NASA CR-160562
JSC-16466
MAR 11 1980

National Aeronautics and Space Administration

Lyndon B. Johnson Space Center
Houston, Texas 77058

EARTH OBSERVATIONS DIVISION

SPACE AND LIFE SCIENCES DIRECTORATE

"AS-BUILT" DESIGN SPECIFICATION FOR CLASSY, AN ADAPTIVE MAXIMUM LIKELIHOOD CLUSTERING METHOD

Job Order 76-662

Prepared By
Lockheed Engineering and Management Services Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15800

February 1980

LEMSCO-14546

1. Report No. JSC-16466	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle "As-Built" Design Specification for CLASSY, an Adaptive Maximum Likelihood Clustering Method		5. Report Date February 1980 6. Performing Organization Code SF3
7. Author(s) C. L. Horton and R. K. Lennington		8. Performing Organization Report No. LEMSCO-14546 10. Work Unit No.
9. Performing Organization Name and Address Lockheed Engineering & Management Services Company, Inc. Systems and Services Division Houston, Texas 77058		11. Contract or Grant No. 9-15800 13. Type of Report and Period Covered User Guide
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058 (J. Sulester, Tech. Monitor)		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This document contains the description of the latest modifications to the CLASSY System and the associate postprocessor MAXLABEL.

17. Key Words (Suggested by Author(s))	18. Distribution Statement		
19. Security Classif. (of this report) Unclassfied	20. Security Classif. (of this page) Unclassified	21. No. of Pages 63	22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161

JSC- 16466

"AS-BUILT" DESIGN SPECIFICATION
FOR
CLASSY, AN ADAPTIVE MAXIMUM LIKELIHOOD CLUSTERING METHOD

Job Order 76-662

Prepared By

C. L. Horton
R. K. Lennington

Approved By

LEMSCO

J. I. Morrow 2-19-80

J. I. Morrow, Supervisor
Software Section

JSC

J. M. Sulester 2/26/80

J. M. Sulester, Technical
Monitor Systems & Facilities
Branch

Prepared By

Lockheed Engineering and Management Services Company, Inc.

For

Earth Observations Division

Space and Life Sciences Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS 77058

February 1980

LEMSCO-14546

CONTENTS

Figures

1. SCOPE

This specification establishes the modifications to the CLASSY program as specified in IDSD Category 1 Job Order Task Agreement, titled CLASSY Program Modification. This modification includes the CLASSY post-processor MAXLABEL and its associated program IMAGE.

2. APPLICABLE DOCUMENTS

The following documents form a part of this specification:

J. O. 76-662 Task Agreement titled: CLASY Program Modification

Technical Memorandum Mathematical Description and Program Documentation for CLASSY, An Adaptive Maximum Likelihood Clustering Method, by R. K. Lennington and M.E. Rassbach Elogic, Inc. Houston, Texas.

Earth Observations Division Version of the Laboratory for Applications of Remote Sensing System (EOD-LARSYS) User Guide for the IBM 370/148 Volumn II - User's Reference Manual.

3. SYSTEM DESCRIPTION

The CLASSY clustering algorithm serves to estimate the component distributions which make up the overall mixture distribution of the data. The present software uses the statistics for these component distributions (clusters) along with a set of labelled data vectors to produce maximum likelihood estimates of the proportion of each labelled class associated with each cluster. These estimates (called betas) are obtained through a fixed point iteration procedure. The estimated betas are used to obtain two different estimates of the proportion of each labelled class in the whole scene. In addition labelled cluster maps are produced for the whole scene using two different techniques.

The maximum Likelihood System of programs consists of the following four programs: CLASSY, DOT, IMAGE and LABEL. (Figure 3-1).

The CLASSY Program creates statistically meaningful clusters from the pixels in a selected area of the segment image tape. The program options are defined in a control card file. The cluster statistical parameters for each iteration are saved on the CLASSY Cluster Statistical Parameter Files for later use by the LABEL program and a CLASSY One Channel Unlabelled Cluster Map is optionally written to tape for use by the Accuracy Assessment Programs. Cluster maps are written at the end of the last iteration and optionally written after each iteration (Figures 3-1 and 3-2).

The DOT program creates a Ground Truth Selected Pixel Values and Names File associated with pixel radiance values from the same segment image tape used by CLASSY, the Ground Truth Description File and the program options as defined on the Dot Control Card File. (Figures 3-1 and 3-5).

The IMAGE program creates a Selected Pixel File from the same segment image tape used by CLASSY and DOT. The LABEL Control Card File is used to define the options. (Figures 3-1 and 3-9).

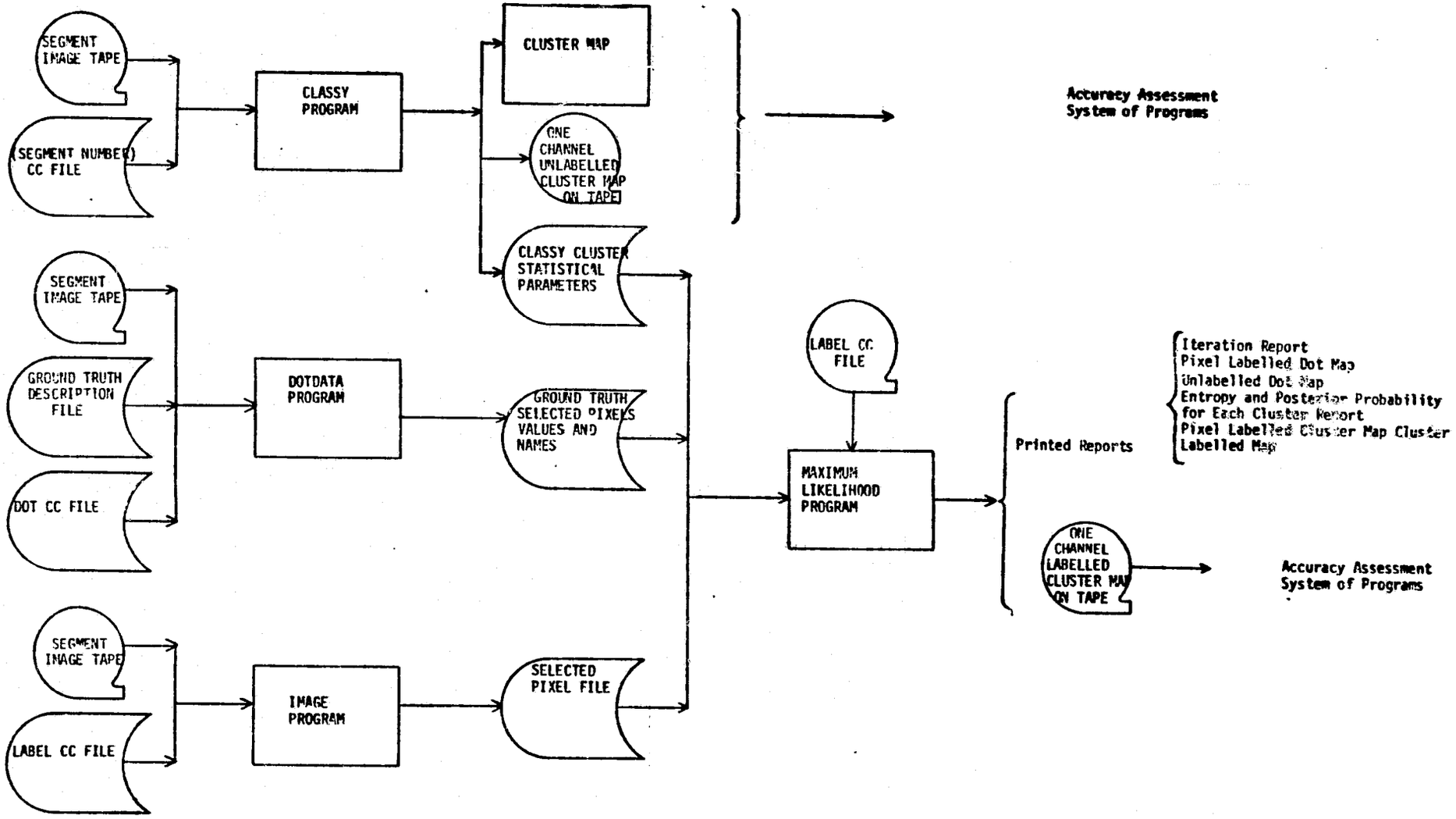
The LABEL program processes the CLASSY Cluster Statistical Parameter File from CLASSY, the Ground Truth Selected Pixel Values and Name File from DOT and the Selected Pixel File from IMAGE to create a One Channel Labelled Cluster Map on tape for use by the Accuracy Assessment Programs and the six following reports:

Iteration Report
Pixel Labelled Dot Map
Unlabelled Dot Map
Entropy and Posterior Probability for each Cluster Report
Pixel Labelled Cluster Map
Cluster Labelled Map

Processing Order -- EXECS: CLASSY or CLASSYN
DOT
LABEL

PROGRAMS: CLASSY
DOT
IMAGE and LABEL

SYSTEM FLOW CHART



CLASSY EXEC FILE

```
>type c)sssu exec
GLOBAL TXTLIB FORTRAN CMSLIB
&IF &STORAGE GE 2040 &SKIP 2
&TYPE DFF STOR 2M REQUIRED
&EXIT
&IF &INDEX EQ 2 &SKIP 3
&IF &3 EQ 1 &SKIP 2
&SK = &2 - 1
TAPE FSF &SK
&CONTROL OFF
GETDISK JSC770 191 350 R B/A PASS AUCOIN
&BEEP = 1
-GET GETDISK TEMP 3M CLEAR DETACH
&IF &RETCODE EQ 0 &SKIP 7
&IF &BEEP LE 7 &SKIP 3
&TYPE NO 25 CYL TEMP DISK AVAILABLE
CP Q T
&EXIT
CP SLEEP 5 MIN
&BEEP = &BEEP + 1
&GOTO -GET
Q DISK D
GETDISK LARSYS
FILEDEF FT02F001 DISK FILE FT02F001 A1 (LRECL 320 BLKSIZE 320  PERM
FILEDEF FT02F002 DISK FILE FT02F002 A1 (LRECL 320 BLKSIZE 320  PERM
FILEDEF 3 TERM (PERM
FILEDEF FT04F001 DISK FILE FT04F001 A4 (LRECL 3060 BLKSIZE 3060 PERM
FILEDEF FT04F002 DISK FILE FT04F002 A4 (LRECL 3060 BLKSIZE 3060 PERM
FILEDEF FT04F003 DISK FILE FT04F003 A4 (LRECL 3060 BLKSIZE 3060 PERM
FILEDEF 6 DISK OUT6 LISTING D (PERM
FILEDEF FT09F001 DISK FILE FT09F001 A1 (LRECL 320 BLKSIZE 320 PERM
FILEDEF FT11F001 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F002 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F003 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F004 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F005 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F006 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F007 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F008 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F009 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F010 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F011 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F012 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F013 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F014 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F015 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F016 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F017 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F018 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F019 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F020 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F021 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F022 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F023 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F024 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F025 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F026 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F027 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F028 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F029 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F030 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT12F001 TAP2 (BLKSIZE 3060 RECFM U DEN 800 PERM
FILEDEF FT12F002 TAP2 (BLKSIZE 12356 RECFM U DEN 800 PERM
FILEDEF FT13F001 DISK FILE FT13F001 A4 (LRECL 320 BLKSIZE 320 PERM
FILEDEF FT14F001 DISK FILE FT14F001 D1 (LRECL 3060 BLKSIZE 3060 PERM
FIL FT16F001 DISK FILE FT16F001 A1 (LRECL 3060 RECFM U BLKSIZE 3060 PERM
FILEDEF FT17F001 TAP3 (BLKSIZE 12356 RECFM U DEN 800 PERM
FILEDEF FT19F001 DISK FILE FT19F001 A1 (LRECL 1860 BLKSIZE 1860 PERM
FILEDEF FT19F002 DISK FILE FT19F002 A1 (LRECL 1860 BLKSIZE 1860 PERM
FILEDEF FT20F001 DISK FILE FT20F001 A4 (LRECL 320 BLKSIZE 320  PERM
FILEDEF FT20F002 DISK FILE FT20F002 A4 (LRECL 320 BLKSIZE 320  PERM
FILEDEF FT21F001 DISK &1 CC A1 (LRECL 80 BLKSIZE 800 PERM
FILEDEF FT21F002 DISK FILE FT21F002 A1 (LRECL 80 BLKSIZE 800 PERM
FILEDEF 22 DISK FILE FT22F001 D (LRECL 800 BLKSIZE 800 XTENT 2100 PERM
FIL FT23F001 DISK &1 1 A4 (LRECL 120 BLKSIZE 130 RECFM VS PERM
FIL FT23F002 DISK &1 2 A4 (LRECL 120 BLKSIZE 130 RECFM VS PERM
FIL FT23F003 DISK &1 3 A4 (LRECL 120 BLKSIZE 130 RECFM VS PERM
FIL FT23F004 DISK &1 4 A4 (LRECL 120 BLKSIZE 130 RECFM VS PERM
FIL FT23F005 DISK &1 5 A4 (LRECL 120 BLKSIZE 130 RECFM VS PERM
FILEDEF 26 DISK FILE MEANS A (PERM
LOAD CLASY BLKCLA (CLEAR START NOMAP
REMOTE E TO HOUSTON
SPOOL E HOLD
PRINT OUT6 LISTING D
&IF &INDEX LT 4 &SKIP 10
CP DET 181
TAPMOUNT &4 TAP1 RI
&IF &INDEX NE 5 &SKIP 3
&IF &5 EQ 1 &SKIP 2
&SP = &5 - 1
TAPE FSF &SP
FIL INMOVE DISK FILE FT16F001 A (BLKSIZE 3060 RECFM U PERM
FIL OUTMOVE TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
MOVEFILE
TAPE WTM 2
CP DET 181
```

SAMPLE CLASSY CONTROL CARD FILE

```
>TYPE 1394 CC A

COMMENT         CLASSY RUN ON SEGMENT 1394
CHANNELS        1,2,3,4,5,6,7,8
DATAFILE        FILE=1
ITER            5
MAP             5
LINES           1-117
NPTS            2
SEGM            1394
PROC            30,1,80
TIME            150
*END*
          (1,1),  (1,1), (196,1),  (196,117), (1,117)
$END*
```

SAMPLE CLASSY EXECUTION of Segment 1394

```
>def stor 2m
STORAGE = 02048K
R; T=0.01/0.01 10:45:37

>classy 1394 5123 1
GLOBAL TXTLIB FORTRAN CMSLIB
JSC770   191 HAS BEEN ATTACHED AS 350.
B (350) R/O
350 HAS BEEN LOGGED IN AS B/A DISK.
DASD 192 DETACHED
YOU ALREADY HAD A VIRTUAL DEVICE 192.  IT IS BEING DETACHED.
TEMP     157 HAS BEEN ATTACHED AS 192.  (003.00 MEGABYTES)
192 HAS BEEN LOGGED IN AS D   DISK.
D (192): 0 FILES; 4 REC IN USE, 3720 LEFT (of 3724), 0% FULL (14 CYL), 3330, R/W
LARSLIB  29C HAS BEEN ATTACHED AS 19C.
Y (19C) R/O
19C HAS BEEN LOGGED IN AS Y   DISK.
EXECUTION BEGINS...
 CLASY STARTED
 PROPORTION RELATIVE TO TOP LEVEL =     1.000000       1
 00-00
 01-00
 02-45 03-55



 PROPORTION RELATIVE TO TOP LEVEL =     1.000000       1
 PROPORTION RELATIVE TO TOP LEVEL =     0.750047       2
 PROPORTION RELATIVE TO TOP LEVEL =     1.000000       1
 INDEX =      2    SYMBOL = ******
 KL=*** INDEX(KL)=  2 NUMBER=  2 RELPRP=  0.0   0.0
 PROPORTION RELATIVE TO TOP LEVEL =     0.816835       2
 00-00
 01-00
 02-82       03-18
 04-37 05-45



 PROPORTION RELATIVE TO TOP LEVEL =     1.000000       1
0***SEPERATE   1 SUPER,SUBS   0  2 SPFAC 0.18557E 02
 00-00
 02-82       03-18
 04-37 05-45



 PROPORTION RELATIVE TO TOP LEVEL =     0.830760       2
 PROPORTION RELATIVE TO TOP LEVEL =     0.484035       5
 00-00
 02-82             03-16
 04-34 05-48
       06-24 07-24


 PROPORTION RELATIVE TO TOP LEVEL =     0.852994       2
0***SEPERATE   2 SUPER,SUBS   0  4 SPFAC 0.53649E 02
 00-00
 04-29 05-56       03-15
       06-45 07-11
```

ORIGINAL PAGE IS
OF POOR QUALITY

CLASSYN EXEC FILE

```
>type classyn exec

&CONTROL OFF
CP SPOOL D CONT NOH TO BATCH
&PUNCH BATCH MACHINE BATJSC
&PUNCH BATCH ID JSC235 JSC235 TENT L
&PUNCH BATCH OUTPUT HOUSTON HOUSTON
&PUNCH EXEC$$
&PUNCH CP SPOOL 9 CLOSE
&PUNCH CP SPOOL 9 TO JSC235 START CL C NOHOLD
&PUNCH GETDISK JSC770 191 291 R B PASS AUCOIN
&PUNCH GETDISK JSC235 191 391 W A PASS WRTE7
&PUNCH EXEC CFILE &1 &2 &3 &4 &5
&PUNCH RELEASE B (DET
&PUNCH CP Q T
&PUNCH CP SPOOL 9 CLOSE STOP
&PUNCH $$
CP SPOOL PUNCH NOCONT CLOSE
CP SPOOL PUNCH HOLD TO RSCS
&EXIT
```

```
>type dot exec                                                    DOT EXEC FILE

&CONTROL OFF
GETDISK ./SC770 191 350 R B/A PASS AUCOIN
GETDISK TEMP 3M CLEAR DETACH
FIL 21 DISK DOT CC A (BLKSIZE 80 RECFM F LRECL 80 PERM
FIL 22 DISK FILE FT22F001 D (LRECL 800 BLKSIZE 800 XTENT 2100 PERM
FIL 3 TERMINAL (PERM
FIL 6 PRINTER ( PERM
FIL FT11F001 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F002 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F003 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F004 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F005 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F006 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F007 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F008 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F009 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F010 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F011 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F012 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F013 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F014 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F015 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F016 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F017 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F018 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F019 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F020 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F021 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F022 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F023 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F024 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F025 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F026 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT11F027 TAP1 (BLKSIZE 12356 RECFM U PERM
FIL FT12F001 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F002 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F003 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F004 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F005 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F006 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F007 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F008 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F009 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F010 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F011 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F012 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F013 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F014 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F015 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F016 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F017 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F018 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F019 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F020 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F021 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F022 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F023 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F024 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F025 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F026 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT12F027 TAP2 (BLKSIZE 12356 RECFM U PERM
FIL FT19F001 DISK &1 DOT2 A (LRECL 80  BLKSIZE 80 PERM
FIL FT19F002 DISK &1 DOT1 A (LRECL 80  BLKSIZE 80 PERM
LOAD MONTOR BLKCOM (CLEAR START NOMAP
```

SAMPLE DOT CONTROL CARD FILE

```
>TYPE DOT CC A

$DOTDATA
CHANNEL       DATA=1,2,3,4,5,6,7,8
DATAFILE      UNIT=11,FILE=7
DOTFIL        OUTPUT/UNIT=19,FILE=1
OPTION        LACIE
OPTION        PRINT
*END
DOT 2  N     1 6 7 8 10 11 13 14 19 21 25 29 33 35 37
DOT 2  N     39 40 41 42 44 45 47 49 50 51 53 54 55
DOT 2  N     56 57 59 61 65 67 73 78 83 84 86 87 88 91
DOT 2  N     92 94 95 99 101 105 109 111 113 121 122 123 125
DOT 2  N     128 129 130 131 132 135 139 141 145 147 149 153 154
DOT 2  N     155 156 159 160 161 162 163 165 166 167 168 169 170
DOT 2  N     171 175 177 179 183 185 189 191 192 193 194 195 196
DOT 2  N     197 203 204 207 208 209 20 22 28 32 34 36 38
DOT 2  N     60 62 64 68 72 100 104 136 140 142 146 152 172
DOT 2  N     174 176 180 182 186 190
DOT 2  S     2 3 4 5 9 12 15 16 17 18 23 27 31 43 46 48
DOT 2  S     52 63 69 71 75 77 79 80 81 82 85 89 90 93 97 103
DOT 2  S     107 115 116 117 118 119 120 124 126 127 133 137 143 151
DOT 2  S     157 158 164 173 181 187 198 199 200 201 202 205 206
DOT 2  S     26 30 58 66 70 76 96 102 106 134 148 150 178 188
$END*
$EXIT
```

SAMPLE DOT EXECUTION of Segment 1394

```
>def stor 2m
STORAGE = 02048K
R; T=0.01/0.02 10:33:06

>dot 1394
JSC770   191 HAS BEEN ATTACHED AS 350.
B (350) R/O
350 HAS BEEN LOGGED IN AS B/A DISK.
DASD 192 DETACHED
YOU ALREADY HAD A VIRTUAL DEVICE 192.  IT IS BEING DETACHED.
TEMP     155 HAS BEEN ATTACHED AS 192.  (003.00 MEGABYTES)
192 HAS BEEN LOGGED IN AS D   DISK.
THE FOLLOWING NAMES ARE UNDEFINED:
 WRTBM
EXECUTION BEGINS...
```

>type label exec

LABEL EXEC FILE

```
&CONTROL OFF
GETDISK JSC770 191 350 R B/A PASS AUCOIN
&BEEP = 1
-GET GETDISK TEMP 3M CLEAR DETACH
&IF &RETCODE EQ 0 &SKIP 7
&IF &BEEP LE 7 &SKIP 3
&TYPE NO 10 CYL TEMP DISK AVAILABLE
CP Q T
&EXIT
CP SLEEP 5 MIN
&BEEP = &BEEP + 1
&GOTO -GET
GETDISK LARSYS
Q DISK *
FILEDEF 3 TERM (PERM
FILEDEF 6 DISK LABEL6 LISTING D (PERM
FILEDEF 8 DISK &1 DOT1 A (LRECL 3060 RECFM V BLKSIZE 3060 PERM
FILEDEF FT11F001 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F002 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F003 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F004 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F005 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F006 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F007 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F008 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F009 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F010 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F011 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F012 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F013 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F014 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F015 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F016 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F017 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F018 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F019 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F020 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F021 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F022 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F023 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F024 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F025 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F026 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F027 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F028 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F029 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF FT11F030 TAP1 (BLKSIZE 3060 RECFM U DEN 1600 PERM
FILEDEF 12 DISK &1 VARM A (LRECL 80 BLKSIZE 80 PERM
FILEDEF FT19F001 DISK FILE FT19F001 A (LRECL 1860 BLKSIZE 1860 PERM
FILEDEF FT19F002 DISK FILE FT19F002 A (LRECL 1860 BLKSIZE 1860 PERM
FILEDEF FT20F001 DISK FILE FT20F001 A (LRECL 320 BLKSIZE 320 PERM
FILEDEF FT20F002 DISK FILE FT20F002 A (LRECL 320 BLKSIZE 320 PERM
FILEDEF FT21F001 DISK LABEL CC A (LRECL 80 BLKSIZE 80 PERM
FILEDEF FT21F002 DISK FILE FT21F002 A (LRECL 80 BLKSIZE 80 PERM
FILEDEF FT22F001 DISK FILE FT22F001 A (LRECL 800 BLKSIZE 800 XTENT 2100 PERM
FIL FT23F001 DISK &1 5 A4 (LRECL 120 BLKSIZE 130 RECFM VS PERM
FIL 24 DISK FILE FT24F001 D4 (LRECL 1200 BLKSIZE 1250 RECFM VS PERM
FILEDEF 25 DISK FILE FT25F001 D (LRECL 132 BLKSIZE 132 PERM
FILEDEF 26 DISK FILE FT26F001 D (LRECL 132 BLKSIZE 132 PERM
FILEDEF 27 DISK FILE FT27F001 D (LRECL 132 BLKSIZE 132 PERM
FILEDEF 28 DISK FILE FT28F001 D (LRECL 132 BLKSIZE 132 PERM
FILEDEF 29 DISK FILE FT29F001 D (LRECL 132 BLKSIZE 132 PERM
FIL FT31F001 DISK FILE FT31F001 D (LRECL 3060 BLKSIZE 3060 RECFM U PERM
GLOBAL TXTLIB FORTRAN CMSLIB
LOAD IMAGE BLKCLA (CLEAR START NOMAP
LOAD MAXLABEL BLKCLA (CLEAR START NOMAP
CP REMOTE E TO HOUSTON
CP SPOOL PRINTER HOLD NOCONT
CP TAG QUE DEV E
L * * D (ALL
PRINT LABEL6 LISTING D
&IF &INDEX LT 2 &SKIP 8
&IF &2 EQ 1 &SKIP 2
&SP = &2 - 1
TAPE FSF &SP (TAP2
FIL INMOVE DISK FILE FT31F001 D (LRECL 3060 BLKSIZE 3060 RECFM U PERM
FIL OUTMOVE TAP2 (  BLKSIZE 3060 RECFM U DEN 1600 PERM
MOVEFILE
TAPE WTM 2 (TAP2
```

```
>TYPE LABEL CC A                    SAMPLE LABEL CONTROL CARD FIL

CHANNELS        1,2,3,4,5,6,7,8
MAP             PIXEL
DATA            FILE = 8
SEGMENT         1394
PROC            07,02,80
*END*
          (1,1),  (1,1), (196,1),  (196,117), (1,117)
$END*
```

SAMPLE EXECUTION OF IMAGE AND LABEL FOR SEGMENT 1394

```
>label 1394
JSC770   191 HAS BEEN ATTACHED AS 350.
'350' REPLACES ' B (350) '
B (350) R/O
350 HAS BEEN LOGGED IN AS B/A DISK.
DASD 192 DETACHED
YOU ALREADY HAD A VIRTUAL DEVICE 192.  IT IS BEING DETACHED.
TEMP     155 HAS BEEN ATTACHED AS 192.  (003.00 MEGABYTES)
192 HAS BEEN LOGGED IN AS D   DISK.
LARSLIB  29C HAS BEEN ATTACHED AS 19C.
Y (19C) R/O
19C HAS BEEN LOGGED IN AS Y   DISK.

EXECUTION BEGINS...
 IMAGE DATA EXTRACTED FROM TAPE, LAST LINE = 117


EXECUTION BEGINS...
```

3.1 HARDWARE DESCRIPTION

The CLASSY clustering program is operational on the IBM 3031. system at Purdue.

3.2 EXEC FILES

3.2.1 CLASSY EXEC FILES

CLASSY is called by using one of the following EXEC files:

CLASSY (segment number), (input tape number), (input file number), (output tape number), (output file number) for interactive runs.

CLASSYN (segment number), (input tape number), (input file number), (output tape number), (output file number) for batch runs.

Where segment number = name of control card file with type CC

input tape number = number of input tape

input file number = number of input file

output tape number = number of output tape

output file number = number of output file

These EXEC files assign (segment number) CC to be the control card input file which specifies the program options. The output files are (segment number) 1, ..., (segment number) (last iteration) and FT16F001.

3.2.2 DOT EXEC FILE

The pixel data files are extracted by executing the following EXEC file to call the DOT DATA processor.

DOT (segment number).

Where segment number = the name of the output files, (segment number) DOT1 and (segment number) DOT2. The name of the control card file is DOT CC.

***** Warning: The tape must be mounted and positioned prior to running the DOT EXEC.

3.2.3 IMAGE AND MAXLABEL EXEC FILE

The post-processor MAXLABEL and its associated program IMAGE are called by the following EXEC file:

LABEL (segment number)

where segment number is the number of the segment.

This EXEC file assigns LABEL CC to be the control card input file which specifies the program options. The report output is sent to the line printer and the labelled cluster map is placed on the D disk as FILE FT31F001.

***** Warning: The tape must be mounted prior to to running the LABEL EXEC.

3.3 CONTROL CARDS

3.3.1 CLASSY CONTROL CARDS

The following control cards are input to the modified CLASSY program, and are analysed by SETUP9. In all cards the "keyword" begins in card column 1 and any parameters on the card are in card columns 11 through 72. Numbers in a series are separated by commas; blanks are optional.

1. "CHANNEL" CARD

    EXAMPLE: CHANNEL 1, 5, 9, 13

The "CHANNEL" card specifies the channel numbers to be used in clustering the multi-channel data vectors. The maximum number of channels allowed is sixteen.

2. "NPTS" CARD

    EXAMPLE: NPTS 2

This card specifies the number of pixels to skip between selected pixels.

3. "DATA" CARD

    EXAMPLE: DATA FILE = 2

This card specifies the input file number. The default value is 1.

4. "MAP" CARD

    EXAMPLE: MAP 1, 3, 5

This card specifies the iteration numbers for which intermediate pixel maps are to be drawn.

5. "LINES" CARD

    EXAMPLE: LINES 1-10, 25-34

This card specifies the lines to be mapped on the intermediate pixel maps.

6. "SEGMENT" CARD

EXAMPLE: SEGMENT 1234

This optional card is used to specify the segment number used in the header of FILE FT31F001. The Accuracy Assessment system of programs requires this data in the header.

7. "PROCESS" CARD

EXAMPLE: PROCESS 2,11,80

This optional card is used to specify the date used in the header of FILE FT31F001. The Accuracy Assessment system of programs requires this data in the header.

8. "*END*" CARD

This mandatory card specifies the end of the control cards.

3.3.2 LABEL CONTROL CARDS

The following control cards are input to the post-processor MAXLABEL and its associated program IMAGE. The cards are analysed by the subroutine SETUPM. In all cards, the "keyword" begins in card column 1 and any parameters are entered from card columns 11 through 72. Numbers in a series are separated by commas; blanks are optional.

1. "CHANNEL" CARD

EXAMPLE: CHANNEL 1, 5, 9, 13

The "CHANNEL" card specifies the numbers of the channels to be used. The maximum number of channels allowed is sixteen.

2. "DATA" CARD

EXAMPLE: DATA FILE =2

This card is used to specify the input tape file number. The default value is 1.

3. "MAPOPT" CARD

EXAMPLES: MAPOPT PIXEL

MAPOPT CLUSTER

This card is used to specify either a Pixel Map or a Cluster Map.

4. "SEGMENT" CARD

EXAMPLE: SEGMENT 1234

This optional card is used to specify the segment number used in the header of FILE FT31F001. The Accuracy Assessment system of programs requires this data in the header.

5. "PROCESS"

EXAMPLE: PROCESS 2,1,80

This optional card is used to specify the date used in the header of FILEFT31F001. The Accuracy Assessment system of programs requires this data in the header.

6. "*END*" CARD

This manditory card specifies the end of the control cards.

3.3.3 FIELD DEFINITION CARDS FOR CLASSY AND MAXLABEL

1. "FLDNAM" CARD

The field definition cards(s) delineate the area on the image data tape to be used by CLASSY and MAXLABEL in terms of pixel coordinates (sample, line) for each vertex of the "field" 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. Coordinate paris are in card column 11 through 72 and are enclosed in parentheses with the pairs separated by commas.

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. 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) are accepted for one field.

EXAMPLE: FLDNAM (2,3) , (2,1), (196,3) , (100,50) *
(196,100) , (1,100) , (20,30)

The result of the above cards is that a rectangular area is read from the image data which bounds the given irregularly shaped field defined above. The coordinates of the rectangular area are (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 only the pixels that are within the actual field defined.

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

2. "$END*" CARD

This mandatory card specifies the end of the field definition cards.

3.4 SOFTWARE DESCRIPTION

3.4.1 CLASSY SUBROUTINES MODIFIED

3.4.1.1 Software Component No. 1 (CLINIT)

Linkage

CLINIT is called from CLASSY.

Interface

Interface is accomplished through a calling argument and the following common blocks:

/CLUS/, /MISC/, /STPAR/, /CLUSTR/, /INITL/ and /MXLL/.

Input

KROT - Index to root link.

Output

Common blocks initialized.

Storage Requirements

N/A

3.4.1.2 SOFTWARE COMPONENT NO. 2 (SETUP9)

Purpose

Read control cards for CLASSY.

Linkages

Interface is accomplished through common blocks /INFORM/, /CLUSTR/, /FILE/, /MAP/, /TIMERR/, and /WRTAP/.

Outputs

NOFSKP	/FILE/	Incremental number of files to skip
NOFEAT	/INFORM/	Number of channels
FETVEC	/INFORM/	Vertices array
NPTS	/CLUSTR/	Number of pixels to skip between pixels
NOCYCL	/CLUSTR/	Number of iterations
MAP	/MAP/	Array of iterations to map
MAPCT	/MAP/	Count of maps
LINES	/MAP/	Matrix of lines to map for each iteration
LINECT	/MAP/	Count of lines to map
TIMEMX	/TIMERR/	Maximum time
VARBL	/WRTAP/	Segment number and date

Storage Requirements

Not applicable.

Description

SETUP9 reads the control cards in CLASSY CC and saves the segment, date, tape and file numbers, map description, lines description, maximum time and number of iterations.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.1.3 SOFTWARE COMPONENT NO. 3 (STATIS)

Purpose

STATIS is the control subroutines for the computation and reporting part of CLASSY.

Linkages

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

Interface

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

Inputs

Temporary scrambled pixel file created by READTP.

Outputs

STATIS calls CLDUMP to output the data.

Storage Requirement

Not applicable

Description

STATIS was modified for the Maximum Likelihood Program to cause data to be calculated and saved at the end of each iteration through the complete data set. The subroutine additions (1) call CALRPR to calculate the relative proportions, and (2) set MXLL in common block /MXLL/ to 1 to cause CLPR to write a file record for each active cluster.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.1.4 SOFTWARE COMPONENT NO. 4 (CLPR)

Purpose

Report intermediate data to a print file; save data for post-processor MAXLABEL.

Linkages

CLPR is called from CLDUMP, ADJUST, SEPER and JOIN. CLPM calls LOCK, MORSTR, FREE, MORSTR and MINV.

Interface

Interface is accomplished through a calling sequence and the following common blocks /CLUS/, /MISC/, /STPAR/ and /MXLL/.

Inputs

KL - Index to LINK
IN - Not used
SUM - Data to be displayed
SKEW - Data to be displayed
KURT - Data to be displayed
RELP - Data to be saved on file
VOLRT - Data to be saved on file
DCON - Data to be saved on file
MEANS - Data to be saved on file
COVAR - Data to be saved on file

Outputs

Report on report file.

Storage Requirement

Not applicable.

Description

CLPR reports to all of the mathematical variables on a report file. The variables RELP, VOLRT and DCON and the MEANS array and COVAR matrix are saved on the file for the post-processor MAXLABEL.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.2 CLASSY SUBROUTINE ADDED

3.4.2.1 SOFTWARE COMPONENT NO. 1 (CALRPR)

Purpose

Calculate the relative probability of each pixel belonging to each class.

Linkages

CALRPR is called by STATIS CALRPR calls ISPLIT.

Interface

Interface is accomplished through the /CLUS/, /MISC/, /STPAR/ and /MXLL/ common blocks.

Inputs

KROTIN	Calling sequence	Index to root link
INDEX	/CLUST/	index to cluster data
LSUBS	/CLUST/	chain of sub-cluster
PROP	/CLUST/	
PRIRCM	/CLUST/	

Outputs

RELPRP /MXLL/ Relative proportions array

Storage Requirement

Not applicable.

Description

CALRPR determines the relative proportion for each link from the root link by dividing the PROP of the current link by the product of PRIRCM for the previous link and the relative proportion for the previous link.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.3 IMAGE PROCESSOR AND SUBROUTINES

3.4.3.1 SOFTWARE COMPONENT NO. 1 PROCESSOR (IMAGE)

Purpose

Create an input file for the MAXLABEL processor from a JSC Universal Image Tape.

Linkages

Interface is accomplished through common blocks /INFORM/ and /CLUSTR/.

Outputs

Pixel file for MAXLABEL Processor.

Storage Requirement

Not applicable.

Description

IMAGE calls SETUP9 to read the LABEL control cards to describe the the pixel data needed for the MAXLABEL processor. The subroutine KREDTP calls the LARSYS subroutines to read the JSC Universal Image Tape and moves the pixel data to the pixel file for the MAXLABEL processor.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.3.2 SOFTWARE COMPONENT NO. 2 (SETUPM)

Purpose

Read control cards for IMAGE and MAXLABEL.

Linkages

Interface is accomplished through common blocks /INFORM/. /CLUSTR/ and /WRTAP/.

Outputs

NOFSKP	/FILE/	Incremental number of files to skip
NOFEAT	/INFORM/	Number of channels
FETVEC	/INFORM/	Vertices array
NPTS	/CLUSTER/	Number of pixels to skip between pixels
VARBL	/WRTAP/	Segment number and date

Storage Requirements

Not applicable.

Description

SETUPM reads the control cards in LABEL CC and saves the segment, date, channel information and map option.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.3.3 SOFTWARE COMPONENT NO. 3 (KREDTP)

Purpose

Read lines of data from image tape.

Linkages

KREDTP is called the post-processor IMAGE. This subroutine calls TAPHDR, FLDINT, LINERD and FDLINT.

Interface

Interface is accomplished through common blocks /INFORM/, /CLUSTR/ and /FILE/.

Inputs

NOFSKP	/FILE/	Incremental number of files to skip
NPTS	/INFORM/	Number of pixels to skip between pixels
NOFEAT	/INFORM/	Number of channels
FETVEC	/INFORM/	Vertices array

Outputs

FILE of pixels as described by control cards.

Storage Requirement

Not applicable.

Description

KREDTP is calls TAPHDR to read the tape header, LAREAD to read the field and vertices information, FLDINT to position tape for this field, LINERD to read lines from the universal format tape.

Flowchart

Not applicable.

Listing

See Appendix A for program.

Error Message

FIELD DEFINATION INFORMATION EXCEEDS 2000 WORDS. END-OF-TAPE REACHED BEFORE END OF FIELD.

Data Mesaages

Vertices listed.

3.4.4 MAXLABEL PROCESSOR AND SUBROUTINES

3.4.4.1 SOFTWARE COMPONENT NO. 1 (MAXLABEL)

Purpose

Maxlabel estimates the probability of observing a particular labelled class given that a particular cluster has been observed (BETA). These estimates are used to produce estimates of the proportion of each labelled class present in the scene. The proportion estimates are computed in two different ways: The first assumes that clusters are to be "bias corrected" using the BETA's. The second assumes that clusters are labelled using the BETA s.

Linkages

MAXLABEL calls SETUPM, ALLPXI, BAPLS, PRTBAP, PRTAP, PRTAP2 and WRTLNS.

Interface

Interface is accomplished through calling sequences, blank common and /FILE/ common blocks.

Inputs.

Control card file LABEL CC
Image input file
CLASSY cluster statisical parameters. (segment number) 1, ...,
(segment number) (last iteration)

Outputs

Iteration Report (Estimated BETA's, cluster labels and propotions of each class present.)

Pixel Labelled Dot Map
Cluster Labelled Dot Map
Unlabelled Dot Map
Entropy and Posterior Probability for each Cluster Report
Pixel Labelled Cluster Map
Cluster Labelled Cluster Map
Unlabelled Cluster Map on Tape

Storage Requirement

Not applicable.

Description

MAXLABEL (1) reads the LABEL CC file to determine the segment number, processing date and data description (2) reads the dot data file and (3) reads the CLASSY cluster statistical parameters file. The iterative process then begins to calculate the estimates. The class and cluster labels are saved for maximum estimates.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.2 SOFTWARE COMPONENT NO. 2 (SETUPM)

See 3.4.3.2 for description of SETUPM.

3.4.4.3 SOFTWARE COMPONENT NO. 3 (READCC)

Purpose

Read CLASSY cluster statistic parameters into common block /CLASY/.

Linkage

READCC is called from MAXLABEL.

Interface

Interface is accomplished through blank common and /CLASY/ blocks.

Inputs

File from CLASSY. (Segment number) (Last iteration number).

Outputs

NOCC	/blank common/	Number of CLASSY clusters
RLPRP	/blank common/	Relative proportion array
CCLRT	/blank common/	Normalization factor elements
CDCON	/blank common/	Normalization factor elements

Storage Requirements

Not applicable.

Description

READCC reads the data from CLASSY describing the clusters.

Flowchart

Not applicable.

Listing

See Appendix B for program.

3.4.4.4 SOFTWARE COMPONENT NO. 4 (ALLPXI)

Purpose

Calculate the probability of X (dot data pixel) given I (Clasy Cluster) for each X in the dot data file.

Linkages

ALLPXI is called by the main program MAXLABEL. ALLPRI calls DOTSQK and GETCC.

Interface

Interface is accomplished through blank common.

Inputs

NOCC	/blank common/	Number of CLASSY Cluster
ITOTDT	/blank common/	Total number of dots
MQ	/blank common/	Number of channels
IDOTS	/blank common/	Dot data pixels
CMEANS	/blank common/	CLASSY cluster mean matrix

Outputs

PXI	/blank common/	Probability of pixel belonging to CLASSY cluster.

Storage Requirement

Not applicable.

Description

The array PXI is created by subtracting the CLASSY cluster mean value from the dot data pixel value for each channel. The product of this array and the covariance matrix for the CLASSY cluster is created. The probability is calculated as P(X.I) = E (-PRODUCT/2.) *E (-DCON/2.)/CVOLRT

Where DCON and CVOLRT are the normalization factor elements of the CLASSY cluster.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.5 SOFTWARE COMPONENT NO. 5 (BAPLS)

Purpose

Sum products in a class for all dot-cluster combinations.

Linkages

BAPLS is called by MAXLABEL.

Interface

Interface is accomplished through blank common.

Inputs

NOCC	/blank common/	Number of CLASSY cluster
NOCAT	/blank common/	Number of classes
ITOTDT	/blank common/	Total number of dots
IDOTS	/blank common/	Dot data pixels
BETA	/blank common/	Prior estiamte
RLPRP	/blank common/	Relative porportion for cluster
PX	/blank common/	Probability of pixel belonging to cluster

Outputs

SLK	/blank common/	Sum of products by class and cluster
SK	/blank common/	Sum of products by cluster

Storage Requirement

Not applicable.

Description

For each dot the following calculations are made:

(1) The class of the dot is determined

(2) The products of BETA (class, cluster)*RLPRP (cluster) *PX (cluster, dot) are summed.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.6 SOFTWARE COMPONENT NO. 6 (PRTELB)

Purpose

Construct and print estimate for each labelled class.

Linkages

PRTELB is called by the main program MAXLABEL.

Interface

Interface is accomplished through blank common.

Inputs

BETA /blank common/ Prior estimate

RLRP /blank common/ Relative proportion

Outputs

Report on report file.

Storage Requirement

Not applicable.

Description

PRTELP creates the sum of the products BETA*RLPRP for each class-category combination.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.7 SOFTWARE COMPONENT NO. 7 (PRTBAP)

Purpose

Construct and print labeled cluster map for each class-cluster.

Linkages

PRTBAP is called by the main program MAXLABEL.

Interface

Interface is accomplished through the blank common and /MPPXL/ common blocks.

Inputs

ITOTDT	/blank common/	Total dots
IDOTS	/blank common/	Dot location
NOCAT	/blank common/	Number of categories
BETA	/blank common/	Prior estimate
RLPRP	/blank common/	Relative proportion
PX	/blank common/	Pixel value
LABELS	/blank common/	Category labels

Outputs

MPXLA /MPPXL/ Pixel labels.

Storage Requirement

Not applicable.

Description

The maximum product for each prior estimate * relative proportion * pixel value is determined. A matrix is constructed of the class labels corresponding to the maximum products. This matrix is printed.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.8 SOFTWARE COMPONENT NO. 8 (PRTAP)

Purpose

Construct and print labeled cluster map for each cluster.

Linkages

PRTAP is called by the main program MAXLABEL.

Interface

Interface is accomplished through blank common and /MPPXL/ common blocks.

Inputs

ITOTDT	/blank common/	Total number of dots
RLRPR	/blank common/	Relative proportion for class
PX	/blank common/	Pixel value for class
NOCC	/blank common/	Number of CLASSY cluster
LBLCST	/blank common/	Labels for CLASSY clusters

Outputs

MPXLA /MPPXL/ Class labels

Storage Requirement

Not applicable.

Description

The maximum product for each relative proportion * pixel value is determined. A matrix is constructed of the class labels corresponding to the maximum products. This matrix is printed. The entropy for each class-CLASSY cluster and the probability for each pixel-CLASSY cluster are calculated and printed.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.9 SOFTWARE COMPONENT NO. 9 (PRTAP2)

Purpose

Print labelled cluser map for each cluster.

Linkages

PRTAP2 is called by the main program MAXLABEL.

Interface

Interface is accomplished through blank common and /MXPPXL/ common blocks.

Inputs

ITOTDT	/blank common/	Total number of dots
NOCC	/blank common/	Number of CLASSY clusters
RLPRP	/blank common/	Relative proportion for CLASSY cluster
PX	/blank common/	Pixel value for CLASSY cluster
LBLCSS	/blank common/	Labels for CLASSY cluster categories

Outputs

Line written to report file of CLASSY cluster categories corresponding to pixel.

Storage Requirement

Not applicable.

Description

The maximum product for each relative proportion * pixel value is determined. A matrix is constructed of the cluster labels corresponding to the maximum products. This matrix is printed.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.10 SOFTWARE COMPONENT NO. 10 (WRTLNS)

Purpose

(1) Read radiance values for lines of original pixel data,
(2) Print heading for reports, and
(3) Call subroutines to calculate report data.

Interface

WRTLNS is called by MAXLABEL. WRTLNS calls WRTHED, PXILN, LNBAP, WRTLN, LNAP, LNAP2 PAGE.

Inputs

Temporary file on unit 24.

Outputs

Pixel Labelled Cluster Map
Cluster Labelled Cluster Map
Unlabelled Cluster Map

Storage Requirement

Not applicable.

Description

WRTLNS reads a description of the lines from the temporary file. Report headings are written to files to be saved for each of the reports. The lines of radiance values are then read, the report calculations made and the report lines written to temporary report files. The temporary report files are then sent to the printer.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.11 SOFTWARE COMPONENT NO. 11 (LNBAP)

Purpose

Print labelled cluster map for each class-cluster,
Pixel cluster = Max (Beta * alpha * P(X.I)).

Linkages

Interface is accomplished through /PXLLN/, /MPPXL/ and blank common blocks.

Inputs

IBEGIN	/PXLLN/	First pixel on line
IEND	/PXLLN/	Last pixel on line
BETA	/blank common/	Prior estimate
RLPRP	/blank common/	Relative proportion
PXLN	/PXLLN/	Probability of pixel given cluster

Outputs

Report line written on labelled cluster map file.

Storage Requirement

Not applicable.

Description

Each pixel in the line of data is processed for every category-cluster combination to determine the maximum sum of the class products for the category where the product = the estimate * the relative proportion * the probability that the pixel belongs to the cluster.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.12 SOFTWARE COMPONENT NO. 12 (PXILN)

Purpose

Calculate the probability of X (dot data pixel) given I (CLASSY cluster) for each X in the line in the image file.

Linkages

PXILN is called by WRTLNS. PXILN calls GETCC and DOTSQK.

Interface

Interface is accomplished through blank common and /PXLLN/.

Inputs

LNDOTS	/PXLLN/	
CMEANS	/blank common/	CLASSY mean array
MQ	/blank common/	Number of channels
CVRIN	/blank common/	CLASSY covariance matrix
DCON	/blank common/	Normalization factor element
CVOLRT	/blank common/	Normalization factor element

Outputs

PXLN /PXLLN/ Probability of X given cluster.

Storage Requirements

Not applicable.

Description

PXILN makes the pixel data calcualtions for each CLASSY cluster by calling GETCC to read the data for the cluster and then processing each dot data pixel using that cluster data to calculate the probability of that dot data pixel given that CLASSY cluster.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.13 SOFTWARE COMPONENT NO. 13 (GETCC)

Purpose

Move one set of CLASSY data from arrays to corresponding scalars.

Linkage

GETCC is called from GETCC.

Interface

Interface is accomplished through blank common and /CLASSY/ blocks.

Input

ICC /blank common/ current class

Output

RELPRP	/blank common/	Relative proportion
CVOLRT	/blank common/	Normalization factor element
CMEANS	/blank common/	CLASSY cluster mean array
CVRIN	/blank common/	Covariance matrix
DCON	/blank common/	Normalization factor element

Storage Requirement

Not applicable.

Description

GETCC moves data for this specified class from the arrays of CLASSY data.

Flowchart

Not applicable.

Listing

See Appendix A for program

3.4.4.14 SOFTWARE COMPONENT NO. 14 (DOTSQK)

Purpose

Calculate the inner product DIFXMN. DIFXMN relative to the metric CVRIN.

Linkages

DOTSQK is called from ALLPXI.

Interface

Interface is accomplished through calling arguments.

Inputs

MQ	Number of channels
DIFXMN	Array of differences between pixel values and CLASSY mean.
CVRIN	CLASSY covariance matrix.

Outputs

DOTSQK enter product DIFXMN. DIFXMN relative to the metric CVRIN.

Storage Requirement

Not applicable.

Description

DOTSQK calculates the inner product DIFXMN. DIFXMN relative to the metric CVRIN.

Flowchart

Not applicable.

Listing

See Appendix A for program.

3.4.4.15 SOFTWARE COMPONENT NO. 15 (PAGE)

Purpose

Read data from temporary file and write report heading and column headings.

Linkages

PAGE is called by WRTLNS.

Interface

Interface is accomplished through a temporary file.

Inputs

IUNIT temporary file unit.

Outputs

Report heading written on report file.

Storage Requirement

Not applicable.

Description

PAGE reads the temporary file and writes the data on the report file.

Flowchart

Not applicable.

Listing

See Appendix A for program.

4. OPERATION

CLASSY, DOTDATA, IMAGE and LABEL are operational on the IBM 3031 computer at LARS, West Lafayette, Indiana.

The programs, EXEC files and CC files can be loaded from tape 3956, file 5.

CLASSY is executed by entering the following commands after signing on the computer system.

```
DEF STOR 2M
TAPMOUNT (Tape Number) TAP1 RO 1600
CLASSY or CLASSYN (Segment Number)
```

Control inputs is read from (Segment Number) CC.

Text output in on the terminal and line printer which are assigned in the EXEC, and the One Channel Unlabelled Tape File is on the output tape designated in the execute statement.

DOTDATA is executed by entering the following commands:

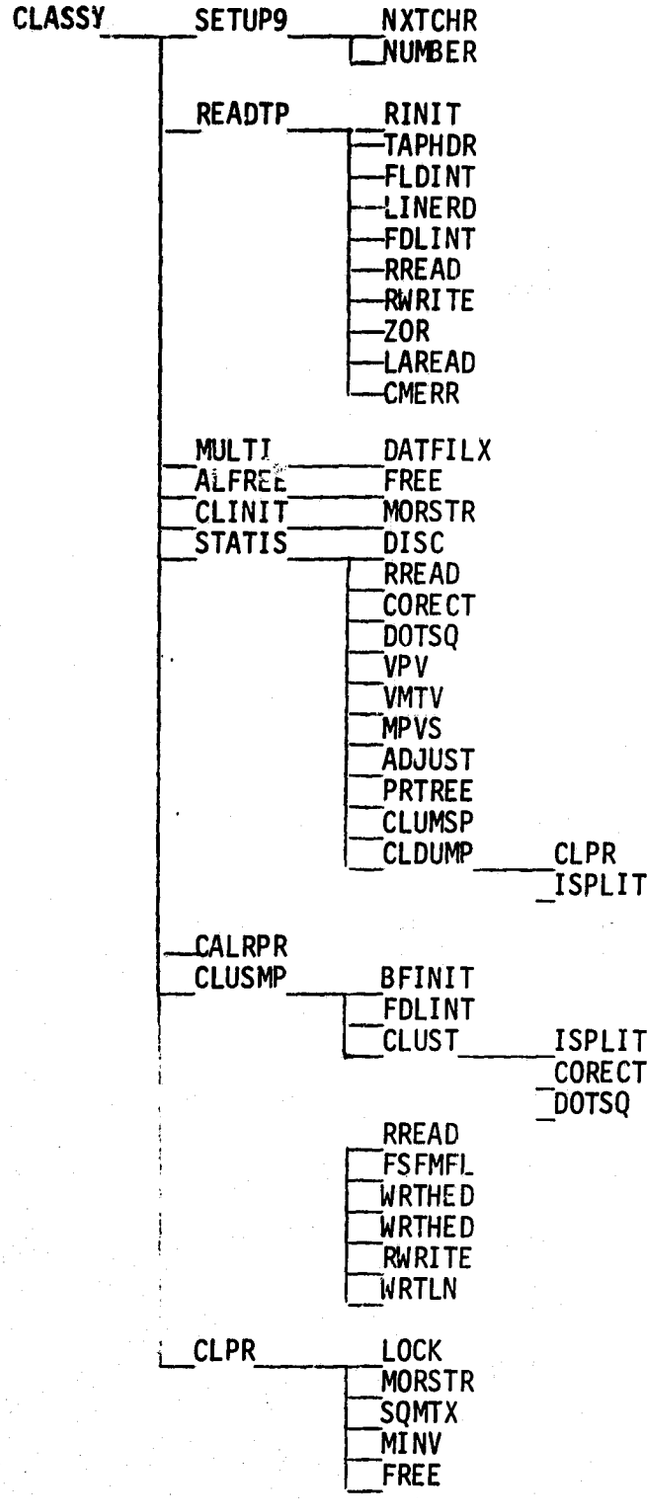
```
TAPE REW
DOT (Segment Number)
```

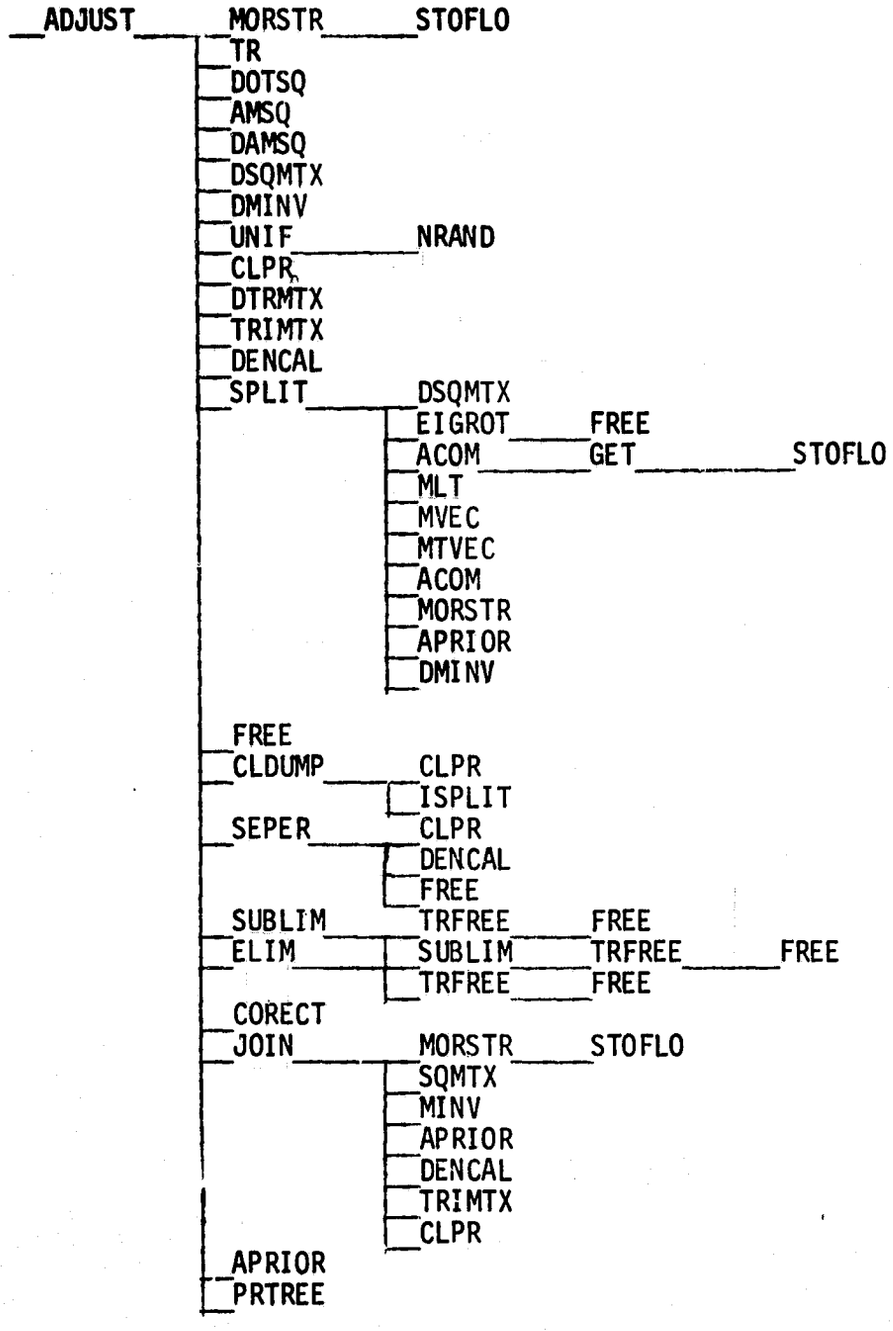
Control input is read from DOT CC.
Text output is on the terminal and line printer, and the Ground Truth Data File is on (Segment Number) DOT1 and (Segment Number) DOT2.

IMAGE and LABEL are executed by entering the following commands:

```
TAPE REW
LABEL (Segment Number)
```

Control input is read from LABEL CC.
Text output is on the terminal and line printer, and the One Channel Labelled Tape File is on the tape designated in the execute statement.


CLASSY
SETUP9
NXTCHR
NUMBER
READTP
RINIT
TAPHDR
FLDINT
LINERD
FDLINT
RREAD
RWRITE
ZOR
LAREAD
CMERR
MULTI
DATFILX
ALFREE
FREE
CLINIT
MORSTR
STATIS
DISC
RREAD
CORECT
DOTSQ
VPV
VMTV
MPVS
ADJUST
PRTREE
CLUMSP
CLDUMP
CLPR
ISPLIT
CALRPR
CLUSMP
BFINIT
FDLINT
CLUST
ISPLIT
CORECT
DOTSQ
RREAD
FSFMFL
WRTHED
WRTHED
RWRITE
WRTLN
CLPR
LOCK
MORSTR
SQMTX
MINV
FREE


ADJUST
MORSTR
STOFLO
TR
DOTSQ
AMSQ
DAMSQ
DSQMTX
DMINV
UNIF
NRAND
CLPR
DTRMTX
TRIMTX
DENCAL
SPLIT
DSQMTX
EIGROT
FREE
ACOM
GET
STOFLO
MLT
MVEC
MTVEC
ACOM
MORSTR
APRIOR
DMINV
FREE
CLDUMP
CLPR
ISPLIT
SEPER
CLPR
DENCAL
FREE
SUBLIM
TRFREE
FREE
ELIM
SUBLIM
TRFREE
FREE
TRFREE
FREE
CORECT
JOIN
MORSTR
STOFLO
SQMTX
MINV
APRIOR
DENCAL
TRIMTX
CLPR
APRIOR
PRTREE

APPENDIX A

LISTINGS OF MODIFIED CLASSY SUBPROGRAMS

LISTINGS OF IMAGE AND MAXLABEL PROGRAMS

```
      SUBROUTINE CLINIT(KROOT)                                          CLI00010
C     THIS ROUTINE CONTAINS THE VARIOUS STATEMENTS NECESSARY TO         CLI00020
C       INITIALIZE THE CLUSTERING ALGORITHM.                            CLI00030
      REAL*8 XTEMP,YTEMP,ZTEMP,DUMK,DUMKO                               CLI00040
      DIMENSION INDEX(27),LSUBS(30),LSUPER(29),IDADJ(24),NSYMB(12),     CLI00050
     1 PCUM(26),PRIRCM(25),CIN(24),CTOT(23),PROP(22),SPEAC(21),         CLI00060
     2 WADJ(20),W(19),OPROP(18),OW(17),VOLIN(16),VOLRT(15),DCON(14),    CLI00070
     3 PORAT(13),DISS(12),PASS(12),PST(11),DCIN(10),PCOND(7),           CLI00080
     4 OPRTOR(9),ODEN(8)                                                CLI00090
      DIMENSION VRIN(475),GEN(999),GREF(999),ALINK(1)                   CLI00100
      EQUIVALENCE (LINK(1),ALINK(1)),(LINK(31),INDEX(27))               CLI00110
      EQUIVALENCE (LINK(3),LSUBS(30))                                   CLI00120
      EQUIVALENCE(LINK(31),LSUPER(29)),  (LINK(31),IDADJ(24)),          CLI00130
     1 (LINK(31),NSYMB(12)),  (LINK(31),PCUM(26)),  (LINK(31),PRIRCM(25)),CLI00140
     2 (LINK(31),CIN(24)),  (LINK(31),CTOT(23)),  (LINK(31),PROP(22)),  CLI00150
     3 (LINK(31),SPEAC(21)),  (LINK(31),WADJ(20)),  (LINK(31),W(19)),   CLI00160
     4 (LINK(31),OPROP(18)),  (LINK(31),OW(17)),  (LINK(31),VOLIN(16)), CLI00170
     5 (LINK(31),VOLRT(15)),  (LINK(31),DCON(14)),  (LINK(31),PORAT(13)),CLI00180
     6 (LINK(31),DISS(12)),  (LINK(31),PASS(12)),  (LINK(31),PST(11)),  CLI00190
     7 (LINK(31),DCIN(10)),  (LINK(31),PCOND(7)),  (LINK(31),VRIN(7)),  CLI00200
     8 (LINK(31),GEN(7)),  (LINK(31),OPRTOR(9)),  (LINK(31),ODEN(8)),   CLI00210
     9 (LINK(31),GREF(4))                                               CLI00220
      COMMON/CLUS/  JUNK(12),NAPL,NTOP,NTHSZM,NWAUT,LINK(14000)         CLI00230
      DIMENSION MXAR(31),LR(3),LV(3)                                    CLI00240
      EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT),                          CLI00250
     1    (LR(3),LOVAR),(LV(1),LSUM),(LV(2),LSKEW),(LV(3),LOSUM)        CLI00260
C                                                                       CLI00270
      COMMON /MISC/ MO,MM,LR,LV,NTNCLS,MXAR,NTINIT,NROOT,EPS,DELT,      CLI00280
     1    AMO,ODCON,XOVFLO,XUNFLO,WADJIN,ELIMTH,SEPTH,VFAC,AMM,SRLTH,   CLI00290
     2    INDXVL,WFAC,NPTSO,PORATH,SPMVTH,OWFAC,GRACTH,AMOFAC,          CLI00300
     3     AMOMIN,AMOMAX,AMORAT,VOLLIM,BIAS,PJOIN,VPJOIN,WSIM,WDELSM,   CLI00310
     4    BETTER,MODE,CORLEN,SPCOR                                      CLI00320
C                                                                       CLI00330
      COMMON /STRAP/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI,   CLI00340
     1     PACCEL(2),MACCEL(2),VACCEL(2)                                CLI00350
C                                                                       CLI00360
      COMMON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD,  SYM(61)  ,        CLI00370
     1                  LMCAT, PRNT(4)  , KLHC , PRTMF, PROUT, TOTPIX,  CLI00380
     2  SCRAM1,BUFPIX,BUFTOT,NBUFSD,JOUMP,LBUF)                         CLI00390
     3, MAXRF, AREA, NWDS, NWURS, NPTS, LBUF, IO1,NOCYCL                CLI00400
C                                                                       CLI00410
      INTEGER TOTWRD,SYM,PRNT,PRTMF,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT   CLI00420
     1 ,CLSNAM                                                          CLI00430
C                                                                       CLI00440
      COMMON /MXLL/ MXLLWT, MXLLEN, RELPRP(200)                         CLI00450
C                                                                       CLI00460
      COMMON /INITL/WTNEW,DEVINI,CHANIN                                 CLI00470
      CHIVAL(DF)=DF*(1.-.222/DF+CONLV*SQRT(.222/DF))**3                 CLI00480
C                                                                       CLI00490
      AMO=MO                                                            CLI00500
C                                                                       CLI00510
C  DEFINE VALUE OF SEPTH IN TERMS OF CHI SQUARE VALUE                   CLI00520
      DFT = AMO * 1                                                     CLI00530
      SEPTH = (CHIVAL(DFT))/2                                           CLI00540
C                                                                       CLI00550
C  WE FIRST SET UP VARIOUS INDEX ARRAYS FOR A PARTICULAR                CLI00560
C       NUMBER OF CHANNELS MO.                                          CLI00570
C       SET UP THE TRIANGULAR POSITION ARRAY MXAR.                      CLI00580
      MM=0                                                              CLI00590
      DO 10 I=1,31                                                      CLI00600
      MXAR(I)=MM                                                        CLI00610
   10 MM=MM+I                                                           CLI00620
      MM=MXAR(MO+1)                                                     CLI00630
      ANM=MM                                                            CLI00640
C NOW WE SET UP THE ORIGIN VECTORS, LR AND LV, OF THE VARIOUS ARRAYS    CLI00650
C     AND VECTORS IN A CLUSTER NODE.                                    CLI00660
      NTNCLS=1                                                          CLI00670
C ***** THIS CONSTANT MUST BE SET TO THE NUMBER OF ARRAYS *****        CLI00680
      DO 21 I=1,3                                                       CLI00690
      LR(I)=NTNCLS                                                      CLI00700
   21 NTNCLS=NTNCLS+MM                                                  CLI00710
      DO 22 I=1,3                                                       CLI00720
      LV(I)=NTNCLS                                                      CLI00730
   22 NTNCLS=NTNCLS+MO                                                  CLI00740
      NSCALS = 25                                                       CLI00750
      NTNCLS=NTNCLS+NSCALS-1                                            CLI00760
C  WE MUST ALSO SET UP SOME THRESHOLDS FOR USE BY THE STATISTICAL       CLI00770
C     SYSTEM.                                                           CLI00780
      SKCHI=(AMO+2.)*(AMO+4.)*CHIVAL(AMO)                               CLI00790
```

```
      URKCHI=AMO*(AMO+4.)*(AMO+6.)/(AMO-.999)*CHIVAL(AMM-1.)                    CLI00800
      TRCHI=CONLV*CONLV*(AMO*(AMO+2.)*(AMO+3.)*M.)                              CLI00810
C   WE NOW CREATE THE HEAD NODE OF THE CLUSTER TREE.  THIS IS NOT               CLI00820
C     AN ACTUAL CLUSTER, AND DOES NOT HAVE STORAGE FOR ANY                      CLI00830
C     OF THE STATISTICAL ARRAYS.                                                CLI00840
      NPTSQ=0                                                                   CLI00850
      KROOT=NOUSED(NSCALS)                                                      CLI00860
C    MAKE FIRST NODE START AT AN ODD NUMBER                                     CLI00870
        IF (MOD(NTOP,2) .NE. 1) NTOP = NTOP + 1                                 CLI00880
      LINK(KROOT)=-262139                                                       CLI00890
      LSUPER(KROOT)=-262142                                                     CLI00900
      IDADJ(KROOT)=999999                                                       CLI00910
      INDEX(KROOT)=-0                                                           CLI00920
      SPFAC(KROOT)=99999.                                                       CLI00930
      W(KROOT)=WTINIT                                                           CLI00940
      OW(KROOT)=-(KROOT)                                                        CLI00950
      PORAT(KROOT)=0.                                                           CLI00960
      PROP(KROOT)=1.                                                            CLI00970
      OPROP(KROOT)=1.                                                           CLI00980
      CIN(KROOT)=W(KROOT)                                                       CLI00990
      OCIN(KROOT)=CIN(KROOT)                                                    CLI01000
      CTOT(KROOT)=0.                                                            CLI01010
      ODEN(KROOT)=W(KROOT)                                                      CLI01020
      PRIPCM(KROOT)=1.                                                          CLI01030
C   NEXT THE INITIAL NODE IS SET UP, TOGETHER WITH SOME CONTROL THRESHOLDS      CLI01040
   57 KFIR=NOUSED(NINCLS)                                                       CLI01050
      DO 54 J=1,M2                                                              CLI01060
      GREF(KFIR+LOVAR+J)=0.                                                     CLI01070
      GREF(KFIR+LKROT+J)=0.                                                     CLI01080
   54 VRIN(KFIR+J)=0.                                                           CLI01090
      DEV2WT=DEVINI*DEVINI*WTINIT                                               CLI01100
      DO 53 J=1,MQ                                                              CLI01110
      GREF(KFIR+LSUM+J)=WTINIT*CHANIN                                           CLI01120
      GREF(KFIR+LOSUM+J)=WTINIT*CHANIN                                          CLI01130
      KLJ=KFIR+MXAR(J+1)                                                        CLI01140
      VRIN(KLJ)=1./DEV2WT                                                       CLI01150
      GREF(KLJ+LOVAR)=DEV2WT                                                    CLI01160
      GREF(KLJ+LKROT)=(MO+2)*DEV2WT                                             CLI01170
   53 GREF(KFIR+LSKEW+J)=0.                                                     CLI01180
      VOLRT(KFIR)=.935762296E-13*(2.506628275*DEVINI)**MQ                       CLI01190
      VOLIN(KFIR)=VOLRT(KFIR)**2                                                CLI01200
C     VOLIN*FAP(DCON)=(2*PI)**MQ*DET(COVARIANCE)=(2*PI/W)**MQ/DET(VRIN)         CLI01210
      ODCON=MQ*ALOG(WTINIT)+60.                                                 CLI01220
      DCON(KFIR)=ODCON                                                          CLI01230
      W(KFIR)=WTINIT                                                            CLI01240
      OW(KFIR)=WTINIT                                                           CLI01250
      CIN(KFIR)=W(KFIR)                                                         CLI01260
      OCIN(KFIR)=CIN(KFIR)                                                      CLI01270
      IDADJ(KFIR)=IDADJIN                                                       CLI01280
      SPFAC(KFIR)=-9999.                                                        CLI01290
      PORAT(KFIR)=0.                                                            CLI01300
      CTOT(KFIR)=0.                                                             CLI01310
      ODEN(KFIR)=W(KFIR)                                                        CLI01320
      PROP(KFIR)=1.                                                             CLI01330
      OPROP(KFIR)=1.                                                            CLI01340
      PRIPCM(KFIR)=1.                                                           CLI01350
      LINK(KFIR)=0                                                              CLI01360
      LSUBS(KFIR)=0                                                             CLI01370
      LSUPER(KFIR)=KROT                                                         CLI01380
      LSUBS(KROOT)=KFIR                                                         CLI01390
      TOTPIX = TOTPRO/MQ                                                        CLI01400
      INDEX(KFIR)=INDXVL                                                        CLI01410
      IDADJ(KFIR)=TOTPIX                                                        CLI01420
C                                                                               CLI01430
C  SET SWITCH FOR MAX LIKLIHOOD LABELING                                        CLI01440
      MXLLWT = 0                                                                CLI01450
      MXLLEN = 23                                                               CLI01460
      PRINT 273,MQ,CONLV,TPCHI,SKCHI,URKCHI,KROOT,KFIR                          CLI01470
  273 FORMAT ('1 CONFIDENCE LEVELS',I4,' CHANNELS',F8.4,'  CHISQUARES',        CLI01480
     1    3E11.5/'   ROOT',I5,'   FIRST',I5)                                   CLI01490
      RETURN                                                                    CLI01500
      END                                                                       CLI01510
```

```
      SUBROUTINE SETUP9                                                 SET00010
C*****************************************************************    SET00020
C                                                                       SET00030
C  THE PURPOSE OF SUBROUTINE SETUP9 IS TO READ AND ANALYZE ALL CARD     SET00040
C  INPUT TO THE PROGRAM                                                 SET00050
C                                                                       SET00060
C*****************************************************************    SET00070
C                                                                       SET00080
      IMPLICIT INTEGER (A-X)                                            SET00090
C                                                                       SET00100
      COMMON /TIMERP/ ORGTIM, TIMEMX                                    SET00110
      REAL ORGTIM, TIMEMX                                               SET00120
C                                                                       SET00130
C                                                                       SET00140
      COMMON /MAP/ MAP(10),LINES(10,2),MAPCT,LINECT,KA                  SET00150
      COMMON /INFORM/HEAD(42), MAPTAP,     DATAPE,     SAVTAP,   MAXFET,SET00160
     1                 PAGSIZ, TAPCHK,     TRNSYM,     TSTSYM,        SET00170
     2                 DUPSYM, TRUSYM,     MAXDIV,     MEMDIV,   SPLMAX,SET00180
     3                 SERIAL, TAPESV,     FILESV,                    SET00190
     4                 MAXCLS, NOCLS2,     MAXFLD,     NOFLD2,   NOFLD3,SET00200
     5                 NOTREC, NOFEAT,     NOFET2,     NOFET4,   VARSIZ,SET00210
     6                 VARSZ2, VARSZ4,     XSIZ  ,     NOSPEC,   NOHIST,SET00220
     *                 NOGRP, DIVSIZ,    KEEPLV,    PRTLEV,   YSIZ  ,  SET00230
     8                 XHGH  , XLOW  ,     SPCMAS,     NOCLS4,   PCTSZ ,SET00240
     9IPLOCK(30),FETVEC(30),FETVC2(30), HISVEC(30),INVERT(30),RESTVC(30)SET00250
      DIMENSION  HED1(10),  HED2(10),  DATE(2),  COMENT(10),  TEMP(10)  SET00260
      EQUIVALENCE  (HED1(1),HEAD(3)),     (DATE(1),  HEAD(15)),        SET00270
     *             (HED2(1),HEAD(20)),    (COMENT(1),HEAD(32))         SET00280
      COMMON /SUPCOM/ INTAPE,STATUS,COL,CODE1,CODE2,CARD(62),ACARD(20)  SET00290
C                                                                       SET00300
      COMMON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYM(61) ,          SET00310
     1              LNCAT, PRNT(4) , KLHC , PRTME, PROUT, TOTPIX,       SET00320
     2  SCRAM1,BUFPIX,BUFTOT,NBUFSD,NDUMP,LBUF)                        SET00330
     3, MAXRF, AREA, NWDS, NWDRS, NPTS, LBUF, IQI,NOCYCL                SET00340
C                                                                       SET00350
      INTEGER TOTWRD,SYM,PRNT,PRTME,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT   SET00360
     1 ,CLSNAM                                                          SET00370
C                                                                       SET00380
C                                                                       SET00390
      INTEGER SEG                                                       SET00400
      COMMON/PRTAP/ICOUNT,FORMT,UNIT,VARBL(600),IREMD                   SET00410
      EQUIVALENCE (VARBL(23),SEG)                                       SET00420
      INTEGER PROCD(3)                                                  SET00430
      EQUIVALENCE (VARBL(20),PROCD(1)),(VARBL(21),PROCD(2)),            SET00440
     1(VARBL(22),PROCD(3))                                              SET00450
      COMMON/FILE/DATFIL                                                SET00460
C                                                                       SET00470
C                                                                       SET00480
      DIMENSION SMPLS(61)                                               SET00490
      DATA  SMPLS/'0','=','/','A','+','1','V','-','<','X','*','0','H', SET00500
     *           'F','>','*','4','S','6','K','L','U','N','9','P','D',  SET00510
     2           'S','T','U','V','W','X','Y','Z','-','/',              SET00520
     3           '-','*','+','+','0','=',':',' ','=','$',')','(','$',  SET00530
     4           '$','>','$',':','?','1H',',','.','.',' ','0' /         SET00540
      DATA BLANK/' '/, KOMMA/','/                                       SET00550
      DIMENSION IDVEC(17)                                               SET00560
      DATA IDVEC/'CHAN','HED1','HED2','DATE','COMM','SEGM','PROC',      SET00570
     * 'NPTS','NPOS','SYM ','PRIN','DATA','ITER','MAP ',                SET00580
     * 'LINE','TIME','END'/                                            SET00590
      DIMENSION EQUVEC(2)                                               SET00600
      DATA KKA/'A'/,KHYPHN/'-'/                                         SET00610
      DATA URCD/'U'/,FRCD/'F'/,EQUVEC/1,'='/                            SET00620
      VARBL(19) = 242                                                   SET00630
      ICNT=0                                                            SET00640
      K0 = 0                                                            SET00650
      KA = KKA                                                          SET00660
      ZERO = 0                                                          SET00670
C  ASSUME VALID FOR MAP AND LINES CARDS                                 SET00680
      LINECT = 1                                                        SET00690
      MAPCT = 1                                                         SET00700
      LINES(1,1) = KA                                                   SET00710
      MAP(1) = KA                                                       SET00720
      DO 5 I=1,61                                                       SET00730
    5 SYM(I)=SMPLS(I)                                                   SET00740
      NOFEAT=0                                                          SET00750
      WRITE(6,600)                                                      SET00760
  600 FORMAT('1',8X,'C L A S S Y   C L U S T E R I N G   P R O G R A M' SET00770
     *   /1X,'VERSION 1.0   FEB 1979   LOCKHEED ELECTRONICS ')//)       SET00780
      WRITE(6,630)                                                      SET00790
```

```
C                                                                       SET00400
CC       PUT THE NEXT CARD IN THE REREAD BUFFER                         SET00410
C                                                                       SET00420
      RPUNIT=30                                                         SET00430
   10 READ(2,1000)(ACARD(I),I=1,20)                                     SET00440
 1000 FORMAT(20A4)                                                      SET00450
      WRITE(RPUNIT,1000)(ACARD(I),I=1,20)                               SET00460
      REWIND RPUNIT                                                     SET00470
C                                                                       SET00480
CCC                                                                     SET00490
CC    READ IN CARD                                                      SET00500
C                                                                       SET00510
      READ(30,490) CODE,CARD                                            SET00520
      REWIND RPUNIT                                                     SET00530
      WRITE(6,550) CODE,CARD                                            SET00540
      COL = 0                                                           SET00550
C  SET NUMBER OF VALID CARD TYPES                                       SET00560
      CNUM = 17                                                         SET00570
C  DETERMINE CARD TYPE                                                  SET00580
      DO 20 I=1,CNUM                                                    SET00590
      IF(CODE .EQ. INVEC(I))GO TO(30,50,70,90,110,120,130,140,150,170,  SET01000
     * 190,260,270,300,350,400,410),I                                   SET01010
   20 CONTINUE                                                          SET01020
C                                                                       SET01030
C  INVALID CARD TYPE                                                    SET01040
      WRITE(6,490)CODE,CARD                                             SET01050
      GO TO 10                                                          SET01060
C                                                                       SET01070
CCC   CHANNEL CARD                                                      SET01080
C                                                                       SET01090
   30 J = NXTCHR(CARD,COL)                                              SET01100
      IF (J .EQ. BLANK)GO TO 10                                         SET01110
      COL=COL-1                                                         SET01120
      NOFEAT = NUMBER(CARD,COL,FETVEC,NOFEAT)                           SET01130
      VARSIZ=(NOFEAT*(NOFEAT+1))/2                                      SET01140
      GO TO 10                                                          SET01150
C                                                                       SET01160
CCC   HED1 CARD                                                         SET01170
C                                                                       SET01180
   50 READ (30,500)HED1                                                 SET01190
      GO TO 10                                                          SET01200
C                                                                       SET01210
CCC   HED2 CARD                                                         SET01220
C                                                                       SET01230
   70 READ (30,500) HED2                                                SET01240
      REWIND RPUNIT                                                     SET01250
      GO TO 10                                                          SET01260
C                                                                       SET01270
CC    DATE CARD                                                         SET01280
C                                                                       SET01290
   90 READ(30,510)DATE                                                  SET01300
      REWIND RPUNIT                                                     SET01310
      GO TO 10                                                          SET01320
C                                                                       SET01330
CC    COMMENT CARD                                                      SET01340
C                                                                       SET01350
  110 READ(30,500)COMENT                                                SET01360
      REWIND RPUNIT                                                     SET01370
      GO TO 10                                                          SET01380
C                                                                       SET01390
C SEGM CARD                                                             SET01400
C                                                                       SET01410
120      J=NXTCHR(CARD,COL)                                             SET01420
      IF(J .EQ. BLANK) GO TO 10                                         SET01430
      COL=COL-1                                                         SET01440
      J=NUMBER(CARD,COL,SEG,0)                                          SET01450
      GO TO 10                                                          SET01460
C                                                                       SET01470
C PROCESSING DATE CARD                                                  SET01480
C                                                                       SET01490
130   J=NXTCHR(CARD,COL)                                                SET01500
      IF(J .EQ. BLANK) GO TO 10                                         SET01510
      COL=COL-1                                                         SET01520
      J=NUMBER(CARD,COL,PROCD,0)                                        SET01530
      GO TO 10                                                          SET01540
C                                                                       SET01550
C  NPTS = NUMBER OF POINTS TO SKIP IN ANALYSIS                          SET01560
C                                                                       SET01570
140   J = NUMBER(CARD,COL,NPTS,0)                                       SET01580
```

```
      NPTS = NPTS - 1                                                    SET01590
      GO TO 10                                                           SET01600
C                                                                        SET01610
C  NPOS CARD--NUMBER OF DRUM POSITIONS FROM WHICH TO OBTAIN DATA FOR     SET01620
C  CLASY3,SO THAT THE DATA WILL BE SCRAMBLED                             SET01630
C                                                                        SET01640
  150 J=NUMBER(CARD,COL,NPOS,K0)                                         SET01650
      GO TO 10                                                           SET01660
C                                                                        SET01670
C  SYMBOL CARD                                                           SET01680
C                                                                        SET01690
  170 ICNT=ICNT+1                                                        SET01700
      IF(ICNT .GT. 61) GO TO 10                                          SET01710
  180 M=NXTCHR(CARD,COL)                                                 SET01720
      IF( M .EQ. BLANK)GO TO 10                                          SET01730
      IF(M .EQ. KOMMA)GO TO 180                                          SET01740
      SYM(ICNT)=M                                                        SET01750
      GO TO 170                                                          SET01760
C                                                                        SET01770
C  PRINT OPTION CARD NO LONGER VALID                                     SET01780
C                                                                        SET01790
190   GO TO 10                                                           SET01800
C.                                                                       SET01810
CCC                                                                      SET01820
CCC     DATA FILE CARD                                                   SET01830
C                                                                        SET01840
  260 M = NXTCHR(CARD,COL)                                               SET01850
      IF(M.EQ.BLANK) GO TO 10                                            SET01860
      IF (M.EQ.ZERO) GO TO 265                                           SET01870
      IF(M.EQ.ERROR) GO TO 267                                           SET01880
  263 WRITE(6,750)                                                       SET01890
  750 FORMAT(' ERROR ON DATA FILE CARD')                                 SET01900
      GO TO 10                                                           SET01910
  265 J = FIND12(CARD,COL,EQUIVEC)                                       SET01920
      IF (J.EQ. -1)  GO TO 263                                           SET01930
      M = NUMBER(CARD,COL,DATAPE,ZERO)                                   SET01940
      COL = COL - 1                                                      SET01950
      GO TO 260                                                          SET01960
A-6 267 J = FIND12(CARD,COL,EQUIVEC)                                     SET01970
      IF (J .EQ. -1)  GO TO 263                                          SET01980
      M = NUMBER(CARD,COL,DATFIL,ZERO)                                   SET01990
      DATFIL = DATFIL - 1                                                SET02000
      IF (DATFIL .LT. 0)  DATFIL = 0                                     SET02010
      COL = COL - 1                                                      SET02020
      GO TO 260                                                          SET02030
C  ITERATION CARD                                                        SET02040
  270 J = NXTCHR (CARD,COL)                                              SET02050
      IF (J .EQ. BLANK) GO TO 10                                         SET02060
      COL = COL - 1                                                      SET02070
      K = NUMBER (CARD,COL,TEMP,K0)                                      SET02080
      NOCYCL = TEMP(1)                                                   SET02090
      GO TO 10                                                           SET02100
C                                                                        SET02110
C  MAP CARD--MAY SPECIFY SERIES OF ITERATIONS TO BE MAPPED OR 'ALL'      SET02120
C    EXAMPLES:  1,3,5,7,9          'ALL'                                 SET02130
300   DO 301 I = 1,10                                                   SET02140
301   MAP(I) = 0                                                         SET02150
      J = NXTCHR (CARD,COL)                                              SET02160
      IF ( J .EQ. BLANK) GO TO 10                                        SET02170
C  IF 'ALL' WAS REQUESTED, GO READ NEXT CARD                             SET02180
      IF (J .EQ. K0) GO TO 10                                            SET02190
C                                                                        SET02200
C  SERIES OF NUMBERS REQUESTED                                           SET02210
      COL = 0                                                            SET02220
      MAPCT = NUMBER (CARD, COL, MAP, K0)                                SET02230
      GO TO 10                                                           SET02240
C                                                                        SET02250
CCC  LINES CARD--LINES TO BE PRINTED.  NUMBERS OR 'ALL' MAY BE SPECIFIED SET02260
C    EXAMPLES:  'ALL'      1,26,51      1-4,26-29,51-54                   SET02270
350   DO 351 I = 1,10                                                   SET02280
      DO 351 J = 1,2                                                     SET02290
351   LINES(I,J) = 0                                                     SET02300
C                                                                        SET02310
C  GET FIRST NONBLANK CHARACTER                                          SET02320
      J = NXTCHR (CARD,COL)                                              SET02330
      IF (J .EQ. BLANK) GO TO 10                                         SET02340
      IF ( J .EQ. K0) GO TO 10                                           SET02350
C                                                                        SET02360
                                                                         SET02370
```

```
C  SERIES OF NUMBERS SPECIFIED                                                  SET02340
355    COL = COL - 1                                                           SET02350
C  UP TO 10 NUMBERS OR SERIES MAY BE SPECIFIED                                 SET02400
       NEXT = 1                                                                SET02410
C  MOVE SERIES OF NUMBERS TO ARRAY 'TEMP', STORE COUNT IN KOUNT                SET02420
       K = 1                                                                   SET02430
       KOUNT = NUMBER (CARD, COL, TEMP,K0)                                     SET02440
       N = 1                                                                   SET02450
C  NUMBER OR SERIES OF NUMBERS SEPARATED BY COMMAS                             SET02460
361    CONTINUE                                                                SET02470
       DO 365 II = N,KOUNT                                                     SET02480
       LINES(NEXT,1) = TEMP(II)                                                SET02490
       LINES(NEXT,2) = TEMP(II)                                                SET02500
365    NEXT = NEXT + 1                                                         SET02510
C  SERIES COULD BE TERMINATED BY BLANK OR HYPHEN                               SET02520
       COL = COL + 1                                                           SET02530
       J = NXTCHR (CARD,COL)                                                   SET02540
       IF (J .EQ. BLANK) GO TO 390                                             SET02550
C  SERIES TERMINATED BY HYPHEN                                                 SET02560
       NEXT = NEXT - 1                                                         SET02570
       KOUNT = NUMBER (CARD, COL, TEMP, K0)                                    SET02580
       LINES(NEXT,2) = TEMP(1)                                                 SET02590
       NEXT = NEXT + 1                                                         SET02600
       IF (KOUNT .EQ. 1) GO TO 390                                             SET02610
C  SERIES OF FORM 1,25-28,51   GO TO PROCESS NEXT NUMBER AFTER HYPHEN ASSIGNED SET02620
C  BEGINNING OF NEXT GROUP OF LINES                                            SET02630
       N = 2                                                                   SET02640
       GO TO 361                                                               SET02650
C                                                                              SET02660
390    LINECT = NEXT - 1                                                       SET02670
       GO TO 10                                                                SET02680
C                                                                              SET02690
C                                                                              SET02700
C  SET MAXIMUM TIME                                                            SET02710
400    J = NUMBER (CARD, COL, ITIME, 0)                                        SET02720
       TIMEMX = ITIME                                                          SET02730
        GO TO 10                                                               SET02740
C                                                                              SET02750
C    *END* CARD                                                                SET02760
C                                                                              SET02770
  410 RETURN                                                                   SET02780
C                                                                              SET02790
C  FORMATS                                                                     SET02800
  480 FORMAT(A4,4X,62A1)                                                       SET02810
  490 FORMAT(' INVALID INPUT CARD--IGNORED'/T5,A4,4X,62A1)                     SET02820
  500 FORMAT(10X,10A6)                                                         SET02830
  510 FORMAT(10X,2A6)                                                          SET02840
  550 FORMAT(5X,A6,4X,62A1)                                                    SET02850
  630 FORMAT(//' INPUT SUMMARY'//)                                             SET02860
       STOP                                                                    SET02870
       END                                                                     SET02880
```

```
      SUBROUTINE STATIS(KROUTN,PV,SUM,SKEW,KURT,OSUM,OVAR)                  STA00010
C                                                                           STA00020
C  PURPOSE                                                                  STA00030
C    (1)  TAKE EACH INPUT POINT AND CLASSIFY IT (ON A FRACTIONAL,           STA00040
C           PROBABILISTIC BASIS.)                                           STA00050
C    (2)  UPDATES THE VARIOUS STATISTICAL PARAMETERS ASSOCIATED WITH THE    STA00060
C           CLASSES INDICATED.                                              STA00070
C    (3)  CALLS ADJUST TO CHECK TO SEE IF ANY OF THESE CLASSES ARE          STA00080
C           POTENTIALLY TWO AND REFER THOSE TO THE ROUTINE ' SPLIT'         STA00090
C                                                                           STA00100
      INTEGER   BUFSIZ , BUFCNT                                             STA00110
C                                                                           STA00120
C  THIS PROGRAM TAKES EACH INPUT POINT AND CLASSIFIES IT                    STA00130
C       (ON A FRACTIONAL, PROBABILISTIC BASIS).  IT THEN                    STA00140
C       UPDATES THE VARIOUS STATISTICAL PARAMETERS ASSOCIATED               STA00150
C       WITH THE CLASSES INDICATED AND CHECKS TO SEE IF                     STA00160
C       ANY OF THESE CLASSES IS POTENTIALLY TWO.  THOSE WHICH               STA00170
C       ARE ARE REFERRED TO THE ROUTINE ' SPLIT'.                           STA00180
      DIMENSION INDEX(27),LSUBS(30),LSUPER(29),IDADJ(28),NSYMB(12),         STA00190
     1 PCUM(26),PRIRCM(25),CIN(24),CTOT(23),PROP(22),SPFAC(21),             STA00200
     2 WADJ(20),W(19),OPROP(18),OW(17),VOLIN(16),VOLRT(15),DCON(14),        STA00210
     3 PORAT(13),DISS(12),PPASS(12),PST(11),OCIN(10),PCOND(7),              STA00220
     4 OPRIOR(9),ODEN(8)                                                    STA00230
      DIMENSION VRIN(475),GEN(999),GREF(999),ALINK(1)                       STA00240
      EQUIVALENCE (LINK(1),ALINK(1)),(LINK(31),INDEX(27))                   STA00250
      EQUIVALENCE (LINK(31),LSUBS(30))                                      STA00260
      EQUIVALENCE(LINK(3)),LSUPER(29)),  (LINK(31),IDADJ(28)),              STA00270
     1 (LINK(31),NSYMB(12)), (LINK(31),PCUM(26)), (LINK(31),PRIRCM(25)),    STA00280
     2 (LINK(31),CIN(24)), (LINK(31),CTOT(23)), (LINK(31),PROP(22)),        STA00290
     3 (LINK(31),SPFAC(21)), (LINK(31),WADJ(20)), (LINK(31),W(19)),         STA00300
     4 (LINK(31),OPROP(18)), (LINK(31),OW(17)), (LINK(31),VOLIN(16)),       STA00310
     5 (LINK(31),VOLRT(15)), (LINK(31),DCON(14)), (LINK(31),PORAT(13)),     STA00320
     6 (LINK(31),DISS(12)), (LINK(31),PPASS(12)), (LINK(31),PST(11)),       STA00330
     7 (LINK(31),OCIN(10)), (LINK(31),PCOND(7)), (LINK(31),VRIN(7)),        STA00340
     8 (LINK(31),GEN(7)), (LINK(31),OPRIOR(9)), (LINK(31),ODEN(8)),         STA00350
     9 (LINK(31),GREF(8))                                                   STA00360
      COMMON/CLUS/ JUNK(12),NARL,NTOP,NTBSZM,NWANT,LINK(14000)              STA00370
      COMMON /MAP/ MAP(10), LINES(10,2), MAPCT, LINECT, KA                  STA00380
      DIMENSION MXAR(3),LR(3),LV(3)                                         STA00390
      EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT),                              STA00400
     1    (LR(3),LOVAR),(LV(1),LSUM),(LV(2),LSKEW),(LV(3),LOSUM)            STA00410
C                                                                           STA00420
      COMMON /MISC/ MQ,M4,LR,LV,NINCLS,MXAR,WTINIT,KROOT,EPS,DELT,          STA00430
     1    AMO,ODCON,XOVFLO,XUNFLO,WADJIN,ELIMTH,SEPTH,VFAC,AMM,SMLTH,       STA00440
     2    INDXVL,VFAC,NPTS0,PORATH,SPMVTH,DIFAC,GFACTM,AMOFAC,              STA00450
     3     AMOMIN,AMOMAX,AMORAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIM,WOELSM,       STA00460
     4    RETTER,MODE,CORLEN,SPCOR                                          STA00470
C                                                                           STA00480
      DIMENSION PACCEL(2),MACCEL(2),VACCEL(2)                               STA00490
C                                                                           STA00500
      COMMON /STPAR/WAIT,CONLV,SKHND,SKCHI,TRHND,TRCHI,URKBND,URKCHI,       STA00510
     1    PACCEL,MACCEL,VACCEL                                              STA00520
C                                                                           STA00530
      COMMON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYM(61) ,              STA00540
     1              LNCAT, PRNT(4) , KLHC , PRTME, PROUT, TOTPIX,           STA00550
     2  SCRAM1,BUFPIX,BUFTOT,NBUFS0,NDUMP,LBUFD                            STA00560
     3, MAXHE, AREA, NWDS, NWDPS, NPTS, LBUF, IQ1,NUCYCL                    STA00570
C                                                                           STA00580
      INTEGER TOTWRD,SYM,PRNT,PRTME,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT       STA00590
     1 ,CLSNAM                                                              STA00600
C                                                                           STA00610
      COMMON /TIMERR/ORGTIM, TIMEMX                                         STA00620
      INTEGER TIMER                                                         STA00630
C                                                                           STA00640
      COMMON /RAND/NX,NXA,NXO                                               STA00650
      COMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200)                             STA00660
      REAL RELPRP                                                           STA00670
CHANGE***     1                                                             STA00680
      REAL*8 ZFAC,ZQ,ZQS                                                    STA00690
      REAL SUM(1),SKEW(1),KURT(1),OSUM(1),OVAR(1)                           STA00700
C                                                                           STA00710
      DIMENSION   REL(30) , COVEC(30) ,  PV(MQ,BUFPIX)                      STA00720
C                                                                           STA00730
      DATA MONTE,AMONTE,PLIM/3,3.,.1/                                       STA00740
      DATA ITHOUS /1000/                                                    STA00750
C  MONTE---CHECK LIK. RATIO 1/3 OF THE TIME, ELIM. WHEN PROP. LT /.1        STA00760
      INTEGER DISC                                                          STA00770
      XP(DIST)=EXP(-.5*DIST)                                                STA00780
C                                                                           STA00790
```

```
C  INITIALIZE SWITCH FOR MAXIMUM LIKEHOOD                                  STA00800
C                                                                          STA00810
      KTRAD = 0                                                            STA00820
C                                                                          STA00830
C       HERE ABOVE GOFT IS THE SQUARE ROOT OF THE COVARIANCE               STA00840
C           MATRIX, AND FFAC IS A POWER OF PI.                             STA00850
      KROT=KROTIN                                                          STA00860
      NPISO = 0                                                            STA00870
      NIT = NOCYCL                                                         STA00880
C                                                                          STA00890
      BMQ=.66666666667*AMQ                                                 STA00900
        KL = LSUBS(KROT)                                                   STA00910
C                                                                          STA00920
       KL=0                                                                STA00930
C                  ***** READ AND PROCESS DATA  NIT  TIMES *****           STA00940
C                                                                          STA00950
      ITER = 0                                                             STA00960
C                                                                          STA00970
 1    ITER = ITER + 1                                                      STA00980
C                                                                          STA00990
C                     *** READ 1 BUFFER OF SCRAMBLED DATA ***              STA01000
C                                                                          STA01010
C  MQ = LENGTH OF ONE VECTOR                                               STA01020
      BUFSIZ = BUFPIX * MQ                                                 STA01030
      NBUFS = TOTWRD/BUFSIZ                                                STA01040
      LBUF = MOD( TOTWRD , BUFSIZ )                                        STA01050
      IF( LBUF .GT. 0 )   NBUFS = NBUFS + 1                                STA01060
C  INADDR = 1ST WORD OF ORIG DATA ON FAST STORAGE                          STA01070
      INADDR = SCRAM1                                                      STA01080
      BUFCNT = 0                                                           STA01090
C  TOTWRD = NO. WORDS IN ORIGINAL DATA ON DRUM                             STA01100
C  MQ = LENGTH OF ONE VECTOR                                               STA01110
      TOTPIX = TOTWRD/MQ                                                   STA01120
C                                                                          STA01130
 50   BUFCNT = BUFCNT + 1                                                  STA01140
C                                                                          STA01150
      NWORDS = BUFSIZ                                                      STA01160
      IF ( LBUF .GT. 0 .AND. BUFCNT .EQ. NBUFS )  NWORDS = LBUF            STA01170
C                                                                          STA01180
C                                                                          STA01190
      CALL RREAD (INADDR, PV, NWORDS, ISTAT)                               STA01200
 10   IF( ISTAT .GT. 0 )   GO TO 10                                        STA01210
C                                                                          STA01220
      INADDR = INADDR + NWORDS                                             STA01230
      NPIXEL = NWORDS/MQ                                                   STA01240
      NDO = NPIXEL                                                         STA01250
C                                                                          STA01260
C                                                                          STA01270
C       *** INSPECT EACH CLASS AND PROCESS EACH OF THE DATA POINTS  ***    STA01280
      DO 399 IDO=1,NDO                                                     STA01290
C                                                                          STA01300
C  CHECK TIME EVERY 1000 POINTS                                            STA01310
      IF (MOD(IDO,ITHOUS) .NE. 0) GO TO 100                                STA01320
C                                                                          STA01330
      INITIM = 0.                                                          STA01340
      ELPTIM = TIMER(INITIM)                                               STA01350
      WRITE (6,9966) ELPTIM,ORGTIM,TIMEMX                                  STA01360
9966  FORMAT (' NEW,ORG,MAX',3F16.5)                                       STA01370
      XX = (ELPTIM-ORGTIM) / 60000.                                        STA01380
      WRITE (6,1111) XX,TIMEMX                                             STA01390
C                                                                          STA01400
      IF (XX .LT. TIMEMX) GO TO 100                                        STA01410
C                                                                          STA01420
C  MAXIMUM TIME EXCEEDED, WRITE MESSAGES AND STOP                          STA01430
      WRITE (6,1111) XX,TIMEMX                                             STA01440
      WRITE (3,1111) XX,TIMEMX                                             STA01450
1111  FORMAT (' TIME ELASPED IS ',F10.3,' MAXIMUM TIME IS ',F10.3)         STA01460
      STOP                                                                 STA01470
C                                                                          STA01480
100   CONTINUE                                                             STA01490
C  ** THIS CODE GETS RANDOM NUMBERS. **                                    STA01500
C  GET NEXT POINT IN SEQUENCE                                              STA01510
C                                                                          STA01520
C                                                                          STA01530
C     WE USE MONTE-CARLO TECHNIQUES FOR LOW PROBABILITY CLASSES(P<PLIM)     STA01540
      PCUM(KROT)=0.                                                        STA01550
C                                                                          STA01560
      IF( INDEX(KL) .NE. 0 .AND. KL .NE. 119)                              STA01570
     *   WRITE(6,1000)   IDO , INDEX(KL) , KL                              STA01580
```

```
 1000 FORMAT(    3X, '**** WARNING FROM STATIS ****  ON THE ', 2X, I5,  STA01590
     * 2X, 'TIME, INDEX(KL)=' , I5, 3X, ', KL=' , I5      )          STA01600
C                                                                     STA01610
      PR(RCM(KROT)=0.                                                 STA01620
      PPASS(KROT)=1.                                                  STA01630
      ISEC=0                                                          STA01640
      KL=LSUBS(KROT)                                                  STA01650
      KFATH=KROT                                                      STA01660
C   GO DOWN CLUSTER TREE                                              STA01670
  130 IF(LSUBS(KL).EQ.0) GO TO 131                                    STA01680
C                                                                     STA01690
C  FIND BOTTOM NODE                                                   STA01700
      PCUM(KL)=0.                                                     STA01710
      PRIRCM(KL)=0.                                                   STA01720
      KFATH=KL                                                        STA01730
      KL=LSUBS(KL)                                                    STA01740
      GO TO 130                                                       STA01750
CHANGE***     4                                                       STA01760
C                                                                     STA01770
C  CALC UNWEIGHTED NORMALIZED VECTOR REL                              STA01780
C   WUSE = CURRENT WEIGHT                                             STA01790
  131 IF(INDEX(KL).LE.0) GO TO 133                                    STA01800
C                                                                     STA01810
C        USE NEW WEIGHTS AND MEANS IF ADJUST HAS BEEN CALLED          STA01820
      CALL CORRECT(REL,PV(1,IDO),W(KL),SUM(KL+1))                     STA01830
      WUSE=W(KL)                                                      STA01840
      PROP(KL)=CIN(KL)/(W(KFATH)-CTOT(KL))                            STA01850
      GO TO 134                                                       STA01860
CHANGE***     4,5                                                     STA01870
  133 CALL CORRECT(REL,PV(1,IDO),OW(KL),OSUM(KL+1))                   STA01880
      WUSE=OW(KL)                                                     STA01890
  134 DISS(KL)=DOTSQ(REL,VRIN(KL+1))*WUSE                             STA01900
      WDISS = DISS(KL) * DCON(KL)                                     STA01910
      IF(ABS(WDISS).LE. 100.) GO TO 531                               STA01920
      PCOND(KL)=0.                                                    STA01930
      GO TO 138                                                       STA01940
  531 CONTINUE                                                        STA01950
       Y = -.5*WDISS                                                  STA01960
      XTEMP = EXP(Y)                                                  STA01970
      PCOND(KL)=XTEMP/VOLPT(KL)                                       STA01980
  138 IF(LSUBS(KL).NE.0) PCUM(KL)=PCUM(KL)/PRIRCM(KL)                 STA01990
      SPUSE=SPFAC(KL)/SPCOR                                           STA02000
      IF(SPUSE.GT.XOVFLO) GO TO 231                                   STA02010
      PST(KL)=PROP(KL)*PCOND(KL)                                      STA02020
C                                                                     STA02030
C  SET KL = LAST NODE IN STRING                                       STA02040
      GO TO 239                                                       STA02050
  231 IF(SPUSE.LT.XOVFLO) GO TO 232                                   STA02060
      PST(KL)=PROP(KL)*PCUM(KL)                                       STA02070
      GO TO 239                                                       STA02080
232   CONTINUE                                                        STA02090
      ZZ=EXP(SPUSE)                                                   STA02100
      PST(KL)=PROP(KL)*(PCOND(KL)+ZZ*PCUM(KL))/(1.+ZZ)                STA02110
  239 PCUM(KFATH)=PCUM(KFATH)+PST(KL)                                 STA02120
      PRIRCM(KFATH)=PRIRCM(KFATH)+PROP(KL)                            STA02130
  139 KL=LINK(KL)                                                     STA02140
      IF(KL)130,149,130                                               STA02150
C   GO UP TREE                                                        STA02160
  149 KL=KFATH                                                        STA02170
      KFATH=LSUPER(KL)                                                STA02180
      IF(KL.NE.KROT) GO TO 131                                        STA02190
C  WE NOW HAVE THE RELEVANT CLASSES AND THEIR PROBABILITIES AVAILABLE. STA02200
C                                                                     STA02210
C  NEXT WE MAKE THE APPROPRIATE INDIVIDUAL FIRST-ORDER STATISTICS ADJ. STA02220
  150 CONTINUE                                                        STA02230
      PCUM(KROT)=PCUM(KROT)/PRIRCM(KROT)                              STA02240
      IF(PCUM(KROT).NE.0.) GO TO 151                                  STA02250
CHANGE***     5                                                       STA02260
      WRITE (6,555) IDO,W(KROT),(PV(KPR,IDO),KPR=1,MO)                STA02270
  555 FORMAT ('0**SUSPECTED BAD DATA POINT--STATIS**IDO=',I5,'   ROOT',STA02280
     1   F10.2/5X,'VECTOR',(5E12.5))                                  STA02290
      WRITE (3,5555) IDO,W(KROT),(PV(KPR,IDO),KPR=1,MO)               STA02300
5555  FORMAT ('**SUSPECTED BAD DATA POINT--STATIS**IDO=',I5,'ROOT',   STA02310
     1  F10.2/'VECTOR=',4F8.3/(7X,4F8.3))                            STA02320
C                                                                     STA02330
      KTBAD = KTBAD + 1                                               STA02340
      IF (KTBAD .LE. 50) GO TO 399                                    STA02350
C                                                                     STA02360
C  ABORT JOB TOO MANY BAD POINTS                                      STA02370
```

```
      WRITE (6,1500)                                                    STA02380
1500   FORMAT (' JOB ABORTED, TOO MANY BAD POINTS')                     STA02390
      STOP                                                              STA02400
C                                                                       STA02410
  151 CONTINUE                                                          STA02420
      KL=ISUBS(KROT)                                                    STA02430
      KFATH=KROT                                                        STA02440
      W(KROT)=W(KROT)+PPASS(KROT)                                       STA02450
      NPTSO=NPTSO+1                                                     STA02460
      KADJ=6                                                            STA02470
  153 CONTINUE                                                          STA02480
      IF(PST(KL).EQ.0.) GO TO 299                                       STA02490
      PPASSK=PPASS(KFATH)                                               STA02500
      P=PST(KL)/(PCUM(KFATH)*PRINCM(KFATH))*PPASSK                      STA02510
      KLO=KL                                                            STA02520
C                                                                       STA02530
  132 IF(P.GE.PLIM)GO TO 140                                            STA02540
      IF(DISC(MONTE).NE.1)GO TO 299                                     STA02550
      PPASSK=PPASSK*AMONTE                                              STA02560
      P=P*AMONTE                                                        STA02570
      GO TO 132                                                         STA02580
CHANGE***      6                                                        STA02590
  140 IF(INDEX(KL).LE.0) GO TO 143                                      STA02600
      CALL CORRECT(REL,PV(1,IDO),W(KL),SUM(KL+1))                       STA02610
      GO TO 144                                                         STA02620
CHANGE***      6.5                                                      STA02630
  143 CALL CORRECT(REL,PV(1,IDO),OW(KL),OSUM(KL+1))                     STA02640
  144 WO=W(KL)                                                          STA02650
      IF(P.GT.1.001.OR.P.LT.0.)PRINT 672,INDEX(KL),KL,INDEX(KFATH),     STA02660
     1  KFATH,IDO,P,PST(KL),PCUM(KFATH),PRINCM(KFATH),PPASSK,           STA02670
     2  PROP(KL)                                                        STA02680
  672 FORMAT(' PROB ERROR(STATIS):',2(I3,I7),I6,' P=',E9.4,             STA02690
     1    20X,'PROM',7E9.4)                                            STA02700
      IF (P.GT.1.) P=.01                                                STA02710
      W(KL)=W(KL)+P                                                     STA02720
      ALO=P/W(KL)                                                       STA02730
      ALPHA=WO*ALO                                                      STA02740
C   HERE WE ADJUST SPFAC AND PORAT.                                     STA02750
      IF(ISUBS(KL).EQ.0) GO TO 611                                      STA02760
      ZO=(PCUM(KL)-PCOND(KL))/(PCUM(KL)+PCOND(KL)+1.E-37)               STA02770
      ZOS=ZO*ZO                                                         STA02780
      PORAT(KL)=PORAT(KL)+P*ZOS                                         STA02790
      SPFAC(KL)=SPFAC(KL)+P*ZO*(2.+ZOS/(1.5-.9*ZOS))                    STA02800
  611 CONTINUE                                                          STA02810
      IF(INDEX(KL).LT.0) GO TO 189                                      STA02820
      VOLIN(KL)=VOLIN(KL)*(1.+ALO*DISS(KL))                             STA02830
      VOLRT(KL)=.5*(VOLRT(KL)+VOLIN(KL)/VOLRT(KL))                      STA02840
C       HERE WE KEEP VOLRT NEAR SQRT(VOLIN) BY NEWTON'S METHOD.         STA02850
      ZFAC=.5/(1./ALO+.5)                                               STA02860
      DCON(KL)=DCON(KL)-ALO*ZFAC*(3.+ZFAC*ZFAC)                         STA02870
C     THE ABOVE APPROXIMATION TO THE LOG IS CORRECTED FOR IN ADJUST     STA02880
C   PROPORTION CALCULATION.                                             STA02890
  189 PROPL=PROP(KL)/PRINCM(KFATH)                                      STA02900
      IF(PROPL.GT..99999) GO TO 192                                     STA02910
      IF(P.GE.PPASSK*PROPL) GO TO 190                                   STA02920
      CTOT(KL)=CTOT(KL)+P/PROPL                                         STA02930
      GO TO 191                                                         STA02940
  192 CTN(KL)=CTN(KL)+1.                                                STA02950
      GO TO 191                                                         STA02960
  190 CTOT(KL)=CTOT(KL)+(PPASSK-P)/(1.-PROPL)                           STA02970
      CTN(KL)=CTN(KL)+(P-PPASSK*PROPL)/(1.-PROPL)                       STA02980
  191 CONTINUE                                                          STA02990
CHANGE***      7                                                        STA03000
      CALL VPV(SUM(KL+1),P,PV(1,IDO))                                   STA03010
      IF(INDEX(KL).LE.0) GO TO 163                                      STA03020
      CALL VMTV(COVEC,VRIN(KL+1),REL)                                   STA03030
      COFI = -ALPHA/(1.+ ALO*DISS(KL))                                  STA03040
      CALL MPVS(VRIN(KL+1),COFI,COVEC)                                  STA03050
      GO TO 164                                                         STA03060
  163 CALL MPVS(OVAR(KL+1),ALPHA,REL)                                   STA03070
  164 CONTINUE                                                          STA03080
C  DISS(KL) CONTAINS THE GAUSSIAN DISTANCE OF THE POINT FROM THE CLUSTER STA03090
C  VRIN IS THE INVERSE COVARIANCE MATRIX   (****) OVER W(KL)     (**** NSTA03095
C                 (THIS INTRODUCES SEVERAL SCALE FACTORS )              STA03100
C  COVEC IS THE CONTRAVARIANT FORM OF THE RELATIVE DISTANCE REL.        STA03110
C             COVEC=VRIN*REL                                            STA03120
C  WE NOW HAVE ALL THE LINEAR AND QUADRATIC STATISTICS, AND PROCEED     STA03130
C    TO CALCULATE THE APPROXIMATE 3RD AND 4TH MOMENTS FOR TESTING.      STA03140
C    THESE MOMENTS ARE NOT CALCULATED EXACTLY: THE SQUARED              STA03150
C       DISTANCE OF A POINT FROM THE MEAN ACTUALLY SHOULD               STA03160
```

```
C         USE ALL THE DATA IN CALCULATING THE MEAN AND                          STA03170
C         COVARIANCE. WHEREAS WE SUBSTITUTE THE CURRENT                         STA03180
C         VALUES INSTEAD.  THUS THE VALUES CALCULATED DEPEND                    STA03190
C         ON THE ORDER THE POINTS ARE READ IN.  THIS IS NOT                     STA03200
C           CRITICAL.                                                           STA03210
C                                                                               STA03220
      WDISS=DISS(KL)*P                                                          STA03230
C      IF (INDEX(KL).LT.0) WRITE (3,9980) WDISS,INDEX(KL)                       STA03240
9980   FORMAT (' WDISS,KL=',F14.4,I6)                                           STA03250
      CALL VPV(SKEW(KL+1),WDISS,REL)                                            STA03260
      CALL MPVS(KURT(KL+1),WDISS,REL)                                           STA03270
298   CONTINUE                                                                  STA03280
C                                                                               STA03290
C  WE NOW ADJUST THE CLASS FOR LARGE-SCALE STATISTICAL EFFECTS.                 STA03300
C     ON AN OCCASIONAL BASIS.   THIS INCLUDES NOMINAL NEWTONS                   STA03310
C     METHOD CORRECTIONS AND TESTING FOR THE POSSIBILITY                        STA03320
C     OF TWO CLUSTERS (USING THE SKEW AND KURT STATISTICS).                     STA03330
      IF(W(KL).GT.(WADJ(KL)+.0005).OR.NPTSO.GE.IDADJ(KL)) KADJ=KL               STA03340
C     IF (KL.EQ.145.AND.W(KL) .LT. 200.5 .AND. W(KL) .GT. 199.5)KADJ=KL         STA03350
      IF (P(KL) .GT.WADJ(KL)) WRITE (6,9988) INDEX(KL),W(KL),WADJ(KL)           STA03360
9988  FORMAT (' STATIS KL, W(KL),WADJ(KL)',I6,2E16.10)                          STA03370
      IF (NPTSO.GE.IDADJ(KL)) WRITE (6,9987) INDEX(KL), NPTSO,IDADJ(KL)         STA03380
9987  FORMAT (' STATIS NPTSO,IDADJ(KL)',3I8)                                    STA03390
  299 PPASS(KL)=P                                                               STA03400
      IF(LSUPS(KL).EQ.0.OR.PCUM(KL).EQ.0.) GO TO 304                            STA03410
      KPATH=KL                                                                  STA03420
      KL=LSUBS(KL)                                                              STA03430
      GO TO 153                                                                 STA03440
  304 KL=LINK(KL)                                                               STA03450
  303 IF(KL) 153,305,153                                                        STA03460
  305 KL=KPATH                                                                  STA03470
      KPATH=LSUPER(KL)                                                          STA03480
      IF(KL.NE.KROT) GO TO 304                                                  STA03490
      IF(KADJ.NE.0) CALL ADJUST(KADJ,SUM,SKEW,KURT,OSUM,OVAR)                   STA03500
      IF(MOD(NPTSO,TOTPTX).NE.0.OR.MODE.EQ.0) GO TO 309                         STA03510
      NXA=NX0                                                                   STA03520
      CALL PRTREE (KROT)                                                        STA03530
      CALL CLDUMP(KROT)                                                         STA03540
  309 CONTINUE                                                                  STA03550
  647 FORMAT(' LOOP IN STATIS:IDO,W(KROT),KL,SECTION',I5,E11.5,2I5              STA03560
     1    /(1X14I5))                                                           STA03570
  399 CONTINUE                                                                  STA03580
      IF( NUFCNT .LT. NVUES )  GO TO 50                                         STA03590
      WRITE  (6,2000) ITER                                                      STA03600
      WRITE (3,2000) ITER                                                       STA03610
2000  FORMAT(////,' NO OF ITERATIONS THROUGH ALL THE DATA = ',I4)               STA03620
      IHOLD = PRINT                                                             STA03630
      PRINT = 3                                                                 STA03640
C  RESET SWITCH TO WRITE MAX LIKELIHOOD LABELING FILE                           STA03650
      MXLLWT = 1                                                                STA03660
      CALL PRTREE (KROT)                                                        STA03670
      CALL CALPRB (KROT)                                                        STA03680
      CALL CLDUMP(KROT)                                                         STA03690
      END FILE 23                                                               STA03700
      MXLLWT = 0                                                                STA03710
C                                                                               STA03720
CC  CALL CLUSTER MAP SUBROUTINE IF THIS MAP REQUESTED FOR THIS ITERATION        STA03730
C                                                                               STA03740
      IF (ITER .EQ. NIT) RETURN                                                 STA03750
C                                                                               STA03760
      IF (MAP(1) .EQ. KA) GO TO 510                                             STA03770
C                                                                               STA03780
      DO 500 I = 1,MAPCT                                                        STA03790
      IF (ITER .EQ. MAP(I)) GO TO 510                                           STA03800
500   CONTINUE                                                                  STA03810
      GO TO 520                                                                 STA03820
C                                                                               STA03830
510   LSTITR = 0                                                                STA03840
      NUFILE = 0                                                                STA03850
      CALL CLUSMP (NUFILE,LSTITR)                                               STA03860
C                                                                               STA03870
520   CONTINUE                                                                  STA03880
      PRINT = IHOLD                                                             STA03890
      KTHAD = 0                                                                 STA03900
      IF ( ITER .LT. NIT)  GO TO 1                                              STA03910
      RETURN                                                                    STA03920
      END                                                                       STA03930
```

```
      SUBROUTINE CLPR(KL,IN,SUM,SKEW,KURT)                                  CLP00010
C  THIS ROUTINE PRINTS OUT ALL THE VARIABLES BELONGING TO SOME              CLP00020
C   CLASS INDEXED BY KL.                                                    CLP00030
      DIMENSION NTR(32)                                                     CLP00040
      DIMENSION INDEX(27),LSUBS(30),LSUPER(29),IDADJ(28),NSYMB(12),         CLP00050
     1 PCUM(26),PRIRCM(25),CIN(24),CTOT(23),PROP(22),SPFAC(21),             CLP00060
     2 WADJ(20),W(19),ODPROP(18),OW(17),VOLIN(16),VOLRT(15),DCON(14),       CLP00070
     3 PORAT(13),DISS(12),PPASS(12),PST(11),OCIN(10),PCOND(7),              CLP00080
     4 OPRTOP(9),ODEN(8)                                                    CLP00090
      DIMENSION VRIN(475),GEN(999),GREF(999),ALINK(1)                       CLP00100
      EQUIVALENCE (LINK(1),ALINK(1)),(LINK(3),INDEX(27))                    CLP00110
      EQUIVALENCE (LINK(31),LSUBS(30))                                      CLP00120
      EQUIVALENCE(LINK(31),LSUPER(29)),  (LINK(31),IDADJ(28)),              CLP00130
     1 (LINK(31),NSYMB(12)),  (LINK(31),PCUM(26)),  (LINK(31),PRIRCM(25)),  CLP00140
     2 (LINK(31),CIN(24)),  (LINK(31),CTOT(23)),  (LINK(31),PROP(22)),      CLP00150
     3 (LINK(31),SPFAC(21)),  (LINK(31),WADJ(20)),  (LINK(31),W(19)),       CLP00160
     4 (LINK(31),ODPROP(18)),  (LINK(31),OW(17)),  (LINK(31),VOLIN(16)),    CLP00170
     5 (LINK(31),VOLRT(15)),  (LINK(31),DCON(14)),  (LINK(31),PORAT(13)),   CLP00180
     6 (LINK(31),DISS(12)),  (LINK(31),PPASS(12)),  (LINK(31),PST(11)),     CLP00190
     7 (LINK(31),OCIN(10)),  (LINK(31),PCOND(7)),  (LINK(31),VRIN(7)),      CLP00200
     8 (LINK(31),GEN(7)),  (LINK(31),OPRTOP(8)),  (LINK(31),ODEN(8)),       CLP00210
     9 (LINK(31),GREF(8)),  (LINK(3),NTR(1))                                CLP00220
      COMMON/CLUS/  JUNK(12),NARL,NTOP,NTHSZM,NWANT,LINK(14000)             CLP00230
      DIMENSION MXAR(3),LR(3),LV(3)                                         CLP00240
      EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT),                              CLP00250
     1    (LR(3),LOVAR),(LV(1),LSUM),(LV(2),LSKEW),(LV(3),LOSUM)            CLP00260
C                                                                           CLP00270
      COMMON /MISC/ MO,MN,LR,LV,NINCLS,MXAR,NTINIT,KROOT,EPS,DELT,          CLP00280
     1    AMO,ODCON,XOVFLO,XUNFLO,WADJIN,ELMTH,SEPTH,VFAC,AMM,SHLTH,        CLP00290
     2    INDXVL,WFAC,NPTSO,POWATH,SPMVTH,OWFAC,GRACTM,AMOFAC,              CLP00300
     3    AMOMIN,AMOMAX,AMORAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIM,WDELSM,        CLP00310
     4    BETTER,MODE,CORLEN,SPCOR                                          CLP00320
C                                                                           CLP00330
      COMMON /STPAR/WAIT,CONLV,SKHND,SKCHI,TKHND,TRCHI,URKBND,URKCHI,       CLP00340
     1    PACCEL(2),MACCEL(2),VACCEL(2)                                     CLP00350
      REAL XTEMP(30)                                                        CLP00360
      REAL SUM(1),SKEW(1),KURT(1)                                           CLP00370
      REAL AMEAN(16),OMEAN(16)                                              CLP00380
C                                                                           CLP00390
      COMMON /WXLL/ MXLLWT, MXLLEN, RELPRP(200)                             CLP00400
      REAL RELPRP, RELP                                                     CLP00410
      IF(KL.EQ.0) RETURN                                                    CLP00420
      PRR=1.                                                                CLP00430
      MXLLEN = 23                                                           CLP00440
      LPCC=LSUPER(KL)                                                       CLP00450
      IF (KL .EQ. 119) LPCC = 119                                           CLP00460
       IF (KL .EQ. 119)  PRR = 0.                                           CLP00470
      IF(KL.NE.119.AND.INDEX(KL).GE.0) PRR=PROP(KL)/PRIRCM(LPCC)            CLP00480
      PRINT 101,IN,INDEX(KL),PRR,W(LPCC),SPFAC(KL),W(KL),OW(KL),            CLP00490
     1 WADJ(KL),IDADJ(KL),PROP(KL),CIN(KL),CTOT(KL),OPROP(KL),OCIN(KL),     CLP00500
     2    ODEN(KL),PORAT(KL),VOLIN(KL),VOLRT(KL),DCON(KL)                   CLP00510
  101 FORMAT('0CLUSTER',I4,' INDEX',I4,'  PROPORTION',F11.5,                CLP00520
     1    ' PARENT',F9.3,/,'  SPLIT',F11.4/                                 CLP00530
     2    ' WEIGHT',F12.3,2X,'WAS',F12.3,                                   CLP00540
     3    4X,'ADJUST',F12.3,'  ID',I6/                                      CLP00550
     4    ' PROPORTION: PROP',F8.5,'  CIN',F8.2,'  CTOT',F8.2/              CLP00560
     5 ' OLD PROP',F9.6,'  CIN',F7.2,'  ODEN',F7.2,'  DIFFER',F7.2/         CLP00570
     6    ' VOLUME',E8.2,'    ROOT',F8.2,'    DCON',F8.2)                   CLP00580
      KD=LINK(KL)                                                           CLP00590
      LPCOC=LSUBS(KL)                                                       CLP00600
C     WRITE (3,9912) KL,LPCO,LPCOC                                          CLP00610
9912  FORMAT (' KL,LPCO,LPCOC',3I6)                                         CLP00620
      IF (LPCO .GE. 0.AND.LPCOC .GE. 0) PRINT 102,KL,INDEX(LPCO),           CLP00630
     1 LINK(KL),INDEX(LPCOC),LSUBS(KL),INDEX(LPCC),LSUPER(KL),NSYMB(KL)     CLP00640
  102 FORMAT('0 LOCATION',I5,'  LINK',I3,I5,'  SUBS',I3,I5,'  SUPER',       CLP00650
     1    I3,I5,'  SYMBOL',I6)                                             CLP00660
      WRITE (3,103) INDEX(KL),NSYMB(KL)                                     CLP00670
      WRITE (3,103) INDEX(KL),NSYMB(KL)                                     CLP00680
  103 FORMAT (' INDEX = ',I6,'     SYMBOL = ',I6)                           CLP00690
      PRINT 112,PST(KL),PCOND(KL),PCUM(KL),PRIRCM(KL)                       CLP00700
  112 FORMAT('0NET PROB',F7.2,'  DIRECT',F7.2,'  CUMS',                     CLP00710
     1    F7.2,' *',F7.2)                                                   CLP00720
C                                                                           CLP00730
      XTEMP = 10.**(-25)                                                    CLP00740
      IF(PCUM(KL).LT.XTEMP.OR.PRIRCM(KL).LT.XTEMP)  PRINT 104,              CLP00750
     1  PCUM(KL),PRIRCM(KL)                                                CLP00760
  104 FORMAT(T29,'CUMS',E10.5,' * ',E10.5,//)                               CLP00770
      IF(INDEX(KL).EQ.0) RETURN                                             CLP00780
      WRITE(6,6)                                                            CLP00790
```

```
      OMUSE=OM(KL)                                                      CLP00800
      IF(INDEX(KL).GE.0) GO TO 5                                        CLP00810
      MUSE=OM(KL)                                                       CLP00820
      OMUSE=M(KL)                                                       CLP00830
    5 TRK=0.                                                            CLP00840
      OVSK=0.                                                           CLP00850
      DO 2 I=1,MO                                                       CLP00860
      LOD=LOCK(I,I)                                                     CLP00870
      TRK=TRK+KURT(KL+LOD)                                              CLP00880
      OVSK=OVSK+GREF(LOSUM+KL+I)**2                                     CLP00890
      OMEAN(I)=GREF(LOSUM+I+KL)/OM(KL)                                  CLP00900
    2 AMEAN(I)=SUM(I+KL)/M(KL)                                          CLP00910
      PRINT 113,(AMEAN(I),I=1,MO)                                       CLP00920
  113 FORMAT(*0 MEAN *,6X,8F7.2/(12X,8F7.2))                            CLP00930
C                                                                       CLP00940
C  SAVE RELATIVE PROPORTION, VOLRT, DCON FOR MAX. LIKELIHOOD LABELING   CLP00950
      NUMBER = IABS(INDEX(KL))                                          CLP00960
      RELP = RELPRP(NUMBER)                                             CLP00970
      WRITE (14,9999) KL, INDEX(KL), NUMBER, RELPRP(NUMBER),RELP        CLP00980
9999  FORMAT (' KL=',I3,' INDEX(KL)=',I3,' NUMBER=',I3,' RELPRP=',2F6.2)CLP00990
      IF(MXLLWT.NE.0.AND.NSYMB(KL).NE.0) WRITE (MXLLFN) RELP, VOLRT(KL),CLP01000
     1  DCON(KL)                                                         CLP01010
C                                                                       CLP01020
C  SAVE MEAN FOR MAX. LIKELIHOOD LABELING                               CLP01030
      IF(MXLLWT.NE.0.AND.NSYMB(KL).NE.0)WRITE(MXLLFN) (AMEAN(I),I=1,MO) CLP01040
C                                                                       CLP01050
       MOS = MO*MO                                                      CLP01060
      LA=MOPSTR(MOS)                                                    CLP01070
C                                                                       CLP01080
C  EXPAND VRIN FROM TRIANG, MAKE AN MO*MO SQ SSYM MATRIX IN LINK(LB)    CLP01090
      LB=MOPSTR(MOS)                                                    CLP01100
      CALL SQMTX(ALINK(LB),VRIN(KL+1))                                  CLP01110
C                                                                       CLP01120
C MULT VRIN BY MUSE                                                     CLP01130
      DO 4 I = 1,MOS                                                    CLP01140
4      ALINK(LB+I-1) = ALINK(LB+I-1) * MUSE                             CLP01150
C                                                                       CLP01160
C  SAVE INVERSE OF COVAR MATRIX FOR MAXIMUM LIKELIHOOD PROGRAM          CLP01170
      DO 1005 I = 1,MO                                                  CLP01180
      IBEG = LB + (I-1)*MO                                              CLP01190
      LAST = IBEG + MO - 1                                              CLP01200
      IF (MXLLWT.NE.0.AND.NSYMB(KL).NE.0)                               CLP01210
     1 WRITE (MXLLFN) (ALINK(II),II=IBEG,LAST)                          CLP01220
1005  CONTINUE                                                          CLP01230
C                                                                       CLP01240
      DO 1004 I =1,MOS                                                  CLP01250
1004   ALINK(LB+I-1) = ALINK(LB+I-1)/MUSE                               CLP01260
C                                                                       CLP01270
C  CALC LA = INVERSE OF LB, CVL = DETERMINATE OF LB                     CLP01280
      CALL MINV(ALINK(LA),ALINK(LB),ALINK(LB),CVL)                      CLP01290
      DO 6 I=1,MOS                                                      CLP01300
    6 ALINK(LA+I-1)=ALINK(LA+I-1)/MUSE                                  CLP01310
      PRINT 116,(ALINK(LA+J-1),J=1,MO)                                  CLP01320
  116 FORMAT(*0 COVARIANCE*,12F7.2/(12X,8F7.2))                         CLP01330
      DO 7 I=2,MO                                                       CLP01340
    7 PRINT 105, I,(ALINK(LA+MO*I+J-MO-1),J=1,MO)                       CLP01350
  105 FORMAT(5X,I5,2X,8F7.2/(16X,8F7.2))                                CLP01360
C                                                                       CLP01370
C                                                                       CLP01380
      IF(TRK.EQ.0.) GO TO 150                                           CLP01390
      PRINT 107,(SKEW(KL+I),I=1,MO)                                     CLP01400
  107 FORMAT(*0 SKEW(*W)  *,1X,8F7.1/(12X,8F7.1))                       CLP01410
      GO TO 200                                                         CLP01420
120      CONTINUE                                                       CLP01430
      DO 300 J=1,MO                                                     CLP01440
      LOD=LOCK(1,J)                                                     CLP01450
      N=LOD+KL                                                          CLP01460
  300 KTEMP(J)=KURT(N)                                                  CLP01470
      PRINT 108, (KTEMP(J),J=1,MO)                                      CLP01480
  108 FORMAT(*0   KURT(*W)    1*,2X,5F13.6/(16X,5F13.6))                CLP01490
      DO 8 I=2,MO                                                       CLP01500
      DO 308 J=1,MO                                                     CLP01510
      LOD=LOCK(I,J)                                                     CLP01520
      LCH=LOD+KL                                                        CLP01530
  308 KTEMP(J)=KURT(LCH)                                                CLP01540
    8 PRINT 105,I, (KTEMP(J),J=1,MO)                                    CLP01550
  150 IF(OVSK.EQ.0..OR.OPROP(KL).EQ.PROP(KL).AND.INDEX(KL).GT.0)        CLP01560
     1    GO TO 200                                                    CLP01570
      PRINT 103,(OMEAN(I),I=1,MO)                                       CLP01580
```

```
 163  FORMAT(/,'     OLD MEAN',6X,5F13.6/(12X,5F13.6))                    CLP01590
      CALL SOMTY(ALINK(LA),GREF(LOVAR+KL+1))                              CLP01600
      DO 156 I=1,MOS                                                      CLP01610
 156  ALINK(LA+I-1)=ALINK(LA+I-1)/OWUSE                                   CLP01620
      PRINT 166,(ALINK(LA+J-1),J=1,MO)                                    CLP01630
 166  FORMAT('0    OLD COVARIANCE',5F13.6/(16X,5F13.6))                   CLP01640
      DO 157 I=2,MO                                                       CLP01650
 157  PRINT 165,I,(ALINK(LA+MO*I+J-MO-1),J=1,MO)                          CLP01660
 200  CALL FREE(LA,MOS)                                                   CLP01670
      CALL FREE(LB,MOS)                                                   CLP01680
      PRINT 109                                                           CLP01690
 109  FORMAT(/)                                                           CLP01700
      RETURN                                                              CLP01710
      END                                                                 CLP01720
```

```
      SUBROUTINE CALPPR (KROTIN)                                        CAL00010
C                                                                       CAL00020
CCC  PURPOSE:  TO CALCULATE THE RELATIVE PROBABILITIES OF THE CLUSTERS  CAL00030
C                                                                       CAL00040
      DIMENSION NTR(32)                                                 CAL00050
      DIMENSION INDEX(27),LSONS(30),LSUPER(29),IDADJ(28),NSYMB(12),     CAL00060
     1 PCUM(25),PWIRCM(25),CIN(24),CTOT(23),PROP(22),SPFAC(21),         CAL00070
     2 WADJ(20),W(19),OPROP(18),OUT(17),VOLIN(16),VOLRT(15),DCON(14),   CAL00080
     3 DUMAT(13),DISS(12),PPASS(12),PST(11),DCIN(10),PCOND(7),          CAL00090
     4 OPPTOP(9),ODEN(8)                                                CAL00100
      DIMENSION VRIN(475),GEN(999),GPEF(999),ALINK(1)                   CAL00110
      EQUIVALENCE (LINK(1),ALINK(1)),(LINK(31),INDEX(27))               CAL00120
      EQUIVALENCE (LINK(3),LSONS(30))                                   CAL00130
      EQUIVALENCE(LINK(3),LSUPER(29)),  (LINK(31),IDADJ(28)),           CAL00140
     1 (LINK(31),NSYMB(12)),  (LINK(31),PCUM(25)),  (LINK(31),PWIRCM(25)),CAL00150
     2 (LINK(31),CIN(24)),  (LINK(31),CTOT(23)),  (LINK(31),PROP(22)),  CAL00160
     3 (LINK(31),SPFAC(21)),  (LINK(31),WADJ(20)),  (LINK(31),W(19)),   CAL00170
     4 (LINK(31),OPPOP(18)),  (LINK(31),OUT(17)),  (LINK(31),VOLIN(16)),CAL00180
     5 (LINK(31),VOLRT(15)),  (LINK(31),DCON(14)),  (LINK(31),DUMAT(13)),CAL00190
     6 (LINK(31),DISS(12)),  (LINK(31),PPASS(12)),  (LINK(31),PST(11)), CAL00200
     7 (LINK(31),DCIN(10)),  (LINK(31),PCOND(7)),  (LINK(31),VRIN(7)),  CAL00210
     8 (LINK(31),GEN(7)),  (LINK(31),OPPTOP(9)),  (LINK(31),ODEN(8)),   CAL00220
     9 (LINK(31),GPEF(4)),  (LINK(31),NTR(31))                          CAL00230
      COMMON/LUS/  JUNK(12),NARL,NTOP,NTHSZM,NWANT,LINK(14000)          CAL00240
      DIMENSION MXAR(3),LR(3),LV(3)                                     CAL00250
      EQUIVALENCE (LR(1),LVRM),(LR(2),LKURT),                           CAL00260
     1    (LR(3),LOVAR),(LV(1),LSUM),(LV(2),LSKEW),(LV(3),LOSUM)        CAL00270
C                                                                       CAL00280
      COMMON /MISC/ MO,M4,LR,LV,NINCLS,MXAR,WTINIT,KROOT,EPS,DELT,      CAL00290
     1    AMO,ODCON,XOVFLO,XUNFLO,XADJIN,ELIMTH,SEPTH,VFAC,AMM,SMLTH,   CAL00300
     2    IMDXVL,MFAC,MPTSO,PDPATH,SPMVTH,UNFAC,GRACTM,AMOFAC,          CAL00310
     3    AMOMIN,AMOMAX,AMOMAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIM,WOELSM,    CAL00320
     4    BETTER,MODE,CORLEN,SPCOP                                      CAL00330
C                                                                       CAL00340
      COMMON /STRAP/WAIT,CONLV,SKHND,SKCHI,TRHND,TRCHI,URKHND,URKCHI,   CAL00350
     1    PACCEL(2),MACCEL(2),VACCEL(2)                                 CAL00360
C                                                                       CAL00370
        COMMON /MXLI/ MXLINT, MXLIEN, RELPRP(200)                       CAL00380
C                                                                       CAL00390
C                                                                       CAL00400
      LOGICAL ISPLIT                                                    CAL00410
C                                                                       CAL00420
      KROT=KROTIN                                                       CAL00430
C     WRITE (3,9991) KROT                                               CAL00440
9991  FORMAT (' ENTERING CALRPR, KROT=',I4)                             CAL00450
CC                                                                      CAL00460
C  RELPRP IS RELATIVE PROPORATION FOR CLUSTER                           CAL00470
      KRTNDX = IABS(INDEX(KROT))                                        CAL00480
      RELPRP(KRTNDX)=1.                                                 CAL00490
C                                                                       CAL00500
C    GO DOWN CLUSTER TREE                                               CAL00510
      KL=LSONS(KROT)                                                    CAL00520
      KFATH=KROT                                                        CAL00530
C                                                                       CAL00540
C  PRIRCM = CUM PRIOR = SUM OF PROPORATIONS FOR ALL SONS                CAL00550
130   KLNDX = IABS(INDEX(KL))                                           CAL00560
      KFNDX = IABS(INDEX(KFATH))                                        CAL00570
      RELPRP(KLNDX)=PROP(KL)/(PRIRCM(KFATH))*RELPRP(KFNDX)              CAL00580
C                                                                       CAL00590
C  CHECK FOR ADDITIONAL SPLITS, UNTIL LAST SPLIT IS FOUND               CAL00600
      IF(.NOT.ISPLIT(KL)) GO TO 131                                     CAL00610
      KFATH=KL                                                          CAL00620
      KL=LSONS(KL)                                                      CAL00630
      GO TO 130                                                         CAL00640
C                                                                       CAL00650
C  END OF SPLITS, CHECK FOR MORE LINKS ON THIS LEVEL                    CAL00660
131   KL=LINK(KL)                                                       CAL00670
      IF (KL .GT. 0) GO TO 130                                          CAL00680
C                                                                       CAL00690
C  NO MORE LINKS ON THIS LEVEL, GO UP ONE LEVEL                         CAL00700
      KL = KFATH                                                        CAL00710
      KLNDX = IABS(INDEX(KL))                                           CAL00720
      RELPRP(KLNDX) = 0.                                                CAL00730
C                                                                       CAL00740
C  RESET INDEX FOR FATHER, TO NEXT LEVEL UP                             CAL00750
      KFATH=LSUPER(KL)                                                  CAL00760
C                                                                       CAL00770
C  KEEP ON PROCESSING NODES UNTIL TOP NODE IS REACHED                   CAL00780
      IF(KL.NE.KROT) GO TO 131                                          CAL00790
```

```
FILE: CALRPP   FORTRAN  A                    CONVERSATIONAL MONITOR SYSTEM

C                                                                      CAL00H00
C       WRITE (3,9992) (RELPRP(I),I=1,7)                               CAL00H10
9992    FORMAT (' LEAVING CALRPR, REL PROP =',7F6.2)                   CAL00H20
        RETURN                                                         CAL00H30
        END                                                            CAL00H40
```

```
      IMPLICIT INTEGER (A-Z)                                                    IMA00010
C                                                                               IMA00020
      DIMENSION FLDINF(6), IDATA(1), FL(12)                                     IMA00030
C                                                                               IMA00040
      DIMENSION IBUFF(196)                                                      IMA00050
C                                                                               IMA00060
C                                                                               IMA00070
      COMMON /INFORM/HEAD(42), MAPTAP,     DATAPE,     SAVTAP,   MAXFET,IMA00080
     1                 PAGSIZ, TAPCHK,     TRNSYM,     TSTSYM,          IMA00090
     2                 DUPSYM, TMPSYM,     MAXDIV,     MINDIV,   SPLMAX,IMA00100
     3                 SERIAL, TAPESV,     FILESV,                      IMA00110
     4                 MAXCLS, NOCLS2,     MAXFLD,     NOFLD2,   NOFLD3,IMA00120
     5                 NOTWRD, NOFEAT,     NOFET2,     NOFET4,   VARSIZ,IMA00130
     6                 VARS72, VARS74,     XSIZ  ,     NOSPEC,   NOHIST,IMA00140
     *                 NOGRP, DIVSIZ,    KEEPLV,    PRTLEV,    YSIZ  ,IMA00150
     8                 XHGH  , XLOW  ,     SPCHAS,     NOCLS3,   PCTSZ ,IMA00160
     9IBLOCK(30),FETVEC(30),FETVC2(30), HISVEC(30),INVERT(30),HESTVC(30)IMA00170
C                                                                               IMA00180
      COMMON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYM(61) ,          IMA00190
     1               LNCAT, PRNT(4) , KLBC , PRTME, PROUT, TOTPIX,      IMA00200
     2  SCRAM1,BUFPIX,BUFTOT,NBUFSD,NOUMP,LBUFD                        IMA00210
     3, MAXRF, AREA, NWDS, NWDRS, NPTS, LBUF, I01,NOCYCL                IMA00220
C                                                                               IMA00230
      INTEGER TOTWRD,SYM,PRNT,PRTME,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT   IMA00240
     1 ,CLSNAM                                                          IMA00250
C                                                                               IMA00260
C                                                                               IMA00270
      EQUIVALENCE (FLDINF(1),LINSTR), (FLDINF(4),SAMSTR),               IMA00280
     *            (FLDINF(2),LINEND), (FLDINF(5),SAMEND),               IMA00290
     *            (FLDINF(3),LININC), (FLDINF(6),SAMINC)                IMA00300
C                                                                               IMA00310
         DIMENSION ARRAY(18000)                                                 IMA00320
C*                                                                              IMA00330
C*    RESERVE 2000 LOCATIONS OF 'ARRAY' FOR FIELD DEFINITION INFORMATION.       IMA00340
C*    THE REMAINDER OF 'ARRAY' IS USED FOR I/O BUFFERS.                         IMA00350
C*                                                                              IMA00360
C*    FIELD INFORMATION STORED AS FOLLOWS                                       IMA00370
C*       ARRAY(1)    =FIRST FIELD NAME FOR THIS CLASS                           IMA00380
C*            (2)    =NO. OF VERTICES FOR THIS FIELD  (NV)                      IMA00390
C*            (3)-(3+NV*2)  = ACTUAL VERTEX NUMBERS                             IMA00400
C*            (3+NV*2)  =TOTAL PIXELS IN THIS FIELD)                            IMA00410
C*            (4+NV*2)-(10+NV*2) = FLDINF BLOCK FOR THIS FIELD                  IMA00420
C                                                                               IMA00430
        CALL SETUPM                                                             IMA00440
C                                                                               IMA00450
        CALL KREDTP                                                             IMA00460
C                                                                               IMA00470
      STOP                                                                      IMA00480
      END                                                                       IMA00490
      SUBROUTINE KREDTP                                                         IMA00500
C                                                                               IMA00510
      IMPLICIT INTEGER (A-Z)                                                    IMA00520
C                                                                               IMA00530
C  PURPOSE: MOVE DATA FROM A UNIVERSAL FORMAT TAPE TO A TEMP DISK FILE          IMA00540
C*     FROM THE IMAGE TAPE                                                      IMA00550
C                                                                               IMA00560
      DIMENSION FLDINF(6),   FL(12)                                             IMA00570
C                                                                               IMA00580
      DIMENSION IBUFF(3200), IDATA(4200)                                        IMA00590
C                                                                               IMA00600
C                                                                               IMA00610
      COMMON /INFORM/HEAD(42), MAPTAP,     DATAPE,     SAVTAP,   MAXFET,IMA00620
     1                 PAGSIZ, TAPCHK,     TRNSYM,     TSTSYM,          IMA00630
     2                 DUPSYM, TMPSYM,     MAXDIV,     MINDIV,   SPLMAX,IMA00640
     3                 SERIAL, TAPESV,     FILESV,                      IMA00650
     4                 MAXCLS, NOCLS2,     MAXFLD,     NOFLD2,   NOFLD3,IMA00660
     5                 NOTWRD, NOFEAT,     NOFET2,     NOFET4,   VARSIZ,IMA00670
     6                 VARS72, VARS74,     XSIZ  ,     NOSPEC,   NOHIST,IMA00680
     *                 NOGRP, DIVSIZ,    KEEPLV,    PRTLEV,    YSIZ  ,IMA00690
     8                 XHGH  , XLOW  ,     SPCHAS,     NOCLS3,   PCTSZ ,IMA00700
     9IBLOCK(30),FETVEC(30),FETVC2(30), HISVEC(30),INVERT(30),HESTVC(30)IMA00710
C                                                                               IMA00720
      COMMON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYM(61) ,          IMA00730
     1               LNCAT, PRNT(4) , KLBC , PRTME, PROUT, TOTPIX,      IMA00740
     2  SCRAM1,BUFPIX,BUFTOT,NBUFSD,NOUMP,LBUFD                        IMA00750
     3, MAXRF, AREA, NWDS, NWDRS, NPTS, LBUF, I01,NOCYCL                IMA00760
C                                                                               IMA00770
      INTEGER TOTWRD,SYM,PRNT,PRTME,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT   IMA00780
     1 ,CLSNAM                                                          IMA00790
```

```
C                                                                       IMA00900
      COMMON /FILE/ NOFSKP                                              IMA00910
C                                                                       IMA00920
      EQUIVALENCE (FLDINF(1),LINSTR), (FLDINF(4),SAMSTR),               IMA00930
     *            (FLDINF(2),LINEND), (FLDINF(5),SAMEND),               IMA00940
     *            (FLDINF(3),LININC), (FLDINF(6),SAMINC)                IMA00950
C                                                                       IMA00960
         DIMENSION ARRAY(14000)                                         IMA00970
C*                                                                      IMA00980
C*    RESERVE 2000 LOCATIONS OF 'ARRAY' FOR FIELD DEFINITION INFORMATION.IMA00990
C*    THE REMAINDER OF 'ARRAY' IS USED FOR I/O BUFFERS.                 IMA01000
C*                                                                      IMA01010
C*    FIELD INFORMATION STORED AS FOLLOWS                               IMA01020
C*       ARRAY(1)    =FIRST FIELD NAME FOR THIS CLASS                   IMA01030
C*            (2)    =NO. OF VERTICES FOR THIS FIELD  (NV)              IMA01040
C*            (3)-(3+NV*2)  = ACTUAL VERTEX NUMBERS                     IMA01050
C*            (3+NV*2)  =TOTAL PIXELS IN THIS FIELD                     IMA01060
C*            (4+NV*2)-(10+NV*2) = FLDINF BLOCK FOR THIS FIELD          IMA01070
C                                                                       IMA01080
C  DEFINE INPUT FILE TO BE UNIT 11                                      IMA01090
      DEFINE FILE 22(2100,200,U,ID)                                     IMA01000
      DATAPE = 11                                                       IMA01010
      CALL TAPHDR(DATAPE,NOFSKP)                                        IMA01020
C                                                                       IMA01030
      NOFLD=0                                                           IMA01040
      IPT=1                                                             IMA01050
C*                                                                      IMA01060
C  READ (1) FIELD NAME, (2) FIELD VERTICES, (3) FIELD INFORMATION       IMA01070
C  (4) NUMBER VERTICES, AND RETURN (5) NUMBER OF CARDS READ             IMA01080
20    ICK = LAREAD(ARRAY(IPT),ARRAY(IPT+2),FLDINF,ARRAY(IPT+1), NOCRDS)  IMA01090
      IF=ARRAY(IPT+1)                                                   IMA01100
      NOFLD=NOFLD+1                                                     IMA01110
      NSAMP=(SAMEND-SAMSTR)/SAMINC+1                                    IMA01120
      FLDSAM=0                                                          IMA01130
      IB=IPT+2                                                          IMA01140
      IE=IB+NV*2-1                                                      IMA01150
      WRITE (6,1600) NOFLD,ARRAY(IPT),NV,SAMINC,LININC,                 IMA01160
     *            (ARRAY(I),I=IB,IE)                                    IMA01170
C*                                                                      IMA01180
C*    POSITION TAPE FOR THIS FIELD                                      IMA01190
      CALL FLDINT(FLDINF,FETVEC,NOFEAT)                                 IMA01200
C                                                                       IMA01210
      KNT=0                                                             IMA01220
C                                                                       IMA01230
C  READ LINES OF DATA FROM UNIVERSAL FORMAT TAPE AND MOVE TO DISK       IMA01240
      DO 70 LINE=LINSTR,LINEND,LININC                                   IMA01250
       LINENO = LINE                                                    IMA01260
C                                                                       IMA01270
C  READ LINE FROM UNIVERSAL FORMAT TAPE (ENDTAP = -1 IF LAST LINE)      IMA01280
      CALL LINERD(IDATA,ENDTAP)                                         IMA01290
      IF(ENDTAP.EQ.-1)GO TO 900                                         IMA01300
C*                                                                      IMA01310
C*    FIND SAMPLE INTERSECTS FOR THIS LINE - NI=NO. OF INTERSECTS       IMA01320
      CALL FDINT(ARRAY(IPT+2),NV,FL,LINE,SAMPS,NI)                      IMA01330
C                                                                       IMA01340
C  WRITE DUMMY FIRST RECORD                                             IMA01350
      NOLNES = (LINEND - LINSTR) / LININC + 1                           IMA01360
C                                                                       IMA01370
C*                                                                      IMA01380
C*    STORE DATA ON THIS LINE INTO OUTPUT BUFFER                        IMA01390
      DO 60 I=1,NI,2                                                    IMA01400
      IB =(FL(I)-SAMSTR)/SAMINC + 1                                     IMA01410
      IE =(FL(I+1)-SAMSTR)/SAMINC + 1                                   IMA01420
      IF(MOD(SAMSTR,SAMINC) .NE. MOD(FL(I),SAMINC))IB=IB+1              IMA01430
      IF(IB .GT. IE)GO TO 60                                            IMA01440
C                                                                       IMA01450
      IWRD = 0                                                          IMA01460
      DO 50 J=IB,IE                                                     IMA01470
      KNT=KNT+1                                                         IMA01480
      DO 50 K=1,NOFEAT                                                  IMA01490
      IWRD=IWRD + 1                                                     IMA01500
      ITEMP = (K-1)*NSAMP + J                                           IMA01510
      IBUFF(IWRD) = IDATA(ITEMP)                                        IMA01520
C                                                                       IMA01530
   50 CONTINUE                                                          IMA01540
C                                                                       IMA01550
C  WRITE LINE ON TEMP DISK FILE                                         IMA01560
      WRITE (24) IWRD, LINENO, IB, IE, (IBUFF(II),II=1,IWRD)           IMA01570
9924   FORMAT (I5,4013)                                                 IMA01580
```

```
   60 CONTINUE                                                                    IMA01590
      FLDSAM=FLDSAM+SAMPS                                                         IMA01600
70    CONTINUE                                                                    IMA01610
71      CONTINUE                                                                  IMA01620
C                                                                                 IMA01630
C  WRITE MESSAGE TO USER                                                          IMA01640
      WRITE (3, 75) LINENO                                                        IMA01650
75    FORMAT (' IMAGE DATA EXTRACTED FROM TAPE, LAST LINE = ',I4)                 IMA01660
C                                                                                 IMA01670
C                                                                                 IMA01680
      RETURN                                                                      IMA01690
C                                                                                 IMA01700
800    WRITE (6,1400)                                                             IMA01710
      STOP                                                                        IMA01720
C                                                                                 IMA01730
1100  FORMAT(10X,A4)                                                              IMA01740
1300  FORMAT(' FIELD DEFINITION INFORMATION EXCEEDS 2000 WORDS')                  IMA01750
1400  FORMAT(' END-OF-TAPE REACHED BEFORE END OF FIELD')                          IMA01760
C                                                                                 IMA01770
1600  FORMAT(1X,I2,4X,A4,12X,I2,10X,I2,6X,I2,5X,                                 IMA01780
     *  5('(',I4,',',I4,')',2X)/2(52X,5('(',I4,',',I4,')',2X)/))                  IMA01790
      END                                                                         IMA01800
      SUBROUTINE SETUP9                                                           IMA01810
C                                                                                 IMA01820
CCC   PURPOSE: READ AND ANALYZE CARDS DESCRIPTING UNIVERSAL FORMAT FILES TO       IMA01830
CCC   SKIP AND LINES OF DATA TO EXTRACT FROM SELECTED FILE                        IMA01840
CCC                                                                               IMA01850
CCC                                                                               IMA01860
      IMPLICIT INTEGER (A-X)                                                      IMA01870
C                                                                                 IMA01880
      COMMON /TIMERR/ ORGTIM, TIMEMX                                              IMA01890
      REAL ORGTIM, TIMEMX                                                         IMA01900
C                                                                                 IMA01910
CC                                                                                IMA01920
      COMMON /MAP/ MAP(10),LINES(10,2),MAPCT,LINECT,KA                            IMA01930
      COMMON /INFORM/HEAD(42), MAPTAP,     DATAPL,     SAVTAP,    MAXFET,IMA01940
     1                 PAGSIZ, TAPCHK,     TMNSYM,     TSTSYM,            IMA01950
     2                 DUPSYM, THNSYM,     MAXDIV,     MINDIV,    SPLMAX,IMA01960
     3                 SPCIAL, TAPESV,     FILESV,                        IMA01970
     4                 MAXCLS, NOCLS2,     MAXFLD,     NOFLD2,    NOFLD3,IMA01980
     5                 NOTFND, NOFEAT,     NOFET2,     NOFET4,    VARSIZ,IMA01990
     6                 VARSZ2, VARSZ4,     XSIZ  ,     NOSPEC,    NOHIST,IMA02000
     *                 NOGRP, DIVSIZ,    KEEPLV,    PRTLEV,    YSIZ  ,  IMA02010
     4                 XHGH  , ALOW  ,     SPCHAS,     NOCLS3,    PCTSZ ,IMA02020
     9IBLOCK(30),FETVEC(30),FETVC2(30), HISVEC(30),INVERT(30),HESTVC(30)IMA02030
      DIMENSION  HED1(10),  HED2(10),  DATE(2),  COMENT(10),  TEMP(10)            IMA02040
      EQUIVALENCE  (HED1(1),HEAD(3)),     (DATE(1),  HEAD(15)),                   IMA02050
     *             (HED2(1),HEAD(20)),     (COMENT(1),HEAD(32))                 IMA02060
      COMMON /SUPCON/ INTAPE,STATUS,COL,CODE1,CODE2,CARD(62),ACARD(20)            IMA02070
C                                                                                 IMA02080
      COMMON/CLUSTR/ IBEGIN,TOT,PO,CLSNAM,IPT,NOFLD, SYM(61) ,                    IMA02090
     1               LNCAT, PRNT(4) , KLBC , PRTHE, PROUT, TOTPIX,              IMA02100
     2  SCRAM1,BUFPIX,BUFTOT,NOFLDS,NOOMP,LBUFD)                                 IMA02110
     3, MAXEF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1,NOCYCL                          IMA02120
C                                                                                 IMA02130
      INTEGER TOT,PO,SYM,PRNT,PRTHE,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT             IMA02140
     1 ,CLSNAM                                                                    IMA02150
CC                                                                                IMA02160
CC                                                                                IMA02170
      INTEGER SEG                                                                 IMA02180
      COMMON/PRTAP/ICOUNT,FORMT,UNIT,VARBL(600),IREMO                             IMA02190
      EQUIVALENCE (VARBL(23),SEG)                                                 IMA02200
      INTEGER PROCD(3)                                                            IMA02210
      EQUIVALENCE (VARBL(20),PROCD(1)),(VARBL(21),PROCD(2)),                      IMA02220
     1(VARBL(22),PROCD(3))                                                        IMA02230
      COMMON/FILE/DATFIL                                                          IMA02240
C                                                                                 IMA02250
CC                                                                                IMA02260
      DIMENSION SYMBLS(61)                                                        IMA02270
      DATA  SYMBLS/'*','=','/','A','+','$','V','-','<','X','*','0','H',          IMA02280
     *             'E','>','1','2','S','3','K','L','9','N','O','P','O',        IMA02290
     2             'U','5','T','6','V','W','X','Y','Z','_','|',',','/',        IMA02300
     3             '-','#','+','S','M','=','$',' ','=','?','(','$','.',        IMA02310
     4             'R','>','$','?','1H','.','.','.','*',' ','0',' '/            IMA02320
      DATA BLANK/' '/, KOMMA/','/                                                 IMA02330
      DIMENSION INVEC(17)                                                         IMA02340
      DATA INVEC/'CHAN','HED1','HED2','DATE','COMM','SEGM','PROC',               IMA02350
     * 'NPTS','NPOS','SYMB','PRIN','DATA','ITER','MAP ',                         IMA02360
     * 'LINE','TIME','END'/                                                      IMA02370
```

```
      DIMENSION EQUIVEC(2)                                                  IMA02380
      DATA KKA/'A'/,KHYPHN/'-'/                                             IMA02390
      DATA DRCD/'D'/,FRCD/'F'/,EQUIVEC/1,'='/                               IMA02400
      VARBL(19) = 242                                                       IMA02410
      ICNT=0                                                                IMA02420
      K0 = 0                                                                IMA02430
      KA = KKA                                                              IMA02440
      ZERO = 0                                                              IMA02450
C  ASSUME 'ALL' FOR MAP AND LINES CARDS                                     IMA02460
      LINECT = 1                                                            IMA02470
      MAPCT = 1                                                             IMA02480
      LINES(1,1) = KA                                                       IMA02490
      MAP(1) = KA                                                           IMA02500
      DO 5 I=1,61                                                           IMA02510
    5 SYM(I)=SMBLS(I)                                                       IMA02520
      NOFEAT=0                                                              IMA02530
      WRITE(6,600)                                                          IMA02540
600   FORMAT (1H1, ' EXTRACT IMAGE DATA FROM UNIVERSAL FORMAT TAPE')        IMA02550
      WRITE(6,630)                                                          IMA02560
C                                                                           IMA02570
C       PUT THE NEXT CARD IN THE PREREAD BUFFER                             IMA02580
C                                                                           IMA02590
      PRUNIT=30                                                             IMA02600
   10 READ(21,1000)(ACARD(I),I=1,20)                                        IMA02610
 1000 FORMAT(20A4)                                                          IMA02620
      WRITE(PRUNIT,1000)(ACARD(I),I=1,20)                                   IMA02630
      REWIND PRUNIT                                                         IMA02640
C                                                                           IMA02650
CC                                                                          IMA02660
CC     READ IN CARD                                                         IMA02670
C                                                                           IMA02680
      READ(30,480) CODE,CARD                                                IMA02690
      REWIND PRUNIT                                                         IMA02700
      WRITE(6,550) CODE,CARD                                                IMA02710
      COL = 0                                                               IMA02720
C  SET NUMBER OF VALID CARD TYPES                                           IMA02730
      CNUM = 17                                                             IMA02740
C  DETERMINE CARD TYPE                                                      IMA02750
      DO 20 I=1,CNUM                                                        IMA02760
      IF(CODE .EQ. INVEC(I))GO TO(30,50,70,90,110,120,130,140,150,170,      IMA02770
     * 190,250,270,300,350,400,410),I                                       IMA02780
   20 CONTINUE                                                              IMA02790
C                                                                           IMA02800
C  INVALID CARD TYPE                                                        IMA02810
      WRITE(6,490)CODE,CARD                                                 IMA02820
      GO TO 10                                                              IMA02830
C                                                                           IMA02840
CC     CHANNEL CARD                                                         IMA02850
C                                                                           IMA02860
   30 J = NXTCHR(CARD,COL)                                                  IMA02870
      IF (J .EQ. BLANK)GO TO 10                                             IMA02880
      COL=COL-1                                                             IMA02890
      NOFEAT = NUMBER(CARD,COL,FETVEC,NOFEAT)                               IMA02900
      VARSIZ=(NOFEAT*(NOFEAT+1))/2                                          IMA02910
      GO TO 10                                                              IMA02920
C                                                                           IMA02930
CC  HED1 CARD                                                               IMA02940
C                                                                           IMA02950
   50 READ (30,500)HED1                                                     IMA02960
      GO TO 10                                                              IMA02970
C                                                                           IMA02980
CC  HED2 CARD                                                               IMA02990
C                                                                           IMA03000
   70 READ (30,500) HED2                                                    IMA03010
      REWIND PRUNIT                                                         IMA03020
      GO TO 10                                                              IMA03030
C                                                                           IMA03040
CC  DATE CARD                                                               IMA03050
C                                                                           IMA03060
   90 READ(30,510)DATE                                                      IMA03070
      REWIND PRUNIT                                                         IMA03080
      GO TO 10                                                              IMA03090
C                                                                           IMA03100
CC  COMMENT CARD                                                            IMA03110
C                                                                           IMA03120
  110 READ(30,500)COMENT                                                    IMA03130
      REWIND PRUNIT                                                         IMA03140
      GO TO 10                                                              IMA03150
C                                                                           IMA03160
```

```
C SEGM CARD                                                                    IMA03170
C                                                                              IMA03180
120     J=NXTCHR(CARD,COL)                                                     IMA03190
        IF(J .EQ. BLANK) GO TO 10                                              IMA03200
        COL=COL-1                                                              IMA03210
        J=NUMBER(CARD,COL,SEG,0)                                               IMA03220
        GO TO 10                                                               IMA03230
C                                                                              IMA03240
C PROCESSING DATE CARD                                                         IMA03250
C                                                                              IMA03260
130     J=NXTCHR(CARD,COL)                                                     IMA03270
        IF(J .EQ. BLANK) GO TO 10                                              IMA03280
        COL=COL-1                                                              IMA03290
        J=NUMBER(CARD,COL,PROCD,0)                                             IMA03300
        GO TO 10                                                               IMA03310
C                                                                              IMA03320
CC  NPTS = NUMBER OF POINTS TO SKIP IN ANALYSIS                                IMA03330
C                                                                              IMA03340
140     J = NUMBER(CARD,COL,NPTS,0)                                            IMA03350
        NPTS = NPTS - 1                                                        IMA03360
        GO TO 10                                                               IMA03370
C                                                                              IMA03380
CC  NPOS CARD,NUMBER OF DRUM POSITIONS FROM WHICH TO OBTAIN DATA FOR           IMA03390
CC  CLASY3,SO THAT THE DATA WILL BE SCRAMBLED                                  IMA03400
C                                                                              IMA03410
150   J=NUMBER(CARD,COL,NPOS,K0)                                               IMA03420
        GO TO 10                                                               IMA03430
CC                                                                             IMA03440
CC   SYMBOL CARD                                                              IMA03450
C                                                                              IMA03460
170   ICNT=ICNT+1                                                              IMA03470
        IF(ICNT .GT. 61) GO TO 10                                              IMA03480
180   M=NXTCHR(CARD,COL)                                                       IMA03490
        IF( M .EQ. BLANK)GO TO 10                                              IMA03500
        IF(M .EQ. KOMMA)GO TO 180                                              IMA03510
        SYM(ICNT)=M                                                            IMA03520
        GO TO 170                                                              IMA03530
C                                                                              IMA03540
CC   PRINT OPTION CARD NO LONGER VALID                                         IMA03550
C                                                                              IMA03560
190     GO TO 10                                                               IMA03570
CCCCC                                                                          IMA03580
        DATA FILE CARD                                                         IMA03590
C                                                                              IMA03600
260   M = NXTCHR(CARD,COL)                                                     IMA03610
        IF(M.EQ.BLANK) GO TO 10                                                IMA03620
        IF (M.EQ.DASH) GO TO 265                                               IMA03630
        IF(M.EQ.EQUAL) GO TO 267                                               IMA03640
263   WRITE(6,750)                                                             IMA03650
750   FORMAT(' ERROR ON DATA FILE CARD')                                       IMA03660
        GO TO 10                                                               IMA03670
265   J = FINDIT(CARD,COL,EQUIVC)                                              IMA03680
        IF (J.EQ. -1)  GO TO 263                                               IMA03690
        M = NUMBER(CARD,COL,DATAFL,ZERO)                                       IMA03700
        COL = COL - 1                                                          IMA03710
        GO TO 261                                                              IMA03720
267   J = FINDIT(CARD,COL,EQUIVC)                                              IMA03730
        IF (J .EQ. -1)  GO TO 263                                              IMA03740
        M = NUMBER(CARD,COL,DATFIL,ZERO)                                       IMA03750
        DATFIL = DATFIL - 1                                                    IMA03760
        IF (DATFIL .LT. 0)  DATFIL = 0                                         IMA03770
        COL = COL - 1                                                          IMA03780
        GO TO 260                                                              IMA03790
C  ITERATION CARD                                                              IMA03800
270     J = NXTCHR (CARD,COL)                                                  IMA03810
        IF (J .EQ. BLANK) GO TO 10                                             IMA03820
        COL = COL - 1                                                          IMA03830
        K = NUMBER (CARD,COL,TEMP,K0)                                          IMA03840
        NOCYCL = TEMP(1)                                                       IMA03850
        GO TO 10                                                               IMA03860
C                                                                              IMA03870
C  MAP CARD--MAY SPECIFY SERIES OF ITERATIONS TO BE MAPPED OR 'ALL'            IMA03880
C     EXAMPLES:  1,3,5,7,9          'ALL'                                      IMA03890
300     DO 301 I = 1,10                                                        IMA03900
301     MAP(I) = 0                                                             IMA03910
        J = NXTCHR (CARD,COL)                                                  IMA03920
        IF ( J .EQ. BLANK) GO TO 10                                            IMA03930
C  IF 'ALL' WAS REQUESTED, GO READ NEXT CARD                                   IMA03940
                                                                               IMA03950
```

```
      IF (J .EQ. KA) GO TO 10                                                    IMA03960
C                                                                                IMA03970
C  SERIES OF NUMBERS REQUESTED                                                   IMA03980
      COL = J                                                                    IMA03990
      MAPCT = NUMBER (CARD, COL, MAP, K0)                                        IMA04000
      GO TO 10                                                                   IMA04010
C                                                                                IMA04020
C                                                                                IMA04030
C  LINES CARD--LINES TO BE PRINTED.  NUMBERS OR 'ALL' MAY BE SPECIFIED           IMA04040
C    EXAMPLES:  'ALL'      1,26,51      1-4,26-29,51-54                          IMA04050
350   DO 351 I = 1,10                                                           IMA04060
      DO 351 J = 1,2                                                             IMA04070
351   LINES(I,J) = 0                                                             IMA04080
C                                                                                IMA04090
C  GET FIRST NONBLANK CHARACTER                                                  IMA04100
      J = NXTCHR (CARD,COL)                                                      IMA04110
      IF (J .EQ. BLANK) GO TO 10                                                 IMA04120
      IF (J .EQ. KA) GO TO 10                                                    IMA04130
C                                                                                IMA04140
C  SERIES OF NUMBERS SPECIFIED                                                   IMA04150
355   COL = COL - 1                                                              IMA04160
C  UP TO 10 NUMBERS OR SERIES MAY BE SPECIFIED                                   IMA04170
      NEXT = 1                                                                   IMA04180
C  MOVE SERIES OF NUMBERS TO ARRAY 'TEMP', STORE COUNT IN KOUNT                  IMA04190
      K = 1                                                                      IMA04200
      KOUNT = NUMBER (CARD, COL, TEMP,K0)                                        IMA04210
      N = 1                                                                      IMA04220
C  NUMBER OF SERIES OF NUMBERS SEPARATED BY COMMAS                               IMA04230
361   CONTINUE                                                                   IMA04240
      DO 365 II = N,KOUNT                                                        IMA04250
      LINES(NEXT,1) = TEMP(II)                                                   IMA04260
      LINES(NEXT,2) = TEMP(II)                                                   IMA04270
365   NEXT = NEXT + 1                                                            IMA04280
C  SERIES COULD BE TERMINATED BY BLANK OR HYPHEN                                 IMA04290
      COL = COL - 1                                                              IMA04300
      J = NXTCHR (CARD,COL)                                                      IMA04310
      IF (J .EQ. BLANK) GO TO 390                                                IMA04320
C  SERIES TERMINATED BY HYPHEN                                                   IMA04330
      NEXT = NEXT - 1                                                            IMA04340
      KOUNT = NUMBER (CARD, COL, TEMP, K0)                                       IMA04350
      LINES(NEXT,2) = TEMP(1)                                                    IMA04360
      NEXT = NEXT + 1                                                            IMA04370
      IF (KOUNT .EQ. 1) GO TO 390                                                IMA04380
C  SERIES OF FORM 1,25-29,51   GO TO PROCESS NEXT NUMBER AFTER HYPHEN AS         IMA04390
C  BEGINNING OF NEXT GROUP OF LINES                                              IMA04400
      N = 2                                                                      IMA04410
      GO TO 361                                                                  IMA04420
C                                                                                IMA04430
390   LINECT = NEXT - 1                                                          IMA04440
      GO TO 10                                                                   IMA04450
C                                                                                IMA04460
C                                                                                IMA04470
C  SET MAXIMUM TIME                                                              IMA04480
400   J = NUMBER (CARD, COL, ITIME, 0)                                           IMA04490
      TIMEMX = ITIME                                                             IMA04500
       GO TO 10                                                                  IMA04510
C                                                                                IMA04520
C   *END* CARD                                                                   IMA04530
C                                                                                IMA04540
 410  RETURN                                                                     IMA04550
C                                                                                IMA04560
C  FORMATS                                                                       IMA04570
 480 FORMAT(A4,4X,62A1)                                                          IMA04580
 490 FORMAT(' INVALID INPUT CARD--IGNORED'/1X,A4,4X,62A1)                        IMA04590
 500 FORMAT(10X,1046)                                                            IMA04600
 510 FORMAT(10X,246)                                                             IMA04610
 550 FORMAT(5X,A6,4X,62A1)                                                       IMA04620
 630 FORMAT(///' INPUT SUMMARY'//)                                               IMA04630
      STOP                                                                       IMA04640
      END                                                                        IMA04650
```

```
C  PURPOSE:  ESTIMATE THE PROBABILITY OF OBSERVING A PARTICULAR          MAX00010
C           LABELED CLASS GIVEN THAT A PARTICULAR CLUSTER HAS           MAX00020
C           BEEN OBSERVED                                               MAX00030
C                                                                       MAX00040
C                                                                       MAX00050
C  SAVE OPTIONS FOR PRINTING MAP                                        MAX00060
      COMMON/FILE/DATFIL, MAPOPT, NOCHAN                                MAX00070
      COMMON  MO, NOCC, NOCAT, ITOTDT, MOP4, MOS, ICC,                  MAX00080
     1  ICLUNT, IOUPUT, IOUT, CATNAM(20), DCON,                         MAX00090
     2 IDOTS(20,209), CVOLPT, RLPRP( 30), CMEANS(15), CVRIN(15R),       MAX00100
     3 PX(30,209), MAP(30), DUTOT(15), LABELS(30),LBLCSS(30),           MAX00110
     4 BETA(20,30),SIGBETA(20,30),SLK(20,30),SK(30),LBLCST(30)          MAX00120
      DIMENSION ICSS(30)                                                MAX00130
C                                                                       MAX00140
      LOGICAL*1 LABEL(30), LABELS, LBLCSS, LBLCST, IBLNK                MAX00150
      DATA LABEL/'1','2','3','4','5','6','7','8','9','A',              MAX00160
     1 'B','C','D','E','F','G','H','I','J','K',                         MAX00170
     2 'L','M','N','O','P','Q','R','S','T','U'/                         MAX00180
        DATA IBLNK/' '/                                                 MAX00190
C                                                                       MAX00200
       DO 1 I = 1,20                                                    MAX00210
 1    LABELS(I) = LABEL(I)                                              MAX00220
C                                                                       MAX00230
C  CALL SETUPM FOR OPTIONS                                              MAX00240
      CALL SETUPM                                                       MAX00250
C                                                                       MAX00260
      ICLUNT = 23                                                       MAX00270
      NOTIFY = 1                                                        MAX00280
      NOCC = 4                                                          MAX00290
      IOUT = 6                                                          MAX00300
      IOUNT = 9                                                         MAX00310
      IUNT = 5                                                          MAX00320
C                                                                       MAX00330
C  SET NUMBER OF CHANNELS FROM 'NOCHAN' SET IN SETUPM                   MAX00340
      MO=NOCHAN                                                         MAX00350
      MOP4 = MO + 4                                                     MAX00360
      MOS = MO**2                                                       MAX00370
C                                                                       MAX00380
C  READ DOT DATA FILE  RECORD 1                                         MAX00390
      READ (IOUNT) NOCAT, NOFEAT, NOFLD, ITOTVT, ITOTDT, NOSUN,        MAX00400
     1        (CATNAM(I), I = 1,NOCAT), ISIZE                           MAX00410
      WRITE (IOUT,45) NOCAT, NOFEAT, NOFLD,  ITOTDT,                   MAX00420
     1  (CATNAM(I),I=1,NOCAT)                                          MAX00430
 45   FORMAT (' NUMBER OF CATEGORIES=',I2,/,' NUMBER OF CHANNELS  =',I1,MAX00440
     1 ' TOTAL NUMBER OF DOTS=',I4,/,' CATEGORY NAMES=',(20(A1,2X)))//) MAX00450
C                                                                       MAX00460
C  SKIP DOT DATA FILE  RECORD 2                                         MAX00470
      READ (IOUNT) I                                                    MAX00480
C                                                                       MAX00490
C  READ DOT DATA FILE  RECORD 3                                         MAX00500
      READ (IOUNT) ((IDOTS(I,J), I=1,ISIZE), J=1,ITOTDT)                MAX00510
C  WRITE DOT DATA                                                       MAX00520
      WRITE (IOUT,44)                                                   MAX00530
44     FORMAT (///,'  COL   ROW                  RADIANCE VALUES',/)    MAX00540
C                                                                       MAX00550
C  PRINT RADIANCE VALUES                                                MAX00560
      DO 47 J = 1,ITOTDT                                                MAX00570
      WRITE (IOUT,46) (IDOTS(I,J),I=1,ISIZE)                            MAX00580
46    FORMAT (I2I4)                                                     MAX00590
47    CONTINUE                                                          MAX00600
C                                                                       MAX00610
C  CHECK   CLASSY AND DOT DATA FILE MUST HAVE SAME NUMBER OF CHANNELS   MAX00620
      IF (MO .EQ. NOFEAT) GO TO 100                                     MAX00630
C                                                                       MAX00640
C                                                                       MAX00650
      WRITE (IOUT,50) MO, NOFEAT                                        MAX00660
 50    FORMAT (//,' NUMBER OF CLASSY CHANNELS = ',I2,                   MAX00670
     1    '   NUMBER OF DOT DATA CHANNELS = ', I2,                      MAX00680
     2 //, ' EXECUTION TERMINATED')                                     MAX00690
      STOP                                                              MAX00700
C                                                                       MAX00710
C                                                                       MAX00720
C                                                                       MAX00730
C  READ VALUES FROM CLASSY                                              MAX00740
100   REWIND ICLUNT                                                     MAX00750
      CALL READCC                                                       MAX00760
C                                                                       MAX00770
C  INITIALIZE BETA VALUES                                               MAX00780
                                                                        MAX00790
```

```
      DO 110 J = 1,NOCC                                                  MAX00800
      DO 110 L = 1,NOCAT                                                 MAX00810
      BETA(L,J) = 1./NOCAT                                               MAX00820
  110  CONTINUE                                                          MAX00830
C                                                                        MAX00840
CCCC                                                                     MAX00850
C   ----------BEGINNING OF ITERATIVE LOOP----------                      MAX00860
C                                                                        MAX00870
  200 CONTINUE                                                           MAX00880
C  SAVE OLD BETA VALUES FOR COMPARISION LATER                            MAX00890
      DO 210 J =1,NOCC                                                   MAX00900
      DO 210 L =1,NOCAT                                                  MAX00910
  210  OBETA(L,J) = BETA(L,J)                                            MAX00920
C                                                                        MAX00930
C    CALC. ALL P(X,I) VALUES FOR ALL DOT DATA-CLUSTER COMBINATIONS       MAX00940
      CALL ALLPXI                                                        MAX00950
C                                                                        MAX00960
CCC  CALC. PRODUCTS FOR ALL PRIOR EST (BETA), CLASY REL PROP. (RELPRP),  MAX00970
CCC    AND CLASS CONDITIONAL PROBABILITIES (P(X,I))                      MAX00980
CCC  CALC. S(L,K) AS SUM OF (BETA*ALPHA*PR(X,I))/SUM ALL BETA*ALPHA*P(X,I)MAX00990
C                                                  FOR ALL CLUSTERS      MAX01000
      CALL BAYES                                                         MAX01010
C                                                                        MAX01020
C                                                                        MAX01030
C    RE-CALCULATE BETAS, SET TO 0. IF LESS THAN .01                      MAX01040
      DO 300 K = 1,NOCC                                                  MAX01050
      DO 300 L = 1,NOCAT                                                 MAX01060
C                                                                        MAX01070
      BETA(L,K) = SLK(L,K) / SK(K)                                       MAX01080
      IF (BETA(L,K) .LT. .01) BETA (L,K) = 0.                            MAX01090
C                                                                        MAX01100
  300  CONTINUE                                                          MAX01110
C                                                                        MAX01120
CC  PRINT THE CURRENT VALUE OF THE LIKELIHOOD FUNCTION                   MAX01130
CCCC     CALL CALFCT                                                     MAX01140
C                                                                        MAX01150
CCC                                                                      MAX01160
C  FIND LARGEST DIFFERCE IN OLD AND NEW BETA,S                           MAX01170
      XMAX = 0.                                                          MAX01180
      DO 500 J = 1,NOCC                                                  MAX01190
      DO 500 L = 1,NOCAT                                                 MAX01200
      ABSDIF=OBETA(L,J) - BETA(L,J)                                      MAX01210
      IF (ABSDIF  .GT. XMAX)     XMAX = ABSDIF                           MAX01220
  500  CONTINUE                                                          MAX01230
C                                                                        MAX01240
C  PRINT  BETA,S                                                         MAX01250
  600  CONTINUE                                                          MAX01260
      WRITE (IOUT,634) NOITER                                            MAX01270
      WRITE (IOUT,605)                                                   MAX01280
  634  FORMAT (' ITERATION NUMBER = ',I2)                                MAX01290
      WRITE (IOUT,610)((L, J, BETA(L,J), L=1,NOCAT), J=1,NOCC)           MAX01300
  605  FORMAT  (/,' CLASS  CLUSTER    BETA')                             MAX01310
  610  FORMAT  (   1X, I3, 5X, I3, 5X,F10.7)                             MAX01320
C                                                                        MAX01330
C  PRINT ESTIMATE FOR EACH LABELED CLASS WHERE EST = SUM(BETA * ALPHA)   MAX01340
      CALL PRTEST                                                        MAX01350
C  PRINT LABELED CLUSTER MAP  WHERE LABEL = MAX (BETA*ALPHA*P(X,I)       MAX01360
CCCC   CALL PRTMAP                                                       MAX01370
C                                                                        MAX01380
CC                                                                       MAX01390
C   PRINT MAXIMUM CLASS FOR EACH CLUSTER                                 MAX01400
      DO 630 K = 1,NOCC                                                  MAX01410
C                                                                        MAX01420
CC  SAVE CLASS AND CLUSTER LABELS FOR MAXIMUM BETAS                      MAX01430
C                                                                        MAX01440
      CSTMAX = 0.                                                        MAX01450
      CSSMAX = 0.                                                        MAX01460
      ICSS(K) = 0                                                        MAX01470
      LBLCST(K) = ISLNK                                                  MAX01480
      LBLCSS(K) = ISLNK                                                  MAX01490
      DO 630 L = 1,NOCAT                                                 MAX01500
      IF (BETA(L,K) .LE. CSTMAX) GO TO 620                               MAX01510
      LBLCST(K) = LABEL(K)                                               MAX01520
      CSTMAX = BETA(L,K)                                                 MAX01530
  620  IF (BETA(L,K) .LE. CSSMAX) GO TO 630                              MAX01540
      LBLCSS(K) = LABEL(L)                                               MAX01550
      ICSS(K) = L                                                        MAX01560
      CSSMAX = BETA(L,K)                                                 MAX01570
  630  CONTINUE                                                          MAX01580
```

```
      WRITE (IOUT,635)  ( I, LBLCSS(I), I = 1,NOCC)                    MAX01590
  635 FORMAT(//,' CLUSTER LABELS=',5(I3,1H=,A1,1H,),(/,16X,6(I3,1H=,A1,MAX01600
     1 1H,)))                                                          MAX01610
C   COMPUTE ESTIMATE OF EACH CLASS USING LABELLED CLUSTERS            MAX01620
      WRITE (IOUT,645)                                                 MAX01630
  645 FORMAT (/////,' LABELLED CLUSTER ESTIMATES',//,                  MAX01640
     1 '   CLASS   ESTIMATE')                                         MAX01650
      DO 655 I=1,NOCAT                                                 MAX01660
      EST = 0.0                                                        MAX01670
      DO 650 K=1,NOCC                                                  MAX01680
      IF (ICSS(K) .EQ. 0) GO TO 650                                    MAX01690
      IF (ICSS(K) .EQ. I) EST=EST+RLPRP(K)                             MAX01700
  650 CONTINUE                                                         MAX01710
      WRITE (IOUT,651) I,EST                                           MAX01720
  651 FORMAT (1H0,I7,3X,F10.7)                                         MAX01730
  655 CONTINUE                                                         MAX01740
C                                                                      MAX01750
C  PRINT LABELLED CLUSTER MAP WHERE LABEL = MAX (ALPHA * P(X,I))       MAX01760
CCC   CALL PRTAP                                                       MAX01770
C                                                                      MAX01780
      NOITER = NOITER + 1                                              MAX01790
       IF (YMAX .GT. .005  .AND. NOITER .LE. 20) GO TO 200             MAX01800
C                                                                      MAX01810
       CALL PRTAP                                                      MAX01820
       CALL PRTAP2                                                     MAX01830
      CALL PRTAP                                                       MAX01840
      CALL WRTLBS                                                      MAX01850
      STOP                                                             MAX01860
      END                                                              MAX01870
C  OPTIONALLY PRINT A LABELED CLUSTER MAP                              MAX01880
      SUBROUTINE ALLPXI                                                MAX01890
C                                                                      MAX01900
CC  PURPOSE: CALCULATE THE PROBABILITY OF X (DOT DATA PIXEL) GIVEN     MAX01910
CC           I(CLASY CLUSTER) FOR EACH X IN THE DOT DATA FILE          MAX01920
CC                                                                     MAX01930
CC  P(X,I) IS BASED ON RELATIONSHIP BETWEEN DOT DATA PTS AND CLUSTERS  MAX01940
C                                                                      MAX01950
      COMMON  MO, NOCC, NOCAT, ITOTDT, MOP4, MOS, ICC,                MAX01960
     1   ICLUNT, IDDUNT, IOUT, CATNAM(20), DCON,                       MAX01970
     2 IDOTS(20,209), CVOLRT, RLPRP( 30), CMEANS(16), CVRIN(136),      MAX01980
     3 PX(30,209), PAP(30), DOTDT(16), LABELS(30),LBLCSS(30),          MAX01990
     4 BETA(20,30),OMETA(20,30),SLK(20,30),SK(30),LBLCST(30)           MAX02000
      LOGICAL*1 LABELS, LBLCSS, LBLCST                                 MAX02010
CC                                                                     MAX02020
CC                                                                     MAX02030
      DIMENSION DIFXMN(4)                                              MAX02040
C                                                                      MAX02050
CC  PX      = PROBABILTY OF EACH DOT DATA PIXEL GIVEN CLASY CLUSTER     MAX02060
CC  ITOTDT = NUMBER OF DOT DATA PIXELS                                 MAX02070
CC  NOCC    = NUMBER OF CLASY CLUSTERS                                 MAX02080
CC  IDOTS   = DOT DATA PIXEL ARRAY                                     MAX02090
CC  ICLUNT = CLASY FILE UNIT                                           MAX02100
CC  MO      = NUMBER OF CHANNELS                                       MAX02110
CC  MOS     = NUMBER OF ELEMENTS IN TRIANGLARIZED ARRAY                MAX02120
CCCC                                                                   MAX02130
CCCC                                                                   MAX02140
CC                                                                     MAX02150
C     *** CALCULATE P(X,I) FOR EACH OF THE CLASY CLUSTERS              MAX02160
      DO 100 I = 1,NOCC                                                MAX02170
      ICC = I                                                          MAX02180
C                                                                      MAX02190
C  READ CLASY RELATIVE PROP, VOLRT, MEANS,AND CO-VARIANCE INVERSE MATRIXMAX02200
      CALL GETCC                                                       MAX02210
C                                                                      MAX02220
C  PROCESS DOT DATA PIXELS                                             MAX02230
      DO 100  IPXL = 1, ITOTDT                                         MAX02240
C                                                                      MAX02250
C  CALCULATE DIFFERENCE IN SPECTRAL VALUES                             MAX02260
      DO 30 J= 1,MO                                                    MAX02270
      DIFXMN(J) = IDOTS(J+4,IPXL) - CMEANS(J)                          MAX02280
30    CONTINUE                                                         MAX02290
C                                                                      MAX02300
C  CALCULATE PRODUCT                                                   MAX02310
      PRODUCT = DOTSQK (MO,DIFXMN,CVRIN)                               MAX02320
C                                                                      MAX02330
C  CALCULATE P(X,I) VALUE FOR CURRENT CLASY CLUSTER                    MAX02340
      IF (PRODUCT.LT.150.) PXI=(EXP(-PRODUCT/2.))*(EXP(-DCON/2.))/CVOLRT MAX02350
      IF (PRODUCT .GE. 150.) PXI = 0.0                                 MAX02360
      PX(I,IPXL)=PXI                                                   MAX02370
```

```
C                                                                       MAX02380
 100  CONTINUE                                                          MAX02390
C                                                                       MAX02400
CC              END OF DOT DATA PIXELS                                  MAX02410
CC              END OF ONE CLASS CLUSTER                                MAX02420
C                                                                       MAX02430
 500   RETURN                                                           MAX02440
       END                                                              MAX02450
       SUBROUTINE CALFCT                                                MAX02460
C                                                                       MAX02470
C PURPOSE:CALC. BETA*ALPHA*P(X,I) FOR EACH DOT-CLASS-CLUSTER COMBINATIONMAX02480
C                                                                       MAX02490
       COMMON  MO, NOCC, NOCAT, ITOTDT, MOP4, MOS, ICC,                MAX02500
      1    ICOUNT, IDOUNT, IOUT, CATNAM(20), DCON,                     MAX02510
      2 IDOTS(20,209), CVOLPT, RLPRP( 30), CMEANS(16), CVRIN(156),      MAX02520
      3 PX(30,209), PAP(30), DOTDT(16), LABELS(30),LBLCSS(30),          MAX02530
      4 BETA(20,30),OBETA(20,30),SLK(20,30),SK(30),LBLCST(30)           MAX02540
       LOGICAL*1 LABELS, LBLCSS, LBLCST                                 MAX02550
C                                                                       MAX02560
C  WRITE HEADING                                                        MAX02570
       WRITE (IOUT,410)                                                 MAX02580
410    FORMAT(///,' CATEGORY NAME  CLUSTER NO.  BETA  REL.PROP  PX(X,I)  MAX02590
     1 FCT  FCTLKH')                                                    MAX02600
       FCTLKH = 1.                                                      MAX02610
C                                                                       MAX02620
C  PROCESS EACH DOT DATA POINT                                          MAX02630
       DO 450 I = 1,ITOTDT                                              MAX02640
       L = IDOTS(4,I)                                                   MAX02650
C                                                                       MAX02660
C  PROCESS CLUSTER                                                      MAX02670
       FCT = 0.                                                         MAX02680
       DO 440 K = 1,NOCC                                                MAX02690
       RELPRP = RLPRP(K)                                                MAX02700
       FCT    = BETA(L,K) * RELPRP * PX(K,I) + FCT                      MAX02710
440    CONTINUE                                                         MAX02720
C                                                                       MAX02730
       FCTLKH = FCTLKH * FCT                                            MAX02740
 450   CONTINUE                                                         MAX02750
C                                                                       MAX02760
C  PRINT LIKELIHOOD FACTOR                                              MAX02770
       WRITE (IOUT,460) FCTLKH                                          MAX02780
460    FORMAT (' LIKELIHOOD FUNCTION = ',E12.3)                         MAX02790
       RETURN                                                           MAX02800
       END                                                              MAX02810
       SUBROUTINE MAPLS                                                 MAX02820
C                                                                       MAX02830
CC  PURPOSE:  SUM PRODUCTS IN A CLASS FOR ALL DOT-CLUSTER COMBINATIONS  MAX02840
C                                                                       MAX02850
       COMMON  MO, NOCC, NOCAT, ITOTDT, MOP4, MOS, ICC,                MAX02860
      1    ICOUNT, IDOUNT, IOUT, CATNAM(20), DCON,                     MAX02870
      2 IDOTS(20,209), CVOLPT, RLPRP( 30), CMEANS(16), CVRIN(156),      MAX02880
      3 PX(30,209), PAP(30), DOTDT(16), LABELS(30),LBLCSS(30),          MAX02890
      4 BETA(20,30),OBETA(20,30),SLK(20,30),SK(30),LBLCST(30)           MAX02900
       LOGICAL*1 LABELS, LBLCSS, LBLCST                                 MAX02910
CC                                                                      MAX02920
CC  INPUT                                                               MAX02930
CC    NOCC   = NUMBER OF CLASS CLUSTERS                                 MAX02940
CC    NOCAT  = NUMBER OF DOT DATA CLASSES                               MAX02950
C                                                                       MAX02960
C  ZERO ACCUMULATORS                                                    MAX02970
       DO 10 L = 1,NOCAT                                                MAX02980
       DO 10 K = 1,NOCC                                                 MAX02990
       SK(K) = 0.                                                       MAX03000
10     SLK(L,K) = 0.0                                                   MAX03010
C                                                                       MAX03020
C  PROCESS FOR EACH DOT                                                 MAX03030
       DO 1000 ID = 1,ITOTDT                                            MAX03040
       L = IDOTS(4,ID)                                                  MAX03050
       SUM = 0.0                                                        MAX03060
C                                                                       MAX03070
C  ACCUMULATE FOR CURRENT CLUSTER FOR ALL DOT-CLASS COMBINATIONS        MAX03080
       DO 100 K = 1,NOCC                                                MAX03090
       PAP(K) = BETA(L,K) * RLPRP(K) * PX(K,ID)                         MAX03100
100    SUM = SUM + PAP(K)                                               MAX03110
C                                                                       MAX03120
CC                                                                      MAX03130
C  SUM PAP'S FOR LABELLED CLASS-CLUSTER COMBINATIONS                    MAX03140
       DO 200 K = 1,NOCC                                                MAX03150
                                                                        MAX03160
```

```
      QUOT = MAP(K) / SUM                                                    MAX03170
      SLK(L,K) = SLK(L,K) * QUOT                                             MAX03180
C  SUM MAP'S FOR LABELLED CLASSES                                            MAX03190
      SK(K) = SK(K) + QUOT                                                   MAX03200
  200 CONTINUE                                                               MAX03210
C                                                                            MAX03220
 1000  CONTINUE                                                              MAX03230
      RETURN                                                                 MAX03240
      END                                                                    MAX03250
      SUBROUTINE PRTELR                                                      MAX03260
C                                                                            MAX03270
CC  PURPOSE: PRINT ESTIMATE FOR EACH LABELED CLASS                           MAX03280
C                                                                            MAX03290
      COMMON  MO, NOCC, NOCAT, ITOTDT, MBP4, MNS, ICC,                       MAX03300
     1    ICLUNT, IDDUNT, IOUT, CATNAM(20), QCON,                            MAX03310
     2 IDOTS(20,209), CVOLRT, RLPRP( 30), CMEANS(16), CVRIN(158),            MAX03320
     3 PX(30,209), MAP(30), DOTDT(16), LABELS(30),LBLCSS(30),                MAX03330
     4 BETA(20,30),OMETA(20,30),SLK(20,30),SK(30),LBLCST(30)                 MAX03340
      LOGICAL*1 LABELS, LBLCSS, LBLCST                                       MAX03350
C                                                                            MAX03360
C  WRITE HEADING                                                             MAX03370
      WRITE (IOUT,10)                                                        MAX03380
  10  FORMAT(//////,' ESTIMATE = BETA * ALPHA',//,'    CLASS    ESTIMATE') MAX03390
C                                                                            MAX03400
      DO 100 L = 1,NOCAT                                                     MAX03410
      SUM = 0.                                                               MAX03420
      DO  50 K = 1,NOCC                                                      MAX03430
   50   SUM = SUM + BETA(L,K) * RLPRP(K)                                      MAX03440
C                                                                            MAX03450
C  WRITE ESTIMATE                                                            MAX03460
      WRITE (IOUT,60) L, SUM                                                 MAX03470
   60   FORMAT (1H0,I7,3X,F10.7)                                              MAX03480
C                                                                            MAX03490
  100   CONTINUE                                                             MAX03500
      RETURN                                                                 MAX03510
      END                                                                    MAX03520
      SUBROUTINE PRTMAP                                                      MAX03530
C  PURPOSE:  PRINT LABELED CLUSTER MAP FOR EACH CLASS-CLUSTER                MAX03540
CC          PIXEL CLUSTER = MAX(BETA*ALPHA*P(X,I)                            MAX03550
C                                                                            MAX03560
      COMMON  MO, NOCC, NOCAT, ITOTDT, MBP4, MNS, ICC,                       MAX03570
     1    ICLUNT, IDDUNT, IOUT, CATNAM(20), QCON,                            MAX03580
     2 IDOTS(20,209), CVOLRT, RLPRP( 30), CMEANS(16), CVRIN(158),            MAX03590
     3 PX(30,209), MAP(30), DOTDT(16), LABELS(30),LBLCSS(30),                MAX03600
     4 BETA(20,30),OMETA(20,30),SLK(20,30),SK(30),LBLCST(30)                 MAX03610
      LOGICAL*1 LABELS, LBLCSS, LBLCST                                       MAX03620
C                                                                            MAX03630
      COMMON /MPPXL/ MPXLA (19,11)                                           MAX03640
      LOGICAL*1 MPXLA                                                        MAX03650
C                                                                            MAX03660
      LOGICAL*1 KBLNK                                                        MAX03670
      DATA KBLNK/' '/                                                        MAX03680
C                                                                            MAX03690
      WRITE (IOUT,10)                                                        MAX03700
  10  FORMAT ('1 PIXEL LABELLED DOT MAP ',//,                                MAX03710
     1 '                1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18',MAX03720
     2 19 ',/)                                                               MAX03730
C                                                                            MAX03740
      DO 20 I = 1,19                                                         MAX03750
      DO 20 J = 1,11                                                         MAX03760
  20  MPXLA(I,J) = KBLNK                                                     MAX03770
C                                                                            MAX03780
C  FIND MAX FOR EACH DOT                                                     MAX03790
      DO 200  N = 1,ITOTDT                                                   MAX03800
      IX = IDOTS (1,N) / 10                                                  MAX03810
      IY = IDOTS (2,N) / 10                                                  MAX03820
      PXLMAX = 0.                                                            MAX03830
C  CLASSES                                                                   MAX03840
      DO 200 L=1,NOCAT                                                       MAX03850
      PIXEL=0                                                                MAX03860
C  CLUSTERS                                                                  MAX03870
      DO 100  I = 1,NOCC                                                     MAX03880
  100 PIXEL = PIXEL + BETA(L,I) * RLPRP(I) * PX(I,N)                         MAX03890
      IF (PIXEL .GT. PXLMAX) MPXLA(IX,IY) = LABELS(L)                        MAX03900
      IF (PIXEL .GT. PXLMAX) PXLMAX = PIXEL                                  MAX03910
  200  CONTINUE                                                              MAX03920
      WRITE (IOUT,210)((J, ( MPXLA(I,J), I=1,19), J=1,11))                   MAX03930
  210  FORMAT (1X,I4,3X,19A1 )                                               MAX03940
      RETURN                                                                 MAX03950
```

```
      END                                                                   MAX03950
      SUBROUTINE PRTAP                                                      MAX03970
C  PURPOSE:  PRINT LABELED CLUSTER MAP FOR EACH CLUSTER                     MAX03980
C            PIXEL CLUSTER LABEL = MAX (ALPHA * P(X,I)                     MAX03990
C                                                                          MAX04000
      COMMON  MO, NOCC, NOCAT, ITOTDT, MAP4, MOS, ICC,                     MAX04010
     1   ICLUST, IDOUT, IOUT, CATNAM(20), DCON,                            MAX04020
     2 IDOTS(20,209), CVOLRT, PLPPP( 30), CMEANS(16), CVRIN(158),          MAX04030
     3 PX(30,209), MAP(30), DOTDT(16), LABELS(30),LBLCSS(30),              MAX04040
     4 BETA(20,30),OMETA(20,30),SLK(20,30),SK(30),LBLCST(30)               MAX04050
      LOGICAL*1 LABELS, LBLCSS, LBLCST, LBLCST                             MAX04060
C                                                                          MAX04070
      COMMON /MPPXL/ MPXLA (19,11)                                         MAX04080
      LOGICAL*1 MPXLA                                                      MAX04090
C                                                                          MAX04100
      DIMENSION SUM(209), INTEG(30), PXEL(30)                              MAX04110
C                                                                          MAX04120
      LOGICAL*1 KBLNK                                                      MAX04130
       DATA KBLNK/' '/                                                     MAX04140
C                                                                          MAX04150
      DO 1 I = 1,30                                                        MAX04160
    1 INTEG(I) = I                                                         MAX04170
C                                                                          MAX04180
      WRITE (IOUT,10)                                                      MAX04190
   10 FORMAT ('1 UNLABELLED DOT MAP'///,                                   MAX04200
     1    '0          1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 1MAX04210
     2R 19'/)                                                              MAX04220
C                                                                          MAX04230
        DO 71 J = 1,19                                                     MAX04240
       DO 70 K = 1,11                                                      MAX04250
   70  MPXLA(J,K) = KBLNK                                                  MAX04260
C                                                                          MAX04270
C  FIND MAX FOR EACH DOT                                                   MAX04280
      DO 150  N = 1,ITOTDT                                                 MAX04290
      IX = IDOTS (1,N) / 10                                                MAX04300
      IY = IDOTS (2,N) / 10                                                MAX04310
      PXLMAX = 0.                                                          MAX04320
C                                                                          MAX04330
C  CLUSTERS                                                                MAX04340
      SUM(N) = 0.                                                          MAX04350
      DO 100  I = 1,NOCC                                                   MAX04360
      PIXEL = PLPPP(I) * PX(I,N)                                           MAX04370
      IF (PIXEL .GT. PXLMAX) MPXLA(IX,IY) = LBLCST(I)                      MAX04380
      IF (PIXEL .GT. PXLMAX) PXLMAX = PIXEL                                MAX04390
      SUM(N) = SUM(N) + PIXEL                                              MAX04400
  100  CONTINUE                                                            MAX04410
  150  CONTINUE                                                            MAX04420
C                                                                          MAX04430
      WRITE (IOUT,210)((J, (MPXLA (I,J), I=1,19), J=1,11))                 MAX04440
  210  FORMAT  (1X,I4,3X, 19A1 )                                           MAX04450
C                                                                          MAX04460
C  PRINT PROBABILITIES FOR PIXELS                                          MAX04470
      MAXCC = NOCC                                                         MAX04480
      IF (MAXCC .GT. 15) MAXCC = 15                                        MAX04490
C                                                                          MAX04500
      WRITE (IOUT,301) (INTEG(I),I=1,MAXCC)                               MAX04510
  301 FORMAT (///,'1 COL ROW ENTROPY',15I5)                                MAX04520
      DO 350 N = 1,ITOTDT                                                  MAX04530
      ENTROP = 0.                                                          MAX04540
      DO 300 I = 1,NOCC                                                    MAX04550
      PXEL(I) = PLPPP(I) * PX(I,N) / SUM(N)                                MAX04560
      IF (PXEL(I).NE.0) ENTROP=ENTROP-PXEL(I)*ALOG(PXEL(I))                MAX04570
  300 CONTINUE                                                             MAX04580
      IX = IDOTS(1,N)/10                                                   MAX04590
      IY = IDOTS(2,N)/10                                                   MAX04600
      WRITE (IOUT,330) IX,IY,ENTROP, (PXEL(I),I=1,NOCC)                    MAX04610
  330 FORMAT (2I4, F9.6, 15F5.2,/,20X,15F5.2)                              MAX04620
  350  CONTINUE                                                            MAX04630
      RETURN                                                               MAX04640
      END                                                                  MAX04650
      SUBROUTINE PRTAP2                                                    MAX04660
C  PURPOSE:  PRINT LABELED CLUSTER MAP FOR EACH CLUSTER                    MAX04670
C            PIXEL CLUSTER LABEL = MAX (ALPHA * P(X,I)                     MAX04680
C                                                                          MAX04690
      COMMON  MO, NOCC, NOCAT, ITOTDT, MAP4, MOS, ICC,                     MAX04700
     1   ICLUST, IDOUT, IOUT, CATNAM(20), DCON,                            MAX04710
     2 IDOTS(20,209), CVOLRT, PLPPP( 30), CMEANS(16), CVRIN(158),          MAX04720
     3 PX(30,209), MAP(30), DOTDT(16), LABELS(30),LBLCSS(30),              MAX04730
     4 BETA(20,30),OMETA(20,30),SLK(20,30),SK(30),LBLCST(30)               MAX04740
```

```
      LOGICAL*1 LABELS, LBLCSS, LBLCST                                  MAX04750
C                                                                       MAX04760
      COMMON /MPXLA/ MPXLA (19,11)                                      MAX04770
      LOGICAL*1 MPXLA                                                   MAX04780
C                                                                       MAX04790
      LOGICAL*1 KBLNK                                                   MAX04800
       DATA KBLNK/' '/                                                  MAX04810
C                                                                       MAX04820
      WRITE (IOUT,10)                                                   MAX04830
10    FORMAT ('1 CLUSTER LABELLED DOT MAP '//,                          MAX04840
     1    '0          1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 1MAX04850
     2R 19',/)                                                          MAX04860
C                                                                       MAX04870
        DO 70 J = 1,19                                                  MAX04880
       DO 70 K = 1,11                                                   MAX04890
70    MPXLA(J,K) = KBLNK                                                MAX04900
C                                                                       MAX04910
C  FIND MAX FOR EACH DOT                                                MAX04920
      DO 150  N = 1,ITOTDT                                              MAX04930
      IX = IDOTS (1,N) / 10                                             MAX04940
      IY = IDOTS (2,N) / 10                                             MAX04950
      PXLMAX = 0.                                                       MAX04960
C                                                                       MAX04970
C  CLUSTERS                                                             MAX04980
      DO 100  I = 1,NOCC                                                MAX04990
      PIXEL = ALPPP(I) * PX(I,N)                                        MAX05000
      IF (PIXEL .GT. PXLMAX) MPXLA(IX,IY) = LBLCSS(I)                   MAX05010
      IF (PIXEL .GT. PXLMAX) PXLMAX = PIXEL                             MAX05020
 100   CONTINUE                                                         MAX05030
150   CONTINUE                                                          MAX05040
C                                                                       MAX05050
      WRITE (IOUT,210)((J, (MPXLA (I,J), I=1,19), J=1,11))              MAX05060
 210   FORMAT  (1X,I4,3X, 19A3 )                                        MAX05070
      RETURN                                                            MAX05080
      END                                                               MAX05090
      SUBROUTINE READCC                                                 MAX05100
C                                                                       MAX05110
C  PURPOSE:  READ CLASY CLUSTER VALUES INTO COMMON BLOCK /CLASY/        MAX05120
C                                                                       MAX05130
      COMMON /MD, NOCC, NOCAT, ITOTDT, MDP4, MDS, ICC,                  MAX05140
     1   ICLUNT, IODUNT, IOUT, CATNAM(20), DCON,                        MAX05150
     2 IDOTS(20,209), CVOLPT, ALPPP( 30), CMEANS(16), CVRIN(136),       MAX05160
     3 PX(30,209), MAP(30), DOTDT(16), LABELS(30),LBLCSS(30),           MAX05170
     4 BETA(20,30),OMEGA(20,30),SLK(20,30),SK(30),LBLCST(30)            MAX05180
      LOGICAL*1 LABELS, LBLCSS, LBLCST                                  MAX05190
C                                                                       MAX05200
      COMMON/CLASY/ CCMENS(16,30),CCVRIN(256,30),CCVLRT(30),CDCON(30)   MAX05210
C                                                                       MAX05220
      NOCC = 1                                                          MAX05230
C                                                                       MAX05240
      WRITE (IOUT,1)                                                    MAX05250
1       FORMAT (1)  INPUT DATA FROM CLASY')                             MAX05260
C  READ CLASY RECORD                                                    MAX05270
10    READ (ICLUNT,END = 500)ALPPP(NOCC), CCVLRT(NOCC),CDCON(NOCC)      MAX05280
9001  FORMAT (F15.8,2F15.8)                                             MAX05290
9000   FORMAT ( F12.5,15F12.5)                                          MAX05300
C                                                                       MAX05310
      WRITE (IOUT,101) NOCC, ALPPP(NOCC), NOCC, CCVLRT(NOCC),           MAX05320
     1  NOCC, CDCON(NOCC)                                              MAX05330
101   FORMAT (//,' ALPPP (',I2,')=',E15.8,/,                           MAX05340
     1 ' VOLRT (',I2,')=',E15.8,/' DCON  (',I2,')=',E15.8)             MAX05350
C                                                                       MAX05360
      READ (ICLUNT,END=600) (CCMENS(I,NOCC),I=1,MD)                     MAX05370
C                                                                       MAX05380
      WRITE (IOUT,102) (CCMENS(I,NOCC),I=1,MD)                          MAX05390
102   FORMAT (/,' MEANS=', (/,3X,8F10.5))                               MAX05400
       WRITE (IOUT,103)                                                 MAX05410
C                                                                       MAX05420
      DO 1000 J = 1,MD                                                  MAX05430
       INDEX = 1 + (J-1)*MD                                             MAX05440
      LAST = INDEX + MD - 1                                             MAX05450
      READ (ICLUNT,END=500)(CCVRIN(I,NOCC), I = INDEX,LAST)             MAX05460
      WRITE (IOUT,104) (CCVRIN(I,NOCC), I=INDEX,LAST)                   MAX05470
1000  CONTINUE                                                          MAX05480
103   FORMAT (' INVERSE OF COVARIANCE MATRIX=')                         MAX05490
104   FORMAT (3X,8F12.7)                                                MAX05500
C                                                                       MAX05510
C  SKIP FIRST 12 CLASSY RECORDS                                         MAX05520
      NOCC = NOCC + 1                                                   MAX05530
```

```
      GO TO 10                                                          MAX05540
C                                                                       MAX05550
C  END OF CLASY FILE                                                    MAX05560
 500   NOCC = NOCC - 1                                                  MAX05570
      WRITE (IOUT,510)                                                  MAX05580
510   FORMAT (//////)                                                   MAX05590
      RETURN                                                            MAX05600
600    WRITE (IOUT,610)                                                 MAX05610
610    FORMAT (' CLASY END OF FILE')                                    MAX05620
       NOCC = NOCC-1                                                    MAX05630
      RETURN                                                            MAX05640
      END                                                               MAX05650
      SUBROUTINE GETCC                                                  MAX05660
C                                                                       MAX05670
C  PURPOSE: FILL ONE SET OF CLASY VALUES USING  'ICC' AS AN INDEX       MAX05680
C                                                                       MAX05690
      COMMON  MO, NOCC, NOCAT, ITOTOT, MAP4, MOS, ICC,                 MAX05700
     1    ICOUNT, IDOUNT, IOUT, CATNAM(20), DCOL,                      MAX05710
     2 IDOTS(20,209), CVOLRT, RLPRP( 30), CMEANS(16), CVRIN(158),       MAX05720
     3 PX(30,209), MAP(30), DOTOT(16), LABELS(30),LHLCSS(30),           MAX05730
     4 BETA(20,30),OMETA(20,30),SLK(20,30),SX(30),LHLCST(30)            MAX05740
      LOGICAL*1 LABELS, LHLCSS, LHLCST                                  MAX05750
      COMMON/CLASY/ CCMENS(16,30),CCVRIN(256,30),CCVLRT(30),CDCON(30)   MAX05760
C                                                                       MAX05770
      RELPRP = RLPRP(ICC)                                               MAX05780
C                                                                       MAX05790
      CVOLRT = CCVLRT(ICC)                                              MAX05800
C                                                                       MAX05810
      DO 10 I = 1,16                                                    MAX05820
 10    CMEANS(I) = CCMENS(I,ICC)                                        MAX05830
C                                                                       MAX05840
      DO 20 I = 1,158                                                   MAX05850
 20    CVRIN(I) = CCVRIN(I,ICC)                                         MAX05860
      DCON=CDCON(ICC)                                                   MAX05870
C                                                                       MAX05880
      RETURN                                                            MAX05890
      END                                                               MAX05900
       FUNCTION DOTSQK (MO,V,AMET)                                      MAX05910
C                                                                       MAX05920
C      CALCULATES THE INNER PRODUCT V.V RELATIVE TO THE METRIC AMET     MAX05930
      REAL V(30),AMET(475)                                              MAX05940
      REAL*8 DDOTSQ,DGDOT                                               MAX05950
C                                                                       MAX05960
      DDOTSQ=0.                                                         MAX05970
      DGDOT=V(1)*V(1)*AMET(1)                                           MAX05980
      DO 10 I=2,MO                                                      MAX05990
       MX = (I-1)*MO                                                    MAX06000
    7 DO 8 J=2,I                                                        MAX06010
    8 DDOTSQ=DDOTSQ+V(I)*V(J-1)*AMET(MX+J-1)                            MAX06020
   10 DGDOT=DGDOT+V(I)*V(I)*AMET(MX+I)                                  MAX06030
C      THE DIAGONALS ARE HANDLED SEPARATELY BECAUSE EACH OFF-           MAX06040
C      DIAGONAL APPEARS TWICE, AND SO MUST BE DOUBLED.                  MAX06050
      DDOTSQ=DDOTSQ+DDOTSQ+DGDOT                                        MAX06060
      DOTSQK = DDOTSQ                                                   MAX06070
      RETURN                                                            MAX06080
      END                                                               MAX06090
```

```
      SUBROUTINE WRTLNS                                                 WRT00010
C                                                                       WRT00020
CC  PURPOSE:  TO READ RADIANCE VALUES FOR LINES OF ORIGINAL PIXEL DATA  WRT00030
CCC           TO PRINT HEADING FOR REPORTS. TO CALL SUBROUTINES TO      WRT00040
CCC           CALCULATE REPORT DATA                                     WRT00050
CC                                                                      WRT00060
      COMMON MQ                                                         WRT00070
C                                                                       WRT00080
      COMMON /TAPEBF/ LINETP(196)                                       WRT00090
C                                                                       WRT00100
       COMMON /PXLLN/ NOPXL, LINENO, IHEGIN, IEND, PXLN(30,196),        WRT00110
     1LNDOTS(20,196),LINE(196),IFSTCL,LSTCOL,LSTPG1,IFSTP2,LNSZP1,LNSZP2WRT00120
      LOGICAL*1 LINE                                                    WRT00130
C                                                                       WRT00140
      COMMON /FILE/ DATFIL, MAPOPT                                      WRT00150
C                                                                       WRT00160
         DIMENSION ICOL(196,3)                                          WRT00170
C                                                                       WRT00180
      DATA KONSTP /'P'/, KONSTC/'C'/                                    WRT00190
      DATA ISIZHD/6/, IEIGHT/8/, ININE/9/, ITEN/10/                     WRT00200
C                                                                       WRT00210
C  RADIANCE INFORMATIONS FOR LINES ON UNIT 24                           WRT00220
      LNUNIT = 24                                                       WRT00230
      REWIND LNUNIT                                                     WRT00240
CC                                                                      WRT00250
C   GENERATE COLUMN HEADINGS                                            WRT00260
      DO 10 I = 1,196                                                   WRT00270
      ICOL(I,1) = I/100                                                 WRT00280
      ICOL (I,2) = MOD(I,100)/10                                        WRT00290
 10   ICOL(I,3) = MOD(I,10)                                             WRT00300
CC                                                                      WRT00310
CC  READ NUMBER OF CHANNELS, FIRST AND LAST COLUMNS TO MAP              WRT00320
      IFSTCL = 1                                                        WRT00330
      LSTCOL = 196                                                      WRT00340
      LSTLNE = 117                                                      WRT00350
CC                                                                      WRT00360
CC  CALCULATE PAGE LIMITS                                               WRT00370
      LSTPG1 = IFSTCL + 109                                             WRT00380
      IFSTP2 = LSTPG1 + 1                                               WRT00390
      IF (LSTCOL .LT. LSTPG1) LSTPG1 = LSTCOL                           WRT00400
C                                                                       WRT00410
C  CALCULATE LINE SIZE FOR HEADINGS AND TEXT                            WRT00420
      LNSZP1 = (LSTPG1 - IFSTP1 + 14) / 4                               WRT00430
      LNSZP2 = (LSTCOL - IFSTP2 + 14) / 4                               WRT00440
C                                                                       WRT00450
C  HEADING--PAGE 1, REPORT 1                                            WRT00460
      WRITE (6,15)                                                      WRT00470
15    FORMAT ('1 PIXEL LABELLED CLUSTER MAP ',/)                        WRT00480
16    FORMAT (I4,' PIXEL LABELLED CLUSTER MAP ',/,I4,36X)               WRT00490
      WRITE (6,30) ((ICOL(I,J),I=IFSTCL,LSTPG1),J=1,3)                  WRT00500
      WRITE (6,11)                                                      WRT00510
11    FORMAT (2X)                                                       WRT00520
C  HEADING--PAGE 2, REPORT 1                                           WRT00530
      IF (IFSTP2 .GT. LSTCOL)GO TO 160                                  WRT00540
      WRITE (25,15) ININE,ININE                                         WRT00550
      WRITE (25,32) LNSZP2, (ICOL(I,1),I=IFSTP2,LSTCOL)                 WRT00560
      WRITE (25,32) LNSZP2, (ICOL(I,2), I=IFSTP2,LSTCOL)                WRT00570
      WRITE (25,32) LNSZP2, (ICOL(I,3), I=IFSTP2,LSTCOL)                WRT00580
      WRITE (25,31) IEIGHT                                              WRT00590
C                                                                       WRT00600
C  WRITE HEADER FOR UNIVERSAL FILE, IF PIXEL OR CLUSTER MAP REQUESTED   WRT00610
        IF (MAPOPT .NE. KONSTP .AND. MAPOPT .NE. KONSTC) GO TO 160      WRT00620
        IZERO = 0                                                       WRT00630
      CALL FSFMEL(31,IZERO,IDUMMY)                                      WRT00640
       ICHAN = 1                                                        WRT00650
      IICHAN = 1                                                        WRT00660
      IPSAMP = 196                                                      WRT00670
      IPERMT = 1                                                        WRT00680
       CALL WRTHED (ICHAN, IICHAN, IPSAMP, IPERMT, 31)                  WRT00690
C                                                                       WRT00700
CC                                                                      WRT00710
C  HEADING--PAGE1, REPORT 2                                             WRT00720
160   WRITE (25, 610) ININE, IEIGHT                                     WRT00730
610   FORMAT (I4, '1 CLUSTER LABELLED CLUSTER MAP          ', /, I4, 32X)WRT00740
      WRITE (26,32) (LNSZP1, (ICOL(I,J),I=IFSTCL,LSTPG1),J=1,3)         WRT00750
      WRITE (26,31) ISIZHD                                              WRT00760
C                                                                       WRT00770
                                                                        WRT00780
                                                                        WRT00790
```

```
C  HEADING--PAGE 2, REPORT 2                                                    WRT00800
      IF (IFSTP2 .GT. LSTCOL) GO TO 180                                         WRT00810
      WRITE (27,610) INLNE, IEIGHT                                              WRT00820
       WRITE (27,31) ISIZHD                                                     WRT00830
      WRITE (27,32) LNSZP2, (ICOL(I,1),I=IFSTP2,LSTCOL)                         WRT00840
      WRITE (27,32) LNSZP2, (ICOL(I,2),I=IFSTP2,LSTCOL)                         WRT00850
      WRITE (27,32) LNSZP2, (ICOL(I,3),I=IFSTP2,LSTCOL)                         WRT00860
      WRITE (27,31) ISIZHD                                                      WRT00870
C                                                                               WRT00880
C  HEADING PAGE 1, REPORT 3                                                     WRT00890
180   WRITE (28,620) ISIZHD, ISIZHD                                             WRT00900
620   FORMAT (I4, '1 UNLABELLED CLUSTER MAP ',/,I4,32X)                         WRT00910
      WRITE (28,32) (LNSZP1, (ICOL(I,J),I=IFSTCL,LSTPG1),J=1,3)                WRT00920
      WRITE (28,31) ISIZHD                                                      WRT00930
C                                                                               WRT00940
CC  HEADING PAGE 2, REPORT 3                                                    WRT00950
      IF (IFSTP2 .GT. LSTCOL) GO TO 200                                         WRT00960
      WRITE (29,620) ISIZHD, ISIZHD                                             WRT00970
      WRITE (29,32) LNSZP2, (ICOL(I,1),I=IFSTP2,LSTCOL)                         WRT00980
      WRITE (29,32) LNSZP2, (ICOL(I,2),I=IFSTP2,LSTCOL)                         WRT00990
      WRITE (29,32) LNSZP2, (ICOL(I,3),I=IFSTP2,LSTCOL)                         WRT01000
      WRITE (29,31) ISIZHD                                                      WRT01010
C                                                                               WRT01020
30    FORMAT (10X,110I1)                                                        WRT01030
31    FORMAT (I4,32X)                                                           WRT01040
32    FORMAT (I4,10X,110I1)                                                     WRT01050
CC                                                                              WRT01060
C  READ RADIANCE VALUES                                                         WRT01070
       NDTAP = 0                                                                WRT01080
200    READ (LNUNIT, END=240) NOPXLV, LINENO, IBEGIN, IEND,                     WRT01090
     1   ((LNDATS(J,I),J=1,MQ),I=IBEGIN,IEND)                                   WRT01100
      IF (LINENO .EQ. LSTLNE) NDTAP = -1                                        WRT01110
CC                                                                              WRT01120
                                                                                WRT01130
      CALL PXILN                                                                WRT01140
C                                                                               WRT01150
      CALL LNMAP                                                                WRT01160
CC                                                                              WRT01170
C  WRITE UNIVERSAL TAPE IF 'PIXEL' MAP REQUESTED                                WRT01180
      IF (MAPOPT .EQ. KONSTP) CALL WRTLN (LINETP, NDTAP)                        WRT01190
C                                                                               WRT01200
      CALL LNAP2                                                                WRT01210
CC                                                                              WRT01220
C  WRITE UNIVERSAL TAPE IF 'CLUSTER' MAP REQUESTED                              WRT01230
      IF (MAPOPT .EQ. KONSTC) CALL WRTLN (LINETP, NDTAP)                        WRT01240
      CALL LNAP                                                                 WRT01250
C                                                                               WRT01260
      GO TO 200                                                                 WRT01270
CC                                                                              WRT01280
C  PRINT REPORTS THAT HAVE BEEN COLLECTED                                       WRT01290
240   IF (IFSTP2 .GT. LSTCOL) GO TO 260                                         WRT01300
      CALL PAGE(25)                                                             WRT01310
C                                                                               WRT01320
C REPORT 2                                                                      WRT01330
260   CALL PAGE(26)                                                             WRT01340
      IF (IFSTP2 .GT. LSTCOL) GO TO 280                                         WRT01350
      CALL PAGE(27)                                                             WRT01360
C                                                                               WRT01370
C  REPORT 3                                                                     WRT01380
280   CALL PAGE(28)                                                             WRT01390
      IF (IFSTP2 .GT. LSTCOL) GO TO 300                                         WRT01400
      CALL PAGE(29)                                                             WRT01410
C                                                                               WRT01420
300   RETURN                                                                    WRT01430
CC                                                                              WRT01440
C  NO DATA ON FILE                                                              WRT01450
2000  WRITE (6, 2010)                                                           WRT01460
2010  FORMAT (' NO DATA ON FILE 24')                                            WRT01470
      STOP                                                                      WRT01480
      END                                                                       WRT01490
      SUBROUTINE PXILN                                                          WRT01500
C                                                                               WRT01510
CC  PURPOSE: CALCULATE THE PROBABILITY OF X (DOT DATA PIXEL) GIVEN              WRT01520
CC        I(CLASS CLUSTER) FOR EACH X IN THE LINE IN THE IMAGE FILE             WRT01530
CC                                                                              WRT01540
CC  P(X,I) IS BASED ON RELATIONSHIP BETWEEN DOT DATA PTS AND CLUSTERS           WRT01550
CC                                                                              WRT01560
      COMMON  MQ, NOCC, NOCAT, ITOTDT, MQP4, MQS, ICC,                          WRT01570
                                                                                WRT01580
```

```
     1   ICLUNT, IDDUNT, IOUT, CATNAM(20), DCON,                          WRT01590
     2 IDOTS(20,209), CVOLRT, RLPRP( 30), CMEANS(16), CVRIN(158),         WRT01600
     3 PX(30,209), RAP(30), DOTDT(16), LABELS(30),LBLCSS(30),             WRT01610
     4 BETA(20,30),OBETA(20,30),SLK(20,30),SK(30),LBLCST(30)              WRT01620
      LOGICAL*1 LABELS, LBLCSS, LBLCST                                    WRT01630
C                                                                         WRT01640
      COMMON /PXLLN/ NOPXL, LINENO, IBEGIN, IEND, PXLN(30,196),           WRT01650
     1LNDOTS(20,196),LINE(196),IFSTCL,LSTCOL,LSTPG1,IFSTP2,LNSZP1,LNSZP2,WRT01660
      LOGICAL*1 LINE                                                      WRT01670
C                                                                         WRT01680
      DIMENSION DIFXMN(8)                                                 WRT01690
C                                                                         WRT01700
C  PX      = PROBABILITY OF EACH DOT DATA PIXEL GIVEN CLASY CLUSTER        WRT01710
C  ITOTDT  = NUMBER OF DOT DATA PIXELS                                     WRT01720
C  NOCC    = NUMBER OF CLASY CLUSTERS                                      WRT01730
C  IDOTS   = DOT DATA PIXEL ARRAY                                          WRT01740
C  ICLUNT  = CLASY FILE UNIT                                               WRT01750
C  MQ      = NUMBER OF CHANNELS                                            WRT01760
C  MQS     = NUMBER OF ELEMENTS IN TRIANGLARIZED ARRAY                     WRT01770
C                                                                         WRT01780
C                                                                         WRT01790
C                                                                         WRT01800
C    *** CALCULATE P(X,I) FOR EACH OF THE CLASY CLUSTERS                  WRT01810
      DO 100 I = 1,NOCC                                                   WRT01820
      ICC = I                                                             WRT01830
C                                                                         WRT01840
C  READ CLASY RELATIVE PROP, VOLRT, MEANS,AND CO-VARIANCE INVERSE MATRIX  WRT01850
      CALL GETCC                                                          WRT01860
C                                                                         WRT01870
C  PROCESS DOT DATA PIXELS                                                WRT01880
      DO 100  IPXL = IBEGIN, IEND                                         WRT01890
C                                                                         WRT01900
C  CALCULATE DIFFERENCE IN SPECTRAL VALUES                                WRT01910
      DO 30 J= 1,MQ                                                       WRT01920
      DIFXMN(J) = LNDOTS(J,IPXL) - CMEANS(J)                              WRT01930
30    CONTINUE                                                            WRT01940
C                                                                         WRT01950
C  CALCULATE PRODUCT                                                      WRT01960
      PRODUCT = DOTSQK (MQ,DIFXMN,CVRIN)                                  WRT01970
C                                                                         WRT01980
C  CALCULATE P(X,I) VALUE FOR CURRENT CLASY CLUSTER                       WRT01990
      IF (PRODUCT.LT.150.) PXILN=(EXP(-PRODUCT/2.))*(EXP(-DCON/2.))/CVOLRTWRT02000
      IF (PRODUCT .GE. 150.) PXILN = 0.0                                  WRT02010
      PXLN(I,IPXL)=PXILN                                                  WRT02020
C                                                                         WRT02030
C                                                                         WRT02040
100   CONTINUE                                                            WRT02050
C                                                                         WRT02060
C           END OF DOT DATA PIXELS                                        WRT02070
C  END OF ONE CLASY CLUSTER                                               WRT02080
C                                                                         WRT02090
500   RETURN                                                              WRT02100
      END                                                                 WRT02110
      SUBROUTINE LNEAP                                                    WRT02120
C  PURPOSE:  PRINT LABELED CLUSTER MAP FOR EACH CLASS-CLUSTER             WRT02130
C          PIXEL CLUSTER = MAX(BETA*ALPHA*P(X,I)                          WRT02140
C                                                                         WRT02150
      COMMON  MQ, NOCC, NOCAT, ITOTDT, MQP4, MQS, ICC,                    WRT02160
     1   ICLUNT, IDDUNT, IOUT, CATNAM(20), DCON,                          WRT02170
     2 IDOTS(20,209), CVOLRT, RLPRP( 30), CMEANS(16), CVRIN(158),         WRT02180
     3 PX(30,209), RAP(30), DOTDT(16), LABELS(30),LBLCSS(30),             WRT02190
     4 BETA(20,30),OBETA(20,30),SLK(20,30),SK(30),LBLCST(30)              WRT02200
C                                                                         WRT02210
      COMMON /TAPEBF/ LINETP(196)                                         WRT02220
      LOGICAL*1 LABELS, LBLCSS, LBLCST                                    WRT02230
C                                                                         WRT02240
      COMMON /PXLLN/ NOPXL, LINENO, IBEGIN, IEND, PXLN(30,196),           WRT02250
     1LNDOTS(20,196),LINE(196),IFSTCL,LSTCOL,LSTPG1,IFSTP2,LNSZP1,LNSZP2,WRT02260
      LOGICAL*1 LINE                                                      WRT02270
C                                                                         WRT02280
C                                                                         WRT02290
      COMMON /MPPXL/ MAPPXL(209)                                          WRT02300
C                                                                         WRT02310
      LOGICAL*1 IBLNK                                                     WRT02320
      DATA IBLNK /' '/                                                    WRT02330
       DATA ISIZMD /5/                                                    WRT02340
C                                                                         WRT02350
C  FIND MAX FOR EACH DOT                                                  WRT02360
      DO 50 I = IBEGIN, IEND                                              WRT02370
```

```
50    LINE(I) = IHLNK                                                    WRT02380
C                                                                        WRT02390
C  SET LAST LINE SWITCH TO 'NOT LAST'                                   WRT02400
C                                                                        WRT02410
      DO 200  N = IHEGIN,IEND                                            WRT02420
      PXLMAX = .0                                                        WRT02430
C  CLASSES                                                               WRT02440
      DO 200 L=1,NOCAT                                                   WRT02450
      PIXEL=0                                                            WRT02460
C  CLUSTERS                                                              WRT02470
      DO 100  I = 1,NOCC                                                 WRT02480
 100  PIXEL = PIXEL + BETA(L,I) * RLPRP(I) * PXLN(I,N)                   WRT02490
      IF (PIXEL .LE. PXLMAX) GO TO 200                                   WRT02500
      LINE(N) = LABELS(L)                                                WRT02510
      LINETP(N) = L                                                      WRT02520
      PXLMAX = PIXEL                                                     WRT02530
 200   CONTINUE                                                          WRT02540
C                                                                        WRT02550
C                                                                        WRT02560
C  WRITE PAGE 1                                                          WRT02570
      WRITE (6,209)(LINENO, ( LINE(I),I=IFSTCL, LSTPG1 ))                WRT02580
209   FORMAT (1X, I4, 5X, 110A1)                                         WRT02590
210   FORMAT (I4, 1X,I4,5X,110A1)                                        WRT02600
C                                                                        WRT02610
C                                                                        WRT02620
C  SAVE PAGE 2 ON UNIT 25 IF THERE IS A PAGE 2                           WRT02630
      IF (IFSTP2 .LE. LSTCOL) WRITE (25,210)                             WRT02640
     1  LNSZP2,  LINENO,(LINE(I),I=IFSTP2,LSTCOL)                        WRT02650
C                                                                        WRT02660
      RETURN                                                             WRT02670
      END                                                                WRT02680
      SUBROUTINE LNAP                                                    WRT02690
C  PURPOSE:  PRINT LABELED CLUSTER MAP FOR EACH LINE                     WRT02700
C         PIXEL CLUSTER LABEL = MAX (ALPHA * P(X,I)                      WRT02710
C                                                                        WRT02720
      COMMON  M0, NOCC, NOCAT, ITOTDT, MOP4, MOS, ICC,                  WRT02730
     1   ICLUST, IODUMI, IOUT, CATNAM(20), OCON,                         WRT02740
     2 IDOTS(20,209), CVOLRT, RLPRP( 30), CMEANS(16), CVRIN(158),        WRT02750
     3 PX(30,209), RLP(30), DOTOT(16), LABELS(30),LBLCSS(30),            WRT02760
     4 BETA(20,30),OBETA(20,30),SLK(20,30),SK(30),LBLCST(30)             WRT02770
C                                                                        WRT02780
      COMMON /TAPEBF/ LINETP(196)                                        WRT02790
C                                                                        WRT02800
      LOGICAL*1 LABELS, LBLCSS, LBLCST                                   WRT02810
      COMMON /PXLLN/ NOPXL, LINENO, IHEGIN, IEND, PXLN(30,196),          WRT02820
     1LNDOTS(20,196),LINE(196),IFSTCL,LSTCOL,LSTPG1,IFSTP2,LNSZP1,LNSZP2WRT02830
      LOGICAL*1 LINE                                                     WRT02840
C                                                                        WRT02850
C                                                                        WRT02860
      LOGICAL*1 KBLNK                                                    WRT02870
      DATA KBLNK/' '/                                                    WRT02880
C                                                                        WRT02890
C                                                                        WRT02900
      DO 70 I = IHEGIN,IEND                                              WRT02910
70    LINE(I) = KBLNK                                                    WRT02920
C                                                                        WRT02930
C  FIND MAX FOR EACH PIXEL ON LINE                                       WRT02940
      DO 150  N = IHEGIN,IEND                                            WRT02950
      PXLMAX = 0.0                                                       WRT02960
C                                                                        WRT02970
C  CLUSTERS                                                              WRT02980
      DO 100  I = 1,NOCC                                                 WRT02990
      PIXEL = RLPRP(I) * PXLN(I,N)                                       WRT03000
      IF (PIXEL .LE. PXLMAX) GO TO 100                                   WRT03010
      LINE(N) = LBLCST(I)                                                WRT03020
      LINETP(N) = I                                                      WRT03030
      PXLMAX = PIXEL                                                     WRT03040
 100   CONTINUE                                                          WRT03050
150   CONTINUE                                                           WRT03060
C                                                                        WRT03070
      WRITE (28,210)LNSZP1, LINENO, (LINE (I), I=IFSTCL, LSTPG1)         WRT03080
210   FORMAT (I4, 1X,I4,5X,110A1)                                        WRT03090
C                                                                        WRT03100
C  SAVE PAGE 2 IF THERE IS ONE                                           WRT03110
      IF (IFSTP2 .LE. LSTCOL) WRITE (29,210)                             WRT03120
     1 LNSZP2, LINENO,(LINE(I),I=IFSTP2,LSTCOL)                          WRT03130
C                                                                        WRT03140
      RETURN                                                             WRT03150
      END                                                                WRT03160
```

```
      SUBROUTINE LNAP2                                                  WRT03170
C  PURPOSE:  PRINT LABELED CLUSTER MAP FOR EACH LINE                    WRT03180
C         PIXEL CLUSTER LABEL = MAX (ALPHA * P(X,I)                     WRT03190
C                                                                       WRT03200
      COMMON  MO, NOCC, NOCAT, ITOTDT, MOP4, MOS, ICC,                  WRT03210
     1   ICLUNT, IDDUNT, IOUT, CATNAM(20), DCON,                       WRT03220
     2 IDOTS(20,209), CVOLRT, RLPRP( 30), CMEANS(16), CVRIN(158),       WRT03230
     3 PX(30,209), RAP(30), DOTDT(16), LABELS(30),LBLCSS(30),           WRT03240
     4 BETA(20,30),OBETA(20,30),SLK(20,30),SK(30),LBLCST(30)            WRT03250
      LOGICAL*1 LABELS, LBLCSS, LBLCST                                  WRT03260
      COMMON /PXLLN/ NOPXL, LINENO, IBEGIN, IEND, PXLN(30,196),         WRT03270
     1LNDOTS(20,196),LINE(196),IFSTCL,LSTCOL,LSTPG1,IFSTP2,LNSZP1,LNSZP2WRT03280
      LOGICAL*1 LINE                                                    WRT03290
C                                                                       WRT03300
CC                                                                      WRT03310
      LOGICAL*1 KBLNK                                                   WRT03320
      DATA KBLNK/' '/                                                   WRT03330
C                                                                       WRT03340
C                                                                       WRT03350
      DO 70 I = IBEGIN,IEND                                             WRT03360
70    LINE(I) = KBLNK                                                   WRT03370
C                                                                       WRT03380
C  FIND MAX FOR EACH PIXEL ON LINE                                      WRT03390
      DO 150  N = IBEGIN,IEND                                           WRT03400
      PXLMAX = 0.0                                                      WRT03410
C                                                                       WRT03420
C  CLUSTERS                                                             WRT03430
      DO 100  I = 1,NOCC                                                WRT03440
      PIXEL = RLPRP(I) * PXLN(I,N)                                      WRT03450
      IF (PIXEL .GT. PXLMAX) LINE(N) = LBLCSS(I)                        WRT03460
      IF (PIXEL .GT. PXLMAX) PXLMAX = PIXEL                             WRT03470
100   CONTINUE                                                          WRT03480
150   CONTINUE                                                          WRT03490
C                                                                       WRT03500
      WRITE (26,210)LNSZP1, LINENO, (LINE (I), I=IFSTCL, LSTPG1)        WRT03510
210   FORMAT (I4,1X,I4,5X,110A1)                                        WRT03520
C                                                                       WRT03530
C  SAVE PAGE 2 IF THERE IS ONE                                          WRT03540
      IF (IFSTP2 .LE. LSTCOL) WRITE (27,210)                            WRT03550
     1  LNSZP2, LINENO,(LINE(I),I=IFSTP2,LSTCOL)                       WRT03560
C                                                                       WRT03570
      RETURN                                                            WRT03580
      END                                                               WRT03590
      SUBROUTINE PAGE(IUNIT)                                            WRT03600
C                                                                       WRT03610
      DIMENSION LINE(196)                                               WRT03620
C                                                                       WRT03630
C  PURPOSE:  READ DATA FOR PAGE FROM TEMPORARY FILE AND WRITE REPORT    WRT03640
C            TITLE AND COLUMN HEADING PRINTED BY WRTLNS                 WRT03650
C                                                                       WRT03660
      REWIND IUNIT                                                      WRT03670
C                                                                       WRT03680
C  PRINT LINES OF REPORT                                                WRT03690
510   READ (IUNIT, 20, END=590) ISIZE, (LINE(I),I=1,ISIZE)              WRT03700
20    FORMAT (I4,32A4)                                                  WRT03710
      WRITE (6,30) (LINE(I), I=1,ISIZE)                                 WRT03720
30    FORMAT (33A4)                                                     WRT03730
      GO TO 510                                                         WRT03740
C                                                                       WRT03750
590   REWIND IUNIT                                                      WRT03760
      RETURN                                                            WRT03770
      END                                                               WRT03780
```