX     0004 I 000031 TOTWRD     0007   000003 TRCHI      0007   000005 TRCHI
0003   000000 TRNSYM     0003   000061 TSTSYM     0007   000026 UPKBND     0007   000077 UPKCHI     0007 R 000014 VACCEL
0003   000103 VARSIZ     0003   000104 VARSZ2     0003   000105 VARSZ4     0004   000063 VFAC       0005 R 000037 VOLIN
0006   000011 VOLLIM     0005 R 000040 VOLRT      0005 R 000050 VRIN       0006   000124 VRJOIN     0005 R 000034 W
0005 R 000013 WADJ       0006   000060 WADJIN     0007   000030 WAIT       0006   000106 WDELSP     0006   000067 WFAC
0006   000105 WSIM       0006   000050 WTINIT     0003   000115 XHGH       0003   000117 XLOW       0006   000056 XOVFLO
0003   000106 XSIZ       0006   000057 XUNFLO     0003   000115 YSIZ

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00100       1*       C************************************************************************      000000
00100       2*       C*******************************************************************           000000
00100       3*       C*                                                                              000000
00100       4*       C*      THIS PROGRAM CLUSTERS THE DATA USING AN ALGORITHM (VARIABLE             000000
00100       5*       C*      CLUSTER COUNT MAXIMUM LIKLIHOOD CLUSTERING) DEVELOPED BY MIKE           000000
00100       6*       C*      PASSBACH OF NASA-FOD.                                                   000000
00100       7*       C*                                                                              000000
00100       8*       C*      THE PROGRAM EXPECTS MULTISPECTRAL SCANNER DATA IN EITHER THE            000000
00100       9*       C*      LARSYS 22 OR THE UNIVERSAL FORMAT.  THE DATA TAPE SHOULD BE             000000
00100      10*       C       ASSIGNED TO UNIT 3.  DRUM ACCESS IS BY RANDIO  RANDOM ACCESS            000000
00100      11*       C*      ROUTINES- RREAD,RWRITE,RINIT                                            000000
00100      12*       C*                                                                              000000
00100      13*       C*******************************************************************           000000
00100      14*       C*                                                                              000000
00101      15*            IMPLICIT INTEGER (A-X)                                                     000000
00103      16*            INCLUDE COMBK1,LIST                                                        000001
00103      16*       COMBK1 PROC                                                                     000001
00104      16*            COMMON /INFORM/HEAD(42), MAPTAP,     CATAPE,     SAVTAP,    MAXFET,        000001
00104      16*           1                 PAGSIZ, TAPCHK,     TRNSYM,     TSTSYM,                   000001
00104      16*           2                 DUPSYM, THRSYM,     MAXDIV,     MINDIV,    SPLMAX,        000001
00104      16*           3                 SERIAL, TAPESV,     FILESV,                               000001
00104      16*           4                 MAXCLS, NOCLS2,     MAXFLD,     NOFLD2,    NOFLD3,        000001
00104      16*           5                 NOTRFD, NOFEAT,     NOFET2,     NOFET4,    VARSIZ,        000001
00104      16*           6                 VARSZ2, VARSZ4,     XSIZ  ,     NOSPEC,    NOHIST,        000001
00104      16*           *                 NOGRP, DIVSIZ,    KEEPLV,    PRTLEV,    YSIZ              000001
00104      16*           8                 XHGH  , XLOW  ,      SPCBAS,     NOCLS3,    PCTSZ ,       000001
00104      16*           9          IBLOCK(30), FETVEC(30), FETVC2(30), HISVEC(30),                  000001
00104      16*           *          INVLRT(30), BESTVC(30)                                           000001
00104      16*       END COMBK1                                                                      000001
00105      17*            INCLUDE CMBK10,LIST                                                        000001
00105      17*       CMBK10 PROC                                                                     000001
00106      17*            PARAMETER MAXPOP=61                                                        000001
00107      17*            INTEGER IBEGIN,TOTWRD,CLSNAM,MAXBF,IPT,NWDS,NOFLD,AREA,                    000001
00107      17*           * SYM,LNCAT,PRNT,KKT,KLBC,PRIME,PROUT,NWDRS,NPTS                            000001
00110      17*            INTEGER TOTPIX,LBUF,IQ1                                                    000001
00111      17*            COMMON/CLUSTR/IBEGIN,TOTWRD,CLSNAM,MAXBF,IPT,NWDS,NOFLD,AREA               000001
00111      17*           *       ,SYM(MAXPOP),LNCAT,PRNT(4),KKT,KLBC,PRIME,PROUT,                    000001
00111      17*           *     TOTPIX,NWDRS,NPTS,LBUF,IQ1                                            000001
00112      17*            DATA PRNT(4)/15/                                                           000001
00112      17*       END CMBK10                                                                      000001
00114      18*            DATA TOP/20000/,TOPID/7000/                                                000001
00114      19*       C    TOPID SHOULD EQUAL VALUE FOR NAREA IN CBLO                                 000001
00117      20*            INCLUDE  MISH,LIST                                                         000001
00117      20*       MISH  PROC                                                                      000001
00120      20*            PARAMETER NTBSIZ=32                                                        000001
```

B-10 47

ORIGINAL PAGE IS
OF POOR QUALITY

```
      NTBSIZ MUST BE A POWER OF TWO.  (NTR & NTBSIZ ARE USED IN THE
        STORAGE ALLOCATOR CALLOC, ENTERED BY GET AND FREE.)
C
C       THE FOLLOWING TWO PARAMETERS GIVE THE NUMBER OF
C         SYMMETRIC MATRICES AND VECTORS IN A CLUSTER NODE.
      PARAMETER NAPRS=3
      PARAMETER NVECS=3
      PARAMETER NSCALS=25
      INTEGER JUNK(12),NIB(NTBSIZ),LINK(9999)
      REAL A(1),POBAT(1),VOLIN(1),VOLRT(1),DCON(1),PPASS(1),PCOND(1)
      INTEGER LSUBS(1),IDADJ(1),INDEX(1),LSUPER(1),NSYMB(1)
      REAL VRIN(9475),GEN(9999),GBEF(9999),ALINK(1)
      REAL PST(1),PCUM(1),DISS(1),WADJ(1),OPROP(11),OW(1),SPFAC(1)
      REAL PRIRCH(1),OPRIOR(1),PROP(1),CIN(1),CTOT(1),OCIN(1),ODEN(1)
C  --THE FOLLOWING STRUCTURES THE CLUSTER NODES.  'GEN' IS
C       USED LATER AS THE BASE FOR ALL THE ARRAYS.  (SEE CLINIT)
C       'GREF' IS USED LIKE 'GEN', BUT INTERNAL TO ROUTINES,
C         RATHER THAN IN CALLING SEQUENCES.
      EQUIVALENCE (LINK(31),LSUBS(1),LSUPER(1)),IDADJ(28),NSYMB(12),
     1   INDEX(1),PRIRCH(25),CIN(24),CTOT(23),
     2     PROP(22),SPFAC(21),WADJ(20),A(19),OPROP(18),OW(17),
     3       VOLIN(16),VOLPT(15),DCON(14),PCUM(13),DISS(12),PPASS(12),
     4         PST(11),OCIN(10),PCOND(9),VRIN(7),GEN(7),OPRIOR(9),ODEN(8),
     5           GBEF(8),NTR(3))
      EQUIVALENCE (LINK(1),ALINK(1))
      COMMON/PLUS/ JUNK,NARL,NTOP,NTRSZM,NWANT,LINK
C  --WE ASSIGN VARIOUS LOCATION PARAMETERS IN CLINIT.  THESE
C       ARE STORED MOSTLY IN THE ARRAYS LR FOR ARRAYS AND LV
C       FOR VECTORS.  THE FOLLOWING EQUIVALENCES ARE PROVIDED
C       FOR CONVENIENCE.  (FOR EXAMPLE, GENLSUM+KL+1) IS THE
C       VECTOR SUMKL+1).
      INTEGER MXARI(31),LR(NARPS),LV(NVECS)
      EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT),
     1 (LR(3),LOVARI),(LV(1),LSUM),(LV(2),LSKEW),(LV(3),LOSUM)
      COMMON/MISC/ MO,MH,LR,LV,NINCLS,MXAR,MTINIT,MXOOT,EPS,DELT,
     1   AMS,OUCON,XOVFLO,WAIJIN,ELIMTH,SEPTH,WFAL,AMM,SBLTH,
     2     INDXVL,WFAC,NPTS0,PGPATH,SPMYTH,OWFAC,GRACTM,AMGFAC,
     3       AMOMAX,AMORAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIM,WDELSM,
     4         BETTER,MODE,COMLEN,SPCOR
      REAL PACCEL(2),WACCEL(2),VACCEL(2)
      COMMON/STPAR/WAIT,CONLV,SKBAR,SKCHI,TRGND,TRCHI,URKBND,URKCHI,
     1   PACCEL,MACCEL,VACCEL
C  MOMAX IS THE MAXIMUM NUMBER OF CHANNELS (30 FOR LARSYS)
      PARAMETER MOMAX=4
C       MOTMAX=(MOMAX*(MOMAX+1))/2
      INTEGER OFT
      PARAMETER MOTMAX=10
      END $ISH
      DIMENSION ARRAY(2700),IDAT(1)
      EQUIVALENCE (LINK(1),IDATA(1))
C**
C**   CALL SETUP9 TO READ INPUT CARDS
C**
C**   CALL SETUP9
C**
C**   CALL READIP TO READ CLASS AND FIELD DEFINITION CARDS AND TO READ
C**   THE FIELDS OF DATA FROM THE IMAGE TAPE AND TO STORE DATA ON DRUM
```

```
      C*
   10 CALL READTP(ARRAY,TOP,LAST,IDATA,TOPID)
      AREA=TOTWRD/NOFEAT
      IF(AREA.LE.700E) GO TO 20
      NWORS=(TOTWRD/NOFEAT)*NOFEAT
      IQ1=TOTWRD/NWORS
      NPTS = NWORS/NOFEAT
      LEUF = TOTWRD - (IQ1*NWORS)
      IF(LEUF.NE.0)IQ1 = IQ1 + 1
      GO TO 50
   20 CONTINUE
      NWORS= TOTWRD/NOFEAT
      NPTS= NWORS/NOFEAT
      IQ1=TOTWRD/NWORS
      LEUF= TOTWRD - (IQ1*NWORS)
      IF(LEUF.NE.0) IQ1=IQ1 + 1
   50 CONTINUE
      C     SET PRINT COUNTERS
      KKT=1
      PRTME=1
      PROUT=1
      LOCAT=0
      LPLACE=IPT + NPTS
      KPLACE = LPLACE + NWORS
      CALL CLASYI(ARRAY(IPT),ARRAY(LPLACE),ARRAY(KPLACE))
      C     CALL MULTI TO PERFORM CLUSTERING
      C
      CALL MULTI(ARRAY(KPLACE))
      CALL CLUSPP(ARRAY,TOP,ARRAY(2001))
      IF(LAST.NE.1) GO TO 10
      STOP
      END

END OF COMPILATION:        1 DIAGNOSTICS.
```

```
...FOR,S  CLASY1,CLASY1
FOR SEE3-04/01/77-11:10:13 (6,)

   SUBROUTINE CLASY1     ENTRY POINT 000600
              CLASY2     ENTRY POINT 000611

   STORAGE USED: CODE(1) 000633; DATA(3) 000111; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   INFORM 000407
    0004   CLUSTR 000123
    0005   CLUS   002017
    0006   MISC   000113
    0007   SIPAP  000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0010   ZOP
    0011   CMERR
    0012   PREAD
    0013   CLDUMP
    0014   NWDU%
    0015   NI02%
    0016   NI01%
    0017   NERR3%

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001      000026 15L       0001      000026 155G      0001      000032 16L       0000      000024 17F       0001      000173 18L
    0001      000267 233G      0001      000367 24L       0001      000267 240G      0001      000343 255G      0001      000355 257G
    0001      000134 35L       0001      000411 36L       0001      000534 40L       0001      000546 41L       0001      000233 5GL
              000037 52F                 000053 53F       0005   R  000022 ALINK      0006      000064 AMM       0006      000075 AMOFAC
    0006      000077 AMOMAX    0006      000076 AMOMIN    0006      000102 AMORAY    0006      000054 AMG       0004   I  000007 AREA
              000361 BESTVC    0006      000107 BETTER    0006      000162 BIAS      0005   R  000027 CIN       0004   I  000002 CLSNAM
              000001 CONLW     0006      000111 CORLEN    0005   R  000330 CTOT      0003      000053 DATAFE    0005   R  000041 DCON
    0006      000053 DELT      0006   R  000043 DISS      0003      000112 DIVSI7    0005   I  000004 DRAID     0003      000062 EUPSYM
    0006      000073 DWFAC     0006      000061 ELIMTH    0006      000056 EPS       0003      000217 FETVC2    0003      000161 FETVEC
    0003      000071 FILESV    0006   R  000056 GEN       0006      000074 GPACTM    0005   R  000047 GREF      0003      000000 HEAD
    0003      000215 HISVEC    0004   I  000000 IBEGIN    0003      000123 IBLOCK    0005   I  000023 IDADJ     0004   I  000022 II
    0000   I  000017 IJ        0004   I  000006 IM        0000   I  000010 IMN       0005   I  000024 INDEX     0006      000066 INDXVL
              000063 INJPS     0003      000313 INVERT    0000   I  000020 IP        0004   I  000004 IPT       0004   I  000122 IQ1
    0004   I  000007 ISTAT     0004   I  000015 IX        0000   I  000021 JI        0003   I  000023 JJ        0005   I  000000 JUNK
    0004   I  000011 K         0003      000113 KEEPLV    0004   I  000112 KKT       0004   I  000113 KLBC      0006      000051 KROOT
    0004   I  000121 LBUF      0004   I  000005 LINK      0006      000103 LKURT     0004   I  000105 LNCAT     0006      000007 LOSUM
              000014 LOVAR     0004   I  000002 LP        0006      000106 LSKEW     0005   I  000021 LSUBS     0006      000005 LSUM
    0005   I  000012 LSUPER    0006   I  000005 LV        0006      000000 LVRIN     0007   R  000032 MACCLL    0003      000052 MAPTAP
    0004   I  000003 MAXRF     0003      000074 MAXCLS    0003      000064 MAXDIV    0003      000055 MAXFLT    0003      000074 MAXFLD
    0003      000065 MINDIV    0006      000001 MM        0006      000110 MODE      0006      000000 MQ        0006   I  000011 MXAR
    0004   I  000006 N         0005      000014 NARL      0005      000010 NINCLS    0005   I  000002 NLAST     0003      000073 NOCLS2
    0003      000161 NOCLS3    0003   I  000150 NOFEAT    0003      000101 NOFET2    0003      000102 NOFET4    0004   I  000006 NOFLD
    0003      000075 NOFLD2    0003      000076 NOFLD3    0003      000111 NOGRP     0003      000110 NOHIST    0003      000107 NOSPEC
    0003      000077 NOTREO    0004   I  000120 NPTS      0006      000070 NPTSO     0000   I  000023 NPEAD     0005   I  000043 NSYMB
```

B-13 50

```
0000 I 000020 NTB        0005   000015 NTBSZM     0003 I 000016 NTEMP      0005   000013 NTOF       0000 I 000000 NTOTAL
0000 I 000010 NU         0005   000017 NWANT      0004 I 000117 NWERS      0004 I 000005 NWDS       0005 R 000045 PCAN
0006   000055 ODCON      0005 R 000047 ODLN       0005 R 000045 OPRIOR     0005 R 000035 OPROP      0005 R 000036 CW
0007 R 000010 PACCEL     0003   000056 PAGSIZ     0005 R 000050 PCOND      0003   000122 PCTSZ      0005 R 000025 PCUM
0006   000103 PJOIN      0005 R 000043 PPASS      0005 R 000042 PORAT      0006   000071 PCRATH     0005 R 000026 PRIRCM
0004 I 000106 PPNT       0000 I 000001 PRNTIM     0005 R 000031 PPOP       0004 I 000115 PROUT      0007   000114 PRTLEV
0004 I 000114 PPTME      0005 R 000044 PST        0003   000054 SAVTAP     0006   000065 SPLTH      0006   000062 SEPTH
0003   000067 SERIAL     0007   000002 SKBNE      0007   000003 SKCHI      0003   000120 SPCPAS     0006   000112 SPLCR
0005 R 000022 SPFAC      0007   000056 SPLMAX     0006   000072 SPMVTH     0004 I 000010 SYM        0003   000057 TAPCHK
0003   000007 TAPESV     0003   000063 THRSYM     0004 I 000116 TOTPIX     0004 I 000001 TOTWRD     0007   000004 TRBND
0007   000005 TRCHI      0003   000060 TRNSYM     0003   000061 TSTSYM     0007   000006 URKEND     0007   000007 URKCHI
0007 R 000014 VACCEL     0003   000103 VARSIZ     0003   000104 VARSZ2     0003   000105 VARSZ4     0006   000063 VFAC
0005 R 000037 VOLIN      0006   000101 VOLLIM     0005 R 000040 VOLRT      0005 R 000050 VRIN       0006   000104 VRJOIN
0005 R 000034 W          0005 R 000073 WADJ       0006   000060 WADJIN     0007   000000 WAIT       0006   000106 WDELSM
0006   000067 WFAC       0006   000105 WSIM       0006   000050 WTINIT     0003   000116 XHGH       0003   000117 XLOW
0006   000066 XOVFLO     0003   000106 XSIZ       0006   000057 XUNFLO     0003   000115 YSIZ       0000 R 000013 Z
0010 R 000000 ZOR        0000 R 000014 ZWRD
```

```
*DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
            SUBROUTINE CLASY1(XWRD,BUFR,DATAB)
      C*************************************************************************
      C*   THE PURPOSE OF SUBROUTINE CLASY1 IS TO SCRAMBLE THE PIXELS AND STORE
      C*   THEM FOR CLASY2 (ENTRY POINT IN CLASY1) TO SEND THEM IN BLOCKS TO
      C*   STATIS.
      C*************************************************************************
            IMPLICIT INTEGER(A-X)
            INCLUDE COMBK1,LIST
      COMBK1 PROC
            COMMON /INFORM/HEAD(42), MAPTAP,     LATAPE,     SAVTAP,    MAXFET,
           1               PAGSIZ, TAPCHK,     TRNSYM,     TSTSYM,
           2               DUPSYM, THRSYM,     MAXDIV,     MINDIV,    SPLMAX,
           3               SERIAL, TAPESV,     FILESV,
           4               MAXCLS, NOCLS2,     MAXFLD,     NOFLD2,    NOFLD3,
           5               NOTRED, NOFEAT,     NOFET2,     NOFET4,    VARSIZ,
           6               VARSZ2, VARSZ4,     XSIZ   ,    NOSPEC,    NOHIST,
           *               NOGRP,  DIVSIZ,   KEEPLV,   PRTLEV,   YSIZ  ,
           8               XHGH   , XLOW   ,    SPCPAS,     NOCLS3,    PCTSZ  ,
           9          IBLOCK(30), FETVEC(30), FETVC2(30), HISVEC(33),
           *          INVERT(30), RESTVC(30)
       END COMBK1
            INCLUDE CMBK1C,LIST
      CMBK1C PROC
            PARAMETER MAXPOP=61
            INTEGER IBEGIN,TOTWRD,CLSNAM,MAXBF,IPT,NWDS,NOFLD,AREA,
           * SYM,LNCAT,PRNT,KKT,KLBC,PRTME,PHOUT,NWDRS,NPTS
             INTEGER TOTPIX,LBUF,IO1
            COMMON/CLUSTR/IBEGIN,TOTWRD,CLSNAM,MAXBF,IPT,NWDS,NOFLD,AREA
           *      ,SYM(MAXPOP),LNCAT,PRNT(4),KKT,KLBC,PRTME,PROUT,
           *    TOTPIX,NWDRS,NPTS,LBUF,IO1
            DATA PRNT(4)/1C/
       END CMBK1C
            INCLUDE MISH,LIST
      MISH  PROC
            PARAMETER NTBSIZ=32
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      C   NTBSIZ MUST BE A POWER OF TWO.  (NTB & NTBSIZ ARE USED IN THE
      C       STORAGE ALLOCATOR CALLOC, ENTERED BY GET AND FREL.)
      C
      C     THE FOLLOWING TWO PARAMETERS GIVE THE NUMBER OF
      C       SYMMETRIC MATRICES AND VECTORS IN A CLUSTER NODE.
            PARAMETER NARRS=3
            PARAMETER NVECS=3
            PARAMETER NSCALS=25
            INTEGER JUNK(12), NTB(NTBSIZ), LINK(999)
            REAL W(1),PORAT(1),VOLIN(1),VOLRT(1),DCON(1),PPASS(1),PCOND(1)
            INTEGER LSUBS(1),IDADJ(1),INDEX(1),LSUPER(1),NSYMB(1)
            REAL VRIN(475),GEN(999),GREF(999),ALINK(1)
            REAL PST(1),PCUM(1),DISS(1),WADJ(1),OPROP(1),OW(1),SPFAC(1)
            REAL PRIRCM(1),OPRIOR(1),PROP(1),CIN(1),CTOT(1),OCIN(1),ODEN(1)
      C  THE FOLLOWING STRUCTURES THE CLUSTER NODES.  'GEN' IS
      C     USED LATER AS THE BASE FOR ALL THE ARRAYS.  (SEE CLINIT)
      C          'GREF' IS USED LIKE 'GEN', BUT INTERNAL TO ROUTINES,
      C               RATHER THAN IN CALLING SEQUENCES.
            EQUIVALENCE (LINK(38),LSUBS(30),LSUPER(29),IDADJ(28),NSYMB(12),
           1    INDEX(27),PCUM(26),PRIRCM(25),CIN(24),CTOT(23),
           2     PROP(22),SPFAC(21),WADJ(20),W(19),OPROP(18),OW(17),
           3     VOLIN(16),VOLRT(15),DCON(14),PORAT(13),DISS(12),PPASS(12),
           4     PST(11),OCIN(10),PCOND(7),VRIN(7),GEN(7),OPRIOR(9),ODEN(8),
           5     GREF(8),NTB(31))
            EQUIVALENCE (LINK(1),ALINK(1))
            COMMON/CLUS/  JUNK,NARR,NTOP,NTBSZM,NWANT,LINK
      C  WE ASSIGN VARIOUS LOCATION PARAMETERS IN CLINIT.  THESE
      C     ARE STORED MOSTLY IN THE ARRAYS LR FOR ARRAYS AND LV
      C     FOR VECTORS.  THE FOLLOWING EQUIVALENCES ARE PROVIDED
      C     FOR CONVENIENCE.  (FOR EXAMPLE, GEN(LSUM+KL+I) IS THE
      C     VECTOR SUM(KL+I) .)
            INTEGER NXARR(3),LR(NARRS),LV(NVECS)
            EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT),
           1    (LR(3),LOVAR),(LV(1),LSUM),(LV(2),LSK(N),(LV(3),LOSUM)
            COMMON /MISC/ MO,MM,LR,LV,NINCLS,MXAR,NTINIT,PROOT,EPS,DELT,
           1    AMC,ODCON,XOVFLO,XUNFLO,WADJIN,CLIMTH,SEPTH,VFAC,AMM,SBLTH,
           2    INDXVL,WFAC,NPTSC,PORATH,SPMVTH,DWFAC,GRACTM,AMOFAC,
           3     AMOMIN,AMOMAX,AMORAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIM,WDELSM,
           4     BETTER,MODE,CORLEN,SPCOR
            REAL PACCEL(2),MACCEL(2),VACCEL(2)
            COMMON /STFAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI,
           1    PACCEL,MACCEL,VACCEL
      C   MQMAX IS THE MAXIMUM NUMBER OF CHANNELS (30 FOR LARSYS)
            PARAMETER MQMAX=4
      C       MQTMAX=(MQMAX*(MQMAX+1))/2
            INTEGER GET
            PARAMETER MQTMAX=10
       END MISH
            REAL DATAB,BUFR
            AENORMAL ZOR
            DIMENSION BUFR(NWORDS),DATAB(NOFEAT,NPTS),XWRD(NPTS)
      C*****************************************************************
            NTOTAL = 0
            TOTPIX=TOTWRD/NOFEAT
            PRNTIM = 0
            NLAST = 0
            NREAD = 0
```

B-15 52

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DRADD = IBEGIN
      DO 5 IM=1,NPTS
    5 XWRD(IM) = IM
C      READ DATA FROM DRUM IN LAREE BLOCK,(BLOCKSIZE = NWDRS)
   15 CONTINUE
      IF(NREAD.EQ.IC1) GO TO 36
      N=1
   16 IF(ISTAT.EQ.1) GO TO 16
      IF(ISTAT.EQ.2)WRITE(6,17)DRADD
   17 FORMAT(1H0,'***ERROR*** RREAD TRY TO READ THE ILLEGAL ADDRES',I10)
      IF(ISTAT.EQ.2)CALL CMERR
      CALL RREAD(DRADD,BUFR(N),NWDRS,ISTAT)
      NREAD=NREAD + 1
      DRADD=DRADD + NWDRS
      NTOTAL = NTOTAL + NWDRS
      IF((TOTWRD-NTOTAL).LE.NWDRS)NLAST=TOTWRD-NTOTAL
      IF(NLAST.GT.0)NWDRS=NLAST
      IF(IMN.NE.1.AND.NREAD.NE.1) GO TO 24
C      SCRAMBLE AND TRANFER THE DATA FOR STATIS
   35 CONTINUE
      K=0
      NU=NPTS
      IF(NREAD.EQ.IC1.AND.LBUF.NE.0)NPTS=LBUF/NOFEAT
      IF(NREAD.EQ.IC1.AND.LBUF.NE.0)NU=NPTS
      IF(IMN.NE.1) GO TO 50
      Z=Z(R(14,927)
   18 ZWRD=Z0P(C)
      IX=NU * ZWRD + 1
      NTEMP=XWRD(NU)
      XWRD(NU)=XWRD(IX)
      XWRD(IX)=NTEMP
      NU=NU - 1
      IF(NU.GT.1)GO TO 18
   50 DO 51 IJ=1,NPTS
      IP=XWRD(IJ)
      IP=IP*NOFEAT - NOFEAT + 1
      DO 49 JI=1,NOFEAT
      K=IP + JI - 1
   49 DATAB(JI,IJ)=BUFR(K)
   51 CONTINUE
      IF(KKT.EQ.0.OR.KKT.EQ.1)WRITE(6,52)
      IF(KKT.EQ.0.OR.KKT.EQ.1)WRITE(6,53)((DATAB(II,JJ),II=1,NOFEAT),JJ=
     * 1,NPTS)
   52 FORMAT(1H0,'***DATAB THE SCRAMBLE DATA ARRAY IS DUMP BELOW
     ****',/)
   53 FORMAT(1H0,12(F9.4))
      IMN=IMN + 1
      GO TO 15
   24 RETURN
      ENTRY CLASYZ(DATAB,NDO)
      NDO=NPTS
      KKT=KKT + 1
      IF(KKT.EQ.10)PRNTIM = PRNTIM + 1
      IF(KKT.EQ.10)GO TO 36
      GO TO 45
   36 IF(KKT.EQ.10.AND.PRNTIM.LE.(IC1-1))KKT=0
      IF(NREAD.GT.IMN)GO TO 35
```

```
                      IF(IEX.EQ.IQ1.AND.KKT.EQ.IC.AND.PRNTIM.LE.(IQ1-1))RETURN
                      IF(IMN.EQ.IQ1.AND.KKT.EQ.IC.AND.PRNTIM.EQ.IQ1)GO TO 40
                      IF(IMA.EQ.IQ1.AND.KKT.EQ.0.AND.PRNTIM.LE.IQ1)RETURN
                      IF(NREAD.EQ.IPN)GO TO 15
   40                 CONTINUE
                      IF(PRNTIM.NE.IQ1)GO TO 41
                      PROUT=2
                      CALL CLOUMPIKL8C1
                      PROUT=1
   41                 CONTINUE
                      RETURN
                      END
```

END OF COMPILATION: 1 DIAGNOSTICS.

B-47

```
@FOR,SUW CLDUMP,CLDUMP
FOR S0E3-04/06/77-13:37:34 (1,2)
-5,5
      LOGICAL ISPLIT,IJSP
-19,19
-20,20
 19   CALL CLPR(KL,LEVEL,GEN(LSUM),GEN(LSKEW),GEN(LKURT))
      IJSP=ISPLIT(KL)
      PRINT 310,IJSP,PROUT
 310  FORMAT(1HO,' ISPLIT,PROUT= ',2I5)

    SUBROUTINE CLDUMP    ENTRY POINT 000146

    STORAGE USED: CODE(1) 000153; DATA(0) 000033; BLANK COMMON(2) 000000

     COMMON BLOCKS:

     0003   CLUS    002017
     0004   MISC    000113
     0005   STPAR   000016
     0006   CLUSTR  000123

    EXTERNAL REFERENCES (BLOCK, NAME)

     0007   ISPLIT
     0010   CLPR
     0011   NPRT$
     0012   NI02$
     0013   NERR3$

    STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001    000023 10L       0001    000121 11L       0001    000114 17L       0001    000036 19L       0000    000007 21DF
     0000    000017 310F      0001    000020 9L        0001    000134 99L       0003 R  000020 ALINK     0004    000064 AMM
     0004    000075 AMOFAC    0004    000077 AMOMAX    0004    000076 AMOMIN    0004    000100 AMORAT    0004    000054 AMQ
     0006 I  000007 AREA      0004    000107 BETTER    0004    000102 BIAS      0003 R  000027 CIN       0006 I  000002 CLSNAM
     0005    000001 CONLV     0004    000111 CORLEN    0003 R  000036 CTOT      0003 R  000041 DCON      0004    000053 DELT
     0003 R  000043 DISS      0004    000073 DWFAC     0004    000061 ELIMTH    0004    000052 EPS       0003 P  000050 GEN
     0004    000074 GRACTM    0003 R  000047 GREF      0006 I  000000 IBEGIN    0003 I  000023 IDADJ     0000 L  000000 IJSP
     0003 I  000024 INDEX     0004    000066 INDXVL    0000    000024 INJP$     0006 I  000004 IPT       0006 I  000122 IQ1
     0007 L  000000 ISPLIT    0003 I  000000 JUNK      0000 I  000006 KEEP      0006 I  000112 KKT       0000 I  000002 KL
     0006 I  000113 KLBC      0000 I  000004 KLIN      0004    000051 KROOT     0000 I  000003 KROT      0006 I  000121 LBUF
     0000 I  000005 LEVEL     0003 I  000020 LINK      0004 I  000003 LKURT     0006 I  000105 LNCAT     0004    000007 LOSUM
     0004    000004 LOVAR     0004 I  000002 LR        0004 I  000006 LSKEW     0003 I  000021 LSUBS     0004 I  000005 LSUM
     0003 I  000022 LSUPER    0004 I  000005 LV        0004    000002 LVRIN     0005 R  000012 MACCEL    0006 I  000003 MAXBF
     0004    000001 MM        0004    000110 MODE      0004    000000 MQ        0004 I  000011 MXAP      0003    000014 NARL
     0004    000010 NINCLS    0000 I  000001 NOFKL     0006 I  000026 NOFLD     0006 I  000120 NPTS      0004    000070 NPTSO
     0003 I  000044 NSYMB     0003 I  000020 NTB       0003    000016 NTBSZM    0003    000015 NTOP      0003    000017 NWANT
     0006 I  000117 NWDRS     0006 I  000005 NWDS      0003 R  000045 OCIN      0004    000055 OCON      0003 R  000047 ODEN
     0003 R  000046 OPRIOR    0003 R  000035 OPROP     0003 R  000036 OW        0005 R  000010 PACCEL    0003 R  000050 PCOND
     0003 R  000025 PCUM      0004    000103 PJOIN     0003 R  000043 PPASS     0003 R  000042 PQRAT     0004    000071 PQRATM
     0003 R  000026 PPIRCM    0006 I  000106 PRNT      0003 R  000031 PROP      0006 I  000115 PROUT     0006 I  000114 PRTME
     0003 R  000044 PST       0004    000065 SBLTH     0004    000062 SEPTH     0005    000002 SKBND     0005    000003 SKCHI
```

ORIGINAL PAGE IS OF POOR QUALITY

```
0004    000112 SPCOR     0003 P 000032 SPFAC     0004    000072 SPMVTM    0006 I 000010 SYM       0006 I 000116 TOTPIX
0006 I  000001 TOTWRD    0005   000004 TRBND     0005    000005 TRCHI     0005   000006 URKBND    0005   000007 URKCHI
0005 R  000014 VACCEL    0004   000063 VFAC      0003 P  000037 VOLIN     0004   000101 VOLLIM    0003 P 000040 VOLPT
0003 R  000050 VRIN      0004   000104 VRJOIN    0003 R  000034 W         0003 P 000033 WADJ      0004   000060 WADJIN
0005    000000 WAIT      0004   000106 WDELSM    0004    000067 WFAC      0004   000105 WSIM      0004   000050 WTINIT
0004    000056 XOVFLO    0004   000057 XUNFLO
```

```
00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101      1*          SUBROUTINE CLDUMP(KLHED)                                                  000000
00101      2*      C   THIS ROUTINE PRINTS OUT ALL THE CLASSES VIA ROUTINE 'CLPR'.               000000
00103      3*          INCLUDE MISH                                                              000000
00131      4*          INCLUDE CMBKID                                                            000000
00140      5*          LOGICAL ISPLIT,IJSP                                                  NEW=000000
00141      6*          NOFKL=0                                                              -01=000000
00142      7*          KL=KLHED                                                                  000000
00143      8*          KROT=LSUPER(KL)                                                           000003
00144      9*          KLIN=LINK(KL)                                                             000005
00145     10*          PRINT 210,INDEX(KL),KL                                                    000007
00151     11*      210 FORMAT('1DUMP OF OBSERVED CLUSTERS FROM',I3,I7)                           000015
00152     12*          LEVEL=0                                                                   000015
00153     13*          GO TO 11                                                                  000016
00154     14*        9 LEVEL=LEVEL+1                                                             000020
00155     15*       10 NSYMB(KL)=0                                                             000023
00156     16*          IF(ISPLIT(KL)) GO TO 19                                                   000024
00160     17*          NOFKL=NOFKL+1                                                             000030
00161     18*          NSYMB(KL)=NOFKL                                                           000034
00162     19*       19 CALL CLPR(KL,LEVEL,GEN(LSUM),GEN(LSKEW),GEN(LKURT))                  NEW=000036
00163     20*          IJSP=ISPLIT(KL)                                                      NEW=000055
00164     21*          PRINT 310,IJSP,PROUT                                                 NEW=000061
00170     22*      310 FORMAT(1HC,' ISPLIT,PROUT= ',2I5)                                    NEW=000070
00171     23*          KEEP=KL                                                              -02=000073
00172     24*          KL=LSUBS(KL)                                                              000072
00173     25*          IF(KL.NE.0.AND.(ISPLIT(KEEP).OR.PROUT.LE.2)) GO TO 9                      000074
00175     26*       17 KL=LINK(KEEP)                                                             000114
00176     27*          IF(KL.EQ.KLIN) GO TO 99                                                   000116
00200     28*       11 IF(KL)10,29,10                                                            000121
00203     29*       29 KL=LSUPER(KEEP)                                                           000122
00204     30*          LEVEL=LEVEL-1                                                             000125
00205     31*          KEEP=KL                                                                   000130
00206     32*          IF(KL.NE.KROT) GO TO 17                                                   000131
00210     33*       99 RETURN                                                                    000134
00211     34*          END                                                                       000152

      END OF COMPILATION:          1 DIAGNOSTICS.
```

B-19 52

```
@FOR,SUW CLPR,CLPR
FOR SGE3-04/06/77-13:38:08 (1,2)
-2,2
-5
      REAL KTEMP(30)
-10,10
      LPCC=LSUPER(KL)
      IF(INDEX(KL).NE.0)PPR=PROP(KL)/PRIRCM(LPCC)
-11,11
      PRINT 101,IN,INDEX(KL),PRR,W(LPCC),SPFAC(KL),W(KL),OW(KL),
-21,22
      LPCD=LINK(KL)
      LPCDC=LSUBS(KL)
      PRINT 102,KL,INDEX(LPCD),LINK(KL),INDEX(LPCDC),LSUBS(KL),
     1   INDEX(LPCC),LSUPER(KL),NSYMB(KL)
-37,37
      LOD=LOCK(I,I)
      TRK=TRK+KURT(KL+LOD)
-46,47
      LA=GET(MQS)
      LB=GET(MQS)
-64,64
      DO 300 J=1,MQ
       LOD=LOCK(I,J)
       N=LOD+KL
 300  KTEMP(J)=KURT(N)
      PRINT 108, (KTEMP(J),J=1,MQ)
-67,67
      DO 308 J=1,MQ
        LOD=LOCK(I,J)
      LCH=LOD+KL
 308  KTEMP(J)=KURT(LCH)
  8   PRINT 105,I, (KTEMP(J),J=1,MQ)

   SUBROUTINE CLPR      ENTRY POINT 00C565

   STORAGE USED: CODE(1) 000706; DATA(0) 000343; BLANK COMMON(2) 000CC0

    COMMON BLOCKS:

    0003   CLUS   002017
    0004   MISC   000113
    0005   STPAR  000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006   GET
    0007   LOCK
    0010   SQMTX
    0011   MINV
    0012   FREE
    0013   NPRT$
    0014   NIO1$
    0015   NIO2$
```

B-20

ORIGINAL PAGE IS OF POOR QUALITY

0016 NERR3$

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

```
0000    000067 101F      0000    000161 102F      0000    000234 105F      0000    000242 107F      0000    000252 108F
0000    000301 109F      0000    000200 112F      0000    000215 113F      0000    000224 116F      0001    000473 15CL
0000    000252 163F      0000    000271 166F      0001    000627 200L      0001    000162 227G      0001    000224 241G
0001    000301 254G      0001    000330 262G      0001    000340 270G      0001    000346 275G      0001    000364 307G
0001    000372 315G      0001    000422 325G      0001    000430 333G      0001    000436 336G      0001    000465 347G
0001    000535 360G      0001    000554 367G      0001    000603 375G      0001    000613 4C3G      0001    000621 41CG
0001    000142 5L        0003 R  000020 ALINK     0000 R  000036 AMEAN     0004    000064 AMM       0004    000075 AMOFAC
0004    000077 AMOMAX    0004    000076 AMOMIN    0004    000100 AMORAT    0004    000054 AMQ       0004    000107 BETTER
0004    000102 BIAS      0003 R  000027 CIN       0005    000001 CONLV     0004    000111 CORLEN    0003 R  000030 CTOT
0000 R  000063 CVL       0003 R  000041 DCON      0004    000053 DELT      0003 R  000043 DISS      0004    000073 DWFAC
0004    000061 ELIMTH    0004    000052 EPS       0003 R  000050 GEN       0006 I  000020 GET       0004    000074 GRACTW
0003 R  000047 GREF      0000 I  000056 I         0003 I  000023 IDADJ     0003 I  000024 INDEX     0004    000046 INDXWL
0000    000306 INJP$     0000 I  000064 J         0003 I  000000 JUNK      0004    000051 KPOOT     0000 R  000000 KTEMP
0000 I  000061 LA        0000 I  000062 LB        0000 I  000066 LCH       0003 I  000020 LINK      0004    000003 LKURT
0007 I  000000 LOCK      0000 I  000057 LOD       0004 I  000007 LOSUM     0004 I  000004 LQVAR     0000 I  000047 LPCC
0000 I  000050 LPCD      0000 I  000051 LPCDC     0004 I  000002 LR        0004    000006 LSKEW     0003 I  000021 LSUPS
0004    000065 LSUM      0003 I  000022 LSUPER    0004 I  000005 LV        0004    000002 LVRIN     0005 R  000012 MACCEL
0004    000001 MM        0004    000110 MODE      0004 I  000000 MQ        0000 I  000060 MQS       0004 I  000011 MXAR
0000 I  000065 N         0003    000014 NARL      0004    000010 NINCLS    0004    000070 NPTSQ     0003 I  000044 NSYMS
0003 I  000020 NTB       0003    000016 NTBSZM    0003    000015 NTOP      0003    000017 NWART     0003 R  000045 OCIN
0004    000055 OCOCN     0003 R  000047 OGEN      0000 R  000042 OMEAN     0003 R  000046 OPRTOR    0003 R  000035 OPPOP
0000 R  000055 OVSW      0003 R  000036 OW        0000 R  000053 OWUSE     0005 R  000010 PACCEL    0003 P  000050 PCONS
0003 R  000025 PCUM      0004    000103 PJOIN     0003 R  000043 PPASS     0003 R  000042 PQRAT     0004    000071 PQRATH
0003 R  000026 PRIPCM    0003 R  000031 PROP      0000 R  000046 PRR       0003 R  000044 PST       0004    000065 SBLTH
0004    000062 SEPTH     0005    000002 SKBND     0005    000003 SKCHI     0004    000112 SPCOR     0003 P  000032 SPFAC
0004    000072 SPMVTH    0005    000004 TREND     0005    000005 TRCHI     0000 R  000054 TPK       0005    000006 URKBND
0005    000007 URKCHI    0005 R  000014 VACCEL    0004    000063 VFAC      0003 R  000037 VOLIN     0004    000101 VOLLIM
0003 P  000040 VOLRT     0003 R  000059 VRIN      0004    000104 VRJOIN    0003 R  000034 W         0003 R  000033 WADJ
0004    000060 WADJIN    0005    000000 WAIT      0004    000106 WDELSM    0004    000067 WFAC      0004    000105 WSIP
0004    000050 WTINIT    0000 R  000052 WUSE      0004    000056 XOVFLO    0004    000057 XUNFLO
```

B-21 58

```
00101      1*            SUBROUTINE CLPR(KL,IN,SUM,SKEW,KURT)                                    000010
00101      2*      C  THIS ROUTINE PRINTS OUT ALL THE VARIABLES BELONGING TO SOME                -D1000010
00101      3*      C   CLASS INDEXED BY KL.                                                       000010
00103      4*            INCLUDE MISH                                                            000010
00131      5*             REAL KTEMP(30)                                                        NEW000010
00132      6*            REAL SUM(1),SKEW(1),KURT(1)                                             000010
00133      7*            REAL AMEAN(MOMAX),OMEAN(MOMAX)                                          000010
00134      8*            IF(KL.EQ.0) RETURN                                                      000010
00136      9*            PRR=1.                                                                  000015
00137     10*            LPCC=LSUPER(KL)                                                        NEW000020
00140     11*            IF(INDEX(KL).NE.0)PRR=PROP(KL)/PRIPCM(LPCC)                            NEW000025
00142     12*            PRINT 101,IN,INDEX(KL),PRR,W(LPCC),SPFAC(KL),W(KL),OW(KL),             NEW000034
00142     13*           1 WADJ(KL),IDADJ(KL),PROP(KL),CIN(KL),CTOT(KL),OPROP(KL),OCIN(KL),      -D2000034
00142     14*           2   OGEN(KL),PQRAT(KL),VOLIN(KL),VOLRT(KL),OCON(KL)                      000034
00167     15*          101 FORMAT('0CLUSTER',I4,' INDEX',I4,'  PROPORTION',F9.6,                 000067
00167     16*           1     '  W=',F9.3,'  SPLIT',F9.4/                                       000067
00167     17*           2  5X,'WEIGHT',F12.3,2X,'WAS',F12.3,                                    000067
00167     18*           3    4X,'ADJUST',F12.3,'  ID',I6/                                       000067
```

```
00167     19*            4   5X,'PROPORTION: PROP',F8.5,'  CIN',E12.5,'  CTOT',E12.5/5X,                    000267
00167     20*            5 'OLD PROP',F9.6,'  CIN',E10.5,'  GDEN',E10.5,'  DIFFER',E10.5/                   000267
00167     21*            6   5X,'VOLUME',E14.6,'  POOT',E14.6,'  DCON',F13.7)                              000267
00170     22*             LPCDC=LINK(KL)                                                                  NEW000067
00171     23*             LPCDC=LSUBS(KL)                                                                 NEW000072
00172     24*             PRINT 102,KL,INDEX(LPCD),LINK(KL),INDEX(LPCDC),LSUBS(KL),                       NEW000074
00172     25*            1   INDEX(LPCC),LSUPER(KL),NSYMB(KL)                                             NEW000074
00204     26*          102 FORMAT('     LOCATION',I5,'  LINK',I3,I5,'  SUBS',I3,I5,'  SUPER',              -02000111
00204     27*            1   I3,I5,'  SYMBOL',I3)                                                         000111
00205     28*             PRINT 112,PST(KL),PCOND(KL),PCUM(KL),PRIRCM(KL)                                 000111
00213     29*          112 FORMAT(5X,'NET PROB',E10.5,'  DIRECT',E10.5,'  CUMS',                          000121
00213     30*            1   E10.5,' *',E10.5)                                                           000121
00214     31*             IF(INDEX(KL).EQ.0) RETURN                                                       000121
00216     32*             WUSE=W(KL)                                                                      000127
00217     33*             OWUSE=OW(KL)                                                                    000132
00220     34*             IF(INDEX(KL).GE.0) GO TO 5                                                      000134
00222     35*             WUSE=CW(KL)                                                                     000137
00223     36*             OWUSE=W(KL)                                                                     000140
00224     37*           5 TPK=0.                                                                          000142
00225     38*             OVSK=0.                                                                         000142
00226     39*             DO 2 I=1,MQ                                                                     000162
00231     40*             LOD=LOCK(I,I)                                                                 NEW000162
00232     41*             TPK=TPK+KURT(KL+LOD)                                                          NEW000167
00233     42*             OVSK=OVSK+GREF(LOSUM+KL+I)**2                                                  -01000175
00234     43*             OMEAN(I)=GREF(LOSUM+I+KL)/OW(KL)                                                000204
00235     44*           2 AMEAN(I)=SUM(I+KL)/W(KL)                                                        000207
00237     45*             PRINT 113,(AMEAN(I),I=1,MQ)                                                     000214
00245     46*         113 FORMAT('0    MEAN',10X,5E13.6/(12X,5E13.6))                                     000227
00245     47*  C      D       PRINT 103,(SUM(KL+I),I=1,MQ)                                               000227
00245     48*  C      D   103 FORMAT('0   SUM(=MEAN*W)',2X,5E13.6/(12X,5E13.6))                          000227
00246     49*             MQS=MQ*MQ                                                                       000227
00247     50*             LA=GET(MQS)                                                                   NEW000232
00250     51*             LB=GET(MQS)                                                                   NEW000236
00251     52*             CALL SCMTX(ALINK(LB),VRIN(KL+1))                                               -02000242
00252     53*             CALL MINV(ALINK(LA),ALINK(LB),ALINK(LB),CVL)                                    000253
00253     54*             DO 6 I=1,MQS                                                                    000301
00256     55*           6 ALINK(LA+I-1)=ALINK(LA+I-1)/WUSE                                                000301
00260     56*             PRINT 116,(ALINK(LA+J-1),J=1,MQ)                                                000304
00266     57*         116 FORMAT('0   COVARIANCE    ',5E13.6/(16X,5E13.6))                              000340
00267     58*             DO 7 I=2,MQ                                                                     000340
00272     59*           7 PRINT 105, I,(ALINK(LA+MQ*I+J-MQ-1),J=1,MQ)                                     000340
00302     60*         105 FORMAT(11X,I5,2X,5E13.6/(16X,5E13.6))                                           000353
00302     61*  C      D       PRINT 106,(VRIN(KL+LOC(I,J)),J=1,MQ)                                       000353
00302     62*  C      D   106 FORMAT('0   VRIN(=COV**-1)',5E13.6/(16X,5E13.6))                           000353
00302     63*  C      D       DO 5 I=2,MQ                                                                000353
00302     64*  C      D     5 PRINT 105,I,(VRIN(KL+LOC(I,J)),J=1,MQ)                                     000353
00303     65*             IF(TPK.EQ.0.) GO TO 150                                                         000353
00305     66*             PRINT 107,(SKEW(KL+I),I=1,MQ)                                                   000355
00313     67*         107 FORMAT('0   SKEW(*W)   ',4X,5E13.6/(12X,5E13.6))                               000372
00314     68*             DO 300 J=1,MQ                                                                 NEW000372
00317     69*              LOD=LOCK(1,J)                                                                NEW000372
00320     70*              N=LOD+KL                                                                     NEW000377
00321     71*       300   KTEMP(J)=KURT(N)                                                              NEW000403
00323     72*             PRINT 108, (KTEMP(J),J=1,MQ)                                                  NEW000407
00331     73*         108 FORMAT('0   KURT(*W)   1',2X,5E13.6/(16X,5E13.6))                              -01000430
00332     74*             DO 8 I=2,MQ                                                                     000430
00335     75*             DO 308 J=1,MQ                                                                 NEW000436
```

B-22 59

```
00340             LOD=LOCK(I,J)                                              NEW000436
00341           LCH=LOD+KL                                                   NEW000443
00342    308    KTEMP(J)=KURT(LCH)                                           NEW000447
00344    8      PRINT 105,I, (KTEMP(J),J=1,MQ)                               NEW000453
00354    150    IF(OVSK.EQ.0..OR.OPROP(KL).EQ.PROP(KL).AND.INDEX(KL).GT.0)  -01C00473
00354         1    GO TO 200                                                 000473
00356           PRINT 163,(OMEAN(I),I=1,MQ)                                  000514
00364    163    FORMAT('     OLD MEAN',6X,5E13.6/(12X,5E13.6))               000540
00365           CALL SOMTX(ALINK(LA),GREF(LOVAR+KL+1))                       000540
00366           DO 156 I=1,MQS                                               000554
00371    156    ALINK(LA+I-1)=ALINK(LA+I-1)/OWUSE                            000554
00373           PRINT 166,(ALINK(LA+J-1),J=1,MQ)                             000557
00401    166    FORMAT('0    OLD COVARIANCE',5E13.6/(16X,5E13.6))            000613
00402           DO 157 I=2,MQ                                                000613
00405    157    PRINT 105,I,(ALINK(LA+MQ*I+J-MQ-1),J=1,MQ)                   000613
00415    200    CALL FREE(LA,MQS)                                            000627
00416           CALL FREE(LB,MQS)                                            000632
00417           PRINT 109                                                    000636
00421    109    FORMAT(/)                                                    000642
00422           RETURN                                                       000642
00423           END                                                          000705

     END OF COMPILATION:        NO  DIAGNOSTICS.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
PROP,SUV,CLUST,CLUST
FOR SEE3-04706/77-13:38:23 (2,3)
-15
C USES PCOND FOR PPASS,DIST FOR DISS(KL)
-18,18
      PCOND(KROT)=1.
-25,25
  130 PCOND(KL)=PROP(KL)/(PRIRCM(KFATH))*PCOND(KFATH)
-34,35
      DIST=DOTSQ(REL,VRIN(KL+1))*W(KL)
      IF(ABS(DIST*DCON(KL)).LE.160)GO TO 531
-38,38
      P=EXP(DIST*DCON(KL))/VOLRT(KL)*PCOND(KL)
-47,47
  149 PCOND(KL)=0.
      KL=KFATH
```

SUBROUTINE CLUST ENTRY POINT 000207

STORAGE USED: CODE(1) 000225; DATA(0) 000055; BLANK COMMON(2) 000000

COMMON BLOCKS:

0003	CLUS	002017
0004	MISC	000113
0005	STPAP	000016
0006	BIGCOM	000001

EXTERNAL REFERENCES (BLOCK, NAME)

0007	ISPLIT
0010	CORECT
0011	DOTSQ
0012	EXP
0013	NERR31

B24 G1

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001		000024	139L	0001		000044	131L	0001		000141	139L	0001		000011	2400
0000		000017	647F	0003	R	000020	ALINK	0004		000064	AMM	0004		000075	AMOFAC
0004		000076	AMOMIN	0004		000100	AMORAT	0004		000054	AMU	0004		000107	BETTER
0006		000000	BIGCOM	0003	P	000027	CIN	0005		000001	CONLV	0004		000111	CORLEN
0003	R	000041	DCON	0004		000053	DELT	0003	R	000043	DISS	0000	R	000014	DIST
0004		000073	DWFAC	0004		000061	ELIMTH	0004		000052	EPS	0003	R	000050	GEN
0003	R	000047	GREF	0003	I	000023	IDADJ	0000	I	000005	IDO	0003	I	000024	INDEX
0000		000035	INJPS	0000	I	000011	ISEC	0007	L	000000	ISPLIT	0000	I	000016	IW
0000	I	000013	KFATH	0000	I	000012	KL	0000	I	000007	KMAX	0004		000051	KROOT
0003	I	000020	LINK	0004		000003	LKURT	0004		000007	LOSLM	0004		000004	LOVAR
0004		000006	LSKEW	0003	I	000021	LSURS	0004		000005	LSUM	0003	I	000022	LSUPER
0004		000002	LVRIN	0005	R	000012	MACCEL	0004		000001	MM	0004		000110	MODE
0004	I	000011	MXAR	0003		000014	NARL	0004		000010	NINCLS	0004		000070	NPTSQ
0003	I	000020	NTB	0003		000016	NTBSZM	0003		000015	NTOP	0003		000017	NWANT
0004		000055	ODCON	0003	R	000047	ODEN	0003	R	000046	OPRIOR	0003	R	000035	OPROP

0001		000105	531L	0004		000077	AMOMAX	0004		000102	BIAS	0004		000036	CTOT
0011	R	000000	DOTSQ	0004		000074	GRACTM	0004		000066	INDXVL	0003	I	000000	JUNK
0003	I	000004	KROT	0004	I	000002	LR	0004	I	000005	LV	0004		000000	MQ
0003	I	000044	NSYMB	0003	R	000045	OCIN	0003		000036	OW				

```
0000 R 000015 P        0005 R 000010 PACCEL    0003 R 000050 PCOND     0003 R 000025 PCUM      0004   000103 PJOIN
0000 R 000006 PMAX     0003 R 000043 PPASS     0003 R 000042 PORAT     0004   000071 PORATH    0003 R 000026 PRIPCM
0003 R 000031 PROP     0003 R 000044 PST       0000 R 000010 PTOT      0000 R 000000 REL       0004   000065 SBLTH
0004   000062 SEPTH    0005   000002 SKBND     0005   000003 SKCHI     0004   000112 SPCOR     0003 R 000032 SPFAC
0004   000072 SPMVTH   0005   000004 TRBND     0005   000005 TRCHI     0005   000006 URKBND    0005 R 000007 URKCHI
0005 P 000014 VACCEL   0004   000063 VFAC      0003 R 000037 VOLIN     0004   000101 VOLLIM    0003 R 000040 VOLRT
0003 R 000050 VRIN     0004   000104 VRJOIN    0003 R 000034 W         0003 R 000033 WADJ      0004   000060 WADJIN
0005   000000 WAIT     0004   000106 WDELSM    0004   000067 WFAC      0004   000105 WSIM      0004   000050 WTINIT
0004   000056 XOVFLO   0004   000057 XUNFLO
```

```
000000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101       1*          SUBROUTINE CLUST(BIGP,NDO,KLOUT,KROTIN,SUM)                                  000002
00101       2*      C   THIS PROGRAM TAKES EACH INPUT POINT AND CLASSIFIES IT,                        000002
00101       3*      C     FOR THE PURPOSE OF GENERATING A MAP.                                       000002
00103       4*            INCLUDE MISH                                                               000002
00131       5*            DIMENSION BIGP(MQ,NDO),KLOUT(NDO),SUM(1)                                   000002
00132       6*            COMMON/BIGCOM/ BIGDUM                                                      000002
00133       7*            REAL PEL(MQMAX)                                                            000002
00134       8*            LOGICAL ISPLIT                                                             000002
00134       9*      C     D      PARAMETER ITLIM=71                                                  000002
00134      10*      C     D      INTEGER KTR(ITLIM)                                                  000002
00134      11*      C     D      COMMON/KTRCOM/IT,KTR                                                000002
00135      12*            XP(DIST)=EXP(-.5*DIST)                                                     000002
00136      13*            KROT=KROTIN                                                                000002
00137      14*            DO 399 IDO=1,NDO                                                           000004
00137      15*      C  INSPECT EACH POINT                                                            000004
00137      16*      C USES PCOND FOR PPASS,DIST FOR DISS(KL)                                  NEW000004
00142      17*            PMAX=-1.                                                                   000011
00143      18*            KMAX=0                                                                     000013
00144      19*            PCOND(KROT)=1.                                                    NEW000014
00145      20*            PTOT=0.                                                           -01000015
00146      21*            ISEC=0                                                                     000016
00146      22*      C     D      IT=1                                                                000016
00147      23*            KL=LSUBS(KROT)                                                             000017
00150      24*            KPATH=KROT                                                                 000021
00150      25*      C   GO DOWN CLUSTER TREE                                                         000021
00151      26*       130    PCOND(KL)=PROP(KL)/(PRIRCM(KPATH))*PCOND(KPATH)                  NEW000024
00152      27*            IF(.NOT.ISPLIT(KL)) GO TO 131                                     -01000031
00154      28*            KPATH=KL                                                                   000035
00155      29*            KL=LSUBS(KL)                                                               000040
00156      30*            GO TO 130                                                                  000042
00157      31*       131  CALL CORFCT(REL,BIGP(1,IDO),W(KL),SUM(KL+1))                              000044
00157      32*      C     D      KTR(IT)=KL                                                          000044
00157      33*      C     D      IT=IT+1                                                             000044
00157      34*      C     D      IF(IT.GT.ITLIM) GO TO 309                                           000044
00160      35*            DIST=DOTSQ(PEL,VRIN(KL+1))*W(KL)                                  NEW000066
00161      36*            IF(ABS(DIST+DCON(KL)).LE..160)GO TO 531                           NEW000076
00163      37*            GO TO 139                                                         -02000103
00164      38*       531  CONTINUE                                                                   000105
00165      39*            P=XP(DIST+DCON(KL))/VOLRT(KL)*PCOND(KL)                           NEW000105
00166      40*            PTOT=PTOT+P                                                       -01000117
00167      41*            IF(P.LE.PMAX.OR.ISPLIT(KL))                                                000121
00167      42*           1   GO TO 139                                                               000121
00171      43*            PMAX=P                                                                     000134
```

B-25 62

```
00172     44*          KMAX=KL                                                    000136
00173     45*      139 KL=LINK(KL)                                                000141
00174     46*          IF(KL)130,149,130                                          000143
00174     47*    C     GO UP TREE                                                 000143
00177     48*      149 PCOND(KL)=0.                                           NEW 000145
00200     49*          KL=KPATH                                               NEW 000146
00201     50*          KPATH=LSUPER(KL)                                           G10 0151
00202     51*          IF(KL.NE.KROT) GO TO 131                                   000153
00204     52*      309 IW=KLOUT(IDO)                                              000155
00205     53*          KLOUT(IDO)=KMAX                                            000157
00206     54*          IF(PTOT.NE.0.) PMAX=PMAX/PTOT                              000161
00206     55*    C     D     IF(II-LI,ITLIM) GO TO 399                            000161
00206     56*    C     D   646 PRINT 647,IDO,W(KROT),KL,ISEC,(KTR(I),I=1,ITLIM)   000161
00210     57*      647 FORMAT('0LOOP IN CLUST:IDO,W(KROT),KL,SECTION',,I5,E11.5,2I5   000171
00210     58*         I    /(1X14I5))                                             000171
00211     59*      399 CONTINUE                                                   000171
00213     60*          RETURN                                                     000171
00214     61*          END                                                        000224

      END OF COMPILATION:          1 DIAGNOSTICS.
```

B-26 63

ORIGINAL

```
@FOR,FL  CMBK10,CMBK10
PDPTIC RL7C-6 04/C6-13:36:57-(C,0)
PEGC01        CMBK10 PROC
   0002              PARAMETER MAXPOP=41                                                    NEW
   0003              INTEGER IBEGIN ,TOTWRD,CLSNAM,MAXBF,IPT,NWDS,NOFLD,AREA,               NEW
   0004             *SYM,LNCAT,PRNT,KWT,KLBC,PRTME,PROUT,NWDRS,NPTS                         NEW
   0005              INTEGER TOTPIX,LBUF,I01                                                NEW
   0006              COMMON/CLUSTR/IBEGIN,TOTWRD,CLSNAM,MAXBF,IPT,NWDS,NOFLD,AREA           NEW
   0007             *      ,SYM(MAXPOP),LNCAT,PRNT(4),KWT,KLBC,PRTME,PROUT,               -05
   0008             *TOTPIX,NWDRS,NPTS,LBUF,I01                                           NEW
   0009              DATA PRNT(4)/10/                                                      +01
   0010        END CMBK10

END PDP  ERRORS : NONE
```

```
PROP,SUV ELIM,ELIM
FOR SCS-047C1777-19:44:01 (0,1)
-12,12
      LS=LSUBS(KEL)
      PRINT 719,INDEX(KEL),INDEX(KMEX),INDEX(LS),INDEX(KF)
-15,15
      LSS=LSUBS(KF)
      IF(KF.EQ.KROOT.AND.LINK(LSS).EQ.0)RETURN
-16,16
      LKI=LINK(LSS)
      IF(LINK(LKI).NE.0.OR.KF.EQ.KROOT)GO TO 5

   SUBROUTINE ELIM       ENTRY POINT 00016C

   STORAGE USED: CODE(1) 000176; DATA(0) 000046; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    CLUS    002017
    0004    MISC    000113
    0005    STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006    SUBLIM
    0007    TRFREE
    001C    NPPT$
    0011    NI02$
    0012    NI01$
    0013    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000132 13L      0001    000136 15L      0001    000070 5L       0000    000020 666F     0001    000075 7L
    0000    000007 719F     0003 R  000020 ALINK    0004    000064 AMM      0004    000075 AMOFAC   0004    000077 AMOMAX
    0004    000076 AMOMIN   0004    000100 AMORAT   0004    000054 AMU      0004    000107 BETTER   0004    000102 BIAS
    0003 P  000027 CIN      0005    000001 CONLV    0004    000111 CORLEN   0003 R  000330 CTOT    0003 R  000041 DCON
    0004    000053 DELT     0003 R  000043 DISS     0004    000073 DWFAC    0004    000061 ELIMTH   0004    000052 EPS
    0003 R  000050 GFN      0004    000074 GRACTM   0003 R  000047 GREF     0003 I  000023 IDADJ    0003 I  000024 INDEX
    0004    000066 INDXVL   0000    000032 INJPS    0003 I  000000 JUNK     0000 I  000005 K        0000 I  000001 KF
    0000 I  000001 KMEX     0000 I  000006 KOLD     0004 I  000051 KROOT    0003 I  000020 LINK     0000 I  000004 LKI
    0004    000003 LKURT    0004    000007 LOSUM    0004    000054 LOVAR    0004 I  000002 LR       0000 I  000002 LS
    0004    000006 LSKEW    0000 I  000003 LSS      0003 I  000021 LSUBS    0004    000005 LSUM     0003 I  000022 LSUPER
    0004 I  000035 LV       0004    000002 LVRIN    0005 R  000012 MACCEL   0004    000001 MM       0004    000110 MODE
    0004    000000 MO       0004 I  000011 MXAR     0003    000014 NARL     0004 I  000010 NINCLS    0004    000070 NPTSO
    0003 I  000043 NSYMB    0003 I  000020 NTB      0003    000016 NTBSZM   0003    000015 NTOP     0003    000017 NWANT
    0003 P  000045 OCIN     0004    000055 ODCON    0003 R  000047 ODEN     0003 R  000046 OPRIOP   0003 R  000035 OPROP
    0003 R  000036 OW       0005 R  000010 PACCEL   0003 R  000050 PCOND    0003 P  000025 PCUM     0004    000103 PJOIN
    0003 P  000043 PPASS    0003 R  000042 PQRAT    0004    000071 PQRATH   0003 R  000026 PPIRCM   0003 P  000031 PROP
    0003 P  000044 PST      0004    000065 SBLTH    0004    000062 SEPTH    0005    000002 SKBND    0005    000033 SKCHI
    0004    000112 SPCOR    0003 R  000032 SPFAC    0004    000072 SPMVTH   0005    000004 TRBND    0005    000005 TRCHI
    0005    000006 URKBND   0005    000007 URKCHI   0005 R  000014 VACCEL   0004    000063 VFAC     0003 R  000037 VOLIN
    0004    000101 VOLLIM   0003 R  000040 VOLRT    0003 R  000050 VRIN     0004    000104 VRJOIN   0003 P  000034 W
```

ORIGINAL PAGE IS OF POOR QUALITY

B-98

```
0003 P 000033 WADJ      0004   000060 WADJIN    0005   030000 WAII      0004   000106 WDELSM     0004   000067 WFAC
0004   000105 WSIM      0004   000050 WTINIT    0004   030056 XOVFLO    0004   030057 XUNFLO
```

```
000000  *DIAGNOSTIC*  THE NAME CEI APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00100   1*            SUBROUTINE ELIM(KEL)                                                    000000
00101   2*      C     THIS ROUTINE ELIMINATES THE CLUSTER KEL FROM THE CLUSTER TREE           000000
00101   3*      C       AND FREES THE STORAGE.                                                000000
00103   4*            INCLUDE MISH                                                            000000
00103   5*      C     D       PRINT 699                                                       000000
00103   6*      C     D  699 FORMAT('0ENTERING ELIM, WHICH MAY CONTAIN ERRORS.'/              000000
00103   7*      C     D      1    '  NO SITUATION REQUIRING THIS ROUTINE HAS BEEN FOUND',     000000
00103   8*      C     D      2    ' IN PRACTICE, SO IT IS INCOMPLETELY DEBUGGED.'/          000000
00103   9*      C     D      3    '       TO AVOID ELIM, SET PARAMETER NOELIM=1.')          000000
00131   10*           KF=LSUPER(KEL)                                                          000000
00132   11*           KMEX=LINK(KEL)                                                          000003
00133   12*           LS=LSUBS(KEL)                                                 NEW000006
00134   13*           PRINT 719,INDEX(KEL),INDEX(KMEX),INDEX(LS),INDEX(KF)          NEW000010
00142   14*       719 FORMAT('0##ELIMINATE',T4,'    LINK,LSUBS,LSUPER=',3I3)          -01000021
00142   15*     C FIRST, USE SUBLIM IF THERE ARE ONLY 2 SUBCLUSTERS AT THIS LEVEL.            000021
00143   16*           LSS=LSUBS(KF)                                                 NEW000021
00144   17*           IF(KF.EQ.KROOT.AND.LINK(LSS).EQ.0)RETURN                      NEW000025
00146   18*           LKI=LINK(LSS)                                                 NEW000043
00147   19*           IF(LINK(LKI).NE.0.OR.KF.EQ.KROOT)GO TO 5                      NEW000047
00151   20*         4 CALL SUBLIM(KF)                                                 -02000061
00152   21*           RETURN                                                                  000064
00153   22*     C   NOW WE REMOVE THE CLUSTER FROM VARIOUS LISTS.                             000064
00153   23*         5 K=LSUBS(KF)                                                           000070
00154   24*           IF(K.EQ.KEL) GO TO 13                                                   000072
00156   25*         7 KOLD=K                                                                000075
00157   26*           K=LINK(K)                                                               000101
00160   27*           IF(K.EQ.0) PRINT 666,KEL,KF,KOLD,LSUBS(KF)                              000105
00167   28*       666 FORMAT('0 **STRUCTURAL ERROR AT ELIM: KEL,KFATH,KOLD,INIT',5I9)         000121
00170   29*           IF(K.NE.KEL) GO TO 7                                                    000121
00172   30*           LINK(KOLD)=LINK(K)                                                      000124
00173   31*           GO TO 15                                                                000130
00174   32*        13 LSUBS(KF)=LINK(K)                                                       000132
00175   33*        15 LINK(KEL)=0                                                             000136
00175   34*     C   NOW DROP THE CLUSTER AND ITS SUBS                                         000136
00176   35*           CALL TREPFE(KEL,NINCLS)                                                 000137
00177   36*           RETURN                                                                  000143
00200   37*           END                                                                     000175
```

END OF COMPILATION: 1 DIAGNOSTICS.

B-29

```
@ASM,SUM LOC,LOC,LOCK
ASM13B RLIB69 04/01-19:42:51-(0,1)

-2,2
LOCK*  L A1,*0,X11
                                                 $(1)      AXRS
          01     000000   27 00 15 13 1 000000   LOCK*  L A1,*0,X11                                    NEW
                 000001   27 00 14 13 1 000001          L A0,*1,X11                                    -01
                 000002   55 00 01 00 0 000014          TG A1,A0
                 000003   74 04 00 00 0 000011          J BIGONE
                 000004   25 17 14 00 000001           AN,XU  A0,1
                 000005   31 17 00 14 0 000001          MSI,XU A0,1,A0
                 000006   73 04 00 00 0 000001          SSA A0,1
                 000007   24 00 14 00 0 000015          A  A0,A1
                 000010   74 04 00 13 0 000003          J 3,X11
                 000011   25 17 15 00 000001     BIGONE  AN,XU  A1,1
                 000012   31 17 01 15 0 000001          MSI,XU A1,1,A1
                 000013   73 04 01 00 0 000001          SSA A1,1
                 000014   24 00 14 00 0 000015          A A0,A1
                 000015   74 04 00 13 0 000003          J 3,X11
                                                           END
END ASM. ERRORS : NONE
```

B-30 67

ORIGINAL PAGE IS OF POOR QUALITY

```
@PDP,FL MISH,MISH
PEPTIC RL70-6 04/01-19:42:47-(0,0)
      MISH  PROC
            PARAMETER NTBSIZ=32
C     NTBSIZ MUST BE A POWER OF TWO.  (NTB & NTBSIZ ARE USED IN THE
C          STORAGE ALLOCATOR CALLOC, ENTERED BY GET AND FREE.)
C
C     THE FOLLOWING TWO PARAMETERS GIVE THE NUMBER OF
C          SYMMETRIC MATRICES AND VECTORS IN A CLUSTER NODE.
            PARAMETER NARRS=3
            PARAMETER NVECS=3
            PARAMETER NSCALS=25
            INTEGER JUNK(12), NTB(NTBSIZ), LINK(999)
            REAL W(1),PCRAT(1),VOLIN(1),VOLPT(1),DCON(1),PPASS(1),PCOND(1)
            INTEGER LSUBS(1),IDADJ(1),INDEX(1),LSUPER(1),NSYMB(1)
            REAL VRIN(475),GEN(999),GREF(999),ALINK(1)
            REAL PST(1),PCUM(1),DISS(1),WADJ(1),OPROP(1),OW(1),SPFAC(1)
            REAL PRIRCM(1),OPRIOR(1),PROP(1),CIN(1),CTOT(1),OCIN(1),ODEN(1)
C     THE FOLLOWING STRUCTURES THE CLUSTER NODES. 'GEN' IS
C          USED LATER AS THE BASE FOR ALL THE ARRAYS. (SEE CLINIT)
C            'GREF' IS USED LIKE 'GEN', BUT INTERNAL TO ROUTINES,
C               RATHER THAN IN CALLING SEQUENCES.
            EQUIVALENCE (LINK(31),LSUBS(30),LSUPER(29),IDADJ(28),NSYMB(12),      NEW
           1   INDEX(27),PCUM(26),PRIRCM(25),CIN(24),CTOT(23),                     -01
           2   PROP(22),SPFAC(21),WADJ(20),W(19),OPROP(18),OW(17),
           3   VOLIN(16),VOLPT(15),DCON(14),PCRAT(13),DISS(12),PPASS(12),
           4   PST(11),OCIN(10),PCOND(7),VPIN(7),GEN(7),OPRIOR(9),ODEN(8),
           5   GREF(8),NTB(31))
            EQUIVALENCE (LINK(1),ALINK(1))
            COMMON/CLUS/  JUNK,NAPL,NTOP,NTBSZM,NWANT,LINK
C     WE ASSIGN VARIOUS LOCATION PARAMETERS IN CLINIT.  THESE
C        ARE STORED MOSTLY IN THE ARRAYS LR FOR ARRAYS AND LV
C        FOR VECTORS.  THE FOLLOWING EQUIVALENCES ARE PROVIDED
C        FOR CONVENIENCE. (FOR EXAMPLE, GEN(LSUM+KL+I) IS THE
C        VECTOR SUM(KL+I) .)
            INTEGER MXAP(31),LR(NARRS),LV(NVECS)
            EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT),
           1   (LR(3),LOVAR),(LV(1),LSUM),(LV(2),LSKEW),(LV(3),LOSUM)
            COMMON /MISC/ MC,MM,LR,LV,NINCLS,MXAP,WTINIT,KROOT,EPS,DELT,
           1   AMC,ODCON,XOVFLO,XUNFLO,WADJIN,ELIMTH,SEPTH,VFAC,AMM,SBLTH,
           2   INDXVL,WFAC,NPISO,PCRATH,SPMVTH,DWFAC,GRACTH,AMGFAC,
           3   AMGMIN,AMGMAX,AMORAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIM,WDELSM,
           4   BETTER,MODE,CORLEN,SPCOR
            REAL PACCEL(2),MACCEL(2),VACCEL(2)
            COMMON /STPAR/WAIT,CONLV,SKRND,SKCHI,TRBND,TRCHI,URKBND,URKCHI,
           1   PACCEL,MACCEL,VACCEL
C     MCMAX IS THE MAXIMUM NUMBER OF CHANNELS (30 FOR LARSYS)
            PARAMETER MCMAX=4
C          MCTMAX=(MCMAX*(MCMAX+1))/2
            PARAMETER MCTMAX=10
```

B-37 68

```
@FOR,SUW MULTI,MULTI
FOR SOE3-04/06/77-13:37:45 (2,3)
-2,2
      SUBROUTINE MULTI(PV)
-6,7
-8,8
      DIMENSION PV(NOFEAT,NPTS)

    SUBROUTINE MULTI     ENTRY POINT 000100

    STORAGE USED: CODE(1) 000111; DATA(0) 000015; BLANK COMMON(2) 000000

     COMMON BLOCKS:

     0003    CLUS    002017
     0004    MISC    000113
     0005    STPAR   000016
     0006    INFORM  000407
     0007    CLUSTR  000123

    EXTERNAL REFERENCES (BLOCK, NAME)

     0010    DATE1X
     0011    ALFREE
     0012    CLINIT
     0013    STATIS
     0014    CLDUMP
     0015    NPRT$
     0016    NIO2$
     0017    NEPR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0000    000000 899F      0003 R 000020 ALINK     0004    000064 AMM       0004    000075 AMOFAC    0004    000077 AMOMAX
     0004    000076 AMOMIN    0004    000100 AMORAT   0004    000054 AMO       0007 I  000007 AREA      0006    000351 BESTVC
     0004    000107 BETTER    0004    000102 BIAS     0003 P  000027 CIN       0007 I  000002 CLSNAM    0005    000001 CONLV
     0004    000111 CORLEN    0003 P  000030 CTOT     0006    000053 DATAPE    0003 R  000041 DCON      0004    000053 DELT
     0003 P  000043 DISS      0006    000112 DIVSIZ   0006    000062 DUPSYM    0004    000073 DWFAC     0004    000061 ELIMTH
     0004    000052 EPS       0006    000217 FETVC2   0006    000161 FETVEC    0006    000071 FILESV    0003 R  000050 GEN
     0004    000074 GPACTM    0003 P  000047 GREF     0006    000000 HEAD      0006    000255 HISVEC    0007 I  000000 IBEGIN
     0006    000123 IBLOCK    0003 I  000023 IDADJ    0003 I  000024 INDEX     0004    000066 INDXVL    0000    000004 INJP$
     0006    000313 INVERT    0007 I  000004 IPT      0007 I  000122 IQ1       0003 I  000000 JUNK      0006    000113 KEEPLV
     0007 I  000112 KKT       0007 I  000113 KLBC     0004 I  000051 KROOT     0007 I  000121 LBUF      0003 I  000020 LINK
     0004 I  000003 LKURT     0007 I  000105 LNCAT    0004 I  000007 LOSUM     0004 I  000004 LOVAR     0004 I  000002 LR
     0004 I  000006 LSKEW     0003 I  000021 LSUPS    0004 I  000005 LSUM      0003 I  000022 LSUPER    0004 I  000005 LV
     0004    000002 LVRIN     0005 R  000012 MACCEL   0006    000052 MAPTAP    0007 I  000003 MAXPF      0006    000072 MAXCLS
     0006    000064 MAXDIV    0006    000055 MAXFET   0006    000074 MAXFLD    0006    000065 MINDIV    0004 I  000001 MM
     0004    000110 MODE      0004 I  000000 MQ       0004 I  000011 MXAR      0003    000014 NARL      0004 I  000010 NINCLS
     0006    000073 NOCLS2    0006    000121 NOCLS3   0006 I  000100 NOFEAT    0006    000101 NOFET2    0006    000102 NOFET4
     0007 I  000006 NOFLD     0006    000075 NOFLD2   0006    000076 NOFLD3    0006    000111 NOGRP     0006    000110 NOHIST
     0006    000107 NOSPLC    0006    000077 NOTRFD   0007 I  000120 NPTS      0004    000070 NPTSO     0003 I  000044 NSYMB
     0003 I  000020 NTB       0003    000016 NTBSZ4   0003    000015 NTOP      0003    000017 NWANT     0007 I  000117 NWERS
     0007 I  000005 NWDS      0003 R  000045 OCIN     0004    000055 ODCON     0003 R  000047 ODEN      0003 R  000046 OPRIOR
```

69 B-82

```
0003 R 000035 OPROP     0003 R 000036 OW        0005 R 000010 PACCEL    0006   000056 PAGSIZ    0003 R 000050 PCOND
0006   000122 PCTSZ     0003 R 000025 PCUM      0004   000103 PJOIN     0003 R 000043 PPASS     0003 R 000042 PORAT
0004   000071 PORATH    0003 R 000026 PRIRCM    0007 I 000106 PRNT      0003 R 000031 PROP      0007 I 000115 PROUT
0006   000114 PRTLEV    0007 I 000114 PRIME     0003 R 000044 PST       0006   000054 SAVTAP    0004   000065 SBLTH
0004   000062 SEPTH     0006   000067 SERIAL    0005   000002 SKBND     0005   000003 SKCHI     0006   000120 SPCBAS
0004   000112 SPCOR     0003 R 000032 SPFAC     0006   000066 SPLMAX    0004   000072 SPMVTH    0007 I 000010 SYM
0006   000057 TAPCHK    0006   000070 TAPESV    0006   000063 THRSYM    0007 I 000116 TOTPIX    0007 I 000001 TOTWRD
0005   000004 TRBND     0005   000005 TRCHI     0006   000060 TRNSYM    0006   000061 TSTSYM    0005   000006 URKBND
0005   000007 URKCHI    0005 R 000014 VACCEL    0006   000103 VARSIZ    0006   000104 VARSZ2    0006   000105 VARSZ4
0004   000063 VFAC      0003 R 000037 VOLIN     0004   000101 VOLLIM    0003 R 000040 VOLRT     0003 R 000050 WPIN
0004   000104 VRJOIN    0003 R 000034 W         0003 R 000033 WADJ      0004   000060 WADJIN    0005   000000 WAIT
0004   000106 WDELSM    0004   000067 WFAC      0004   000105 WSIM      0004   000050 WTINIT    0006   000116 XHGH
0006   000117 XLOW      0004   000056 XOVFLO    0006   000106 XSIZ      0004   000057 XUNFLO    0006   000115 YSIZ

00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00100      1*       C   MULTI
00101      2*             SUBROUTINE MULTI(PV)                                                      NEW000000
00103      3*             INCLUDE MISH,LIST                                                          -01000000
00103      3*       MISH  PROC                                                                         000000
00104      3*             PARAMETER NTBSIZ=32                                                          000000
00104      3*       C     NTBSIZ MUST BE A POWER OF TWO.  (NTB & NTBSIZ ARE USED IN THE                000000
00104      3*       C        STORAGE ALLOCATOR CALLOC, ENTERED BY GET AND FREE.)                       000000
00104      3*       C                                                                                  000000
00104      3*       C     THE FOLLOWING TWO PARAMETERS GIVE THE NUMBER OF                              000000
00104      3*       C        SYMMETRIC MATRICES AND VECTORS IN A CLUSTER NODE.                         000000
00105      3*             PARAMETER NARRS=3                                                            000000
00106      3*             PARAMETER NVECS=3                                                            000000
00107      3*             PARAMETER NSCALS=25                                                          000000
00110      3*             INTEGER JUNK(12), NTB(NTBSIZ), LINK(999)                                     000000
00111      3*             REAL W(1),PORAT(1),VOLIN(1),VOLRT(1),DCON(1),PPASS(1),PCOND(1)               000000
00112      3*             INTEGER LSUBS(1),IDADJ(1),INDEX(1),LSUPER(1),NSYMB(1)                        000000
00113      3*             REAL VRIN(475),GEN(999),GREF(999),ALINK(1)                                   000000
00114      3*             REAL PST(1),PCUM(1),DISS(1),WADJ(1),OPROP(1),OW(1),SPFAC(1)                  000000
00115      3*             REAL PRIRCM(1),OPRIOR(1),PROP(1),CIN(1),CTOT(1),OCIN(1),ODEN(1)              000000
00115      3*       C  THE FOLLOWING STRUCTURES THE CLUSTER NODES. 'GEN' IS                            000000
00115      3*       C     USED LATER AS THE BASE FOR ALL THE ARRAYS. (SEE CLINIT)                      000000
00115      3*       C        'GREF' IS USED LIKE 'GEN', BUT INTERNAL TO ROUTINES,                      000000
00115      3*       C          RATHER THAN IN CALLING SEQUENCES.                                       000000
00116      3*             EQUIVALENCE (LINK(31),LSUBS(30),LSUPER(29),IDADJ(28),NSYMB(11),              000000
00116      3*            1   INDEX(27),PCUM(26),PRIRCM(25),CIN(24),CTOT(23),                           000000
00116      3*            2   PROP(22),SPFAC(21),WADJ(20),W(19),OPROP(18),OW(17),                       000000
00116      3*            3   VOLIN(16),VOLRT(15),DCON(14),PORAT(13),DISS(12),PPASS(12),                000000
00116      3*            4   PST(11),OCIN(10),PCOND(7),VRIN(7),GEN(7),OPRIOR(9),ODEN(6),               000000
00116      3*            5   GREF(8),NTB(31))                                                          000000
00117      3*             EQUIVALENCE (LINK(1),ALINK(1))                                               000000
00120      3*             COMMON/CLUS/  JUNK,NAPL,NTOP,NTBSZM,NWANT,LINK                               000000
00120      3*       C  WE ASSIGN VARIOUS LOCATION PARAMETERS IN CLINIT.  THESE                         000000
00120      3*       C     ARE STORED MOSTLY IN THE ARRAYS LR FOR ARRAYS AND LV                         000000
00120      3*       C     FOR VECTORS.  THE FOLLOWING EQUIVALENCES ARE PROVIDED                        000000
00120      3*       C     FOR CONVENIENCE. (FOR EXAMPLE, GEN(LSUM+KL+I) IS THE                         000000
00120      3*       C     VECTOR SUM(KL+I) .)                                                          000000
00121      3*             INTEGER MXAR(31),LR(NARRS),LV(NVECS)                                         000000
00122      3*             EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT),                                     000000
00122      3*            1   (LR(3),LOVAR),(LV(1),LSUM),(LV(2),LSKEW),(LV(3),LOSUM)                    000000
```

```
00123     3*          COMMON /MISC/ MQ,MM,LR,LV,NINCLS,MXAB,LINIT,KROOT,EPS,DELT,
00123     3*         1   AMC,ODCON,XOVFLO,XUNFLO,LADJIN,ELIMTH,SEPTH,VFAC,AMM,SBLTH,
00123     3*         2    INDXVL,WFAC,NPTSO,PORATH,SPMVTH,DWFAC,GRACTM,AMOFAC,
00123     3*         3    AMOMIN,AMOMAX,AMORAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIM,WDELSM,
00123     3*         4   BETTER,MODE,CORLEN,SPCOR
00124     3*          REAL PACCEL(2),MACCEL(2),VACCEL(2)
00125     3*          COMMON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI,
00125     3*         1    PACCEL,MACCEL,VACCEL
00125     3*     C    MQMAX IS THE MAXIMUM NUMBER OF CHANNELS (30 FOR LARSYS)
00126     3*          PARAMETER MQMAX=4
00126     3*     C      MQTMAX=(MQMAX*(MQMAX+1))/2
00127     3*          PARAMETER MQTMAX=10
00130     3*          INTEGER GET
00130     3*      END MISH
00131     4*          INCLUDE COMBK1,LIST
00131     4*     COMBK1 PROC
00132     4*          COMMON /INFORM/HEAD(42), MAPTAP,     DATAPE,     SAVTAP,   MAXFET,
00132     4*         1                PAGSIZ, TAPCHK,     TRNSYM,     TSTSYM,
00132     4*         2                DUPSYM, THRSYM,     MAXDIV,     MINDIV,   SPLMAX,
00132     4*         3                SERIAL, TAPESV,     FILESV,
00132     4*         4                MAXCLS, NOCLS2,     MAXFLD,     NOFLD2,   NOFLD3,
00132     4*         5                NOTRFD, NOFEAT,     NOFET2,     NOFET4,   VARSIZ,
00132     4*         6                VARSZ2, VARSZ4,     XSIZ  ,     NOSPEC,   NOHIST,
00132     4*         *                NOGRP, DIVSIZ,   KEEPLV,   PRTLEV,   YSIZ  ,
00132     4*         8                XHGH  , XLOW  ,     SPCBAS,     NOCLS3,   POTSZ ,
00132     4*         9            ' IBLOCK(30), FETVEC(30), FETVC2(30), HISVEC(30),
00132     4*         *              INVERT(30), BESTVC(30)
00132     4*      END COMBK1
00133     5*          INCLUDE CMBK10,LIST
00133     5*     CMBK10 PROC
00134     5*          PARAMETER MAXPOP=61
00135     5*          INTEGER IBEGIN ,TOTWRD,CLSNAM,MAXBF,IPT,NWDS,NOFLD,AREA,
00135     5*         *SYM,LNCAT,PRNT,KKT,KLBC,PPTME,PROUT,NWDRS,NPTS
00136     5*          INTEGER TOTPIX,LBUF,IQ1
00137     5*          COMMON/CLUSTR/IBEGIN,TOTWRD,CLSNAM,MAXBF,IPT,NWDS,NOFLD,AREA
00137     5*         *      ,SYM(MAXPOP),LNCAT,PRNT(4),KKT,KLBC,PRTME,PROUT,
00137     5*         *TOTPIX,NWDRS,NPTS,LBUF,IQ1
00140     5*          DATA PRNT(4)/10/
00140     5*      END CMBK10
00142     6*          DIMENSION PV(NOFEAT,NPTS)                                  NEW00060
00143     7*          CALL DATFIX                                                -03000000
00144     8*          MQ=NOFEAT
00145     9*          MM=(MQ*(MQ+1))/2
00146    10*          CALL ALFREE(KLBC,NINCLS)
00147    11*          CALL CLINIT(KLBC)
00150    12*          KROOT=KLBC
00150    13*     C      PRINT 636,KLBC,NINCLS,LR,LV
00150    14*     C  636 FORMAT(' MULTI.I: KLBC,NINCLS',2I5,' LR,LV',3I3,I5,3I3)
00151    15*          CALL STATIS(KLBC,PV,GEN(LSUM),GEN(LSKEW),GEN(LKURT),GEN(LOSUM),
00151    16*         *  GEN(LOVAR))
00151    17*     C K: PRINT OUT ALL CLASSES.
00152    18*          PROUT=3
00153    19*          CALL CLDUMP(KLBC)
00154    20*          PROUT=1
00155    21*          PRINT 899
00157    22*      899 FORMAT(' EXIT FROM MULTI')
00160    23*          RETURN
```

B-34

ORIGINAL PAGE IS
OF POOR QUALITY

```
                                                                    000113
00161     24*           END

          END OF COMPILATION:          1 DIAGNOSTICS.
```

B-35 72

```
FTOP,CUW SEPER,SEPER
FOR SOE3-04/01777-19:44:46 (0,1)
-13,13
      LPS=LSUPER(KL)
      LBS=LSUPS(KL)
      PRINT 571,INDEX(KL),INDEX(LPS),INDEX(LBS),SPFAC(KL)

   SUBROUTINE SEPER     ENTRY POINT 000217

   STORAGE USED: CODE(1) 000242; DATA(0) 000046; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   CLUS    002017
    0004   MISC    000113
    0005   STPAR   000016

   EXTERNAL REFERENCES (BLOCK, NAME)

    0006   CLPR
    0007   DENCAL
    0010   FREE
    0011   NPRT$
    0012   NI01$
    0013   NI02$
    0014   SEER3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000067 10L      0001    000044 20L      0001    000062 29L      0000    000007 571F     0001    000144 614L
    0000    000022 664F     0001    000072 666L     0003 R  000020 ALINK    0004    000064 AMM      0004    000075 AMOFAC
    0004    000077 AMOMAX   0004    000076 AMOMIN   0004    000100 AMORAT   0004    000054 AMO      0004    000107 BETTER
    0004    000102 BIAS     0003 R  000027 CIN      0005    000001 CONLV    0004    000111 CORLEN   0003 P  000030 CTOT
    0003 R  000041 DCON     0004    000053 DELT     0003 R  000043 DISS     0004    000073 DWFAC    0004    000061 ELIMTH
    0004    000052 EPS      0003 R  000050 GEN      0004    000074 GRACTM   0003 R  000047 GREF     0003 I  000023 IDADJ
    0003 I  000024 INDEX    0004    000066 INDXVL   0000    000034 INJP$    0003 I  000000 JUNK     0000 I  000005 K
    0000 I  000006 KOLD     0004    000051 KROOT    0000 I  000002 KS       0000 I  000001 LBS      0003 I  000020 LINK
    0004 I  000003 LKURT    0004    000007 LOSUM    0004    000004 LOVAR    0000 I  000000 LPS      0004 I  000002 LR
    0004 I  000006 LSKEW    0003 I  000021 LSUPS    0004 I  000005 LSUM     0003 I  000022 LSUPER   0004 I  000005 LV
    0004    000002 LVRIN    0005 R  000012 MACCEL   0004    000001 MM       0004    000110 MODE     0004    000000 MQ
    0004 I  000011 MXAR     0003    000014 NARL     0004 I  000010 NINCLS   0004    000070 NPTSQ    0003 I  000043 NSYMB
    0003 I  000020 NTB      0003    000016 NTBSZ    0003    000015 NTOP     0003    000017 NWANT    0003 P  000045 OCIN
    0004    000055 OCON     0003 R  000047 ODEN     0003 R  000046 OPRIOR   0003 R  000035 OPROP   0003 R  000036 OW
    0005 R  000010 PACCEL   0000 P  000004 PASSF    0003 R  000050 PCOND    0003 R  000025 PCUM     0004    000103 PJOIN
    0003 P  000043 PPASS    0003 R  000042 PGRAT    0004    000071 PGRATH   0003 R  000026 PRIRCM   0003 R  000031 PROP
    0003 R  000044 PST      0000 R  000003 RATIO    0004    000065 SBLTH    0004    000062 SEPTH    0005    000002 SKBND
    0005    000003 SKCHI    0004    000112 SPCOR    0003 R  000032 SPFAC    0004    000072 SPMVTH   0005    000004 TRBND
    0005    000005 TRCHI    0005    000006 URKBND   0005    000007 URKCHI   0005 R  000014 VACCEL    0004    000063 VFAC
    0003 R  000037 VOLIN    0004    000101 VOLLIM   0003 R  000040 VOLRT    0003 R  000050 VRIN     0004    000104 VRJOIN
    0003 P  000034 W        0003 P  000033 WADJ     0004    000060 WADJIN   0005    000000 WAIT     0004    000106 WDELSM
    0004    000067 WFAC     0004    000105 WSIM     0004    000050 WTINIT   0004    000056 XOVFLO   0004    000057 XUNFLO
```

B-36 73

```
000000 *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
000101      1*            SUBROUTINE SEPER(KL)                                                    000000
000101      2*      C   THIS ROUTINE IS CALLED WHENEVER IT HAS BEEN DECIDED THAT A                000000
000101      3*      C     CLUSTER SHOULD BE SPLIT FOR GOOD.  THE CLUSTER HAS PREVIOUSLY           000000
000101      4*      C     BEEN SPLIT BY THE ROUTINE SPLIT, AND SUFFICIENT STATISTICS              000000
000101      5*      C     HAVE NOW BEEN GATHERED TO CONFIRM THAT THE CLUSTER CAN BE               000000
000101      6*      C     SPLIT UP ON A STATISTICALLY SIGNIFICANT BASIS.                          000000
000101      7*      C   THE ROUTINE TAKES THE CLUSTER AT KL, AND BRINGS UP ALL ITS               000000
000101      8*      C     DAUGHTER CLUSTERS TO THE SAME LEVEL AS KL ITSELF.  KL IS              000000
000101      9*      C     THEN ELIMINATED.                                                      000000
000103     10*            INCLUDE MISH                                                          000000
000103     11*      C     D       PARAMETER ITLM=40                                             000000
000103     12*      C     D       DIMENSION KTR(ITLM)                                           000000
000131     13*            LPS=LSUPER(KL)                                                  NEW000000
000132     14*            LBS=LSUBS(KL)                                                   NEW000002
000133     15*            PRINT 571,INDEX(KL),INDEX(LPS),INDEX(LBS),SPFAC(KL)             NEW000004
000141     16*        571 FORMAT('0###SEPERATE  ',I3,' SUPER,SUBS',2I3,'  SPFAC',E11.5)   -01000020
000141     17*      C     D       IT=0                                                          000020
000142     18*            KS=LSUPER(KL)                                                         000020
000143     19*            RATIO=PROP(KL)/PRIRCM(KL)                                             000022
000144     20*            PASSF=PST(KL)/(PCUM(KL)*PRIRCM(KL))                                   000025
000144     21*      C   FIRST WE FIND KL IN THE LIST OF LSUBS OF KS, AND REMOVE IT.             000025
000145     22*            K=LSUBS(KS)                                                           000033
000146     23*            IF(K.NE.KL) GO TO 20                                                  000035
000155     24*            LSUBS(KS)=LINK(KL)                                                    000037
000151     25*            GO TO 29                                                              000042
000152     26*        20  KOLD=K                                                               000044
000152     27*      C     D       KTR(IT)=K                                                     000044
000152     28*      C     D       IT=IT+1                                                       000044
000152     29*      C     D       IF(IT.GT.ITLM) GO TO 671                                      000044
000153     30*            K=LINK(K)                                                             000047
000154     31*            IF(K.LE.0) GO TO 666                                                  000051
000156     32*            IF(K.NE.KL) GO TO 20                                                  000053
000160     33*        25  LINK(KOLD)=LINK(KL)                                                  000056
000160     34*      C   PROCESS EACH SUBCLUSTER.                                              000056
000161     35*            29 K=LSUBS(KL)                                                        000062
000162     36*            KOLD=KL                                                               000064
000163     37*            10 CONTINUE                                                           000067
000164     38*            IF (K.GT.0) GO TO 614                                                 000067
000166     39*           666 PRINT 664,KL,K,KOLD                                                000072
000173     40*           664 FORMAT('0BAD SUBINDEX IN SEPER: KL,K, K OLD=',2I6,I12)             000103
000174     41*            CALL CLPP(KL,666,GEN(LSUM),GEN(LSKEW),GEN(LKURT))                     000103
000175     42*            CALL CLPR(KOLD,666,GEN(LSUM),GEN(LSKEW),GEN(LKURT))                   000120
000176     43*            RETURN                                                                000140
000177     44*           614 CONTINUE                                                           000144
000177     45*      C     D       KTR(IT)=-K                                                    000144
000177     46*      C     D       IT=IT+1                                                       000144
000177     47*      C     D       IF(IT.GT.ITLM) GO TO 671                                      000144
000200     48*            LSUPER(K)=KS                                                          000146
000201     49*            CALL DENCAL(K,RATIO,W(KL))                                            000150
000202     50*            PST(K)=PST(K)*PASSF                                                   000157
000203     51*            KOLD=K                                                                000163
000204     52*            K=LINK(K)                                                             000166
000205     53*            IF(K.NE.0) GO TO 10                                                   000170
000205     54*      C   NOW ADD THE SUBCLUSTER LIST OF KL TO THAT OF KS                         000170
000207     55*            LINK(KOLD)=LSUBS(KS)                                                  000172
```

```
00210        LSUBS(KS)=LSUBS(KL)                                                     000174
00211        CALL FREE(KL,NINCLS)                                                    000176
00211  C  D        PRINT 771,KL,LSUBS(KL),LINK(KL),LSUPER(KL),KS,LSUBS(KS),         000176
00211  C  D     1     LINK(KS),LSUPER(KS)                                            000176
00211  C  D  771 FORMAT(' LEAVING SEPER: KL,LSUBS,LINK,LSUPER',4I6/18X,'KS',4I9)     000176
00212        RETURN                                                                  000202
00212  C  D  671 PRINT 672,KL,KS,KOLD,KTR                                            000202
00212  C  D  672 FORMAT('0LOOP IN SEPER: KL,KS,KOLD,,KTR=',3I5/(3X10I6))              000202
00212  C  D        RETURN                                                           000202
00213        END                                                                     000241

END OF COMPILATION:          1 DIAGNOSTICS.
```

ORIGINAL PAGE IS OF POOR QUALITY

```
@FOR,SLW SETUP9,SETUP9
FOR SCE3-04/06/77-13:37:53 (1,2)
-19,19
  4              '!','/',';','/',1H',',',' ','0'/

     SUBROUTINE SETUP9     ENTRY POINT 000271

     STORAGE USED: CODE(1) 000275; DATA(0) 000157; BLANK COMMON(2) 000000

      COMMON BLOCKS:

      0003     INFORM 000407
      0004     SUPCOM 000103
      0005     CLUSTR 000123

     EXTERNAL REFERENCES (BLOCK, NAME)

      0006     NXTCHR
      0007     NUMBER
      0010     NWDU$
      0011     NI02$
      0012     NRDU$
      0013     NI03$
      0014     NERR2$
      0015     NERR3$

     STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0001     000014 10L       0001     000161 110L      0001     000172 130L      0001     000003 14GG      0001     000202 150L
      0001     000041 161G      0001     000212 170L      0001     000220 180L      0001     000236 190L      0001     000260 260L
      0001     000100 30L       0000     000116 480F      0000     000120 490F      0001     000126 50L       0000     000130 500F
      0000     000132 510F      0000     000134 550F      0000     000137 630F      0001     000137 70L       0001     000150 90L
      0005 I   000007 AREA      0003     000351 BESTVC    0000 I   000107 BLANK     0004 I   000005 CARD      0005 I   000002 CLSNAM
      0000 I   000112 CODE      0004     000003 CODE1     0004     000004 CODE2     0004 I   000002 COL       0003 I   000037 COMENT
      0003     000053 DATAPE    0003 I   000016 DATE      0003     000112 DIVSIZ    0003     000062 DUPSYM    0003     000213 FETVC2
      0003 I   000161 FETVEC    0003     000071 FILESV    0003     000000 HEAD      0003 I   000002 HED1      0003 I   000023 HED2
      0003     000255 HISVEC    0000 I   000111 I         0005 I   000000 IBEGIN    0003     000123 IBLOCK    0000 I   000110 ICNT
      0000     000151 INJPS     0004     000000 INTAPE    0000 I   000075 INVEC     0003     000313 INVFRT    0005 I   000004 IPT
      0005 I   000122 IOI       0000 I   000113 J         0003     000113 KEEPLV    0005 I   000112 KKT       0005 I   000113 KLBC
      0005 I   000121 LPUF      0005 I   000105 LNCAT     0003 I   000115 M         0003     000052 MAPTAP    0005 I   000003 MAXBF
      0003     000072 MAXCLS    0003     000064 MAXDIV    0003     000055 MAXFET    0003     000074 MAXFLD    0003     000065 MINDIV
      0003     000073 NOCLS2    0003     000121 NOCLS3    0003 I   000100 NOFEAT    0003     000101 NOFET2    0003     000102 NOFET4
      0005 I   000006 NOFLD     0003     000075 NOFLD2    0003     000076 NOFLD3    0003     000111 NOGRP     0003     000110 NOHIST
      0003     000107 NOSPLC    0003     000077 NOTPFD    0000 I   000114 NPOS      0005 I   000120 NPTS      0007 I   000000 NUMBER
      0005 I   000117 NWORS     0005 I   000005 NWDS      0006 I   000000 NXTCHR    0003     000056 PAGSIZ    0003     000122 PCTSZ
      0005 I   000106 PRNT      0005 I   000115 PROUT     0003     000114 PRTLEV    0005 I   000114 PRTPE     0003     000054 SAVTAP
      0003     000067 SERIAL    0000 T   000000 SMBLS     0003     000120 SPCBAS    0003     000066 SPLMAX    0004     000001 STATUS
      0005 I   000010 SYM       0003     000057 TAPCHK    0003     000070 TAPESV    0003     000063 THRSYM    0005 I   000116 TOTPIX
      0005 I   000001 TOTWRD    0003     000060 TRNSYM    0003     000061 TSTSYM    0003 I   000103 VARSIZ    0003     000104 VARSZ2
      0003     000105 VARSZ4    0003     000116 XHGH      0003     000117 XLOW      0003     000106 XSIZ      0003     000115 YSIZ
```

~~B-39~~ 76

```
00101        1*           SUBROUTINE SETUP9                                                      000000
00101        2*     C******************************************************************         000000
00101        3*     C*                                                                          000000
00101        4*     C*    THE PURPOSE OF SUBROUTINE SETUP9 IS TO READ AND ANALYZE ALL CARD      000000
00101        5*     C*    INPUT TO THE PROGRAM                                                  000000
00101        6*     C*                                                                          000000
00101        7*     C******************************************************************         000000
00101        8*     C*                                                                          000000
00103        9*           IMPLICIT INTEGER (A-X)                                                000000
00104       10*           INCLUDE COMBK1,LIST                                                   000000
00104       10*     COMBK1 PROC                                                                 000000
00105       10*           COMMON /INFORM/HEAD(42), MAPTAP,     DATAPE,     SAVTAP,   MAXFET,    000000
00105       10*          1              PAGSIZ, TAPCHK,     TRNSYM,     TSTSYM,                 000000
00105       10*          2              DUPSYM, THRSYM,     MAXDIV,     MINDIV,   SPLMAX,       000000
00105       10*          3              SERIAL, TAPESV,     FILESV,                             000000
00105       10*          4              MAXCLS, NOCLS2,     MAXFLD,     NOFLD2,   NOFLD3,       000000
00105       10*          5              NOTRFD, NOFEAT,     NOFET2,     NOFET4,   VARSIZ,       000000
00105       10*          6              VARSZ2, VARSZ4,     XSIZ  ,     NOSPEC,   NOHIST,       000000
00105       10*          *              NOGRP, DIVSIZ,    KEEPLV,    PRILEV,    YSIZ   ,        000000
00105       10*          8              XHGH   , XLOW   ,   SPCBAS,     NOCLS3,   PCTSZ ,       000000
00105       10*          9        IBLOCK(30), FETVEC(30), FETVC2(30), HISVEC(30),               000000
00105       10*          *        INVERT(30), BESTVC(30)                                        000000
00105       10*      END COMBK1                                                                 000000
00106       11*           INCLUDE COMBK4,LIST                                                   000000
00106       11*     COMBK4 PROC                                                                 000000
00107       11*           DIMENSION  HED1(10),  HED2(10),  DATE(2),  COMENT(10)                 000000
00110       11*           EQUIVALENCE  (HED1(1),HEAD(3)),     (DATE(1),  HEAD(15)),             000000
00110       11*          *             (HED2(1),HEAD(20)),    (COMENT(1),HEAD(32))              000000
00110       11*      END COMBK4                                                                 000000
00111       12*           INCLUDE COMBK8,LIST                                                   000000
00111       12*     COMBK8 PROC                                                                 000000
00112       12*           COMMON /SUPCOM/ INTAPE,STATUS,COL,CODE1,CODE2,CARD(62)                000000
00112       12*      END COMBK8                                                                 000000
00113       13*           INCLUDE CMBK10,LIST                                                   000000
00113       13*     CMBK10 PROC                                                                 000000
00114       13*           PARAMETER MAXPOP=61                                                   000000
00115       13*           INTEGER IBEGIN ,TOTWRD,CLSNAM,MAXBF,IPT,NWDS,NOFLD,AREA,              000000
00115       13*          *SYM,LNCAT,PRNT,KKT,KLBC,PRIME,PROUT,VWDRS,NPTS                        000000
00116       13*           INTEGER TOTPIX,IBUF,IO1                                               000000
00117       13*           COMMON/CLUSTR/IBEGIN,TOTWRD,CLSNAM,MAXBF,IPT,NWDS,NOFLD,AREA          000000
00117       13*          *     ,SYM(MAXPOP),LNCAT,PRNT(4),KKT,KLBC,PRIME,PROUT,                 000000
00117       13*          *TOTPIX,NWDRS,NPTS,LBUF,IO1                                            000000
00120       13*           DATA PRNT(4)/10/                                                      000000
00120       13*      END CMBK10                                                                 000000
00122       14*           DIMENSION SMBLS(MAXPOP)                                               000000
00123       15*           DATA  SMBLS/'1','2','3','4','5','6','7','8','9','A','B','C','D',      000000
00123       16*          *            'E','F','G','H','I','J','K','L','M','N','O','P','Q',      000000
00123       17*          2            'R','S','T','U','V','W','X','Y','Z','$','#','&','/',      000000
00123       18*          3            '-','*','.','$','@','=','£','?','>','<',')','(',':',      000000
00123       19*          4            '!','/',';','/',1H','','','0'/                            NEWC00000
00123    *DIAGNOSTIC*  A LIST IN THE ABOVE STATEMENT IS TOO LONG.
00125       20*           DATA PRNT/ 2,1,3/                                                     -01000000
00125    *DIAGNOSTIC*  A LIST IN THE ABOVE STATEMENT IS TOO LONG.
00127       21*           PARAMETER CNUM=10                                                     000000
00130       22*           DIMENSION INVEC(CNUM)                                                 000000
00131       23*           DATA INVEC/'CHANNE','HED1','HED2','DATE','COMMEN','NPTS','NPOS' ,     000000
```

B-40 77

```
00131     24*          * 'SYMBOL','PRINT','*END*'/                                    000300
00133     25*           DATA BLANK/1H /,NPTS/10/                                      000003
00136     26*           ICNT=0                                                        000000
00137     27*           DO 5 I=1,MAXPOP                                               000003
00142     28*         5 SYM(I)=SMBLS(I)                                               000003
00144     29*           NOFEAT=0                                                      000005
00145     30*           WRITE(6,630)                                                  000006
00147     31*        10 READ(5,480)CODE,CARD                                          000014
00153     32*           WRITE(6,550)CODE,CARD                                         000024
00157     33*           COL=0                                                         000035
00160     34*           DO 20 I=1,CNUM                                                000041
00163     35*        20 IF(CODE .EQ. INVEC(I))GO TO(30,50,70,90,110,130,150,170,190,  000041
          36*          * 260),I                                                      000041
00166     37*           WRITE(6,490)CODE,CARD                                         000065
00172     38*           GO TO 10                                                      000076
00172     39*     C*                                                                  000076
00172     40*     C     CHANNEL CARD                                                  000076
00172     41*     C*                                                                  000076
00173     42*        30 J = NXTCHR(CARD,COL)                                          000100
00174     43*           IF (J .EQ. BLANK)GO TO 10                                     000104
00176     44*           COL=COL-1                                                     000106
00177     45*           NOFEAT = NUMBER(CARD,COL,FETVEC,NOFEAT)                       000111
00200     46*           VARSIZ=(NOFEAT*(NOFEAT+1))/2                                  000120
00201     47*           GO TO 10                                                      000124
00201     48*     C*                                                                  000124
00201     49*     C*    HED1 CARD                                                     000124
00201     50*     C*                                                                  000124
00202     51*        50 READ (30,500)HED1                                             000126
00205     52*           GO TO 10                                                      000135
00205     53*     C*                                                                  000135
00205     54*     C*    HED2 CARD                                                     000135
00205     55*     C*                                                                  000135
00206     56*        70 READ (30,500) HED2                                            000137
00211     57*           GO TO 10                                                      000146
00211     58*     C*                                                                  000146
00211     59*     C*    DATE CARD                                                     000146
00211     60*     C*                                                                  000146
00212     61*        90 READ(30,510)DATE                                              000150
00215     62*           GO TO 10                                                      000157
00215     63*     C*                                                                  000157
00215     64*     C*    COMMENT CARD                                                  000157
00215     65*     C*                                                                  000157
00216     66*       110 READ(30,500)COMENT                                            000161
00221     67*           GO TO 10                                                      000170
00221     68*     C*                                                                  000170
00221     69*     C*    NPTS CARD,NUMBER OF DATA POINTS FOR EACH CHANNEL RETURNED TO  000170
00221     70*     C*    CLASY3 EACH CALL TO CLASY2                                    000170
00221     71*     C*                                                                  000170
00222     72*       130 J=NUMBER(CARD,COL,NPTS,0)                                     000172
00223     73*           GO TO 10                                                      000200
00223     74*     C*                                                                  000200
00223     75*     C*    NPOS CARD,NUMBER OF DRUM POSITIONS FROM WHICH TO OBTAIN DATA FOR  000200
00223     76*     C*    CLASY3,SO THAT THE DATA WILL BE SCRAMBLED                     000200
00223     77*     C*                                                                  000200
00224     78*       150 J=NUMBER(CARD,COL,NPOS,0)                                     000202
00225     79*           GO TO 10                                                      000210
00225     80*     C*                                                                  00210
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00225      61*      C*     SYMBOL CARD                                                    000210
00225      62*      C*                                                                    000210
00226      83*        170 ICNT=ICNT+1                                                     000212
00227      84*            IF(ICNT .GT. MAXPOP) GO TO 10                                   000214
00231      85*        180 M=NXTCHR(CARD,COL)                                              000220
00232      86*            IF( M .EQ. BLANK)GO TO 10                                       000224
00234      87*            IF(M .EQ. ',')GO TO 180                                         000227
00236      88*            SYM(ICNT)=M                                                     000232
00237      89*            GO TO 170                                                       000234
00237      90*      C*                                                                    000234
00237      91*      C      PRINT OPTION CARD                                              000234
00237      92*      C*                                                                    000234
00240      93*        190 J=NXTCHR(CARD,COL)                                              000236
00241      94*            IF (J .EQ. BLANK)GO TO 10                                       000242
00243      95*            COL=COL-1                                                       000244
00244      96*            J=NUMBER(CARD,COL,PRNT,0)                                       000247
00245      97*            GO TO 10                                                        000256
00245      98*      C*                                                                    000256
00245      99*      C*     *END* CARD                                                     000256
00245     100*      C*                                                                    000256
00246     101*        260 RETURN                                                          000260
00247     102*        480 FORMAT(A6,4X,62A1)                                              000274
00250     103*        490 FORMAT(' INVALID INPUT CARD--IGNORED'/15,A6,4X,62A1)            000274
00251     104*        500 FORMAT(10X,10A6)                                                000274
00252     105*        510 FORMAT(10X,2A6)                                                 000274
00253     106*        550 FORMAT(5X,A6,4X,62A1)                                           000274
00254     107*        630 FORMAT(//' INPUT SUMMARY'//)                                    000274
00255     108*            END                                                             000274

      END OF COMPILATION:          2 DIAGNOSTICS.
```

B-42 79

```
@FOR,S  STATIS,STATIS
FOR S1E3-04/21/77-21:10:17 (L,)

SUBROUTINE STATIS     ENTRY POINT 001152

STORAGE USED: CODE(1) 001205; DATA(0) 000260; BLANK COMMON(2) 000000

 COMMON BLOCKS:

  0003    CLUS     002217
  0004    MISC     000113
  0005    STPAR    000015
  0006    CLUSTR   000123
  0007    RAND     000003

EXTERNAL REFERENCES (BLOCK, NAME)

  0010    DISC
  0011    CLASY2
  0012    CORECT
  0013    DOTSC
  0014    VPV
  0015    VMTV
  0016    MPVS
  0017    ADJUST
  0020    CODUMP
  0021    NWDU$
  0022    NI02$
  0023    EXP
  0024    NPRT$
  0025    NI01$
  0026    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

  0000     000126 1F        0001     000064 130L      0001     000076 131L      0001     000450 132L      0001     000135 133L
  0001     000162 134L      0001     000222 138L      0001     000420 140L      0001     000447 143L      0001     000472 144L
  0001     000320 151L      0001     000354 153L      0001     000020 156G      0001     000751 163L      0001     000772 164L
  0001     000642 189L      0001     000671 190L      0001     000711 191L      0001     000664 192L      0001     000245 231L
  0001     000256 232L      0001     000272 239L      0001     000343 263G      0001     001042 299L      0001     001062 304L
  0001     001127 309L      0001     001127 399L      0001     000310 52L       0001     000211 531L      0000     000142 555F
  0001     000610 611L      0000     000076 647F      0000     000162 672F      0003 R   000023 ALINK     0000 R   000117 ALOW
  0004 R   000125 ALPHA     0004     000064 AMM       0004     000075 AMOFAC    0004     000077 AMOMAX    0004     000076 AMCMIN
  0004 R   000075 AMONTE    0004     000100 AMORAT    0004 R   000054 AMC       0006 I   000007 AREA      0004     000107 BETTER
  0004     000102 BIAS      0000 R   000100 BMU       0003 R   000027 CIN       0006 I   000002 CLSNAM    0003 R   000125 COFI
  0005     000001 CONLV     0004 R   000111 CORLEN    0000 R   000036 COVEC     0003 R   000030 CTOT      0003 R   000041 DCON
  0004     000053 DELT      0010     000000 DISC      0003 R   000043 DISS      0012 R   000000 DOTS$     0004     000073 DWFAC
  0004     000061 ELIMTH    0004     000052 EPS       0003 R   000050 GEN       0004     000074 GRACIM    0003 R   000047 GREF
  0005 I   000000 IBEGIN    0003 I   000023 IDADJ     0000 I   000102 I0        0003 I   000024 INDEX     0004     000066 INDXVL
  0000     000230 INJP$     0006 I   000004 IPT       0006 I   000122 IO1       0000 I   000104 ISEC      0003 I   000000 JUNK
  0003 I   000113 KADJ      0000 I   000105 KFATH     0006 I   000112 KKT       0000 I   000103 KL        0006 I   000113 KLBC
  0000 I   000112 KPR       0004     000051 KROOT     0000 I   000077 KROT      0006 I   000101 LBUF      0003 I   000020 LINK
  0004     000003 LKURT     0006 I   000105 LNCAT     0004     000007 LOSUM     0004     000004 LOVAR     0004 I   000002 LR
```

B-43 80

```
0004     000016 LSKEW     0007 I 000021 LSUBS     0004     000005 LSUM      0003 I 000022 LSUPER    0004 I 000005 LV
0004     000002 LVRIN     0005 R 000016 MACELL    0006 I 000003 MAXBF     0004     000001 MP        0004 I 000110 MODE
0000 I 000074 MONTE       0004 I 000000 MQ        0004 I 000011 MXAR      0003     000014 NARL      0003 I 000101 NDO
0004     000010 NINCLS    0006 I 000006 NOFLD     0006 I 000125 NPTS      0004 I 000070 NPTSU     0003 I 000043 NSYMB
0003 I 000020 NTS         0007     000026 NTESZM    0003     000015 NTOP      0003     000017 NWANT     0005 I 000117 NWORS
0006 I 000065 NWDS        0007     000000 NX        0007 I 000001 NXA       0007 I 000002 NXO       0003 R 000045 OCIN
0004     000055 ODCON     0007 R 000047 ODEN      0003 R 000046 OPRIOR    0003 R 000035 OPROF     0002 R 000036 OW
0000 R 000115 P           0005 R 000010 PACCLL    0003 R 000050 PCOND     0003 R 000025 PCUM      0004     000153 PJCIN
0000 R 000076 PLIM        0003 R 000043 PPASS     0000 R 000114 PPASSK    0003 R 000042 PLRAT     0004     000071 PORATH
0003 R 000026 PPIPCM      0006 I 000106 PPNT      0003 R 000031 PROP      0000 R 000124 PROFL     0006 I 000115 PROUT
0000 I 000130 PPTME       0003 R 000044 PST       0000 R 000000 REL       0004     000065 SPLTH     0004     000062 SEPTH
0005     000002 SKRND     0005     000003 SKCHI     0004 R 000112 SPCOR     0003 R 000032 SPFAC     0004     000072 SPMVTH
0000 R 000110 SPUSE       0006 I 000010 SYM       0006 I 000116 TOTPIX    0006 I 000001 TOTWRD    0005     000004 TRBND
0005     000005 TRCHI     0005     000006 UPKBND    0005     000007 URKCHI    0005 R 000014 VACCEL    0004     000063 VFAC
0003 R 000007 VOLIM       0004     000101 VOLLIM    0003 R 000040 VOLRT     0003 R 000050 VRIN      0004     000104 VRJOIN
0003 R 000034 W           0003 R 000033 WADJ      0004     000060 WADJIN    0005     000000 WAIT      0004     000106 WDELSM
0000 R 000107 WDISS       0004     000067 WFAC      0000 R 000116 WO        0004     000105 WSIM      0004     000050 WTINIT
0000 R 000156 WUSE        0004 R 000056 XOVFLO    0004 R 000057 XUNFLO    0000 R 000123 ZFAC      0000 R 000121 ZQ
0000 R 000122 ZOS         0000 R 000111 ZZ
```

```
*DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
      SUBROUTINE STATIS(KROTIN,PV,SUM,SKEW,KURT,OSUM,OVAR)
C  THIS PROGRAM TAKES EACH INPUT POINT AND CLASSIFIES IT
C     (ON A FRACTIONAL, PROBABILISTIC BASIS).  IT THEN
C     UPDATES THE VARIOUS STATISTICAL PARAMETERS ASSOCIATED
C     WITH THE CLASSES INDICATED AND CHECKS TO SEE IF
C     ANY OF THESE CLASSES IS POTENTIALLY TWO.  THOSE WHICH
C     ARE ARE REFERRED TO THE ROUTINE 'SPLIT'.
      INCLUDE HISH
      INCLUDE CMBKIC
      COMMON /RAND/NX,NXA,NXO
CHANGE***
      REAL SUM(1),SKEW(1),KURT(1),OSUM(1),OVAR(1)
      DIMENSION REL(30),PV(MQ,NPTS),COVEC(30)
      DATA MONTE,AMONTE,PLIM/3,3.,.1/
C   D     PARAMETER ITLIM=71
C   D     INTEGER KTR(ITLIM)
C   D     COMMON/KTRCOM/IT,KTR
      INTEGER GTSC
      XF(DIST)=EXP(-.5*DIST)
C      HERE ABOVE GDET IS THE SQUARE ROOT OF THE COVARIANCE
C        MATRIX, AND FFAC IS A POWER OF PI.
      KROT=KROTIN
C   D     CALL CLPR(LSUPS(KROT),1,SUM,SKEW,KURT)
      VPO=.666666666667*AMO
C     CALL CLASY2 AND GET NDO  VECTORS,MQ IN LENGTH
   50 CONTINUE
      CALL CLASY2(PV,NDO)
      DO 399 IDO=1,NDO
C  ** THIS CODE GETS RANDOM NUMBERS. **
C  GET NEXT POINT IN SEQUENCE
C  INSPECT EACH CLASS AND INSERT POINT
C    WE USE MONTE-CARLO TECHNIQUES FOR LOW PROBABILITY CLASSES(P<PLIM)
      PCUM(KROT)=0.
```

```
      IF(INDEX(KL).GT.3.AND.KL.NE.119)WRITE(6,1)IO,INDEX(KL),KL
1     FORMAT(1H0,'***WARNING ON THE ',I3,' TIME INDEX(KL)=',I3,' KL=',I3
     *)
      PPIRCM(KROT)=0.
      PPASS(KROT)=1.
      ISEQ=0
C     D       II=1
      KL=LSUBS(KROT)
      KFATH=KROT
C     GO DOWN CLUSTER TREE
  130 IF(LSUBS(KL).EQ.0) GO TO 131
      PCUM(KL)=0.
      PPIRCM(KL)=0.
      KFATH=KL
      KL=LSUBS(KL)
      GO TO 130
CHANGE***      4
  131 IF(INDEX(KL).LE.0) GO TO 133
      CALL COPECT(REL,PV(1,IDO),W(KL),SUM(KL+1))
      WUSE=W(KL)
C     D       KTR(II)=KL
C     D       II=II+1
C     D       IF(II.GT.ITLIM) GO TO 150
C     D       PRINT 639,KL,LSUBS(KROT),W
      PROP(KL)=CIN(KL)/(W(KFATH)-CTOT(KL))
      GO TO 134
CHANGE***      4.5
  133 CALL COPECT(REL,PV(1,IDO),OW(KL),CSUM(KL+1))
      WUSE=OW(KL)
  134 DISS(KL)=DCTSQ(REL,VRIN(KL+1))*WUSE
C     D       PRINT 530,IDO,NDO,DISS(KL),(PV(II),II=1,MO)
C     D   530 FORMAT(' I,N,DISS,PV',2I4,E11.6,1X,5F8.3/(32X,5F8.3))
      WDISS=DISS(KL)*W(KL)/(W(KL)-CORLEN*(2.+DISS(KL)))*DCON(KL)
      IF(ABS(WDISS).LT. 100.) GO TO 531
C     D       IF(KL.EQ.LSUBS(KROT)) PRINT 631,W(KL),DISS(KL),(REL(II),II=1,MO)
C     D   631 FORMAT(' STATIS: WILD PT. W',F9.2,'  DISS',E10.4/7X,'REL',
C     D    1    5F10.4/(10X,5F10.4))
      PCOND(KL)=0.
      GO TO 138
  531 CONTINUE
      PCOND(KL)=XP(WDISS)/VOLRT(KL)
  138 IF(LSUBS(KL).NE.0) PCUM(KL)=PCUM(KL)/PRIRCM(KL)
      SPUSE=SPFAC(KL)/SPCOP
      IF(SPUSE.GT.XUNFLO) GO TO 231
      PST(KL)=PROP(KL)*PCOND(KL)
      GO TO 239
  231 IF(SPUSE.LT.XOVFLO) GO TO 232
      PST(KL)=PROP(KL)*PCUM(KL)
      GO TO 239
  232 ZZ=EXP(SPUSE)
      PST(KL)=PROP(KL)*(PCOND(KL)+ZZ*PCUM(KL))/(1.+ZZ)
C     D       IF(IDO.GT.5000.AND.IDO.LT.5100.OR.
C     D    1    PST(KL).LT.0..OR.PCUM(KFATH).LT.0.) PRINT 753,P,PST(KL),
C     D    1    PCUM(KFATH),KL,KFATH,IDO
  239 PCUM(KFATH)=PCUM(KFATH)+PST(KL)
      PPIRCM(KFATH)=PPIRCM(KFATH)+PROP(KL)
  139 KL=LINK(KL)
```

```
C     D       PRINT 639,KL,LSUBS(KROT),K
C     D   639 FORMAT(' KL,LSUBS(KROT),K=',2I9,2I4)
        IF(KL)130,149,132
C   GO UP TREE
  149 KL=KFATH
        KFATH=LSUPER(KL)
        IF(KL.NE.KROT) GO TO 131
C  WE NOW HAVE THE RELEVANT CLASSES AND THEIR PROBABILITIES AVAILABLE.
C
C  NEXT WE MAKE THE APPROPRIATE INDIVIDUAL FIRST-ORDER STATISTICS ADJ.
  150 CONTINUE
C     D       ISUBC=IT
        PCUM(KROT)=PCUM(KROT)/PRIRCM(KROT)
        IF(PCUM(KROT).NE.0.) GO TO 151
CHANGE***     5
        PRINT 555,IDO,W(KROT),(PV(KPR,IDO),KPR=1,MQ)
  555 FORMAT('0**SUSPECTED BAD DATA POINT--STATIS**IDO=',I5,'   ROOT',
     1   F10.2,5X,'VECTOR',(5E12.6))
        GO TO 399
  151 CONTINUE
        KL=LSUBS(KROT)
        KFATH=KROT
        W(KROT)=W(KROT)*PPASS(KROT)
        NPTS0=NPTS0+1
        KADJ=1
  153 CONTINUE
C     D       KTK(IT)=KL
C     D       IT=IT+2
C     D       IF(IT.GT.ITLIM) GO TO 399
C     D       PRINT 639,KL,LSUBS(KROT),K
        IF(PST(KL).EQ.0.) GO TO 299
        PPASSK=PPASS(KFATH)
        P=PST(KL)/(PCUM(KFATH)*PRIRCM(KFATH))*PPASSK
C     D       PRINT 753,P,PST(KL),PCUM(KFATH),KL,KFATH,IDO
C     D   753 FORMAT(' P,PST,PTOT,KL,KFATH,IDO=',2E13.6,E13.6,2I5,I6)
C       KLO=KL
  132 IF(P.GE.PLIM)GO TO 140
        IF(DISC(MONTE).NE.1)GO TO 299
        PPASSK=PPASSK*AMONTE
        P=P*AMONTE
        GO TO 132
CHANGE***     5
  140 IF(INDEX(KL).LE.0) GO TO 143
        CALL CORECT(REL,PV(1,IDO),W(KL),SUM(KL+1))
        GO TO 144
CHANGE***     5
  143 CALL CORECT(REL,PV(1,IDO),OW(KL),OSUM(KL+1))
  144 WO=W(KL)
        IF(P.GT.1.01.OR.P.LT.0.)PRINT 672,INDEX(KL),KL,INDEX(KFATH),
     1  KFATH,IDO,P,PST(KL),PCUM(KFATH),PRIRCM(KFATH),PPASSK,
     2  PROP(KL)
  672 FORMAT(' PROB ERROR(STATIS):',2(I3,I7),I6,' P=',E9.4,
     1    2X,'FROM',7E9.4)
        IF (P.GT.1.1) P=.01
        W(KL)=W(KL)*P
        ALOW=P/W(KL)
        ALPHA=WO*ALOW
```

B-46

ORIGINAL PAGE IS OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      C        COVARIANCE, WHEREAS WE SUBSTITUTE THE CURRENT                  CDC772
      C        VALUES INSTEAD.  THUS THE VALUES CALCULATED DEPEND             CDC772
      C        ON THE ORDER THE POINTS ARE READ IN.  THIS IS NOT              CDC772
      C          CRITICAL.                                                    CDC772
      C                                                                       CDC772
           WDISS=DISS(KL)*P                                                   CDC776
           CALL VPV(SKEW(KL+1),WDISS,REL)                                     CDC1C1
           CALL MPVS(KURT(KL+1),WDISS,REL)                                    CDC1C8
      C                                                                       CDC1C8
      C  WE NOW ADJUST THE CLASS FOR LARGE-SCALE STATISTICAL EFFECTS,         CDC1C8
      C    ON AN OCCASIONAL BASIS.    THIS INCLUDES NOMINAL NEWTONS           CDC1C8
      C    METHOD CORRECTIONS AND TESTING FOR THE POSSIBILITY                 CDC1C8
      C    OF TWO CLUSTERS (USING THE SKEW AND KURT STATISTICS).              CDC1C8
           IF(W(KL).GT.WADJ(KL).OR.NPTS0.GE.IDADJ(KL)) KADJ=KL                CDC1C8
      C    D     IF(IDO.EQ.NDO) CALL ADJUST(KL,SUM,SKEW,KURT,GEN(LOSUM),      CDC1D0
      C    D    1     GEN(LOVAR))                                             CDC1D0
      C    D     IF (IDO.LE.5.OR.IDO.EQ.NDO) CALL MCHECK(KL)                  CDC1D0
         299 PPASS(KL)=P                                                      CDC1E2
           IF(LSUBS(KL).EQ.0.OR.PCUM(KL).EQ.0.) GO TO 304                     CDC1E4
           KPATH=KL                                                           CDC1E8
           KL=LSUBS(KL)                                                       CDC1E8
           GO TO 153                                                          CDC1E8
      C    304 KL=LINK(KL)                                                    CDC1E8
         303 IF(KL) 152,305,153                                               CDC1E8
         305 KL=KPATH                                                         CDC1E8
           KPATH=LSUPER(KL)                                                   CDC1E8
      C    D     IF(KPATH.EQ.0) GO TO 646                                     CDC1E8
           IF(KL.NE.KROT) GO TO 304                                           CDC1E8
           IF(KADJ.NE.0) CALL ADJUST(KADJ,SUM,SKEW,KURT,OSUM,OVAR)           CDC1E8
           IF(MOD(NPTS0,10*PIX).NE.0.OR.MODE.EQ.0) GO TO 309                  CDC1E8
           NXA=NXO                                                            CDC1E8
           CALL CLDAMP(KROT)                                                  CDC1E8
         309 CONTINUE                                                         CDC1E8
      C    D     IF(IT.LT.ITLIM) GO TO 399                                    CDC1E8
      C    D   646 PRINT 647,IDO,W(KROT),KL,ISEC,(KTR(I),I=1,IT)             CDC1E8
         647 FORMAT('0LOOP IN STATIS:IDO,W(KROT),KL,SECTION',15,E11.5,215     CDC1E8
           1     /(1X,10I5))                                                  CDC1E8
      C    D     IF(IDO.GT.5.AND.IDO.LT.NDO-5.AND.MOD(IDO,7).NE.3)GO TO 399   CDC1E8
      C    D     CALL MCHECK(KLO)                                             CDC1E8
      C    D     CALL CLPR(KLO,1,SUM,SKEW,KURT)                               CDC1E8
         399 CONTINUE                                                         CDC1E8
      CHANGE***     6                                                        CDC1E8
           IF(KKT.NE.10) GO TO 50                                             CDC1E8
           RETURN                                                             CDC1E8
           END                                                                CDC1E8

END OF COMPILATION:          1 DIAGNOSTICS.
```

```
$FOR,SOL TR,TR
FOR SOE3-04/01/77-19:45:04 (0,1)
-12,17
      MXA=MXAR(I)
 15   TR=TR-AM(MXA + I)*AMET(MXA+I)

 FUNCTION TR           ENTRY POINT 000105

 STORAGE USED: CODE(1) 000124; DATA(0) 000035; BLANK COMMON(2) 000000

  COMMON BLOCKS:

  0003   CLUS   002017
  0004   MISC   000113
  0005   STPAR  000016

 EXTERNAL REFERENCES (BLOCK, NAME)

  0006   NEPR3$

 STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

  0001   000035 1340      0001   000046 1420      0003 R 000020 ALINK     0004   000064 AMM      0004   000075 AMOFAC
  0004   000077 AMOMAX    0004   000076 AMOMIN    0004   000100 AMORAT    0004   000054 AMQ      0004   000107 BETTER
  0004   000102 BIAS      0003 R 000027 CIN       0005   000001 CONLV     0004   000111 CORLEN   0003 R 000030 CTOT
  0003 R 000041 DCON      0004   000053 DELT      0003 R 000043 DISS      0004   000073 EMFAC    0004   000061 ELIMTH
  0004   000052 EPS       0003 R 000050 GEN       0004   000074 GRACTM    0003 R 000047 GREF     0004 I 000001 I
  0003 I 000023 IDADJ     0003 I 000024 INDEX     0004   000066 INDXVL    0000   000035 INOPS    0003 I 000020 JUNK
  0004   000051 KPOOT     0003 I 000025 LINK      0004   000003 LKURT     0004   000007 LOSUM    0004   000004 LOVAR
  0004 I 000002 LP        0004   000006 LSKEW     0003 I 000021 LSUBS     0004   000005 LSUM     0003 I 000022 LSUPER
  0004 I 000005 LV        0004   000002 LVRIN     0005 R 000012 MACCEL    0004 I 000001 MM       0004   000110 MODE
  0004 I 000000 M9        0000 I 000002 MXA       0004 I 000011 MXAR      0003   000014 NARL     0004   000010 NINCLS
  0004   000070 NPTSO     0003 I 000043 NSYMB     0003 I 000020 NTB       0003   000016 NTBSZM   0003   000015 NTOP
  0003   000017 NWANT     0003 R 000045 OCIN      0004   000055 ODCON     0003 R 000047 ODEN     0003 R 000046 OPRIOP
  0003 R 000035 OPROP     0003 R 000036 OW        0005 R 000010 PACCEL    0003 R 000050 PCOND    0003 R 000025 PCUM
  0004   000103 PJOIN     0003 R 000043 PPASS     0003 R 000042 PORAT     0004   000071 PORATH   0003 R 000026 PRIRCM
  0003 R 000031 PROP      0003 R 000044 PST       0004   000065 SBLTH     0004   000062 SEPTH    0005   000002 SWEND
  0005   000003 SKCHI     0004   000112 SPCOR     0003 R 000032 SPFAC     0004   000072 SPMVTH   0000 R 000000 TR
  0005   000004 TRAND     0005   000005 TRCHI     0005   000006 URKBND    0005   000007 URKCHI   0005 R 000014 VACCEL
  0004   000063 VFAC      0003 R 000037 VOLIN     0004   000101 VOLLIM    0003 R 000040 VOLPT    0003 R 000050 VRIM
  0004   000104 VRJOIN    0003 R 000034 W         0003 R 000033 WADJ      0004   000060 WADJIN   0005   000000 WAIT
  0004   000106 WDELSM    0004   000067 WFAC      0004   000105 WSIM      0004   000050 WTINIT   0004   000056 XOVFLO
  0004   000057 XUNFLO

 00000  *DIAGNOSTIC*  THE NAME GET APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
 00101      1*          FUNCTION TR(AM,AMET)                                                    000023
 00101      2*     C      CALCULATES THE TRACE OF THE MATRIX AM RELATIVE TO THE METRIC AMET.    000023
 00103      3*          INCLUDE MISH                                                            000023
 00131      4*          REAL AM(475),AMET(475)                                                  000023
 00132      5*          TR=  AM(1)*AMET(1)*.5                                                   000023
```

B-49 8/6

```
00133      6*          DO 10 I=2,MM                                                  000035
00136      7*       10 TR=TR+AM(I)*AMET(I)                                           000036
00140      8*          TR=TR+TR                                                      000041
00143      9*    C        WE MUST DOUBLE THE OFFDIAGONAL TERMS (SEE COMMENT IN FUNCTION DOT)  000041
00140     10*    C        NOW SUBTRACT DIAGONALS.                                   000041
00141     11*          DO 15 I=2,MO                                                  000046
00144     12*          MXA=MXAR(I)                                                NEW000046
00145     13*       15 TR=TR-AM(MXA + I)*AMET(MXA+I)                              NEW000055
00147     14*          RETURN                                                     -01000063
00150     15*          END                                                           000123

 END OF COMPILATION:        1 DIAGNOSTICS.
```

B-50
87

APPENDIX C

SAMPLE OUTPUT

```
DUMP OF OBSERVED CLUSTERS FROM  0     119

CLUSTER   0 INDEX   0  PROPORTION 1.000000  W*      .000  SPLIT*********

WARNING **UNNORMALIZED NUMBER(S) BEING EDITED FOR OUTPUT**
     WEIGHT    58800.001  WAS        .001     ADJUST          .000  ID999999
     PROPORTION: PROP 1.00000  CIN   .10000-02  CTOT    .00000
     OLD PROP 1.000000  CIN .10000-02  ODEN .10000-02  DIFFER .00000

WARNING **UNNORMALIZED NUMBER(S) BEING EDITED FOR OUTPUT**

WARNING **UNNORMALIZED NUMBER(S) BEING EDITED FOR OUTPUT**

WARNING **UNNORMALIZED NUMBER(S) BEING EDITED FOR OUTPUT**
     VOLUME     .000001-39   ROOT     .000001-39   DCON       .0000000
     LOCATION  119  LINK  0*****  SUBS 20  907  SUPER********  SYMBOL  0

WARNING **UNNORMALIZED NUMBER(S) BEING EDITED FOR OUTPUT**

WARNING **UNNORMALIZED NUMBER(S) BEING EDITED FOR OUTPUT**
     NET PROB .00000-39   DIRECT .00000-39   CUMS .39209+03 * .10003+01

 ISPLIT,PROUT=     1    3

CLUSTER    1 INDEX  20  PROPORTION   .159261  W*58800.001  SPLIT*********
     WEIGHT      533.136  WAS      386.234    ADJUST        792.468  ID 63753
     PROPORTION: PROP  .15931  CIN   .45802+03  CTOT   .55851+05
     OLD PROP  .159697  CIN .32704+03  ODEN .20237+04  DIFFER .00000
     VOLUME   .494152-22  ROOT      .707965-11   DCON   31.0823653
     LOCATION  907  LINK 21  974  SUBS***    0  SUPER  0  119  SYMBOL  1
     NET PROB .83355-03   DIRECT .52323-02   CUMS .00000    * .00000
                                             CUMS     0 *         0


     MEAN              .244191+02    .220744+02    .220631+02    .280189+02

     COVARIANCE        .151249+01    .375306+00   -.153323+00   -.264263+00
                  2    .375306+00    .931926+00    .367113+00    .332197+00
                  3   -.153323+00    .367113+00    .136749+01    .135775+01
                  4   -.264263+00    .332197+00    .135775+01    .247297+01

     SKEW(*W)         -.189279+02   -.423267+02   -.186871+02    .113485+03

     KURT(*W)     1    .621717+03    .317013+03   -.143623+03   -.797597+02
                  2    .317013+03    .679747+03    .190359+03    .265770+03
                  3   -.143623+03    .190359+03    .989500+03    .971920+03
                  4   -.797597+02    .265770+03    .971920+03    .197572+04

     OLD MEAN          .244380+02    .220928+02    .220780+02    .280100+02

     OLD COVARIANCE    .196399+01    .389956+00   -.155557+00   -.278417+00
                  2    .389956+00    .969695+00    .357242+00    .326909+00
                  3   -.155557+00    .357242+00    .136530+01    .132724+01
                  4   -.278417+00    .326909+00    .132724+01    .246794+01
```

62 84

```
ISPLIT,PROUT=      0     7

CLUSTER   1 INDEX  21  PROPORTION  .017117  W*56800.001  SPLIT*********
     WEIGHT       192.093  WAS       117.817    ADJUST       255.634  ID 60272
     PROPORTION: PROP   .01712  CIN    .12728+03  CTOT    .51365+05
     OLD PROP   .017605  CIN .76710+02  ODEN .43263+04  DIFFER .00000
     VOLUME     .167813-23   KCOT     .409650-10   DCON    30.4185677
     LOCATION  974  LINK 17  639  SUBS***     0  SUPER  0  119  SYMBOL  2
     NET PROB .19774+01   DIRECT .11549+03    CUMS .0000   * .00000
                                              CUMS      0 *      0

   MEAN              .247375+02    .214028+02    .213331+02    .251397+02

   COVARIANCE        .214522+01    .973832+00   -.223327+00   -.175067+01
                2    .973832+00    .164586+01    .454767+00    .594253+00
                3   -.223327+00    .464767+00    .277522+01    .510245+01
                4   -.175767+01    .594253+00    .510245+01    .130666+02

   SKEW(*W)          .112030+03   -.369928+02    .891402+02   -.130702+03

   KURT(*W)     1    .982721+03    .457817+03    .171443+03   -.100396+04
                2    .457817+03    .607137+03    .177583+03    .198254+03
                3    .171443+03    .177583+03    .126173+04    .181904+04
                4   -.100396+04    .198254+03    .181904+04    .635322+04

   OLD MEAN          .247821+02    .215096+02    .213441+02    .251481+02

   OLD COVARIANCE    .225691+01    .952064+00   -.350735+00   -.198594+01
                2    .952064+00    .177156+01    .439803+00    .513850+00
                3   -.350735+00    .439803+00    .276711+01    .510723+01
                4   -.198594+01    .513850+00    .510723+01    .129797+02

ISPLIT,PROUT=      0     3

CLUSTER   1 INDEX  17  PROPORTION  .142164  W*56800.001  SPLIT -25.7294
     WEIGHT       657.725  WAS       384.050    ADJUST       788.100  IL 62814
     PROPORTION: PROP   .14221  CIN    .59099+03  CTOT    .54643+05
     OLD PROP   .138918  CIN .34786+03  ODEN .24544+04  DIFFER .12673+01
     VOLUME     .139525-22   ROOT     .373533-11   DCON    30.2206948
     LOCATION  639  LINK-10  371  SUBS 43 1108  SUPER  0  119  SYMBOL  3
     NET PROB .34972-09   DIRECT .24592-08   CUMS .10566-05 * .10128+01

   MEAN              .159431+02    .187757+02    .168655+02    .180105+02

   COVARIANCE        .117877+01    .922541+00    .643798+00    .813283+00
                2    .922541+00    .115379+01    .634979+00    .639340+00
                3    .643798+00    .634979+00    .605899+00    .540090+00
                4    .813283+00    .637340+00    .540090+00    .169201+01

   SKEW(*W)          .255284+02    .729395+01    .221810+02   -.253776+02

   KURT(*W)     1    .141696+04    .117224+04    .820552+03    .105645+04
                2    .117224+04    .131956+04    .790859+03    .880514+03
```

C-3 90

```
              3     .677772+03    .798859+03    .645351+03    .692648+03
              4     .179645+04    .888514+03    .692348+03    .193214+04

   OLD MEAN         .258687+02    .189146+02    .168832+02    .130127+02

   OLD COVARIANCE   .125836+01    .967684+00    .661623+00    .787530+00
              2     .967684+00    .127667+01    .663574+00    .608216+00
              3     .661623+00    .663574+00    .671201+00    .536065+00
              4     .787530+00    .608216+00    .536065+00    .174804+01

 ISPLIT,PROUT=      0     3

CLUSTER    1 INDEX -10  PROPORTION  .115314  W*58800.001  SPLIT -42.9731
    WEIGHT      931.929  WAS        691.595     ADJUST      1798.147  ID 62463
    PROPORTION: PROP  .11535  CIN    .75447+03  CTOT    .52137+05
    OLD PROP   .115347  CIN .55400+03  ODEN .47668+04  DIFFER .19082+02
    VOLUME     .804056-23   ROOT      .283559-11   DCON    32.3689790
    LOCATION  371  LINK 16  706  SUBS 48 1577  SUPER  0  119  SYMBOL  *
    NET PROB .10444-01   DIRECT .90541-01   CUMS .71194+02 * .11200+01

   MEAN             .262200+02    .197986+02    .163757+02    .161847+02

   COVARIANCE       .111537+01    .381424+00    .957464+00    .113333+01
              2     .381424+00    .986630+00    .674306+00    .833040+00
              3     .957484+00    .674306+00    .101638+01    .196463+01
              4     .113333+01    .833040+00    .196463+01    .333824+01

   SKEW(*W)        -.772521+02   -.741650+02   -.133592+03   -.289516+03

   KURT(*W)   1     .147827+04    .587151+03    .149416+04    .186600+04
              2     .587151+03    .115244+04    .115422+04    .158029+04
              3     .149416+04    .115420+04    .230446+04    .293492+04
              4     .186600+04    .158029+04    .293492+04    .443702+04

   OLD MEAN         .262147+02    .197624+02    .163804+02    .162197+02

   OLD COVARIANCE   .111525+01    .385717+00    .991062+00    .113689+01
              2     .385717+00    .968412+00    .694624+00    .875438+00
              3     .991062+00    .694624+00    .162137+01    .196956+01
              4     .113689+01    .875438+00    .196956+01    .323092+01

 ISPLIT,PROUT=      0     7

CLUSTER    1 INDEX  15  PROPORTION  .090132  W*58800.001  SPLIT -43.8563
    WEIGHT      576.300  WAS        321.239     ADJUST       662.478  ID 61951
    PROPORTION: PROP  .09016  CIN    .48929+03  CTOT    .53372+05
    OLD PROP   .087163  CIN .26808+03  ODEN .30813+04  DIFFER .11277+02
    VOLUME     .206329-22   ROOT      .454234-11   DCON    30.0344720
    LOCATION  706  LINK 14  144  SUBS 46 1041  SUPER  0  119  SYMBOL  5
    NET PROB .44395-04   DIRECT .49241-03   CUMS .14892-04 * .10359+01

   MEAN             .273305+02    .207649+02    .181555+02    .182596+02
```

C4 91

```
COVARIANCE      .120491+01   .644868+00   .117258+01   .131583+01
            2   .644868+00   .171231+01   .927047+00   .107653+01
            3   .117258+01   .927047+00   .176075+01   .191185+01
            4   .131583+01   .107653+01   .191185+01   .246139+01

SKEW(*W)       -.845190+02  -.227343+02  -.184875+03  -.213409+03

KURT(*W)    1   .134550+04   .634449+03   .151085+04   .175906+04
            2   .634449+03   .927069+03   .109178+04   .134794+04
            3   .151085+04   .109178+04   .227453+04   .266340+04
            4   .175906+04   .134794+04   .266340+04   .337106+04

OLD MEAN        .273362+02   .227596+02   .182114+02   .183430+02

OLD COVARIANCE  .122274+01   .748716+00   .119421+01   .131779+01
            2   .748716+00   .118601+01   .981393+00   .110216+01
            3   .119421+01   .981393+00   .177553+01   .183853+01
            4   .131779+01   .110216+01   .183853+01   .241185+01


ISPLIT,PROUT=      0    3

CLUSTER    1 INDEX  14  PROPORTION  .178724   W*58900.001  SPLIT*********
    WEIGHT       1211.320  WAS      1039.368    ADJUST     2702.200  ID 63734
    PROPORTION: PROP  .17376  CIN    .12202+04  CTOT   .52030+05
    OLD PROP   .178841  CIN .11384+04  ODEN .56253+04  DIFFER .00000
    VOLUME     .564358-22   ROOT     .751238-11   DCON    31.7569318
    LOCATION   144  LINK 15  438  SUBS***    0  SUPIR  0  119  SYMBOL  6
    NET PROB .16141-06  DIRECT .90286-08  CUMS .15952+05 * .10029+01

MEAN            .283813+02   .237969+02   .222934+02   .172373+02

COVARIANCE      .169773+01   .648572+00   .194256+01   .136348+01
            2   .648572+00   .142523+01   .926863+00   .909831+00
            3   .194256+01   .926863+00   .325918+01   .207082+01
            4   .136348+01   .909831+00   .207082+01   .253826+01

SKEW(*W)        .480341+02  -.125611+03   .129354+02  -.861008+02

KURT(*W)    1   .188751+04   .553720+03   .193258+04   .132148+04
            2   .553720+03   .130751+04   .766285+03   .849073+03
            3   .193258+04   .766285+03   .319878+04   .205673+04
            4   .132148+04   .849073+03   .205673+04   .226910+04

OLD MEAN        .283831+02   .231046+02   .223063+02   .172521+02

OLD COVARIANCE  .171554+01   .653767+00   .193009+01   .135861+01
            2   .653767+00   .145356+01   .937486+00   .912041+00
            3   .193009+01   .937486+00   .324906+01   .205065+01
            4   .135861+01   .912041+00   .205065+01   .255063+01


ISPLIT,PROUT=      0    3

CLUSTER    1 INDEX  15  PROPORTION  .297289   W*56830.001  SPLIT*********
```

ORIGINAL PAGE IS OF POOR QUALITY

C-5 92

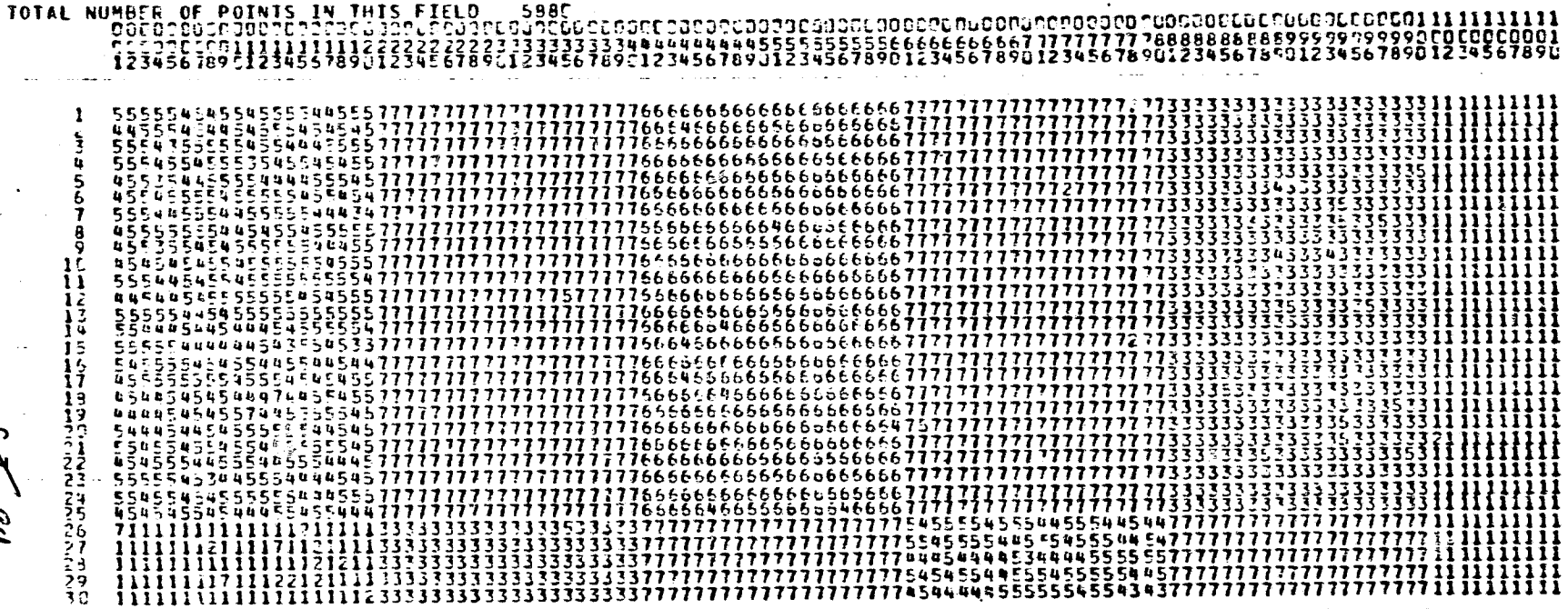
```
 WEIGHT    3369.666  WAS     1659.905    ADJUST    4315.753  ID 58967
 PROPORTION: PROP  .29738  CUM  .33370+04  CTOT   .49581+05
 OLD PROP  .297171  CIN .16417+04  ODEN .55459+04  DIFFER .00000
 VOLUME    .152962-21   ROOT    .123678-10   DCON   29.537926
 LOCATION  438  LINK***      0  SUBS***      0  SUPER   0   119  SYMBOL  7
 NET PROB .39022+03   DIRECT .13122+04   CUMS .51699+04 * .99521+00

MEAN              .253367+02    .210946+02    .202328+02    .226950+02

COVARIANCE        .184812+01    .717034+00    .108135+01    .139348+01
             2    .717034+00    .180428+01    .536412+00    .654312+00
             3    .108135+01    .536412+00    .117996+01    .106108+01
             4    .139348+01    .654312+00    .106108+01    .176069+01

SKEW(*W)          -.272165+03    .288031+03   -.430796+03   -.158198+03

KURT(*W)    1     .157035+05    .561123+04    .100230+05    .127347+05
            2     .561123+04    .123593+05    .431592+04    .491342+04
            3     .100230+05    .431592+04    .103270+05    .980183+04
            4     .127347+05    .491342+04    .980183+04    .154934+05

OLD MEAN          .255450+02    .210032+02    .202372+02    .226945+02

OLD COVARIANCE    .192517+01    .730219+00    .107758+01    .139330+01
             2    .730219+00    .184722+01    .536079+00    .654863+00
             3    .107758+01    .536079+00    .124827+01    .105755+01
             4    .139330+01    .654863+00    .105755+01    .183398+01


 ISPLIT,PROUT=    0    3
EXIT FROM MULTI
```

ORIGINAL PAGE IS
OF POOR QUALITY

C-6 93

LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

TOTAL NUMBER OF POINTS IN THIS FIELD 5980

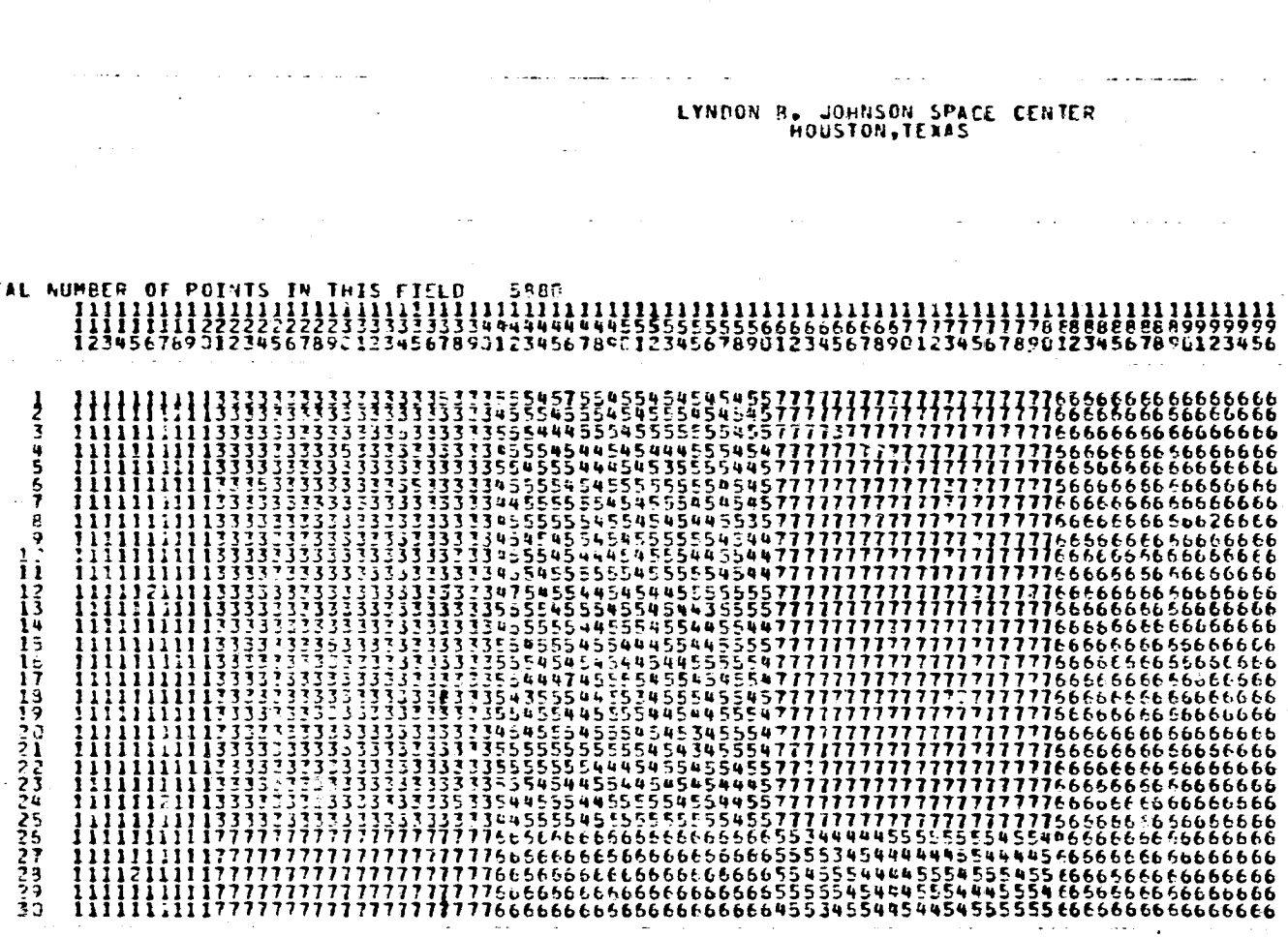


2-94

ORIGINAL PAGE IS
OF POOR QUALITY

LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

TOTAL NUMBER OF POINTS IN THIS FIELD 5886



POINTS PER CLUSTER IN THIS FIELD

CLUSTER	SYMBOL	POINTS
1	1	684
2	2	16
3	3	1171
4	4	445
5	5	763
6	6	1069

ORIGINAL PAGE IS OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

TOTAL NUMBER OF CLUSTERS = 7

TOTAL NUMBER OF POINTS = 5880

CLUSTER	SYMBOL	POINTS IN CLUSTER
1	1	684
2	2	16
3	3	1171
4	4	445
5	5	763
6	6	1269
7	7	1892

GPMD

PMD 0331-04/01- 02:10:40
SYSTEPLIB%. LEVEL 71-3

BANK- $1BANK SEGMENT- $MAIN$ ELEMENT- TREE $I 011 AT ADDRESS 001000 CREATED ON: 03 MAR 77 AT 10:18:00

BANK- $1BANK SEGMENT- $MAIN$ ELEMENT- ACOM $I 011 AT ADDRESS 001436 CREATED ON: 02 MAR 77 AT 15:02:04

BANK- $1BANK SEGMENT- $MAIN$ ELEMENT- MTVEC $I 011 AT ADDRESS 00163C CREATED ON: 03 MAR 77 AT 09:59:24