1

100780

LARSFRIS

PROGRAM ABSTRACTS

Module Name	LARS Program Abstract Number	Module Name	LARS Program Abstract Number
ARBASE	301 - 474	COSPEC	334 - 570
BATRD	055 - 108	COVIN	158 - 276
BCDVAL	001 - 5	CPFUNC	003 - 12
BIPCLA	404 - 682	CTLWRD	004 - 15
BIPDIV	406 - 684	DECCLS	124 - 202
BIPELL	403 - 681	DISPLY	241 - 372
BIPLTR	402 - 680	DISPY1	242 - 383
BIPMEN	405 - 683	DISPY2	243 - 386
BIPRDR	401 - 679	DIVERG	303 - 479
BIPSUP	400 - 678	DIVPRT	304 - 482
BISPLT	335 - 574	DIVRG1	305 - 492
BLOAD	100 - 170	DIVRG2	306 - 495
BONDSU	122 - 194	ERMNAM	101 - 172
BPROFILE	056 - 124	ERPRNT	102 - 174
BSTCHK	302 - 476	ERRINT	103 - 178
CCHIS	181 - 322	EXCOMD	057 - 126
CGROUP	179 - 317	FILBUF	366 - 628
CHANEL	002 - 9	FIXFLD	176 - 306
CHANGE	384 - 658	FLDBN	244 - 389
CHASUP	271 - 435	FLDBOR	231 - 352
CHTAPE	385 - 661	FLDCOV	323 - 532
CLACOM	151 - 253	GADLIN	005 - 22
CLAINT	152 - 254	GADRUN	006 - 29
CLARDR	153 - 258	GATHER	371 - 645
CLASS	154 - 262	GBIPSUP	083 - 163
CLASUP	150 - 251	GCLASUP	071 - 152
CLSCHK	123 - 197	GCLUSUP	072 - 153
CLSDEC	142 - 248	GCOL	191 - 328
CLSFY1	155 - 265	GCOMSUP	084 - 164
CLSFY2	156 - 268	GENALL	054 - 106
CLSHIS	321 - 525	GENALLDV	089 - 169
CLSSPC	322 - 528	GETDAT	307 - 498
CLUCOM	171 - 289	GETINV	308 - 501
CLUMP	173 - 294	GETSHW	309 - 504
CLUMP1	174 - 298	GGRHSUP	073 - 154
CLUPRO	175 - 301	GHISSUP	074 - 155
CLURDR	172 - 290	GLARSMN	069 - 150
CLUSUP	170 - 286	GLIN	192 - 332
CMGRAY	218	GLOCOM	104 - 180
CMSFNC	035 - 103	GMERSUP	085 - 165
COMCOM	381 - 650	GPICSUP	076 - 156
COMPAR	383 - 655	GPRISUP	077 - 157
COMRDR	382 - 651	GRESSUP	078 - 158
COMSUP	380 - 648	GRATSUP	086 - 166
CONTEX	157 - 273	GRHIST	125 - 205
COPY	261 - 420	GRHRDR	193 - 336
BIPCOM	407 - 685		

Module Name	LARS Program Abstract Number	Module Name	LARS Program Abstract Number
GRHSUP	190 - 325	PICRDR	233 - 357
GRPSCN	126 - 208	PICSUP	230 - 350
GRUNSUP	079 - 159	PIC1	234 - 362
GSAMSUP	080 - 160	POLMER	340 - 587
GSECSUP	087 - 167	POLSET	341 - 590
GSEPSUP	081 - 161	PRACRE	252 - 414
GSMOSUP	088 - 168	PRICOM	246 - 395
GSTASUP	082 - 162	PRIINT	247 - 396, 399
GTDATE	007 - 33	PRIRDR	248 - 402
GTSERL	159 - 279	PRISUP	240 - 369
HEADER	127 - 213	PROCES	106 - 184
HISRDR	201 - 346	PRTHED	249 - 405
HISSUP	200 - 344	PRTPCT	250 - 408
HISTD	128 - 216	RATCOM	391 - ~~665~~ 668
IBCD	025 - 92	RATIO	394 - 675
IDNAME	008 - 36	RATRDR	392 - 669
INVPNT	180 - 320	RATSUP	390 - 665
LAREAD	129 - 222	RATTAP	393 - 672
LARSMN	105 - 181	RDFLDS	131 - 230
LARS12	009 - 38	READMX	354 - 602
LEARN	324 - 535	REDFLD	337 - 580
LINRDR	194 - 339	REDSAV	133 - 234
LIST	058 - 131	REDSTA	134 - 238
LLARSMN	070 - 151	REFERENC	061 - 137
LOGICOPS	014 - 86	RESCOM	262 - 424
LOGLIK	368 - 634	RESCOP	264 - 428
MCONTX	161 - 283	RESRDR	263 - 425
MDIST	177 - 311	RESSUP	260 - 417
MERCOM	331 - 561	RTBSUP	107 - 185
MERINT	332 - 562	RTMAIN	028 - 95
MERRDR	333 - 565	RUNCOM	272 - 438
MERSTT	338 - 582	RUNERR	120 - 188
MERSUP	330 - 559	RUNLS	062 - 140
MEXPAN	339 - 584	RUNSUP	270 - 431
MMTAPE	130 - 226	SAMCLS	367 - 631
MOUNT	010 - 41	SAMCOM	281 - 451
MOVBYT	015 - 84	SAMINT	282 - 452
NEWS	059 - 133	SAMINV	283 - 455
PACCT	029 - 97	SAMRDR	284 - 458
PBID	060 - 135	SAMSUP	280 - 448
PCHFLD	178 - 314	SECCOM	361 - 611
PCHSTA	325 - 549	SECHO1	370 - 640
PCTTAL	245 - 392	SECINT	363 - 620
PICCOM	232 - 356	SECPRT	365 - 625
PICINT	235 - 366	SECRD	364 - 623

Module Name	LARS Program Abstract Number
SECRDR	362 → 615
SECSUP	360 - 609
SEPCOM	310 - 507
SEPINT	311 - 508
SEPRDR	312 - 511
SEPSUP	300 - 471
SMCLS1	285 - 461
SMCLS2	286 - 464
SMMULT	287 - 469
SMOCOM	351 - 595
SMOINT	352 - 596
SMOOTH	355 - 605
SMORDR	353 - 598
SMOSUP	350 - 593
SMOTAB	356 - 608
SPACE	063 - 142
STACOM	326 - 552
STAINT	327 - 553
STARDR	328 - 556
STASUP	320 - 523
STAT	135 - 241
STATE	032 - 101
STATRD	336 - 578
STATS	251 - 411
STATS2	369 - 637
SUBGEN	275 - 446
SYMSET	313 - 515
TAPOP	011 - 45
TERMTEST	068 - 148
THRESC	160 - 280
TIMER	016 - 87
TRARDR	273 - 439
TRAXEQ	274 - 443
TSPACE	136 - 244
TSTREQ	012 - 75
TYPEITEM	064 - 145
UNIDAT	108 - 187
UNMNT	121 - 196
URADST	013 - 77
USER	314 - 518
WRTMTX	017 - 89
WRTTRN	137 - 246

LARS Program Abstract 001

MODULE IDENTIFICATION

Module Name: BCDVAL Function Name: SYSTEM SUPPORT

Purpose: Interprets and stores BCD and fullword numeric values.

System/Language: CMS/FORTRAN

Author: E.M. Rodd Date: 10/11/72

Latest Revisor: Date:

MODULE ABSTRACT

BCDVAL will move a string of characters, separated by commas into a vector.

IVAL converts numeric characters to integer representations and stores the values. There may be a set of integers separated by commas.

FVAL is like IVAL except that it works with floating point numbers.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

BCDVAL

CALL BCDVAL (CARD,COL,VEC,VECSZ,*)

Input Arguments:

CARD - L*1, Card image of card being interpreted

COL - I*4, Column number preceding column containing next useable information (i.e., start interpretation at column COL)

VECSZ - I*4, Number of values to be found, interpreted and stored.

Output Arguments:

VEC - I*4, VEC (I) will be returned with the I'th character found. Note that if the card contains more than one character between commas, the first of those characters will be in VEC (I).

VECSZ - I*4, If values less than VECSZ (as input) are found, VECSZ will be returned with the number of values found.

Non-Standard Return will be taken if:

a. Right parenthesis is missing

b. Right parenthesis found and no left parenthesis was used

c. More than VECSZ values found

d. Syntax errors detected in attempt to compute COL, which include:

   1. Two commas found before next non-blank

   2. Left parenthesis found before another non-blank

   3. Sequence blank followed by non-blank with no comma

COL I*4 is returned as the column number preceding the next control parameter.

BCDVAL is used to get a set of single characters separated by commas, and put them into VEC, one character per element of VEC. The set of characters may be enclosed in parenthesis.

If no characters are found (i.e., COL comes in as 72, the first character encountered is a comma or the rest of the card is blank), VECSZ is returned as zero.

IVAL

CALL IVAL (CARD,COL,VEC,VECSZ,*)

Input Arguments:

CARD - Same as those for BCDVAL

COL - Same

VECSZ - Same (except integer values, not characters)

Output Arguments:

VEC - I*4, VEC(I) will contain as an integer the I'th integer found on the card

VECSZ - Same

COL - Same

Non-Standard Return will be taken if:

a. Same as those for BCDVAL

b. No values are found

c. A non-numeric character is found (other than the comma used to separate values or the asterisk used to indicate repetition)

IVAL is used to interpret a set of integer values separated by commas and which may be in parenthesis.

FVAL

CALL FVAL (CARD,COL,VEC,VECSZ,*)

Arguments are the same as for IVAL. The difference is that FVAL places the results in VEC in floating point representation and can interpret numbers containing decimal points.

Note that in all three entries the form "n*value" can be used to indicate n repetitions of the value.

2. Internal Description

This subroutine uses FORTRAN character manipulation consisting of moving single characters from a LOGICAL*1 array (CARD) to a fullword by equivalencing a LOGICAL*1 variable to the first byte of the fullword. Comparisons can then be performed using the fullword variable. The internals make use of machine representations of numbers and characters. The program uses the same code for all entries with branches.

Program is mostly a loop:

a. Loops over the columns

b. When a comma is found, the value is placed in VEC

c. When right parenthesis is detected or the end of the card is reached, the program returns.

d. The loop is run through until a value is found. Then COL is updated, and the loop is reentered starting at COL to find the next value.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 002

MODULE IDENTIFICATION

Module Name: CHANEL Function Name: SYSTEM SUPPORT

Purpose: Interprets channel card

System/Language: CMS/FORTRAN

Author: T. Ransom Date:

Latest Revisor: E. M. Rodd Date: 12/12/72

MODULE ABSTRACT

CHANEL interprets the channel card and returns all information contained on the card in a form usable by functional programs.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CHANEL

CALL CHANEL (CARD,COL,NCR,CSEL,CSET,CHAN,*)

Input Arguments:

CARD - I*4, 18 fullword buffer containing card image of CHANNELS card to be interpreted.

COL - I*4, The column number in CARD preceding the beginning of channels information on the CHANNELS card.

CSEL - I*2, CSEL(I) = calibration code for channel I if channel I has been selected (by a previous call to CHANEL). Otherwise CSEL(I) = 0. This means that for the first call to CHANEL, CSEL(I) = 0 for all I.

CSET - R*4, 90 fullword vector containing calibration values for channels selected. CSET(3*I-2) = C0 for channel I, CSET(3*I-1) = C1 for channel I and CSET(3*I) = C2 for channel I. Values for all channels not yet selected or for values not explicitly specified on a previous CHANNELS card are =-50000.0. This means that for the first call to CHANEL, CSET(I) =-50000.0 for all I.

CHAN - I*2, CHAN(I) = channel number of the I'th channel selected by previous CHANNEL cards. This means that for the first call to CHANEL, CHAN(I) = 0 for all I.

Output Arguments:

COL - I*4, For a normal return, the number of the last column processed. For a non-standard return, the last column number processed before the error was detected.

NCR - I*4, For a normal return, the number of entries in CHAN (i.e., the total number of channels selected thus far). For a non-standard return, a column number known to be after the column containing the error.

CSEL - Same definition as for input except that it is updated by the CHANNEL card being interpreted.

CSET - Same definition as for input except that it is updated by the CHANNEL card being interpreted.

CHAN - Same definition as for input except that it is updated by the CHANNEL card being interpreted.

Non-Standard Return:

RETURN 1 is executed when an error is detected on the CHANNELS card. Errors detected are for syntax, invalid calibration codes and invalid channel number. Channel number must be between 1 and 30 and calibration code must be between 1 and 7.

CHANEL is called to interpret the CHANNELS card. Successive calls can be made to CHANEL for successive CHANNELS cards and CHANEL will add information from each card to the arrays. This means that the arrays CSEL, CSET and CHAN must be initialized before the first call. Invalid data in these arrays can cause unpredictable (and difficult to locate) errors.

2. Internal Description

CHANEL interprets the entire card. It contains an internal subroutine beginning at statement 200 which processes the next item and then returns to the part of the program which called it.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

The CHANNEL card is described in the control card dictionary for each function which uses it.

6. Flowchart

Not Applicable

LARS Program Abstract 003

MODULE IDENTIFICATION

Module Name: CPFUNC Function Name: SYSTEM SUPPORT

Purpose: Executes CP console functions

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd Date: 08/01/72

Latest Revisor: Date:

MODULE ABSTRACT

This program allows FORTRAN programs to execute CP console functions. If an error occurs, an error code is passed to the caller.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CPFUNC

CALL CPFUNC (NCHAR, TEXT, ERROR)

Input Arguments:

NCHAR - I*4 The number of characters in the CP console function to be executed.

TEXT - This is a literal containing the text of the CP console function. It may be input in two ways: one is as a literal in the call (e.g. 'DETACH 181') and the other is that TEXT may be an array containing the text in characters.

Output Arguments:

ERROR - I*4 The error code returned from CP. If NCHAR is greater than 99, ERROR will be returned = 100+ the error code from CP and only the first 99 characters of the console function will be sent to CP.

Note that if NCHAR does not match the number of characters in TEXT, strange results could occur.

2. Internal Description

CPFUNC first executes the CMS CONWAIT function to clear all the buffered output to the terminal. This is done so that a response from CP will appear in the correct place on the terminal output. Then NCHAR is checked. If it is greater than 99, it is set to 99 and R2 initialized to 100 rather than 0. NCHAR itself is not changed, only the register used internally is changed. Then that number of characters is moved from TEXT to the buffer of the CMS CPFUNCTN function. The X'FF' fence is then added after the last character. Then CPFUNCTN is executed via SVC 202 and R15 is added to R2 to form the final error code. This value is placed in ERROR and CPFUNC returns.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 004

MODULE IDENTIFICATION

Module Name: CTLWRD Function Name: SYSTEM SUPPORT

Purpose: READS and interprets control cards

System/Language: CMS/FORTRAN

Author: P. Spencer Date:

Latest Revisor: E. M. Rodd Date: 08/02/72

MODULE ABSTRACT

CTLWRD and its entry points are used to read and interpret control and data cards. CTLWRD is the only entry point which reads cards. It also interprets keywords on control cards. CTLPRM interprets control parameters. DATCRD interprets keywords. BCDFIL moves a character string from a control or data card to an array. LOCATE locates a certain character CTLWRD recovers from errors wherever possible.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CTLWRD

CALL CTLWRD (CARD,COL,LIST,LISTSZ,CODE,READIN,ERRCOR)

Input Arguments:

LIST - L*1, Vector of character strings containing keywords being searched (LIST is a list of 4 character strings.)

LISTSZ - I*4, Number of elements in LIST.

READIN - I*4, Unit number from which control cards are being read.

ERRCOR - I*4, = 0 if the control card is to be read from unit READIN

= 1 if CTLWRD has been called to read a corrected card from the terminal

= 3 if CTLWRD has been called to read an additional control card from the terminal (that is, a card to supply required information missing from the supplied control cards). ERRCOR=3 differs from ERRCOR=1 only in the value of ERRCOR returned and in error handling.

Output Arguments:

CARD - L*1, Vector containing the card read in EBCDIC (Card should be demensioned CARD(80))

COL - I*4, Column number of the column before the control parameter. If no control parameter, COL = 72

CODE - I*4, Character string number in LIST which matched the keyword

ERRCOR - I*4, = 0 if the keyword was found and ERRCOR ≠ 3 on input.

= 2 if EOF was read on the normal input unit (READIN).

= 3 if ERRCOR came in = 3.

= 4 if 'KILL' is read.

CTLWRD will read a card from the unit (READIN or TYPEWR) and determine the keyword by matching it to an element of LIST. It does its own error recovery (see Internal Description).

CTLPRM

CALL CTLPRM (CARD,COL,LIST,LISTSZ,CODE,*)

Input Arguments:

CARD - L*1, Card image.

COL - I*4, Column number before first character of control parameter.

LIST - Same as in CTLWRD (except LIST has control parameters).

LISTSZ - Same as in CTLWRD.

Output Arguments:

COL - I*4, Column number before the next non-blank after a comma, or the column number of a left parenthesis.

CODE - I*4, Same as in CTLWRD.

A non-standard return is made for unrecognized control parameter or syntax error.

CTLPRM determines the identity of a control parameter.

DATCRD

CALL DATCRD (CARD,COL,LIST,LISTSZ,CODE,*)

Input Arguments:

CARD - Same as in CTLPRM.

COL - I*4, Must be input as 0.

LIST - Same as in CTLWRD.

LISTSZ - Same as in CTLWRD.

Output Arguments:

COL - Same as in CTLPRM.

CODE - Same as in CTLWRD.

A non-standard return is taken for unrecognized keyword or syntax errors.

DATCRD determines the identity of the keyword on a data card. DATCRD DIFFERS from CTLWRD and CTLPRM in error detection and recovery (see internals).

BCDFIL

CALL BCDFIL (CARD,COLST,COLEN,VEC,VECSZ)

Input Arguments:

CARD - Same as CTLPRM.

COLST - I*4, Column number in which character string starts.

COLEN - I*4, Column number where character string ends.

VECSZ - I*4, Number of bytes of VEC to be filled.

Output Arguments:

VEC - L*1, Filled with the characters between COLST and COLEN from CARD. If VECSZ is greater than the number of characters from COLST to COLEN, VEC is padded with blanks to fill VECSZ characters.

BCDFIL is used to move a character string from an input card to an array. No possible error conditions.

LOCATE

CALL LOCATE (CARD,COL,CHLIST,LISTSZ,CODE,*)

Input Arguments:

CARD - Same as in CTLPRM.

1. CTLWRD and DATCRD if a comma immediately follows the keyword with no intervening blanks.

2. In CTLPRM if two commas are detected with no intervening non-blanks.

3. In CTLPRM if a left parenthesis is detected after a comma with no intervening non-blanks.

4. In CTLPRM if a non-blank is found separated from the control parameter by blanks only and not a comma nor left parenthesis.

Below is the action taken when an unrecognized keyword or syntax error is detected.

CTLWRD - The card is typed and an error message is written via ERPRNT. The terminal is unlocked to type in the corrected control card. At this time, if the response is a Carriage Return, and ERRCOR $\neq$ 3 the card is ignored and the next control card is read from unit READIN. If ERRCOR = 3 and the response is Carriage Return, error 536 is written via ERPRNT and the terminal is unlocked to accept the card.

CTLPRM - The card is typed and an error message is written via ERPRNT and a RETURN 1 is executed.

DATCRD - If the keyword is unrecognized, a RETURN 1 is executed. If a syntax error is detected, a message is written via ERPRNT and then RETURN 1 is executed.

3. Input Description

Only CTLWRD reads cards from the card reader or typewriter.

4. Output Description

1. CTLWRD prints the card image read in

2. CTLWRD and CTLPRM type an erroneous card

3. Messages (via ERPRNT) =

   102 - Unrecognized keyword

   103 - Unrecognized control parameter

   104 - Syntax error

   536 - Additional card must be supplied! Type correct card.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract

MODULE IDENTIFICATION

Module Name: GADLIN Function Name: SYSTEM SUPPORT

Purpose: Obtains a data line from a Multispectral Image Storage Tape

System/Language: CMS/FORTRAN

Author: T.L. Phillips Date: 01/30/70

Latest Revisor: J.S. Buis Date: 07/10/79

MODULE ABSTRACT

GADLIN reads a line of data from a Multispectral Image Storage Tape and calibrates the data. Errors are reflected to the caller via error codes. Entry RLCP retrieves the calibration parameters used by the last call to GADLIN.

GADLIN determines whether the MIST tape is in Universal or LARSYS format through common block UNIDAT and reads a line of data accordingly.

1. Module Usage

GADLIN

CALL GADLIN (BLOCK,CSEL,CSET,ID,DATUNT,NCD,NSD,BDATA,
RDATA,ROLL,ERROR)

Input Arguments:

BLOCK - I*2 Each call to GADLIN returns data from one line of data. BLOCK(1) defines the desired line of data. BLOCK(2) defines the first sample of data desired. BLOCK(3) defines the last sample of data desired. BLOCK(4) defines the sample interval.

CSEL - I*2 Multispectral Image Storage Tape can contain any number of channels up to 30. A particular run contains ID(5) channels. If data is not desired from Nth channel, CSEL(N) should be set to 0. Otherwise CSEL(N) contains the calibration code. The valid codes are defined below.

CSEL(N)	Data From Channel N Calibrated on
1	C0
2	C1
3	C2
4	C0 and C1
5	C0 and C2
6	C1 and C2
7	No Calibration

CSET - R*4 Calibration information from N channels of data. The calibration information for channel i must be stored for C0 in CSET(1,i), for C1 in CSET(2,i), and for C2 in CSET(3,i). The use of these variables is discussed in the output section. CSET should be the information taken from the 10th record of the tape unless the values were explicitly computed by the caller.

ID - I*4 This is the ID record from the Multispectral Image Storage Tape. It is 200 fullwords. For the definition of the ID array, see the data set description of the Multispectral Image Storage Tape in the System Manual.

DATUNT - I*4 The unit must be defined as a FORTRAN unit and the same as was used in GADRUN.

NCD - I*4 NCD is the number of channels dimensioned in RDATA and must be greater than or equal to the number of selected channels.

NSD - I*4 NSD is the number of samples dimensioned in RDATA and must be greater than or equal to [BLOCK(3) - BLOCK(2)]/ BLOCK(4) + 1 + 6 for calibration values.

BDATA - BDATA is an array used by GADLIN as a buffer to read the bytes of Data from each requested channel on the tape. BDATA must be dimensioned for NCR*ID(6) bytes: where NCR is the number of channels requested. (i.e., number of non-zero entries in CSEL)

Output Arguments:

RDATA - R*4 RDATA must be dimensioned at least (NSD*NCD) in the calling program. The RDATA array is used as an output to the calling program for the samples of data requested. The calibrated or uncalibrated data samples requested are stored in consecutive real fullword locations for the channels requested. The calibration parameters are stored in the last six real fullwords (NSD-5 to NSD) for each channel requested.

All data is calibrated according to the calibration code by the equation:

RDATA(I,J) = A*RAW(I,J) + B

Where:

RDATA(I,J) = the calibrated value for the I'th sample requested for the J'th channel requested. Used this way, think of RDATA as a two-dimensional array dimensioned (NSD,NCD).

RAW(I,J) = the raw (uncalibrated) data point for the I'th sample requested for the J'th channel. This raw value has been converted from the byte integer format of the data tape to floating point fullword format.

A and B are calculated according to the following algorithm depending upon the calibration code. A and B are calculated for each channel.

The following notation is used in the expressions for A and B:

C0 = CSET(1,N) where N is the channel number.
C1 = (2,N)
C2 = (3,N)
K0 = the value of C0 taken from the tape record for this line for this channel. K0 = (ID(6)-5)th byte of data for the channel.
K1 = (ID(6)-3)th byte of data for this channel.
K2 = (ID(6)-1)th byte of data for this channel.

Calibration Code	A	B
1	1.0	C0-K0
2	1.0	C1-K1
3	1.0	C2-K2
4	$\frac{C0-C1}{K0-K1}$	$\frac{K0*C1-K1*C0}{K0-K1}$
5	$\frac{C0-C2}{K0-K2}$	$\frac{K0*C2-K2*C0}{K0-K2}$
6	$\frac{C1-C2}{K1-K2}$	$\frac{K1*C2-K2*C1}{K1-K2}$
7	1.0	0.0

ROLL - I*4, The roll parameter on the tape. (See the Multispectral Image Storage Tape documentation in the User's Manual.)

ERROR - I*4, ERROR returns are as follows:

ERROR	DESCRIPTION OF ERROR
0	No Error exists.
1	Data line requested does not exist on tape.
2	Incorrect byte count in data record requested. Two read retries were made.
3	Parity check error occurred.
4	Hardware parity error occurred on reading tape.
5	Some error combination of errors 2 and 3 or 4. Data delivered is probably in error.
6	Tape Unit was not assigned.
7	BLOCK(1) is less than or equal to 0.
8	BLOCK(2) is less than or equal to 0.
9	BLOCK(4) is less than or equal to 0.
10	BLOCK(2) is greater than BLOCK (3).
11	A channel flag is less than 0 or greater than 10.
12	No channels were selected or selected channel was not run.
13	The number of channels requested is greater than the number of channels dimensioned in RDATA.

ERROR	DESCRIPTION OF ERROR
14	The number of samples requested ((BLOCK(3)-BLOCK(2))/BLOCK(1)+7) is greater than the number of samples dimensioned (NSD).
15	Data in requested line does not exist (ROLL = 0).
16	Data cannot be calibrated as requested.
17	If requested line cannot be located on tape.

NOTE: All error conditions will return very questionable data or no data at all except for ERROR = 12 which returns only the ROLL parameter and no data.

RLCP

CALL RLCP (I,A,B)

Input Arguments:

I - INTEGER*4, Channel number of the channel for which calibration values are requested.

Output Arguments:

A - REAL*4, The value of A used in the last call to GADLIN.

B - REAL*4, The value of B used in the last call to GADLIN.

2. Internal Description

When the MIST tape is in LARSYS format, GADLIN uses TOPRV to read the data tape, but when the tape is in Universal format TOPRD is used. Only those channels requested are actually transferred into main storage. Sub-routine URADST is used to unpack the data (and calibration) values from the byte integer format into fullword floating point format. GADLIN assumes only that the tape is positioned somewhere in the correct run. It reads the line at which the tape is positioned and determines if that is the correct line, or if the tape must be moved backward or foward.

3. Input Description

The data tape is read via a call to TOPRV or TOPRD, depending on if the tape is in LARSYS or Universal format. The tape is positioned to the proper line by TOPFS to move the tape foward, or by either TOPBS or the combination of TOPRF and TOPFS to move the tape backward.

4. Output Description

Not applicable

5. Supplemental Information

If the MIST tape is of Universal format, GADRUN has created a LARSYS formatted ID record from the information in the Universal header record. The format of the header record is documented under LACIE Input Tape Header Record Format, Appendix D, in the EOD LARSYS & Universal tape format documentation notebook.

6. Flowchart

Not applicable

LARS Program Abstract 006

MODULE IDENTIFICATION

Module Name: GADRUN Function Name: SYSTEM SUPPORT

Purpose: Locates data runs in the runtable and mounts tape

System/Language: CMS/FORTRAN

Author: T. L. Phillips Date: 09/09/69

Latest Revisor: J.S. Buis Date: 07/10/79

MODULE ABSTRACT

GADRUN and entry GETRUN are used to find the tape containing a given run number and then mount the tape and position it at the correct file and validate that the correct run is present. Entry GETRUN differs from GADRUN only in that it first searches a user table of runs before the system RUNTABLE.

GADRUN determines whether the tape is in Universal or LARSYS format, and shares this information with GADLIN through common block UNIDAT.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GADRUN

CALL GADRUN (RUNSEL,DATUNT,ID,ERROR)

Input Arguments:

RUNSEL - INTEGER*4, The run number of the run GADRUN is to locate and mount.

DATUNT - INTEGER*4, The DSRN to be used for the desired data tape.

ID - INTEGER*4, On input to GADRUN ID(1) should contain the tape number of any Multispectral Image Storage Tape currently mounted on DSRN DATUNT. If no tape is currently mounted, ID(1) = 0. If a tape is mounted, ID(2) = the file number it is currently positioned in.

Output Arguments:

ID - INTEGER*4, The full 200 word ID record from the desired run. This array is described in the description of the Multispectral Image Storage Tape.

ERROR - INTEGER*4, The error code. See section four for a description of these errors.

GETRUN

CALL GETRUN (RUNSEL,DATUNT,ID,ERROR,RUNTAB,IMARK)

All arguments are the same except the following two input arguments have been added.

RUNTAB - A user runtable dimensioned (10,3). It contains up to 10 entries and is of the form:

RUNTAB (i,1) - Run number
RUNTAB (i,2) - Tape number
RUNTAB (i,3) - File number

IMARK - The number of entries in the array RUNTAB.

GETRUN and GADRUN function the same except that GETRUN first searches the user supplied runtable before the system runtable.

2. Internal Description

GADRUN and GETRUN locate the run in the user or system runtable. If the run is not found, an error code is returned. They then determine if the correct tape is mounted. If so, it is positioned correctly. If not, subroutine MOUNT is called to mount the correct tape with the ring out. The tape is then positioned. When a tape is mounted, the ID record of the first file is read to check the tape number, and if the tape number in this record does not agree with the tape number requested, CPFUNC is called to send the operator a message indicating the wrong tape was mounted. The tape is then unloaded via TOPRU and MOUNT is called again.

3. Input Description

The system runtable is read to locate tapes corresponding to the given run. This data set is described in the data organization section of the System Manual.

The Multispectral Image Storage Tape is read to obtain the ID record of the desired run.

If the MIST tape is in Universal format, GADRUN will construct a LARSYS formatted ID record from the information in the Universal header record. The format of the Universal header record is documented under LACIE Input Tape Header Record Format, Appendix D, in the EOD LARSYS and Universal tape format documentation notebook.

4. Output Description

Both entries will produce the messages:

10035 SEARCHING FOR RUN nnnnnnnn

10036 DESIRED RUN FOUND nnnnnnnn

Depending on which format the tape is, GADRUN will produce one of the following messages:

If data tape is in LARSYS format,

INNN Data is in LARSYS format.

If data tape is in Universal format,

INNNN Data is in universal format.

0 - No errors

1 - EOF was detected where ID record was expected.

2 - The correct ID record was read with a wrong length indication.

3 - The expected ID record was read but with a tape parity error.

4 - The expected ID record was read but with a hardware parity error.

5 - The expected ID record was read but with a parity error and wrong length count.

6 - The DSRN passed to GADRUN is invalid.

7 - The ID record was read from the expected tape and file number but contains the wrong run number.

8 - The requested run number was not found.

5. Supplemental Information

GADRUN uses the routine MOUNT to request the operator to mount the tape. All positioning and mounting of the tape is done using TAPOP.

If the data is in Universal format, the user must supply a run number. If the data is in LACIE Universal format, the user-supplied run number must consist of the acquisition number, followed by three zeros, where the acquisition # is the Julian date the data was collected.

6. Flowchart

Not Applicable

LARS Program Abstract 007

MODULE IDENTIFICATION

Module Name: GTDATE Function Name: SYSTEM SUPPORT

Purpose: Fetches date and time of day

System/Language: CMS/ASSEMBLER

Author: P. E. Anuta Date: 09/09/68

Latest Revisor: P. W. Spencer Date: 02/26/75

MODULE ABSTRACT

GTDATE and IGTDAT fetch the date and entry point GETIME fetches the time of day. Both GTDATE and GETIME return 12 character EBCDIC strings ready for printing by 3A4 format. IGTDAT returns the month, day and year in 3 successive locations as integers. The OS TIME macro is used to get the time of day and date.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GTDATE

CALL GTDATE (DATE)

Output Arguments:

DATE - This is a 3 fullword area. The date is returned in characters ready for printing with 3A4 format. The format is 'MMMM dd,yyyy'.

IGTDAT

CALL IGTDAT (DATE)

DATE - This is a 3 fullword area. The date is returned as month in word 1, day in word 2, and the last 2 digits of the year in word 3.

GETIME

CALL GETIME (TIME)

Output Arguments:

TIME - This is a 3 fullword area. The time is returned in characters ready for printing with 3A4 format. The format is 'hh mm ss $\genfrac{}{}{0pt}{}{P}{A}$ M '. The time is on a 12 hour clock, and the PM or AM is specified.

2. Internal Description

All 3 entries use the OS TIME macro to access the date and time. See the IBM reference manual 'SUPERVISOR SERVICES' for the description of this macro. (IBM manual GC28-6646-6). GTDATE and GETIME then manipulate this information into the desired format.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 008

MODULE IDENTIFICATION

Module Name: IDNAME Function Name: SYSTEM SUPPORT

Purpose: Obtain USERID and USERNAME from CP UTABLE

System/Language: CMS/ASSEMBLER

Author: Howard Grams Date: 11/10/72

Latest Revisor: Date:

MODULE ABSTRACT

IDNAME is a fortran callable routine used to obtain the 8-character USERID and 16-character USERNAME from the CP UTABLE and return them to the caller.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

IDNAME

CALL IDNAME (USERID,NAME)

Output Arguments:

USERID - Is an 8-character array (e.g. REAL*8, or INTEGER*4 dimensioned 2).

NAME - Is a 16-character array (e.g. REAL*8 dimensioned 2, or INTEGER*4 dimensioned 4).

Upon return, USERID and NAME will be filled in with information extracted from the CP UTABLE.

2. Internal Description

A diagnose instruction is issued with code 100. Code 100 is a LARS-defined code to extract USERID, ACCTNG, and VMUSER1-4 (which at LARS contains the user's name) from CP. See LARS system file 9010 for more details.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

This routine depends on the non-standard CP module PRIVLGED which supports LARS-defined diagnose codes.

6. Flowchart

Not Applicable

LARS Program Abstract 009

MODULE IDENTIFICATION

Module Name: LARS12 Function Name: SYSTEM SUPPORT

Purpose: Interprets fixed form field description card

System/Language: CMS/FORTRAN

Author: W. Simmons Date: 03/24/69

Latest Revisor: E. M. Rodd Date: 12/12/72

MODULE ABSTRACT

LARS12 interprets the fixed format field description card.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LARS12

CALL LARS12 (CARD,INFO,*)

Input Arguments:

CARD - L*1, An eighty character vector (usually I*4 and 20 fullwords in the calling routine) containing a card image of the field description card to be interpreted.

Output Arguments:

INFO - I*4, 17 fullword array of following format.

INFO(1) = number from columns 1-8 of card (run number)

INFOR(2-3) = 8 character description from columns 11-18 of card.

INFO(4) = number from cols 21-25 of card (first line)

INFO(5) = number from cols 26-30 of card (last line)

INFO(6) = number from cols 31-35 of card (line interval)

INFO(7) = number from cols 36-40 of card (first sample)

INFO(8) = number from cols 41-45 of card (last sample)

INFO(9) = number from cols 46-50 of card (sample interval)

INFO(10-17) = 30 characters from columns 51-80 of card and two blanks at the end.

RETURN 1 is executed if any non-numeric character is found in columns 1-8 or 21-50. When RETURN 1 is executed, INFO(1) and INFO(2) contain the column numbers of the field (or fields) in error.

The numbers in the numeric fields need not be right or left justified and can contain blanks between significant digits.

2. Internal Description

LARS12 uses FORTRAN LOGICAL variables to manipulate and examine each byte of the card it is decoding. If any non-numeric is found in a field which should be numeric, RETURN 1 is executed.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 0010

MODULE IDENTIFICATION

Module Name: MOUNT Function Name: SYSTEM SUPPORT

Purpose: Auto wait for tape program

System/Language: CMS/Assembler

Author: E. M. Rodd Date: 08/08/72

Latest Revisor: William C. Zurney Date: 08/14/75

MODULE ABSTRACT

This program is a FORTRAN callable subroutine for requesting and waiting for a tape to be mounted.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Syntax:

```
CALL MOUNT (tape, dsrn, ringflag, unittype)
```

Required Parameters:

```
tape     -- Integer *4 - tape number
            or 8 character- tape number
dsrn     -- Integer *4 - FORTRAN data set reference
                         number on which the tape is to mou
ringflag - 2 characters - 'RI' - Request tape with
                                 ring in.
                          'RO' - Request tape with
                                 ring out.
```

Optional Parameter:

```
unittype -- characters

             'ANY' - Request any tape drive
            '9TRK' - Request a 9-track tape drive
            '7TRK' - Request a 7-track tape drive
             '800' - Request a 800BPI 9-track tape drive
            '1600' - Request a 1600BPI 9-track tape drive
            '6250' - Request a 6250BPI 9-track tape drive
            The default is the unittype corresponding to
            the mode parameter of the FILEDEF issued for
            this fortran DSRN.  If no mode parameter
            exists, '9TRK' is assumed.
```

Example:

```
CALL MOUNT (1000, 12, 'RO','800')
```

This requests the operator to mount tape 1000 with the ring out on the unit corresponding to FORTRAN data set 12. The tape drive to be used is requested to be a 9-track drive with 800BPI capability.

2. Internal Description

MOUNT is a routine which is fortran callable: MOUNT takes the parameters passed to it and forms a parameter list to be used in calling the LARS routine TAPMOUNT.

MOUNT does very little error checking. It does, however, insure that the dsrn argument is legal. By legal, it is meant that the dsrn is a valid fortran unit number and that the dsrn is a tape unit. If this error occurs, or TAPMOUNT detects an error; then MOUNT terminates the job by calling the fortran exit routine EXIT.

To insure that the fortran I/O routines are aware that a new tape is to be present, MOUNT calls the entry point TOPRU in the LARS subprogram TAPOP to unload the old tapes. At this time an interface is required between MOUNT and the LARS routine TAPMOUNT to have TAPMOUNT type the informational messages with (MOUNT) at the end. This was done to maintain compatability with the pre-existing LARSYS error and information message document, because MOUNT has been rewritten to call TAPMOUNT to do the work. The interface consists of the added parameter (MOUNT), which follows the unittype parameter in MOUNT's TAPMOUNT parameter list.

3. Input Description

Not Applicable

4. Output Description

As described in the Internal Description section, MOUNT delegates most message writing and error detection to TAPMOUNT. MOUNT does check for a valid dsrn and produces the following message, if an error is found, on both the user's terminal and the printer.

E0331 BAD DATA SET REFERENCE NUMBER xxx (MOUNT)

5. Supplemental Information

FORTRAN Library Subprograms Required:

1) EXIT - Service subroutine to terminate execution.

LARS Support Routines Used:

1) TAPMOUNT - Request and wait for tape mount.

LARS Subprograms Used:

1) TAPOP - A multifunctional program to read, write, and position tape files. Entry point TOPRU is used to unload a tape..

CMS Support Routines Used:

1) TYPLIN - Types message on user's terminal

2) PRINTR - Prints message on printer

CMS Tables Used:

1) FCB - File Control Blocks

CMS Macros Used:

1) CMSYSREF - CMS general information

2) CMSCB - DSECT for FCB organization

6. Flowchart

Not Applicable

LARS Program Abstract 011

MODULE IDENTIFICATION

Module Name: TAPOP Function Name: SYSTEM SUPPORT

Purpose: A multifunction program to read, write, and position tape files.

System/Language: CMS/ASSEMBLER

Author: Paul E. Anuta Date: 11/9/69

Latest Revisor: Howard L. Grams Date: 01/28/74

MODULE ABSTRACT

TAPOP is a subroutine callable by FORTRAN for reading and writing records on tape, reading tape records specially formatted for multi-spectral data, positioning tape files, and acquiring information about the tape drive being used. The entry points are:

- TOPRD - to read a record of any length
- TOPRV - to read a variable number of channel records from one physical block as defined for the multispectral data tape.
- TOPWR - to write a record of any length
- TOPRW - to rewind the tape
- TOPRU - to rewind and unload the tape
- TOPEF - to write a tape mark (end of file)
- TOPFF - to forward file 1 file beyond current file
- TOPRF - to refile to beginning of current file
- TOPBF - to back file 1 file before current file
- TOPBS - to backspace a specified number of records
- TOPFS - to forward space a specified number of records
- TCLOSE - to logically disconnect tape unit from program
- RINGIN - to determine if tape file-protect ring is in or out of the tape reel
- GTUNIT - to determine the device address of the tape drive being used.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Function

The program is to be used for reading and writing data blocks of arbitrary length on tape. Rewind, rewind and unload, backspace, forward space, backspace and forward space file, and write end of file mark functions are also implemented.

Parameters

General definitions pertinent to all TAPOP calls - (all scalar arguments are integer full words).

UNIT = FORTRAN data set reference number on which the operation is to be performed. Standard unit assignments are available from the LARS Data Handling Staff or in the CMS User's Guide.

COUNT = Exact byte count of record to be written or read for the TOPRD, TOPWR read/write calls. For backspace and forward space it is the number of physical tape records to be passed over. The program tests to see if this count was actually read or written. For read, two modes of wrong count response are provided (see TOPRD). If the block size is unknown for read, COUNT can be set larger than the expected record size. The true count will be returned. The particulars of each function are discussed in detail below.

ERROR = Return codes as defined in Table 1.

IDATA = Data buffer where record is read into or from for read/write operations. It must be dimensioned at least COUNT bytes for TOPRD and TOPWR. For read variable (TOPRV) it must be dimensioned (NSC)x(NC) (see discussion of TOPRV below).

This program utilizes input/output functions supplied by CMS and interaction between the two systems exists. Specifically the mode set function for 7-track tape units and dual-density 9-track units controlled by the CMS FILEDEF command. The FILEDEF command may be used in order to override the standard unit assignments. Also, TAPOP tries to maintain compatibility with FORTRAN I/O processing. However, there are some restrictions for mixing FORTRAN I/O functions and TAPOP functions. (See the section on Supplemental Information).

2. Internal Description - TAPOP Entry Point

READ

CALL TOPRD (UNIT,COUNT,ERROR,IDATA,NRTRY,WLF)

Input Arguments:

UNIT - Fortran Data Set Reference Number

COUNT - Number of bytes to be read

IDATA - Buffer to hold data to be read

NRTRY - Number of retrys to be executed if an error occurs

WLF - Wrong length flag

Output Arguments:

ERROR - Error Code

COUNT - Count Transmitted (if WLF = 0)

Data in IDATA Array

One record of COUNT bytes in length is read from the FORTRAN tape unit defined by UNIT. Data is placed in core starting in the first byte of the IDATA array. Table 1 describes the error codes returned in ERROR. NRTRY defines the number of retrys which will be attempted in case a read parity error is detected or a wrong length condition is encountered. The WLF (wrong length flag) controls handling of wrong length conditions, i.e., the block size encountered differs from that specified by count. Note that the NRTRY capability may or may not be applicable depending on the use of WLF and whether a parity error or a wrong length condition is encountered. It is expected that in normal use the wrong length flag will be off (WLF=0) and two retries for parity error will be desired; thus, these default options are supplied. If WLF is omitted from the argument list, WLF=0 is assumed. If NRTRY is omitted, NRTRY=2 is assumed and WLF must be omitted; that is, the program takes the first argument after IDATA to be NRTRY and the second to be WLF. A table of block, count and error code returns for TOPRD is presented in Table 2 on page 11.

If WLF=0, no retrys are carried out if the block size encountered differed from that specified by COUNT. If the block is transmitted to core and the block size is returned in COUNT; error code 2 is returned or code 5 when a parity error is also encountered. If the block was longer than COUNT, only COUNT bytes are transmitted to core and code 2 is returned or code 5 for parity error.

If WLF≠0, NRTRY retries are executed if the block size is not equal to that specified by COUNT. If a parity error also occurred reading the record, the retry process for the parity error is carried out and the retrys for wrong count are not carried out. If the block was shorter than COUNT, the operation is the same as for WLF=0 except for the retrys. If, however, the block was longer than COUNT, the true long count is returned in COUNT but only COUNT bytes are transmitted to core and the rest of the record is skipped. This feature is implemented by backspacing over the record and rereading it to count the number of bytes in the long block. This feature supplies information on the long block size to the programmer without requiring a buffer to hold the data beyond COUNT bytes.

If the length of a block is unknown, COUNT can be set to the full size of the buffer IDATA before TOPRD is called and the actual block size is returned in COUNT. Also, as long as COUNT is not larger than the buffer, the core space above the buffer can never be wiped out by a read.

WRITE

CALL TOPWR (UNIT,COUNT,ERROR,IDATA)

Input Arguments:

UNIT - Fortran Data Set Reference Number

COUNT - Number of bytes to be written

IDATA - Data block to be written

Output Arguments:

Data record on tape

ERROR - Return code

COUNT - Bytes actually transmitted

One record of COUNT bytes in length is written on the FORTRAN tape unit defined by UNIT. Data is fetched from core starting at the first byte of IDATA. The ERROR returns are discussed in Table 1.

READ VARIABLE

CALL TOPRV (UNIT,NSC,ERROR,IDATA,NRTRY,NC,CSEL,LNID)

Input Arguments:

UNIT - Fortran Data Set Reference Number

NSC - Number of samples per channel

NRTRY - Number of parity error retrys

NC - Number of channels in tape record

CSEL - Halfword channel flag array

Output Arguments:

Requested data in IDATA

ERROR - Return code

LNID - 4 byte line ID record (line number and roll angle)

This entry expects the tape to be formatted for multispectral data (see Data Organization section V of the LARSYS System Manual). Data from 1 to 30 channels in a tape record can be read. This function expects a tape record format of 2 bytes for line number and 2 bytes for roll angle, then NC sets of NSC bytes each. Each channel must have NSC bytes. NC (Number of Channels) must be less than or equal to 30. CSEL is a halfword integer flag array that indicates which of up to 30 channels are to be read. That is, if CSEL is non zero, the ith set of NSC bytes in the tape record is read into core. If CSEL (i)=0, that channel is skipped. If all CSEL (i)=0, only the line number and roll angle are read into core.

Data is read into consecutive core locations starting with the first byte of IDATA. Data from consecutive or non-consecutive tape channels are read into consecutive core locations. This allows the programmer to dimension his input array only as large as necessary for the number of channels he wishes to read. For example, if an 18-channel tape is to be read and channels 1,6,12, and 18 are desired, IDATA would have to be dimensioned at least NSC*4 bytes, NC=18, CSEL(1)=1, CSEL(6)=1, CSEL(12)=1, CSEL(18)=1, all other CSEL(i)=0, CSEL would be dimension Integer*2 18. The first four bytes of the record are read and placed in LNID. Channel data starts in IDATA(1). NRTRY = the number of retrys which will be attempted in case a read error is detected (see Table 1).

BACKSPACE

CALL TOPBS (UNIT,COUNT,ERROR)

Input Arguments:

UNIT - Fortran data set reference number

COUNT - Number of physical records to be backspaced

Output Arguments:

ERROR - Return code

COUNT - Number of physical records actually backspaced

The number of physical records indicated by COUNT are backspaced on FORTRAN unit defined by UNIT. An EOF mark counts as one record. Backspace is halted if an EOF is passed and the number of records backspaced at this point are output in COUNT. Only error codes 0,1, or 2 can be returned (see Table 1).

REWIND

CALL TOPRW (UNIT)

The tape on the FORTRAN unit indicated is rewound to the load point. The file is not closed.

REWIND AND UNLOAD

CALL TOPRU (UNIT)

The tape on the FORTRAN unit indicated is rewound and unloaded. The file is closed. No wait is issued.

FORWARD SPACE

CALL TOPFS (UNIT,COUNT,ERROR)

Input Arguments:

UNIT - FORTRAN Data Set Reference Number

COUNT - Number of physical records to forward space

Output Arguments:

ERROR - Return code

COUNT - Number of physical records actually forward spaced

The tape on the FORTRAN unit is forward spaced COUNT physical records. If an EOF mark is passed, forward space is halted. Error return codes 0, 1 can occur.

END FILE

CALL TOPEF (UNIT,ERROR)

A tape mark (end-of-file) is written on the tape on the indicated unit. Error codes 0 or 1 can be returned. The ERROR argument is optional.

FORWARD FILE

CALL TOPFF (UNIT)

The tape on the FORTRAN unit given in the argument is advanced to the next file. The tape ends up positioned past the next end-of-file mark on the tape. No indication is given of hitting an end-of-file mark or the end-of-tape mark.

REFILE

CALL TOPRF (UNIT)

The tape on the indicated FORTRAN unit is backed up to the beginning of the file in which the tape is presently positioned. If the present file is the first file on the tape, the tape is positioned at the load point. If it is the second or a succeeding file, the tape ends up just past the end-of-file mark at the beginning of the file.

BACKFILE

CALL TOPBF (UNIT)

The tape on the indicated FORTRAN unit is backed up to the beginning of the file preceding the one in which the tape is presently positioned. If the present file is the first, then the result is the same as TOPRF (i.e., at the load point). Calling TOPBF from file 2 puts the tape at the load point. From file 3 or more the final position is just past the tape mark at the beginning of the previous file.

TCLOSE

CALL TCLOSE (UNIT)

This routine should be called after the caller is finished using the specified unit. It must be called once for each unit used. This function will disconnect the specified unit from the user's program and will prevent FORTRAN from closing the unit (this prevents FORTRAN from repositioning the tape or writing a Tape Mark for an output file). TCLOSE performs no tape positioning.

RINGIN

CALL RINGIN (UNIT,FLAG)

Input Arguments:

UNIT - FORTRAN Data Set Reference Number

Output Arguments:

FLAG - = 0 if the file-protect ring is in
= 1 if the file-protect ring is out

This entry allows the FORTRAN programmer to check whether or not a file-protect ring is inserted in the tape mounted on the specified unit.

GTUNIT

CALL GTUNIT (UNIT,ADDR)

Input Arguments:

UNIT - FORTRAN Data Set Reference Number

Output Arguments:

ADDR - The tape unit address (i.e. 180) that is currently assigned to UNIT is returned. If the unit is not assigned, the results are unpredictable. ADDR must be printed or used in Z format (it's a hexadecimal number).

This entry determines the tape unit address (the virtual device address) currently associated with the specified FORTRAN Data Set Reference Number.

Table 1

Tape Operations Program (TAPOP) error return codes for read, write, backspace, forward space, and end-of-file calls.

Code	Meaning
0	Operation was successful.
1	Read: End-of-File mark was read. Write: End-of-Tape mark encountered. The write operation was carried out past end-of-tape mark. Backspace and Forward Space: End-of-file mark encountered at some point during backup. EOF mark is counted as a block. Backup or forward space was halted when EOF was passed. End File: End-of-Tape mark encountered. EOF record past the EOT marker.
2	Read (TOPRD): The byte count of the block read is not the same as that specified by calling program. The indicated number of retrys was carried out only if WLF $\neq$ 0. The count for the block is always returned in the COUNT argument. The number of bytes transmitted is never more than the number requested. That is, if the block was shorter than requested, all the data is transmitted and the short count is returned. If the block was longer, only the number of bytes requested is transmitted and the rest of the record is skipped. The total length of the long record is returned in COUNT if WLF $\neq$ 0. If WLF = 0, the return COUNT is never more than the initial value supplied. The return counts and codes for TOPRD are indicated in Table 2. Read Variable (TOPRV): For read variable, the count is the individual channel count and there can be up to 30 channels. Wrong count in this case indicates that the last channel did not have the correct count, but the error could be due to a wrong count in any of the previous channels. Results in this case are unpredictable. NSC in the argument list is not changed. Write: Code 2 should never be observed for write.

Backspace: The load point was reached before backspace of the number of blocks specified was completed. The number of blocks actually backspaced is returned in COUNT.

Forward Space: Code 2 is not returned for forward space. No program indication is available if tape is forward spaced to the end-of-the reel. Only the tape indicator (T1) light is illuminated on the unit in this case.

3 A vertical, longitudinal, cyclic, or skew read or write parity check error was sensed. For read the specified number of retries were carried out. Tape is positioned past the block. The data from the bad record is in core for read. For write the bad parity data is on the tape. Retry for write was as follows: the tape was backed up over the bad record and an attempt was made to rewrite the record. This was done 10 times, after which the tape was backed up over the bad record, a 21-inch gap was erased and a new set of ten retires to write the record was made.

4 Any one or more of a set of "special" error conditions cause a code 4 return. For 9 track units, the error is in the I/O interface and is due to a channel data check, bus out check, equipment check, data check for control operation, or chaining check. These are all hardware parity errors on read or write and occur very rarely.

The above errors also apply to the 7 track unit, however, frequent Code 4 errors may occur for the 7 track unit when reading tapes written by the LARS A/D System. This is due to data convertor check errors. Details of the data convertor error are explained on page 20 of the 2400 series tape unit manual A22-6866. It occurs when the number of characters in a record is not divisible by four and the recovery procedure is not applicable (see above reference). This can be due to bad A/D operation, bad tape, or faulty reading. Wrong length will usually accompany this error.

5 Both wrong length and parity or data convertor check errors occurred during the operation.

6 The I/O request was invalid for the unit specified. This can be due to using a FORTRAN unit number for a tape unit not defined by FILEDEF command or permanently assigned. Also, the device may not be attached or the file-protect ring may be out for a write operation.

Initial Condition		Result After Call to TOPRD			
BLOCK=Bytes in Record COUNT=Requested Bytes	WLF	Number of Bytes Transmitted to core	Value of COUNT Returned in Argument	Error Return Code-No Parity Error	Error Return Code-Parity Err.
BLOCK = COUNT	0	BLOCK	BLOCK	0	3 or 4
	1	BLOCK	BLOCK	0	3 or 4
BLOCK.LT.COUNT	0	BLOCK	BLOCK	2	5
	1	BLOCK	BLOCK	2	5
BLOCK.GT.COUNT	0	COUNT	COUNT	2	5
	1	COUNT	BLOCK	2	5

Table 2

Parameter Returns for TOPRD Count and Parity Errors

3. Input Description

TAPOP may be used to read tapes of almost any format. It is especially designed to read multispectral data tapes formatted as defined in the Data Organization Section of the LARSYS System Manual.

4. Output Description

TAPOP may be used to write tapes with any number of files and with any size records (no automatic blocking/deblocking of logical records from a physical record is performed). TAPOP usually returns an error code which the calling program may use to produce a message. As a result, TAPOP does not normally produce any messages when used with LARSYS. TAPOP does send a message to the user for TOPRV and RINGIN calls that determine the tape is not ready or not attached to the virtual machine. This message is:

"IØ5Ø TAPE UNIT 181 NOT ATTACHED OR NOT READY (TAPOP)"

If the user receives this message, he should notify the CP operator. The program will wait until the tape is ready and then proceed automatically. (Note: When TAPOP is used within LARSYS, all tapes will be attached and ready before TAPOP is called). TAPOP uses the CMS function TAPEIO for most of its operations. As a result, for TOPRW only, the following messages may appear if TAPOP is used without LARSYS:

"TAPn NOT READY YET" - if the tape is attached but not ready (the program will wait)

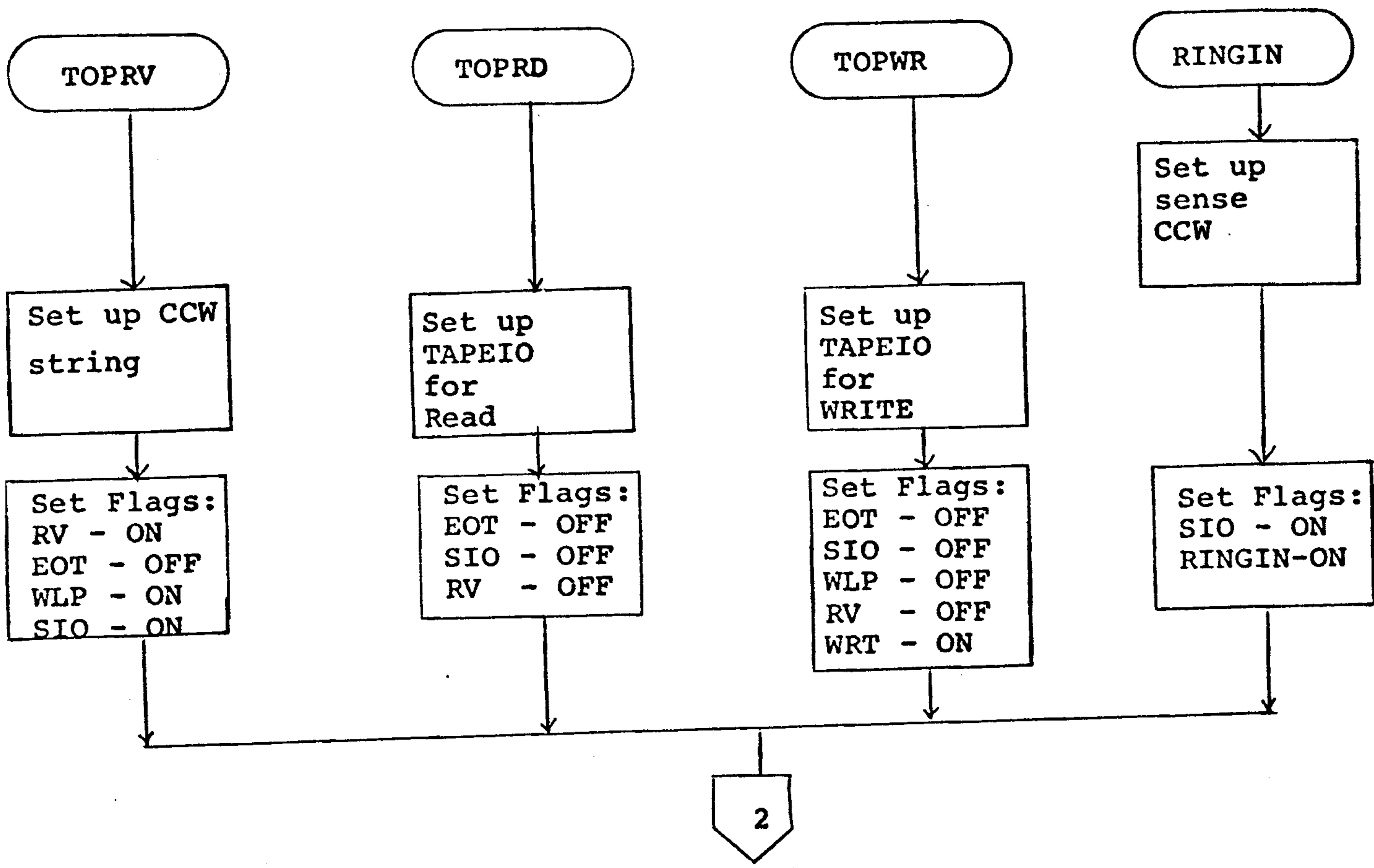
"(OK - READY NOW)" - when the tape becomes ready

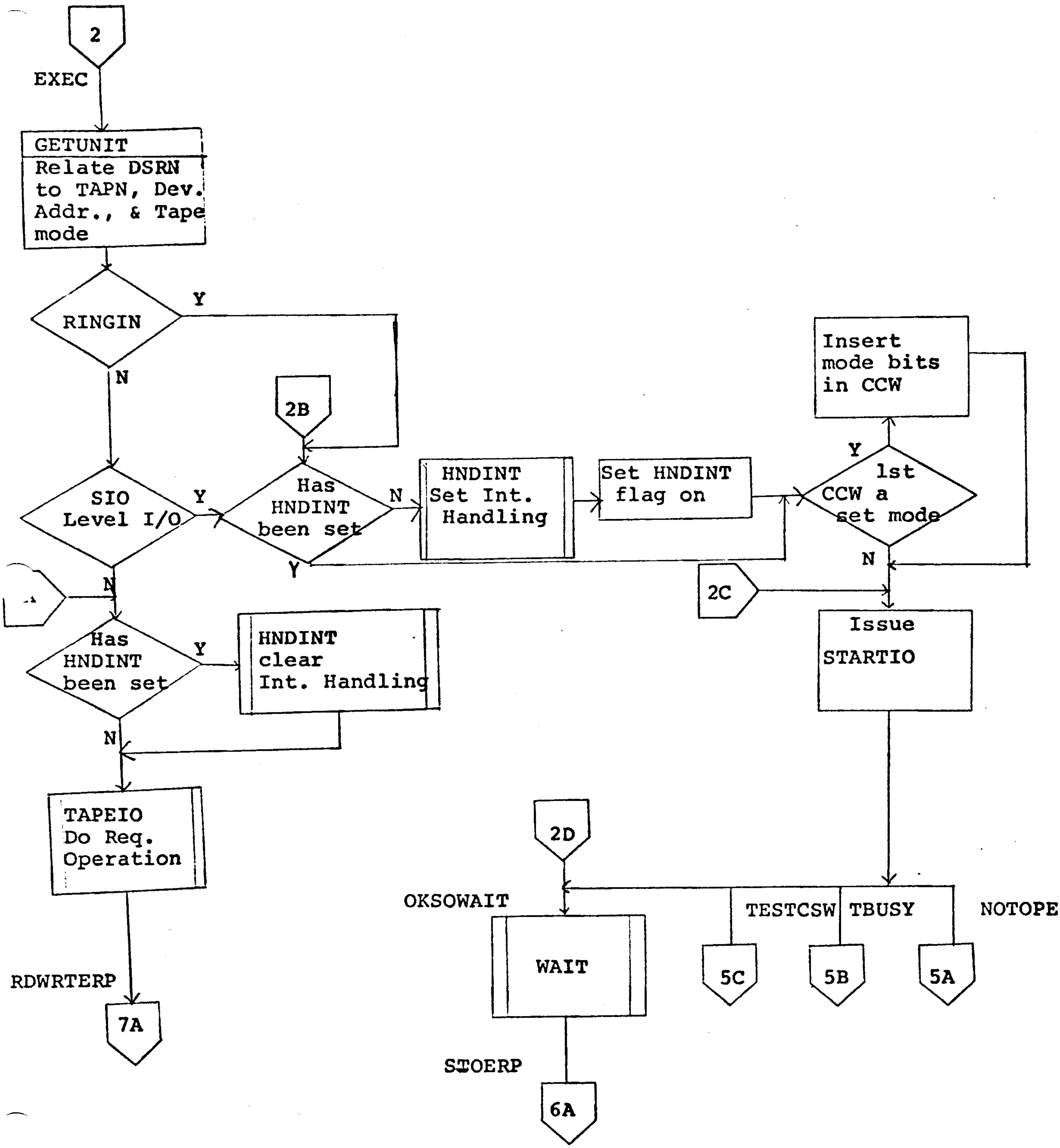
"TAPn NOT ATTACHED" - if the tape has not been attached (TAPOP will return ERROR=6)

5. Supplemental Information

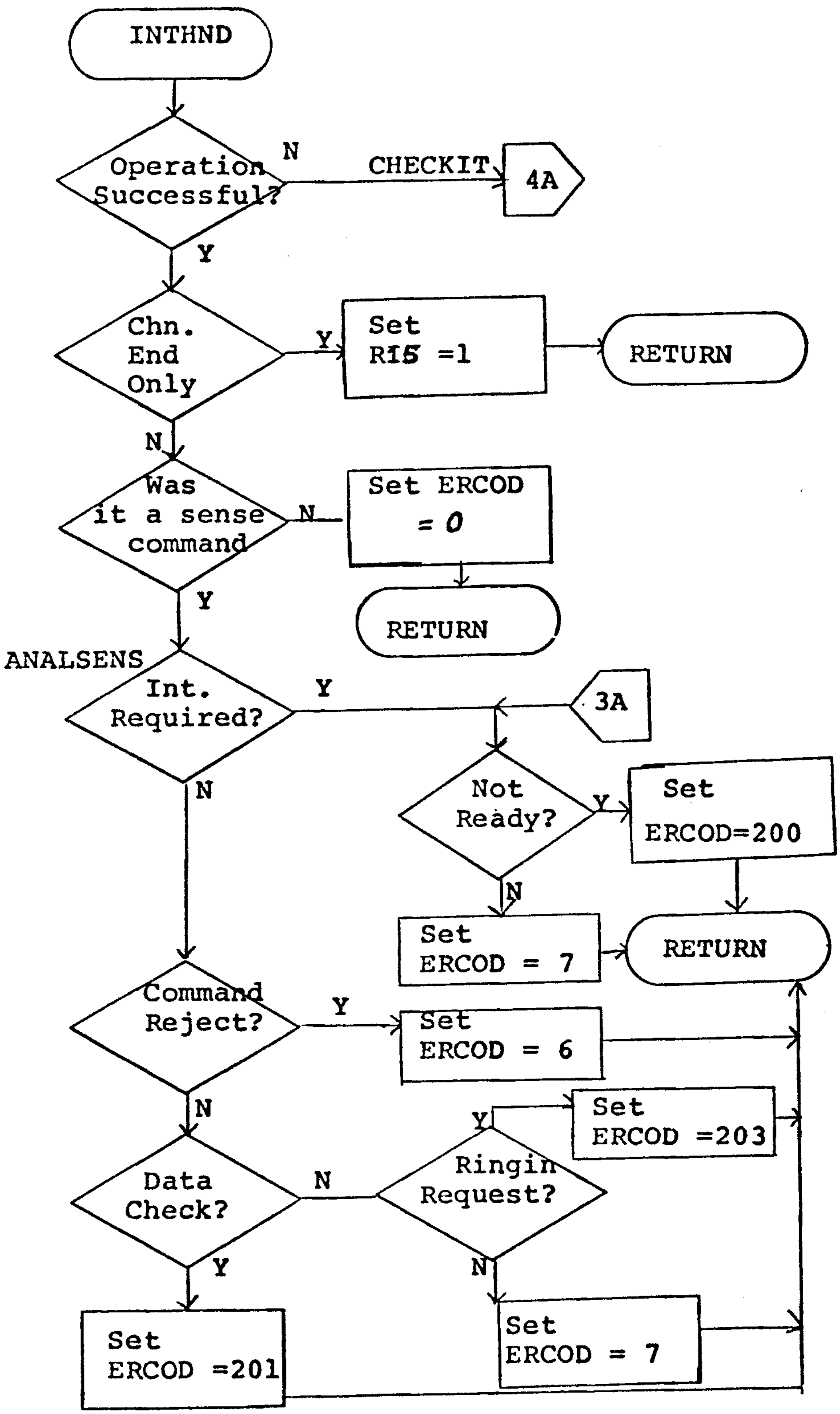
All parameters used by TAPOP are supplied via the calling parameter list. No variables in any COMMON are used. TAPOP does not call any other LARSYS subroutines. However, it does utilize several CMS and FORTRAN functions such as TAPEIO, WAIT, TYPLIN, HNDINT, SYSREF, NUCON and IHCUATBL.

6. Flowcharts



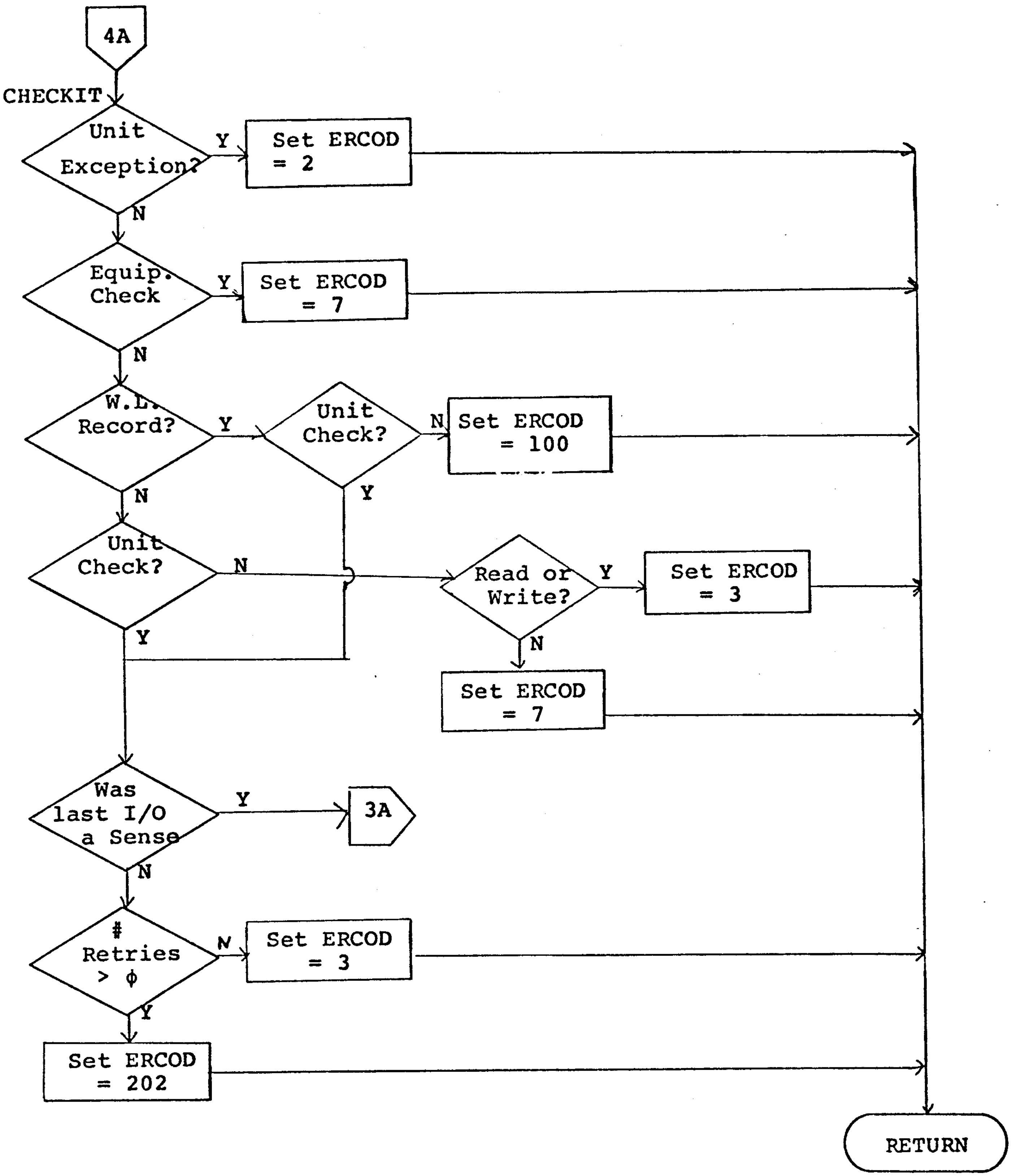

2
EXEC
GETUNIT
Relate DSRN to TAPN, Dev. Addr., & Tape mode
RINGIN
Y
N
2B
SIO Level I/O
Y
Has HNDINT been set
N
Y
HNDINT Set Int. Handling
Set HNDINT flag on
Insert mode bits in CCW
Y
1st CCW a set mode
N
2C
Issue STARTIO
N
Has HNDINT been set
Y
HNDINT clear Int. Handling
N
TAPEIO Do Req. Operation
RDWRTERP
7A
2D
OKSOWAIT
WAIT
STOERP
6A
TESTCSW
TBUSY
NOTOPE
5C
5B
5A

INTHND - HANDLES TAPE INTERRUPTS ASYNCHRONOUSLY FOR
SIO LEVEL I/O

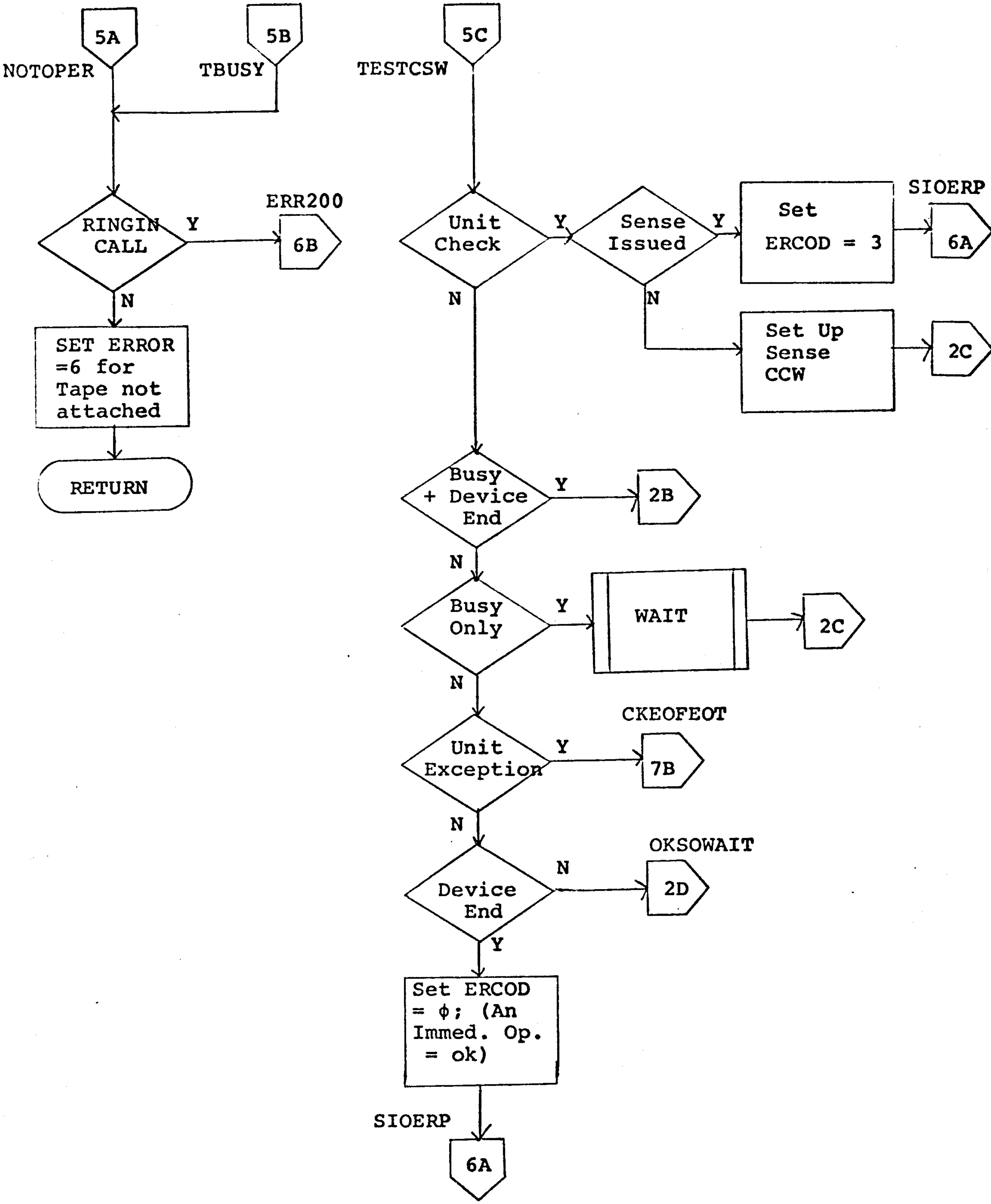


Upon Exit:

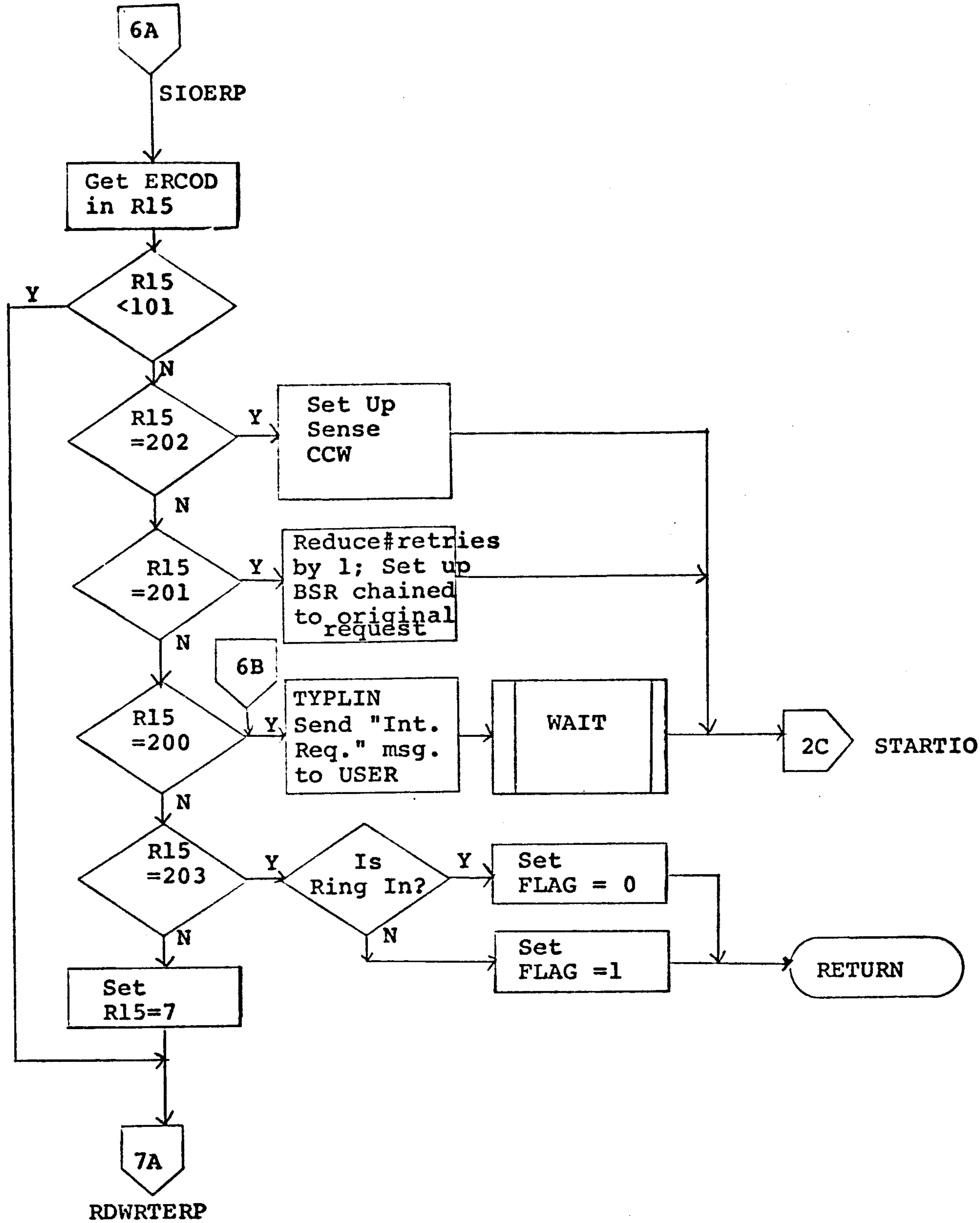
ERCOD	= 0	Successful
	= 2	EOF or EOT
	= 3	Permanent I/O Error
	= 6	Tape is file protected
	= 7	Serious Tape Error
	= 100	Incorrect Length Record
	= 200	Intervention Required
	= 201	Permanent I/O Error, but should retry it
	= 202	Sense Operation is Recommended
	= 203	Ringin Test (sense has been issued)


4A
CHECKIT
Unit Exception?
Y
Set ERCOD = 2
N
Equip. Check
Y
Set ERCOD = 7
N
W.L. Record?
Y
Unit Check?
N
Set ERCOD = 100
Y
N
Unit Check?
N
Read or Write?
Y
Set ERCOD = 3
N
Set ERCOD = 7
Y
Was last I/O a Sense
Y
3A
N
Retries > ϕ
N
Set ERCOD = 3
Y
Set ERCOD = 202
RETURN

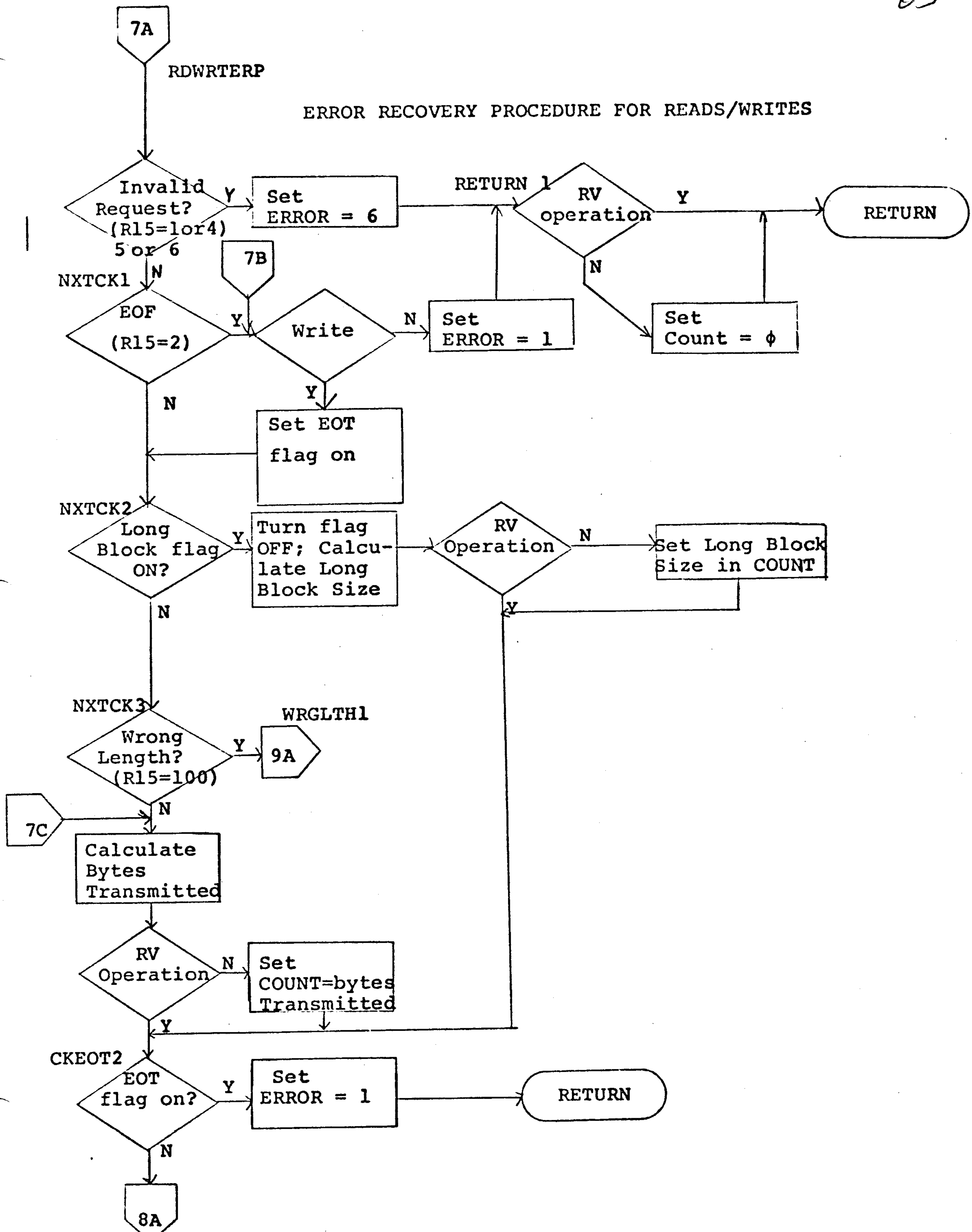
STARTIO ERROR ANALYSIS

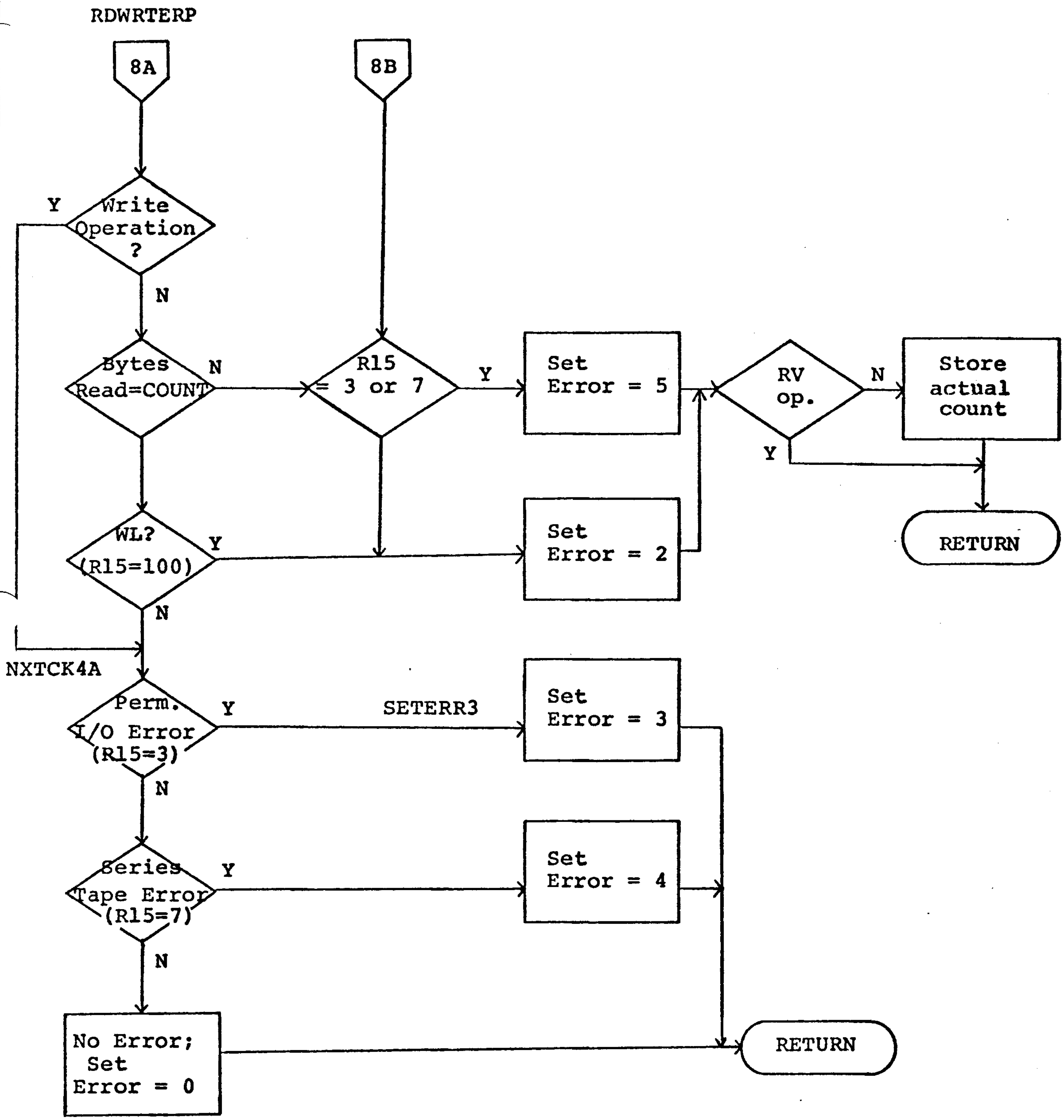


ANALYZE RESULTS OF SUCCESSFUL STARTIO

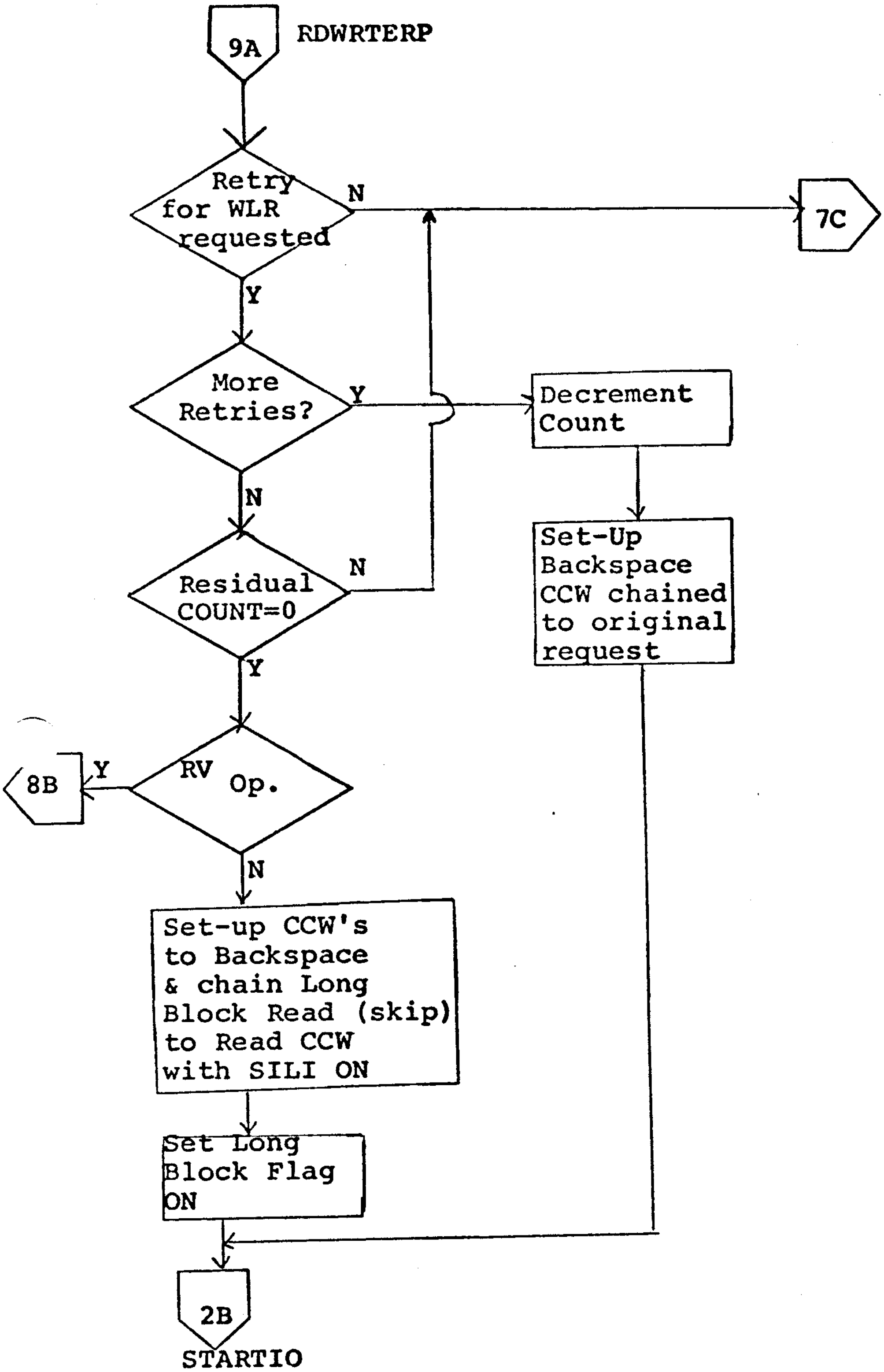


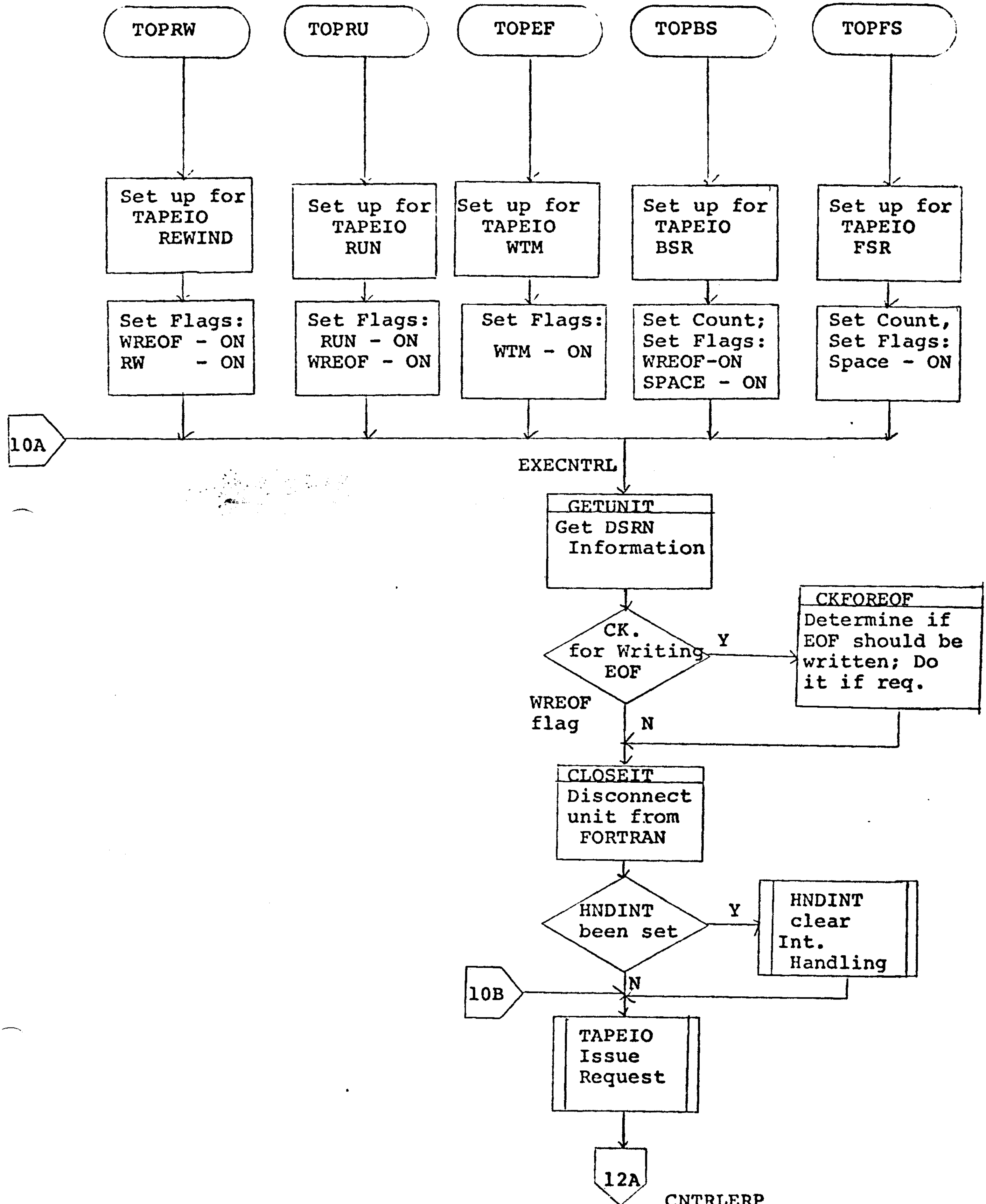
ERROR RECOVERY PROCEDURE FOR READS/WRITES

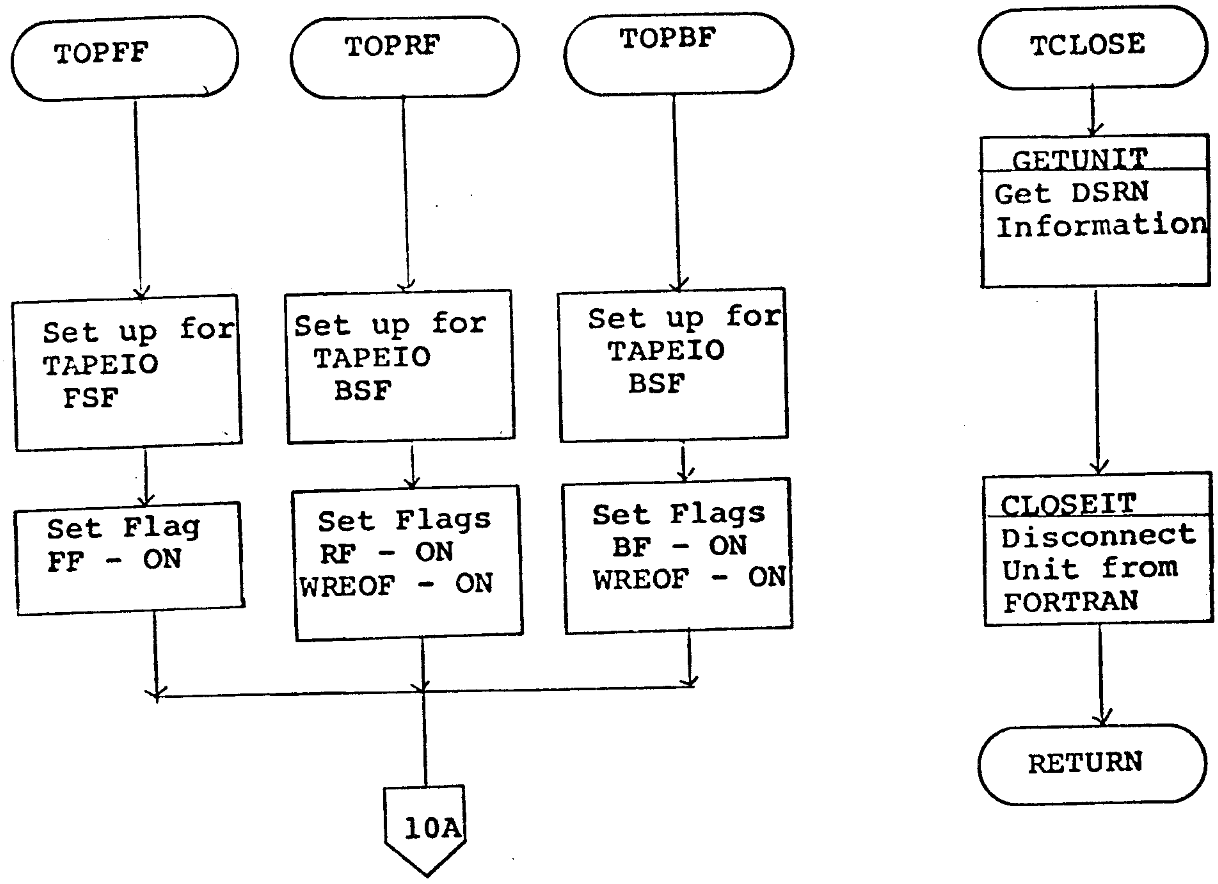


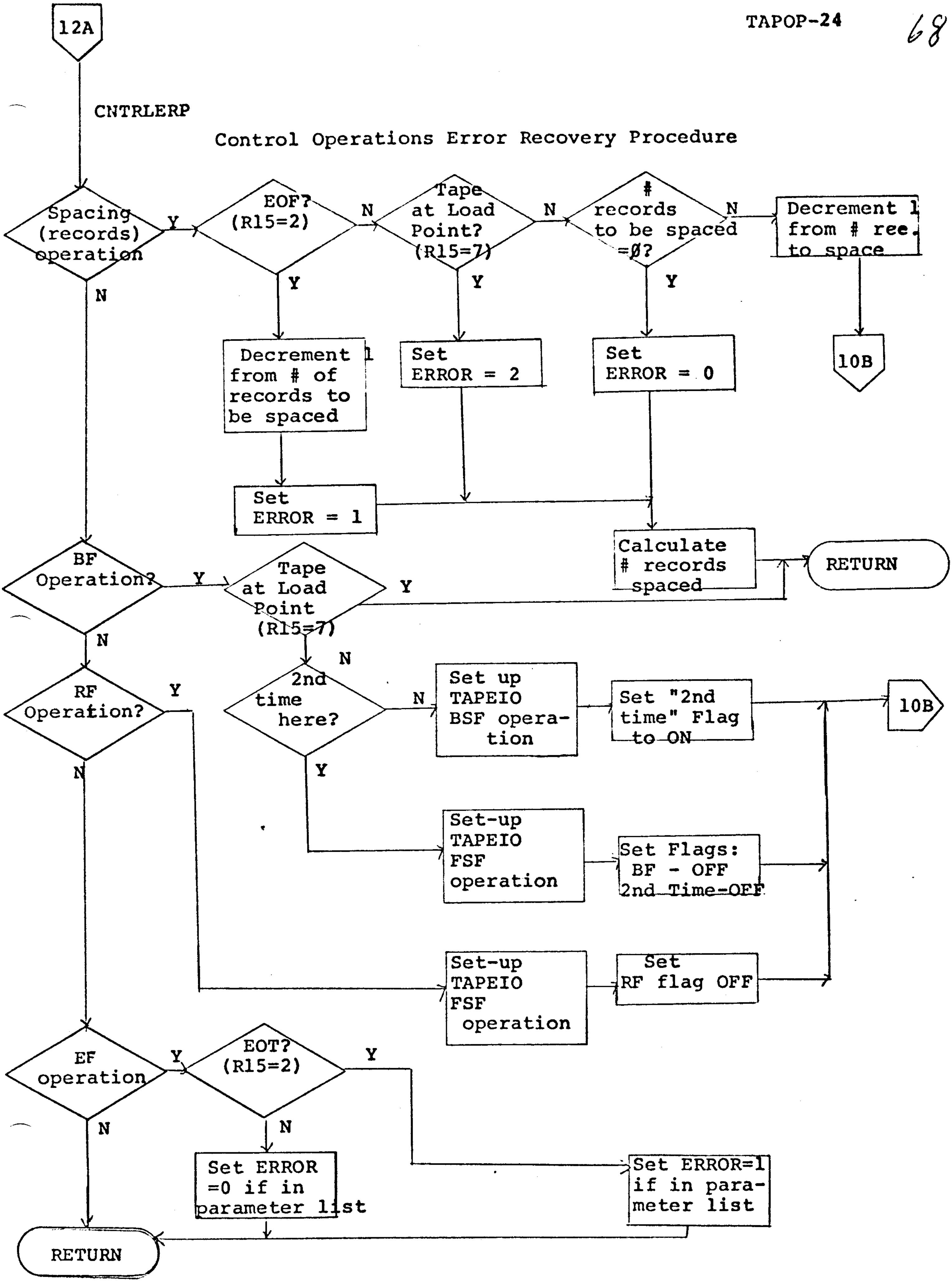


Revised September 1974


9A
RDWRTERP
Retry for WLR requested
N
7C
Y
More Retries?
Y
Decrement Count
Set-Up Backspace CCW chained to original request
N
Residual COUNT=0
N
Y
RV Op.
Y
8B
N
Set-up CCW's to Backspace & chain Long Block Read (skip) to Read CCW with SILI ON
Set Long Block Flag ON
2B
STARTIO


TOPRW
TOPRU
TOPEF
TOPBS
TOPFS
Set up for TAPEIO REWIND
Set up for TAPEIO RUN
Set up for TAPEIO WTM
Set up for TAPEIO BSR
Set up for TAPEIO FSR
Set Flags: WREOF - ON RW - ON
Set Flags: RUN - ON WREOF - ON
Set Flags: WTM - ON
Set Count; Set Flags: WREOF-ON SPACE - ON
Set Count, Set Flags: Space - ON
10A
EXECNTRL
GETUNIT
Get DSRN Information
CK. for Writing EOF
WREOF flag
Y
N
CKFOREOF
Determine if EOF should be written; Do it if req.
CLOSEIT
Disconnect unit from FORTRAN
HNDINT been set
Y
N
HNDINT clear Int. Handling
10B
TAPEIO Issue Request
12A
CNTRLERR


TOPFF
TOPRF
TOPBF
TCLOSE
GETUNIT
Get DSRN Information
Set up for TAPEIO FSF
Set up for TAPEIO BSF
Set up for TAPEIO BSF
Set Flag FF - ON
Set Flags RF - ON WREOF - ON
Set Flags BF - ON WREOF - ON
CLOSEIT
Disconnect Unit from FORTRAN
RETURN
10A


12A
CNTRLERP
Control Operations Error Recovery Procedure
Spacing (records) operation
Y
N
EOF? (R15=2)
N
Y
Tape at Load Point? (R15=7)
N
Y
records to be spaced =Ø?
N
Y
Decrement 1 from # rec. to space
10B
Decrement 1 from # of records to be spaced
Set ERROR = 2
Set ERROR = 0
Set ERROR = 1
Calculate # records spaced
RETURN
BF Operation?
Y
N
Tape at Load Point (R15=7)
Y
N
RF Operation?
Y
N
2nd time here?
N
Y
Set up TAPEIO BSF operation
Set "2nd time" Flag to ON
10B
Set-up TAPEIO FSF operation
Set Flags: BF - OFF 2nd Time-OFF
Set-up TAPEIO FSF operation
Set RF flag OFF
EF operation
Y
N
EOT? (R15=2)
Y
N
Set ERROR =0 if in parameter list
Set ERROR=1 if in parameter list
RETURN

6. Supplemental Information

A. Intermixing of FORTRAN and TAPOP I/O Functions

Use of FORTRAN I/O statements with TAPOP functions can successfully be accomplished; however, certain precautions must be observed. The only time TAPOP should be used with FORTRAN I/O is when the file positioning capabilities of TAPOP are required. Otherwise, standard CMS/FORTRAN capabilities can be used to read or write any size record (also blocking/deblocking) and, also, handle limited multiple file requirements. Refer to the proper CMS documentation for details. General guidelines for successful intermixing follow:

* Use TOPRW as the first operation on any tape unit.

* Use only FORTRAN Reads/Writes with TOPEF, TOPRW, TOPFF, TOPRF, TOPBF, TOPRU, RINGIN and TCLOSE as required. Do not use TOPBS or TOPFS with FORTRAN created files.

* Multiple FORTRAN created files on the same tape must have the same record characteristics as specified by the FILEDEF command for that tape unit.

* In order to process two or more files sequentially (i.e. read all of file 1, then read file 2) a call to TCLOSE or TOPRF must be executed after reading one file and before reading the next file in order to prevent FORTRAN from incrementing the file ID (i.e. FT11FØØ1 to FT11FØØ2).

* Use TOPRU or TCLOSE when you are finished with the tape so that FORTRAN will not try to close the tape unit when it processes the STOP statement.

B. Automatic Writing of a Tape Mark (End-of-File)

TOPRW, TOPRU, TOPRF, and TOPBF examine the last operation performed on the specified unit and if it was a write or a write end-of-file, then a tape mark (end-of-file) will be written on the tape prior to the execution of the called function. If the last operation was not a write or write end-of-file, then the tape mark will not be written. For either case, the tape will be positioned the same.

C. System Dependencies

TAPOP is very system dependent in that it interfaces quite closely with CMS and with the FORTRAN library modules IHCUATBL, IHCECOMH (IBCOM#) and IHCEFIOS (FIDCS).

CMS Dependencies:

* All tape control operations and normal reads/writes are handled by using the TAPEIO function.

* The read variable function, RINGIN, and retry operations for reads use STARTIO level I/O utilizing the HNDINT function of CMS.

* The TYPLIN function is used to type a message.

* The CMS module TAPEIO was modified to support a variable number of retries and to allow wrong length record detection.

* TAPEIO will normally produce error messages at the typewriters. This is not desirable for use within TAPOP since TAPOP returns error codes and allows the caller to provide his own error messages. To inhibit messages from TAPEIO, the KT (Kill Typing) flag in the CMS nucleus is turned on prior to invoking TAPEIO and then turned off after TAPEIO completes. This is identical to the user issuing the KT command of CMS. (The coding is dependent on the KT flag being located X'SE2' bytes beyond the beginning of NUCON.) The one exception to the above is for TOPRW which does not use the KT flag since TOPRW is often the first call to TAPOP and TAPEIO REWIND can produce only messages related to a tape not being attached or ready. This allows the user to be informed if these conditions are present.

* The GETUNIT section of code associates the given DSRN (Data set reference number) to a TAPN for use with TAPEIO and to a device address for use with SIO. This section was written such that entry GTUNIT (UNIT, ADDR) is used as an externally callable routine and GETUNIT is used internally within TAPOP.

  LARSYS, as well as FORTRAN, issues CMS FILEDEF's for all DSRN's. The FILEDEF builds a FCB (file control block) with entries of FTXXF001 for the DD name and TAPn for the data set name. Therefore, GETUNIT can simply search the FCB's for the desired DSRN within the DD name field and, if there is a match, pick up the corresponding TAPn from the data set name field. The TAPn is returned to the caller. Also, the mode of the tape unit (in FCBMODE) is returned to internal caller.

At this point, the actual device address of the tape (required for SIO) is determined by searching the device table (DEVTAB) in the CMS nucleus. The search can be made on TAPn and, when a match is found, pick up the corresponding device address. The device address is returned to the caller.

Both DEVTAB and FCBTAB have their addresses in SYSREF. The CMS macro NUCON is used to reference these pointers and the macro CMSCB is used to reference fields within the FCB's.

* The error and I/O completion handling routines are dependent on TAPEIO return codes and I/O control blocks. There are four distinct sections of code involved with error handling. There are two sections to provide the interrupt handling routine for the HNDINT function. The third section starts at RDWRTERP and performs unique TAPOP error recovery for Read/Write operations (i.e., handling the "Wrong Length Record" condition) and, also, converts TAPEIO return codes into TAPOP error codes. The fourth section of code performs error analysis for control operations; plus, for record spacing controls looping for the number of records specified. This section is labeled CNTRLERP. For all error codes, the TAPEIO form of error code is converted to TAPOP ERROR codes.

With CMS, the routine specified by HNDINT SET is executed asynchronously of other TAPOP code and must perform all basic interrupt handling and error recovery. This routine (called INTHND) is branched to and must return to CMS nucleus module IOINT. As a result, a section of code labeled SIOERP must be executed immediately after the WAIT (after a SIO) is satisfied. Since the interrupt handling code, INTHND, is executed prior to the WAIT being satisfied, these two routines interact closely with each other. Several exits from this code continue into the RDWRTERP routine. The RDWRTERP will be entered immediately after completion of TAPEIO. Likewise, the CNTRLERP code is entered immediately after a TAPEIO for a control operation.

CMS (and OS) FORTRAN Dependencies:

* FORTRAN I/O status bits are not maintained by TAPOP. TAPOP will always close (go to CLOSEIT) any unit referenced in a TAPOP call. This allows FORTRAN I/O operations to cause the unit to be opened properly (input or output) as required.

For CMS, the data set assignment table is IHCUATBL. It contains entries for each DSRN among which is a pointer to a Unit Block. The Unit Block has the I/O status bits, DCB, buffer addresses, etc. The Unit Block for DSRN only exists if FORTRAN has OPENed the unit; otherwise, the Unit Block pointer is an odd number (invalid address). For reference on this subject, consult pages 239-242 of the OS FORTRAN PLM (GY28-6638).

* The CKFOREOF section of code depends on a flag being set to indicate if a TOPWR or TOPEF was the previous operation for the specified DSRN. In the current TAPOP, an unused byte (the 6th byte), in the IHCUATBL entries is used for this flag. If this byte is used by FORTRAN in the future (unlikely), then a modification will be required.

* The CLOSEIT section of code is used to logically disconnect the tape from FORTRAN. This is accomplished by making the Unit Block pointer in IHCUATBL an odd number (invalid address) and then freeing the storage occupied by the Unit Block.

* IBCOM/FIOCS use double buffering of input. As a result, the buffers are usually primed one record ahead. This causes a problem if FORTRAN READ's are being used and then a TOPRW, TOPFF, TOPBF, or TOPRF is issued and then more FORTRAN READ's are executed. Therefore, for each of the TAPOP functions, a branch to CLOSEIT is used in order to force the erasure of the buffers.

D. Use of Base Registers

TAPOP essentially uses one base register to cover all of TAPOP. This simplifies the addressability within TAPOP. However, TAPOP currently is almost 4096 bytes; as a result, if any significant additions are made to TAPOP in the future, the base register usage may need to be altered.

E. TAPEIO Modifications

The TAPEIO module of CMS was modified for two reasons:

(1) TAPEIO must handle a variable number of retries so that the NRTRY parameter of TAPOP can be supported and (2) TAPEIO must recognize the "Incorrect Length" condition and then return a code indicating the condition.

Both of these new features are activated by the use of a modified parameter list. If the current unmodified parameter list is used with the new TAPEIO, it will function exactly as the unmodified TAPEIO would do. A new return code of 100 in R15 will indicate "Incorrect Length". The TAPEIO UPDATE listing is included. The revised calling sequence (for READ's only) is unchanged except for the use of the high-order byte in the "number of bytes read" field to signal TAPEIO that the two new features are desired. If this byte contains X'Ø1', then an additional parameter containing the desired number of retries must be added. The revised parameter list is shown below:

```
PLIST DC      CL8'TAPEIO'
      DC      CL8'READ'
      DC      CL4'TAP1'
      DC      XL'93' mode set
      DC      AL3(BUFFER) I/O buffer address
      DC      F'buffer size' buffer length (in bytes)
      DC      X'01'   Indicates optional "number of retries"
      DC      XL3'00' parameter is used.  Upon return, the
                      full word will contain the number of
                      bytes actually read.
      DC      F'number of retries'
```

The attached TAPEIO UPDATE listing is for CMS Release 3.1 TAPEIO with PTF A28836CA applied.

```
FILE. . .      TAPEIO   UPDATE   A1

./ I 00103000                                                             TAP0001
         CLI   PRECSZ,X'01'         IS THIS A LARS TOPRD       LARS-JUN72 TAP0002
         BNE   CKBUFSZ              NO, GO CHECK BUFFER SIZE   LARS-JUN72 TAP0003
         NI    RDWRCCW+4,NOSILI     TURN OFF SILI IN CCW       LARS-JUN72 TAP0004
CKBUFSZ  EQU   *                                               LARS-JUN72 TAP0005
./ I 00126000                                                             TAP0006
         CLI   PRECSZ,X'01'         IS THIS A TOPRD CALL       LARS-JUN72 TAP0007
         BNE   STDTAPIO             NO                         LARS-JUN72 TAP0008
         L     COUNT,PNRTRY         GET VAR NUM OF RETRIES     LARS-JUN72 TAP0009
         BAL   LINK,TAPWAIT2        START THE TAPE AND WAIT    LARS-JUN72 TAP0010
         OI    RDWRCCW+4,SILI       RESTORE SILI IN CCW        LARS-JUN72 TAP0011
         B     STDRETRN                                        LARS-JUN72 TAP0012
STDTAPIO EQU   *                                               LARS-JUN72 TAP0013
./ I 00127000                                                             TAP0014
STDRETRN EQU   *                                               LARS-JUN72 TAP0015
./ I 00164000                                                             TAP0016
         TM    STATUS+1,ICL         INCORRECT LEIGTH           LARS-JUN72 TAP0017
         BO    12(,R13)             TAKE IC RETURN             LARS-JUN72 TAP0018
./ I 00256000                                                             TAP0019
*                                                                         TAP0020
WRONGLTH OI    RDWRCCW+4,SILI       RESTORE SILI IN CCW        LARS-JUN72 TAP0021
         LH    R15,CSW+6             GET BYTES NOT READ        LARS-JUN72 TAP0022
         SLL   R15,16                STRIP OFF SIGN BITS       LARS-JUN72 TAP0023
         SRL   R15,16                                          LARS-JUN72 TAP0024
         L     TEMP,PBUFFSZ         GET BUFFER SIZE            LARS-JUN72 TAP0025
         SR    TEMP,R15              SUBTRACT AMOUNT NOT READ  LARS-JUN72 TAP0026
         ST    TEMP,PRECSZ          SET RECORD READ SIZE       LARS-JUN72 TAP0027
         LA    R15,100               INCORRECT LENGTH ERROR    LARS-JUN72 TAP0028
         BR    R14                  EXIT                       LARS-JUN72 TAP0029
./ I 00264000                                                             TAP0030
*                                   (RETURNED TO CALLER.)      LARS-JUN72 TAP0031
*        FOR LARS, A VALUE OF X'01' IN THE HIGH ORDER BYTE     LARS-JUN72 TAP0032
*        WHEN TAPEIO IS CALLED WILL INDICATE AN ADDITIONAL     LARS-JUN72 TAP0033
*        PARAMETER TO CONTROL THE NUMBER OF RETRIES FOR        LARS-JUN72 TAP0034
*        READ OPERATIONS.  ALSO, THE PRESENCE OF THIS          LARS-JUN72 TAP0035
*        PARAMETER WILL ALLOW INCORRECT LENGTH CONDITIONS      LARS-JUN72 TAP0036
*        TO BE DETECTED FOR READ OPERATIONS.                   LARS-JUN72 TAP0037
                                                                          TAP0038
PNRTRY   DC    F'0'                 NUMBER OF RETRIES          LARS-JUN72 TAP0039
./ I 00302000                                                             TAP0040
NOSILI   EQU   X'DF'                MASK FOR NO SILI           LARS-JUN72 TAP0041
./ I 00343000                                                             TAP0042
         B     WRONGLTH             HANDLE INCORRECT LENGTH    LARS-JUN72 TAP0043
```

LARS Program Abstract 012

MODULE IDENTIFICATION

Module Name: TSTREQ Function Name: SYSTEM SUPPORT

Purpose: Checks status of STOP/SUSPEND flag

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd Date: 08/01/72

Latest Revisor: Date:

MODULE ABSTRACT

TSTREQ is called by FORTRAN to return the contents of a switch in the CMS USERSECT area and clean the switch.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised October 1980

1. Module Usage

TSTREQ

CALL TSTREQ (ICOMD)

Output Arguments:

ICOMD - INTEGER*4, Returned with the content of a half-word switch in the CMS USERSECT area. This switch is set by the 'STOP' and 'SUSPEND' commands. If the 'STOP' command has been issued since last call to TSTREQ, ICOMD is returned = 1. If the 'SUSPEND' command has been issued, ICOMD = 2. If both have been issued, the most recent one is reflected to TSTREQ.

The call to TSTREQ clears the switch so that any call after the switch is cleared will return ICOMD = 0 if neither 'STOP' nor 'SUSPEND' has been issued. The value of ICOMD for the first call to TSTREQ after ipling CMS is unpredictable.

2. Internal Description

TSTREQ uses the second halfword of the word called USER4 which is referenced by the CMS USERSECT macro. The first halfword of USER4 is unchanged. TSTREQ sets ICOMD = second halfword of USER4 and then sets this halfword = 0.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 013

MODULE IDENTIFICATION

Module Name: URADST Function Name: SYSTEM SUPPORT

Purpose: Unpacks data from Multispectral Image Storage Tapes

System/Language: CMS/ASSEMBLER

Author: Paul Anuta Date: 09/09/69

Latest Revisor: E. M. Rodd Date: 09/07/72

MODULE ABSTRACT

URADST takes input data in the format of the Multispectral Image Storage Tapes, unpacks it and converts to floating point full words.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

URADST

CALL URADST (RDATA,BDATA,NCR,NS,NSD,ISAM,LSAM,SINT)

Input Arguments:

BDATA - Raw data from the Multispectral Image Storage Tapes. This area is assumed to contain the data from the Multispectral Image Storage Tape for one line for NCR channels. The data for each channel is exactly as it appears on the Multispectral Image Storage Tape (including the last six bytes of calibration data). BDATA is as output from a call to TOPRV and thus does not contain the four bytes of line id information.

NCR - INTEGER*4, The number of channels in BDATA.

NS - INTEGER*4, The number of bytes per channel in BDATA (the number of data samples per channel + 6 for the calibration data).

NSD - INTEGER*4, Number of words available in the output array (RDATA)/NCR. NSD must be at least as large as the number of samples per line [(LSAM-ISAM)/SINT+1] + 6 for the calibration values.

ISAM - INTEGER*4, First sample number (column number) in a line requested.

LSAM - INTEGER*4, Last sample number (column number) requested in a line.

SINT - INTEGER*4, Sample interval.

Output Arguments:

RDATA - REAL*4, The output array to contain the unpacked data values and calibration values. RDATA(1) = sample ISAM from the first of the NCR channels. RDATA [(LSAM-ISAM)/SINT+1] = sample LSAM from the first of the NCR channels.

RDATA [(LSAM-ISAM)/SINT+2] = first of the six calibration values. RDATA(NSD+1) = sample ISAM from the second of the NCR channels etc.

If LSAM is greater than the last data sample in RDATA, it is assumed to be equal to the last data sample on the tape. The value of LSAM returned to the caller is unchanged.

If ISAM is greater than the last value in BDATA, only the calibration data will be returned in RDATA.

2. Internal Description

URADST operates by use of an internal subroutine (the loop beginning with label LOOP) which converts from integer to floating point. The code from NEXTCHAN to DONE is used simply to manipulate various indices to use this internal subroutine to read the 6 calibration values.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

The code is not the most compact or efficient. Much effort is taken to use the four statement internal subroutine to read calibration values when it would be better to simply repeat those instructions.

6. Flowchart

Not Applicable

80

LARS Program Abstract 014

MODULE IDENTIFICATION

Module Name: LOGICOPS Function Name: SYSTEM SUPPORT

Purpose: Provides logical operations for FORTRAN programs

System/Language: CMS/ASSEMBLER

Author: L. E. England Date: 03/70

Latest Revisor: P. A. Legare Date: 08/28/72

MODULE ABSTRACT

This is a function subprogram which returns the result of a logical operation on one or two input parameters. AND, OR, and COMPLEMENT are the logical operations available. The arguments may be Fullwords, Halfwords, or bytes. Two arguments are required as input for the AND and OR functions, and one argument is required for the COMPLEMENT. The Fullword arguments may be REAL*4 or INTEGER*4. Different entries are provided for all data types.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LOGICOPS

Input Arguments:

This is a function subroutine to provide logical operations to FORTRAN Programs. The logical AND, OR, and COMPLEMENT operations are provided. Each operation is performed on a bit-by-bit basis (each bit is independent of other bits). The parameters may be full words (32 bits), half words (16 bits,) or bytes (8 bits).

Entries to process integer fullwords

```
IWORD = IAND(WD1, WD2)

or

IWORD = IOR(WD1, WD2)

or

IWORD = ICOMPL(WD1)
```

where

IWORD is type INTEGER*4
WD1 and WD2 are INTEGER*4

The symbols IAND, IOR, and ICOMPL must be defined INTEGER*4 in the calling program

Entries to process real fullwords

```
WORD = AND(WD1, WD2)

or

WORD = OR(WD1,WD2)

or

WORD = COMPL(WD1)
```

where

WORD, WD1, and WD2 are REAL*4

The symbols AND, OR, and COMPL must be defined as REAL*4 in the calling program

Entries to process halfwords

IHWORD = IANDH(HWD1,HWD2)

or

IHWORD = IORH(HWD1,HWD2)

or

IHWORD = ICOMPH(HWD1)

where

IHWORD, HWD1, and HWD2 are INTEGER*2

The symbols IANDH, IORH, and ICOMPH must be defined as INTEGER*2 in the calling program

Entries to process bytes

IBYTE = IANDB(BY1,BY2)

or

IBYTE = IORB(BY1,BY2)

or

IBYTE = ICOMPB(BY1)

where

IBYTE, BY1, and BY2 are LOGICAL*1

The symbols IANDB, IORB, and ICOMB must be defined as LOGICAL*1 in the calling program

2. Internal Description

LOGICOPS executes the various functions by use of the machine instructions for AND and OR operations.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

84

LARS Program Abstract 015

MODULE IDENTIFICATION

Module Name: MOVBYT Function Name: SYSTEM SUPPORT

Purpose: Byte manipulation

System/Language: CMS/ASSEMBLER

Author: P. Swain Date: 07/70

Latest Revisor: E. M. Rodd Date: 09/26/72

MODULE ABSTRACT

MOVBYT is used to move bytes (and pack or unpack) from an input buffer to an output buffer.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

MOVBYT

```
CALL MOVBYT (INBUF,INDIS,ININC,OUTBUF,OUTDIS,OUTINC,
             NOBYTS)
```

Input Arguments:

INBUF - Can be any data type since MOVBYT simply moves bytes around. The first byte of INBUF is the first byte of the input buffer.

INDIS - I*4 The displacement in bytes from the first byte to be moved to the output buffer. If the first byte of INBUF is to be moved, INDIS = 0.

ININC - I*4 The increment of the bytes in the input buffer to be moved. This may be equal to zero in which case the same byte will be moved into all output buffer bytes.

OUTDIS - I*4 The displacement in bytes from the first byte of OUTBUF into which the first byte is to be moved. If the first byte of OUTBUF is to be used, OUTDIS = 0.

OUTINC - I*4 Increment of the bytes in OUTBUF which are to have bytes moved into them. OUTINC can equal zero though this would simply cause successive moves to overlay the same byte.

NOBYTS - I*4 The number of bytes to be moved from INBUF to OUTBUF.

Output Arguments:

OUTBUF - Can be any data type since it is just the bytes moved from the input buffer.

2. Internal Description

Not Applicable

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

87

LARS Program Abstract 016

MODULE IDENTIFICATION

Module Name: TIMER Function Name: SYSTEM SUPPORT

Purpose: Gives time of day in 100th sec.

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd Date: 08/01/72

Latest Revisor: Date:

MODULE ABSTRACT

TIMER is a function subprogram used to return the time of day in increments of 1/100th of seconds. Timer uses the OS TIME macro.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

TIMER

J = TIMER(I)

Input Arguments:

I - The value of I is irrelevant and is not changed. it is used only so that the compiler recognizes TIMER as an external function without the use of an EXTERNAL statement.

Output Arguments:

J - INTEGER*4, The time of day expressed as number of hundredths of seconds since midnight.

TIMER is useful for interval timing. Since CMS timer is not updated as frequently as the timer on a real machine, successive calls to TIMER can produce the same value. Timing intervals less than five seconds is not recommended using TIMER for this reason.

2. Internal Description

TIMER executes the OS TIME macro with the BIN option which automatically places the desired value in RO. See form GC28-6647-5.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 017

MODULE IDENTIFICATION

Module Name: WRTMTX Function Name: SYSTEM SUPPORT

Purpose: Prints out lower half of symmetric matrix.

System/Language: CMS/FORTRAN

Author: E. M. Rodd Date: 12/12/72

Latest Revisor: Date:

MODULE ABSTRACT

WRTMTX prints out the lower triangular portion of a symmetric matrix. The columns and rows are labeled with the spectral band values.

1. Module Usage

WRTMTX

CALL WRTMTX (MATICE,SIZE,FRQCAL,BCD,FETVEC)

Input Arguments:

MATICE - REAL*4, The input matrix. This is one dimensional and contains only the lower triangular portion of the symmetric matrix. The order of elements is c_{11}, c_{21}, c_{22}, c_{31}, etc.

SIZE - INTEGER*4, The size of the matrix. The full matrix is SIZE x SIZE.

FRQCAL - REAL*4, The FRQCAL array from GLOCOM. This includes words 51-200 of the ID record from the data tape.

BCD - INTEGER*4, The values from MATICE will all be printed with format F9.x where x is the EBCDIC value in BCD. Thus to print with F9.2, BCD must be input as '2'.

FETVEC - INTEGER*2, FETVEC(I) = channel number of I'th row and column of MATICE. FETVEC is used to index FRQCAL. Thus WRTMTX could be used to print any numbers for the row and column labels. However, the row and the column labels are called 'SPECTRAL BANDS' on the printout.

DWRTMX

CALL DWRTMX (DMATIC,SIZE,FRQCAL,BCD,FETVEC)

Input Arguments:

The arguments are the same as for WRTMTX except that DMATIC is R*8.

2. Internal Description

Not Applicable

3. Input Description

Not Applicable

4. Output Description

A printout is generated of the lower triangular portion of the matrix. The printer is double spaced before the first line.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 025

MODULE IDENTIFICATION

Module Name: IBCD Function Name: SYSTEM SUPPORT

Purpose: Converts full word signed integer numbers to their BCD character equivalent.

System/Language: CMS/ASSEMBLY

Author: W. R. Simmons Date: 02/10/70

Latest Revisor: S. McAhren Date: 11/22/72

MODULE ABSTRACT

Program to change integer number from numeric (computer) notation to printed (literal) notation.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

IBCD

CALL IBCD (I,OUTBUF,ND)

Input Arguments:

I - INTEGER*4, The variable I is a full word signed integer number.

ND - INTEGER*4, The variable ND is a positive full word integer and must be equal to the number of bytes in OUTBUF.

Output Arguments:

OUTBUF - The decimal character representation of I is returned in OUTBUF. The subroutine considers OUTBUF on the byte basis and therefore no boundary alignment is required. The low order decimal digit will be returned in the NDth byte of OUTBUF, and the next higher order digit in the (ND-1)th byte, etc. If I contains more than ND decimal digits, OUTBUF will be filled with ND astericks (*). Unused high order digits in OUTBUF are set to blank.

2. Internal Description

This program takes each digit of an integer number and converts it to BCD code. It places the code in the appropriate byte. If the number of digits exceed the bytes allocated, astericks (in BCD code) are placed in each of the bytes.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Examples of use of CALL IBCD (I,OUTBUF,ND)

Input: I = 183, OUTBUF = one full word integer, ND = 4

Output: OUTBUF = b183 or 40F1F8F3 in hexidecimal where:

b = blank

Input: I = 183, OUTBUF = 2 full words integer array
ND = 6

Output: OUTBUF = bbb183xx or 404040F1F8F3xxxx in hexidecimal where:

b = blank

x = unchanged

Note: The first and third call statement arguments may be constants, but the second argument must be a variable containing at least the number of bytes specified in argument 3.

6. Flowchart

Not Applicable

LARS Program Abstract 028

MODULE IDENTIFICATION

Module Name: RTMAIN Function Name: SYSTEM SUPPORT

Purpose: Returns to main program

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd Date: 01/08/73

Latest Revisor: Date:

MODULE ABSTRACT

RTMAIN returns to a FORTRAN main program at the statement after its last call.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

RTMAIN has no arguments and is FORTRAN callable. If any assembly language programs are in the chain between the caller of RTMAIN and the FORTRAN main, RTMAIN will produce unpredictable results. RTMAIN returns to the main program at the statement after its last call to a subroutine.

2. Internal Description

RTMAIN uses the save area chain to branch back to the main program. It identifies the main program by the first word of its epilog. Word 1 of a FORTRAN program's save area points to the epilog (it is for this reason that an assembler program in the chain will cause unpredictable results). The first instruction of a main program's epilog is:

'L 13,4(13)' or X'58D0D004'.

This is not the case for subroutines. When the main program is identified, R14 is retrieved from its save area and its registers R2-R12 are restored. Then a branch is made to R14 which is the return address stored when main called a subroutine.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

97

LARS Program Abstract 029

MODULE IDENTIFICATION

Module Name: PACCT Function Name: SYSTEM SUPPORT

Purpose: Creates LARSYS usage account cards

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd Date: 01/11/73

Latest Revisor: Date:

MODULE ABSTRACT

PACCT initializes timing information.

PACCT1 generates the usage account card for a function.

TPACCT generates a data run usage account card.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

PACCT

CALL PACCT (FUNC,CHKOUT)

Input Arguments:

FUNC - INTEGER*4, A four character function code indicating the function about to be executed (e.g., '*CLA')

CHKOUT - INTEGER*4, The control card checkout flag. It is .TRUE. if the control card checkout option is in effect for this function.

The function code and an indication of checkout mode appear on the usage account card.

PACCT1

CALL PACCT1

PACCT1 has no arguments.

PACCT is called to initialize timing data for a function. It is called prior to execution of the function. PACCT1 is called immediately after the function has completed. PACCT1 will produce program interrupts if PACCT has not been called before PACCT1.

TPACCT

CALL TPACCT (RUN,TAPE,FILE)

Input Arguments:

RUN - INTEGER*4, Multispectral Image Storage Tape run number

TAPE - INTEGER*4, The tape number containing the run

FILE - INTEGER*4, The file containing the run.

TPACCT is called from GADRUN after the run has been found in the runtable. Thus it is called once for each time a new run is made available to a user.

2. Internal Description

PACCT uses the diagnose instruction with code X'C' to get timer information (see the CP PLM). PACCT1 again uses the diagnose for timing information. It then uses packed decimal instructions to compute the elapsed time. The elapsed time, the total CPU time and the virtual CPU time are converted to characters and the LARS supplied CP function PCHACNT is executed to punch the usage account card. The CP function is executed via the CMS CPFUNCTN command.

TPACCT converts the run, tape and file numbers to characters and uses the PCHACNT command to produce the usage account card.

3. Input Description

Not Applicable

4. Output Description

PACCT1 produces the function usage account card of the format:

COL	Entry
1-50	Standard information produced by PCHACNT.
51-54	Function code taken from input parameter FUNC to PACCT.
55	'C' if CHKOUT was input to PACCT as .TRUE. and blank if CHKOUT was .FALSE.
56-62	Elapsed clock time in seconds. This is in integer character format right justified.
63-70	Virtual CPU time used by the function in milliseconds. Also a right justified character format integer.
71-78	Total CPU time used by the function in milliseconds. Also in characters.
79-80	'L3'

TPACCT produces a tape usage account card of the format:

COL	Entry
1-50	Standard information produced by PCHACNT.
51-58	Run number as an eight digit integer in character form.
59-62	Tape number as a right justified four digit integer in character form.
63-66	File number also as a four digit integer.
67-78	Blank
79-80	'GR'

5. Supplemental Information

Correct results make it imperative to be certain that PACCT is called prior to the function.

6. Flowchart

Not Applicable

LARS Program Abstract 032

MODULE IDENTIFICATION

Module Name: STATE   Function Name: SYSTEM SUPPORT

Purpose: Determine if a file exists

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd   Date: 02/13/73

Latest Revisor: Date:

MODULE ABSTRACT

STATE determines if a specific file exists. It is FORTRAN callable.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

STATE

CALL STATE ('file description', &nnn)

Input Arguments:

'file description' is an eighteen character literal of the form

1-8 filename of file (padded with blanks on the right if less than eight characters)

8-16 filetype of the file (also padded with blanks)

17-18 filemode of the file (or '* ' if the file can be on any disk)

Non-Standard Returns:

RETURN 1 is executed if the file does not exist. Otherwise a normal RETURN is made.

2. Internal Description

STATE executes the CMS STATE function.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 035

MODULE IDENTIFICATION

Module Name: CMSFNC Function Name: SYSTEM SUPPORT

Purpose: Executes CMS commands

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd Date: 08/01/72

Latest Revisor: Louis Lang Date: 08/01/80

MODULE ABSTRACT

This program allows FORTRAN programs to execute CMS Commands. If an error occurs, an error code is passed to the caller.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CMSFNC

CALL CMSFNC (NCHAR, TEXT, ERROR)

Input Arguments:

NCHAR - I*4 The number of characters in the CMS command to be executed.

TEXT - This is a literal containing the text of the CMS command. It may be input in two ways: one is as a literal 8 byte-aligned on left in the call (e.g. 'DETACH 181') and the other is that TEXT may be an array containing the text in characters.

Output Arguments:

ERROR - I*4 The error code returned from CP. If NCHAR is greater than 99, ERROR will be returned = 100+ the error code from CP and only the first 99 characters of the console function will be sent to CP.

Note that if NCHAR does not match the number of characters in TEXT, strange results could occur.

2. Internal Description

CMSFNC first executes the CMS CONWAIT function to clear all the buffered output to the terminal. This is done so that a response from CMS will appear in the correct place on the terminal output. Then NCHAR is checked. If it is greater than 99, it is set to 99 and R2 initialized to 100 rather than 0. NCHAR itself is not changed, only the register used internally is changed. Then that number of characters is moved from TEXT to the buffer of the CMS command. The X'FF' fence is then added after the last character. Then the command is executed via SVC 202 and R15 is added to R2 to form the final error code. This value is placed in ERROR and CMSFNC returns.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

106

LARS Program Abstract 054

MODULE IDENTIFICATION

Module Name: GENALL Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Generates all load modules.

System/Language: CMS/EXEC

Author: E. M. Rodd Date: 01/09/73

Latest Revisor: Paul W. Spencer Date: 02/01/74

MODULE ABSTRACT

GENALL generates all LARSYS load modules including the root.

The load maps will be printed unless the parameter NOMAP is present.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GENALL has no parameters. It generates all function load modules including the root. If NOMAP is specified, it is passed as a parameter to each individual module generation EXEC routine called by GENALL, indicating to the individual routine that no load map be printed. Thus NOMAP in GENALL requests no load maps be printed. If no parameter is specified, all load maps are printed. A CLOSIO PRINTER OFF is issued before the generation begins and CLOSIO PRINTER ON is issued afterwards so that all load maps, if requested, will be printed in a single printer file.

A NOMAP parameter is available when using an individual module generation EXEC routine. The parameter controls the printing of the load map.

2. Internal Description

GENALL executes all the module generation EXECs.

3. Input Description

Not Applicable

4. Output Description

The load maps are produced by the individual module generation EXECs. GENALL types the name of each load module before executing the generation EXEC. The normal typeout of the generation EXEC is typed, i.e. typewriter messages are produced to indicate the usual unresolved reference to the LARSYS dummy routine PROCES; and to indicate the highest storage location used by the created load module.

5. Supplemental Information

GENALL must be used whenever a program module in the root is altered.

6. Flowchart

Not Applicable

LARS Program Abstract 055

MODULE IDENTIFICATION

Module Name: BATRD

Function Name: LARSYS Executive Routines

Purpose: LARSYS Batch Monitor Utility

System/Language: CMS/Assembler

Author: E. M. Rodd

Date: 08/15/72

Latest Revisor: Louann Grady

Date: 04/01/75

MODULE ABSTRACT

This program's two entry points perform the reading and interpretation of batch header cards, the communication with the operator and the batch controller batch machine, BATCH, of the status of the batch machine, the automatic waiting for the next job to be entered and the generation of accounting information.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

For LARSYS, the batch monitor is controlled by BPROFILE (an EXEC file on the C-disk of the batch machine). See LARS Abstract 056. BPROFILE creates a new EXEC file (.TEMP. .NAME.) on the batch machine's P-disk, and alters it to EXCOMD EXEC P1 (there is already a different EXCOMD on the C-disk). The sole purpose of this new EXCOMD is to call BPROFILE automatically when a batch job is finished. This setup allows BPROFILE to be started once manually, and then automatically after 'ipl larsys' for all following batch jobs.

BATRD is used by BPROFILE to control the batch monitor. There are two entry points in BATRD (BATRD and BATEND).

STARTUP

When LARSYS is IPL'ed, EXCOMD is automatically invoked if it exists. Since BPROFILE has normally previously created an EXCOMD on the P-disk of the batch machine, that EXCOMD is automatically invoked, and in turn invokes BPROFILE. BPROFILE requests a system time limit if one was not present in its parameters, erases ACCT BATCH if present, detaches 192 if present, clears 191, recreates the EXCOMD file, and executes BATRD.

The normal method of running a batch machine is disconnected, obtained by invoking BPROFILE by the sequence:

```
LOGIN  19C C
EXEC  BPROFILE  nnnnnnnn SEC DISC.
```

If the timer information is not present, message B10 will ask for it. If DISC is not included, the batch machine will run in connected mode. This timer and disconnect information is saved and reused in all future re-creations of EXCOMD EXEC.

BATRD

After initialization, BPROFILE operates in a control loop, reading in user runs, executing them, and punching account cards. It uses the BATRD module to perform major parts of the process.

The first step is to call BATRD to read batch header cards and initialize accounting information. BATRD starts by checking the DRYUP flag (first halfword of the word USER4 in SYSREF in the CMS nucleus). If the DRYUP command has been issued by the operator and the DRYUP flag is on, then message B01 is sent to the operator and the machine is logged out.

If the DRYUP flag was not on, the CP function SET CARDSAVE ON is called to save the input file in the event of machine failure. Then the CMS function CARDRD is used to read the first batch header card. If an error exists, exit is taken from the read. If the cause was an empty card reader, then WAIT is executed, if not then a new attempt is made to read a BATCH ID card.

When a batch card is the first card read, then BATRD interprets both it and the following batch header cards. The first card is checked to be sure that it does contain 'BATCH' 'ID' followed by the ID field and USERNAME field. If either is missing, BER is executed and the job is flushed. The given ID is checked against the CP directory for validity using the LARS-defined diagnose code 108. (Refer to LARS System file 9010). If invalid, the job is flushed. If valid, the corresponding ACCOUNT NO is obtained for later use on the account card. If everything is in order, the USERID, ACCOUNT NO, and USERNAME are placed in local storage. The username is also inserted into the CP UTABLE of the virtual machine using LARS-defined diagnose code 104. The internal subroutine FINDFLD is always used to find the start of the next field on the card since the cards are free format. If any errors are encountered, BER is executed. R4 is a flag to indicate which card is being interpreted. It is 0 if an ID card, 1 if working on the TIME or OUTPUT cards and scanning has not reached the site-id fields, and -1 if the site-id field has been reached on the OUTPUT card. This flag is needed because the same code is used to read all cards and to check to see that the card starts with 'BATCH'. The flag is used to branch after that point.

An optional BATCH TIME card may follow the BATCH ID card. This card contains 'BATCH' 'TIME' and the maximum desired CPU time in seconds (up to 8 digits). The program right adjusts the number and fills in blanks with zeros.

When the BATCH OUTPUT card is recognized, the CP functions REMOTE E OFF and REMOTE D OFF are executed before reading any site-ids given on the OUTPUT card for which new remotes may be executed. Thus the default is to send output to the computer site. After the header cards have been successfully interpreted, the operator receives a message indicating that a run has started for a certain user. Then a diagnose is issued to cause CP to execute the timer function. The userid, account number, date and time, and CPU time used thus far by the batch machine are stored in an output buffer.

The output buffer is completed by storing the CPU time limit. The limit from the BATCH TIME card (if one was present) is compared to the machine time limit from BPROFILE (passed as a parameter to BATRD), and the smaller is used. If only the user or the machine time limit is present, that one is used. If neither is present, the time limit is set to the largest possible time (14.6 hours). BATRD calls the CMS FREE function to obtain sufficient free storage for a section of code called TIMINT. Then this code is copied into this free storage and executed. TIMINT sets the timer value to the specified time limit and alters the external new PSW address to an address in TIMINT which is to receive control when the time limit is reached or when an external interrupt is received. Since TIMINT is in free storage, it will remain there even during execution of LARSYS.

TIMINT handles all external interrupts. External interrupts can be generated by the timer and also by the operator who issues the interrupt along with a special code which sets the external old PSW interruption code (bits 24-31). Altogether there are four uses of the external interrupt and corresponding interruption codes. These are:

Bit pattern (bits 24-31)	Use
10000000	TIMER - stop job execution
00100000	DRYUP - set first halfword of DUSER4 to 1
00010000	STOP - set second halfword of DUSER4 to 1
00001000	SUSPEND - set second halfword of DUSER4 to 2

All of these are valid when the batch machine is active, but only the DRYUP has meaning when the machine is idle.

When an external interrupt is received, control passes to an address in TIMINT. There the interruption code is checked to be sure that it is valid. If it is valid the appropriate action is taken. Otherwise execution continues at the point where the external interrupt was received.

If the batch machine is entering the idle state it sets hexloc 94 in the address field of the external new PSW (bits 48-63). At this location code has been moved which will test for a DRYUP interruption code - the only valid one. If it is valid then a diagnose code 8 is issued to use CP-function to inform the operator that the batch machine is logging out. Then a second diagnose code 8 is used to perform the actual logout. For any other external interrupt the batch machine returns to the idle state.

As BATRD reads through the header cards and collects accounting information, it is also constructing a message in a buffer to send to the batch controller batch machine, BATCH, to indicate that a job is starting. If in the process of reading the header cards BATRD finds that this job was not originally submitted to

BATCH, it will generate the extra header card needed and transfer the deck to BATCH. The deck is then flushed and another job begins to be processed.

The card punched to inform BATCH that a job is starting has the following format:

Col.	Entry
1-5	'BATCH'
6	Blank
7-11	'START'
12-15	Blank
16-23	Batch machine name
24	Blank
25-32	Userid
33	Blank
34-41	Time of job submission
42-80	Blank

Control is then returned to BPROFILE which acquires a T-disk and issues the RUN LARSYS command.

BATEND

On completion of the run, BPROFILE loads BATRD and executes it, starting at the entry point BATEND to punch the CP accounting card and punch an information card to BATCH indicating the job has finished. BATEND first executes internal subroutine CLEARFIL to clear this job from the reader. BATEND reads the starting information from the file ACCT BATCH (if the file does not exist, SER is executed). The diagnose is executed to get CP to get the current timer information. To get the CPU time used by this run, the CPU time from the file ACCT BATCH is subtracted from the current CPU time. Then the CPU time is converted to characters and then a check is made to see if the monitor is running on a IBM 360 or 370. If it is a 360 then the LARS CP function PCHARD is executed to punch an account card with the following format:

Col.	Entry
1-8	Userid
9-16	Account Number
17-28	Date and Time started
39-40	Date and Time ended
41-48	CPU time used in milliseconds
49-53	'BATCH'
54-68	Blank
69-76	Userid
77-78	Blank
79-80	X'FF'

If the machine is a 370, then the present time and the start time are converted to numbers from characters and the total attach time is computed. Then the LARS CP function PCHCARD is executed to punch an account card in the following format:

Col.	Entry
1-8	Userid
9-16	Account Number
17-28	Date and Time ended
29-32	Attach time in seconds (binary)
33-36	Total CPU time used in milliseconds
37-40	Virtual CPU time used in milliseconds
41-68	Blank
69-76	Userid of batch virtual machine
77-78	Blank
79-80	'Bl'

BATEND then checks to see how it gained control. This is done by checking the status byte called ENDTYP If BATEND was called by BPROFILE after completion of the batch run, then this byte is 0. If the job exceeded its time limit, then the byte is 1 and message I0037 is printed on the user's output.

Message number I0039 is then printed to the printer informing the user of the starting and ending times for the job and the total CPU time used.

While executing the BATEND code, information is collected in order to punch a card informing BATCH that a job is finishing execution. This card has the following format:

Col.	Entry
1-5	'BATCH'
6	Blank
7-12	'FINISH'
13-15	Blank
16-23	Batch machine name
24	Blank
25-32	Userid
33	Blank
34-41	Time of job submission
42	Blank
43-50	Type of job termination
51-80	Blank

BATEND then sends a message to the operator indicating that a batch job has ended for this user. If the job ended by exceeding the time limit, then the message will indicate so. Finally, BATEND closes the printer and punch and re-ipl's LARSYS.

2. Internal Description

BATRD employs the following internal subroutines:

FINDFLD - Finds the next non-blank in the buffer card. The position within CARD is in R7 and the address to which FINDFLD is to return is in R8. If FINDFLD reaches column 72 a check is made to see which batch header card is being interpreted. Unless the scan is as far as the site-id fields of the OUTPUT card (which is optional), an error (B02) is detected.

SIDLIN - Finds the length of the parameter in characters. The length is placed into R8 form where it will be used to execute a MVC instruction via EX instruction. The return address is in R9.

WAIT - Reduces the number of pages allocated on the paging device for an idle batch machine to one. This is accomplished by setting up enough code in page 0 to force a re-IPL whenever an I/Q interrupt occurs. Then a diagnose code 10 is issued to CP to release all pages except for page 0.

UNEXE - Sends error B03 indicating that an unexpected error has occured on card reading which includes the error code. It then logs out.

BER - Sends message B02 indicating batch header and error, and then executes CLEARFIL internal subroutine with return address set to label BEG (start interpretation of next set of batch header cards).

CLEARFIL - Clears a job from the virtual card reader. BATRD sets CARDSAVE ON before it begins reading a batch job. This is done so that if the machine goes down while processing this job, the card file will still be in the reader when the machine comes back up. CLEARFIL closes and readies the reader and then issues a CP SET CARDSAVE OFF. It then reads the first card. It then closes and readies the reader. This is necessary to treat the reader as a reader on a real machine (where it is physically necessary to read one card file to get to another). CLEARFIL returns to the address passed to it in R4.

CPFUN - Is used to execute CP functions. If the first four characters of the CP function are 'M CP', then the following message is also typed at the terminal using the CMS TYPLIN function. On input R5 is the address of the CP function, and R6 contains the length in bytes. R3 contains the return address. CPFUN uses the (CMS CPFUNCTN) function to execute CP functions.

TYPMSG - Is part of CPFUN. It will type a message on the terminal. R5 contains the address of the message R6 the length, and R3 the return address. ('M CP' is stripped off message before printing.)

3. Input Description

Reads batch header cards and the ACCT BATCH file.

4. Output Description

Writes ACCT BATCH and various error messages.
Punches information cards to BATCH.

5. Supplemental Information

These are the messages the CP operator can receive from the batch monitor. All are of the form 'Bnn text'. Note that all messages also appear on the batch terminal if it is connected.

B01 LOGGING OUT AS A RESULT OF DRYUP COMMAND.

This message is issued when the batch machine is being logged out at the completion of a user run because the operator has issued the DRYUP command.

B02 ERROR IN BATCH HEADER CARDS ID = XXXXXXXX CARD FOLLOWS

This message is issued when a syntax error is detected in a batch header card. If the error is in the OUTPUT card, then the userid will be printed. If the error is in the ID card, the ID is, of course, not known and will not be printed. The card image of the erroneous card is then printed for the operator. (The batch monitor should be able to recover from this error by flushing the run and going on to the next run in the queue.)

B03 UNEXPECTED ERROR IN CARD READER - ERRCODE IS nnnn-LOGGING OUT

When this message is printed, the monitor is in serious trouble. The best hope of recovery is to get on the batch machine and from CP close and ready the reader and re-ipl LARSYS and thus restart the monitor. This error can occur if only one batch header card was given and no data cards. It can also occur when hardware errors occur.

B04 A BATCH JOB FOR XXXXXXXX (nnnnnnnn) HAS STARTED

This message indicates that the header cards have been interpreted and the job is about to start. XXXXXXXX is the userid and nnnnnnnn is the name from the BATCH ID card.

B05 ACCT FILE DOES NOT EXIST

This message means that BATEND could not find the file ACCT BATCH written by BATRD. The machine is then logged out. This could occur if an I/O error had occured when the file was being written.

B06 BATCH JOB ENDED FOR XXXXXXXX

This message is printed when a job has ended. It does not mean that LARSYS terminated error free. If this message has the form B06 BATCH JOB ENDED FOR XXXXXXXX - ENDED BY EXCEEDING TIME LIMIT then the job exceeded either the machine or user time limit.

B07 NO TEMPORARY DISKS ARE AVAILABLE

This message indicates that when BPROFILE attempted to acquire a T-disk, none were available. The batch virtual machine is logged out. The job indicated in the last B04 message was not run but is saved in the virtual reader of the batch machine to be run when the machine is logged in again.

B08 SYNTAX ERROR IN TIME LIMIT SPEC.

This message is produced when the user time limit specification on a BATCH TIME card is incorrect. This will occur if the number contains imbedded blanks or invalid characters. The batch job is not run.

B09 BATCH MACHINE CAN'T CONTINUE. '.TEMP. .NAME.' ALREADY EXISTS

Each time BPROFILE is run, the batch machine's P-disk is erased and a new file '.TEMP. .NAME.' is created and later altered to EXCOMD EXEC. If a '.TEMP. .NAME.' already exists on a supposedly erased P-disk something has gone wrong. The job is logged out.

B10 WHAT IS TO BE THE JOB TIME LIMIT, IN SECONDS?

If a machine time limit was not included in the call to BPROFILE, this message will request a time limit. Only the number of seconds need be typed in. This time limit will be used on all future executions of BATRD.

B11 UNEXPECTED ERROR IN CARD PUNCH - ERRCODE IS nnnn-LOGGING OUT

This message is produced if there was a punch error when a "BATCH START' or 'BATCH FINISH' card was being punched or when a job entered directly to the batch machine is being transfered to the batch controller virtual machine, BATCH.

B12 INCOMPLETE DECK RECEIVED - FLUSHED FROM READER

This message is produced if a file contained only a 'BATCH ID' card. This should actually never occur since the batch controller virtual machine, BATCH, also checks for incomplete decks.

Two messages may appear on the printed output;

I0037 BATCH RUN TERMINATED DUE TO EXCEEDING CPU TIME LIMIT OF XXXXXXXX SECONDS (BATEND)

The batch machine may have a CPU time limit on all jobs run on it, or the use may have specified a time limit on the BATCH TIME card. (The system picks the smaller of the two). I0038 indicates this time limit was exceeded. Output may be incomplete.

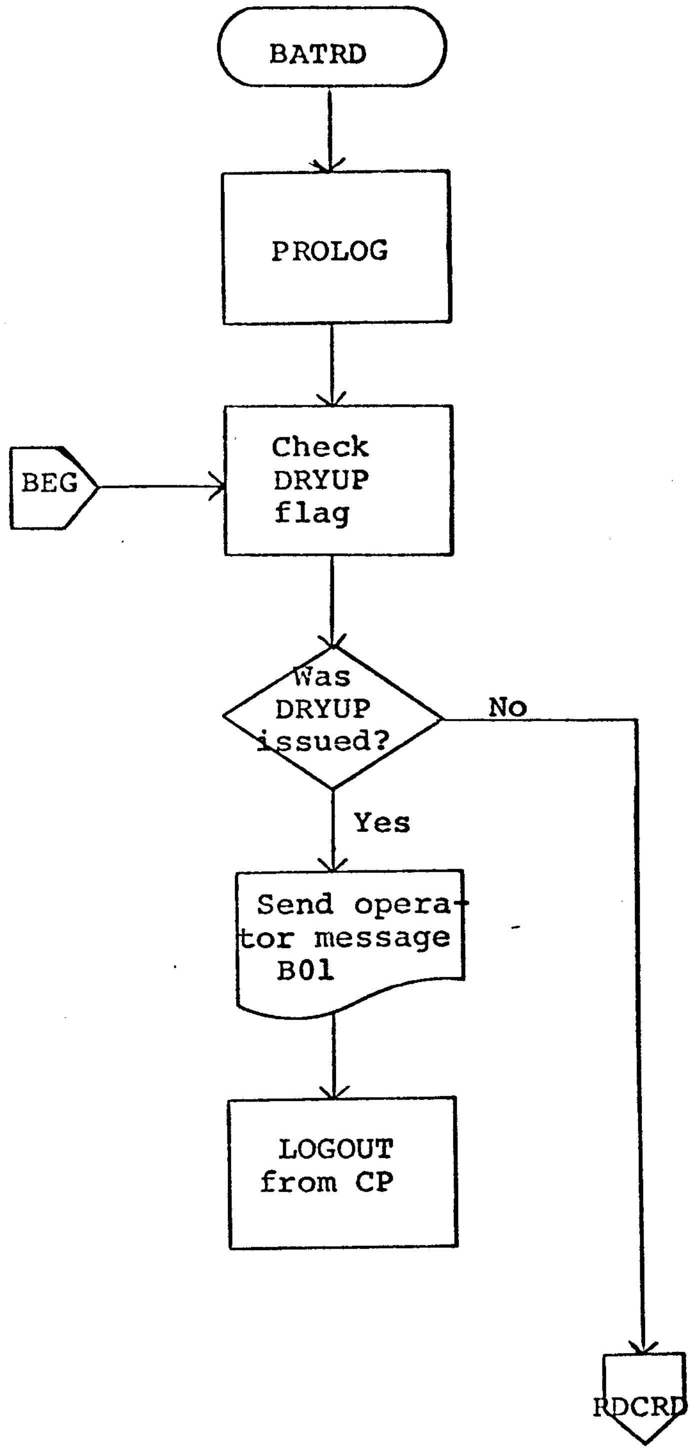
I0038 BATCH RUN TERMINATED DUE TO SYNTAX ERROR ON TIME CARD

The message is printed whenever an error is observed in the BATCH TIME card. The printed output is often the only output that the user will see, so I0038 tells why the job was not run.

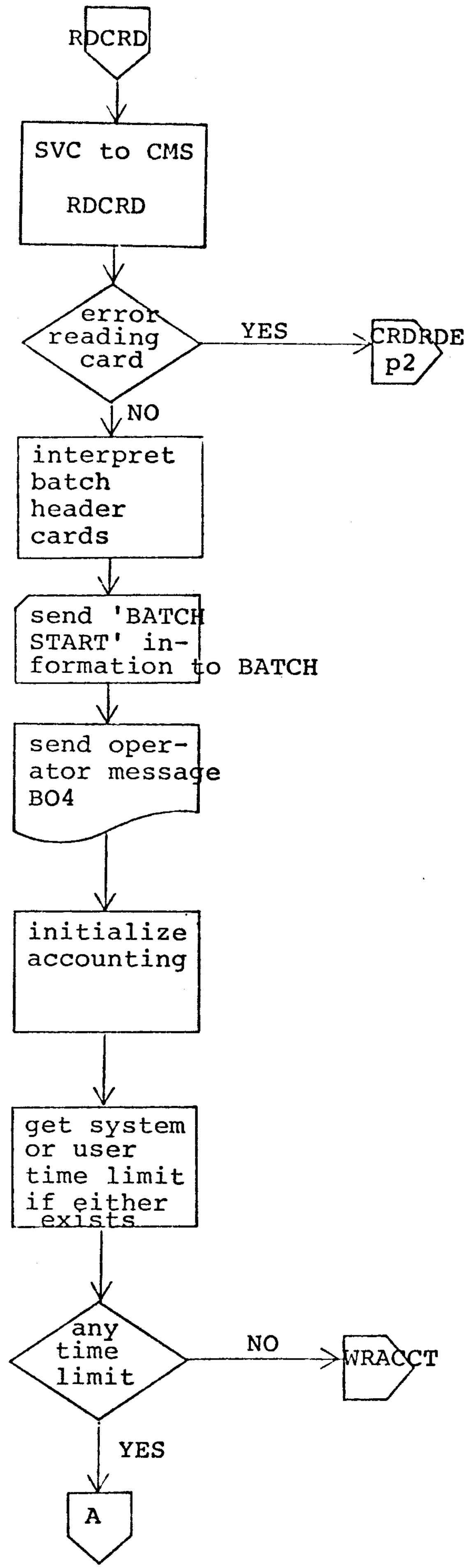
I0039 BATCH JOB STARTED HH.MM.SS ENDED HH.MM.SS
TOTAL CPU TIME USED X.XX SECONDS. (BATEND)

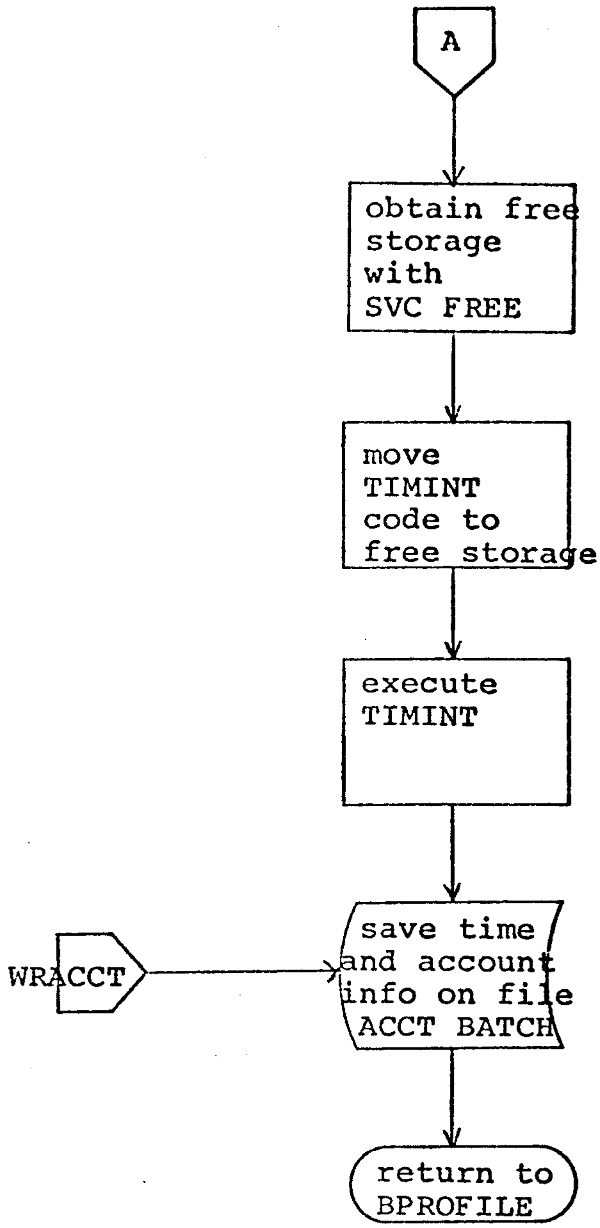
This message is printed at the end of each batch job to inform the user of the amount of CPU time used and charged for, as well as the time of day the job was run.

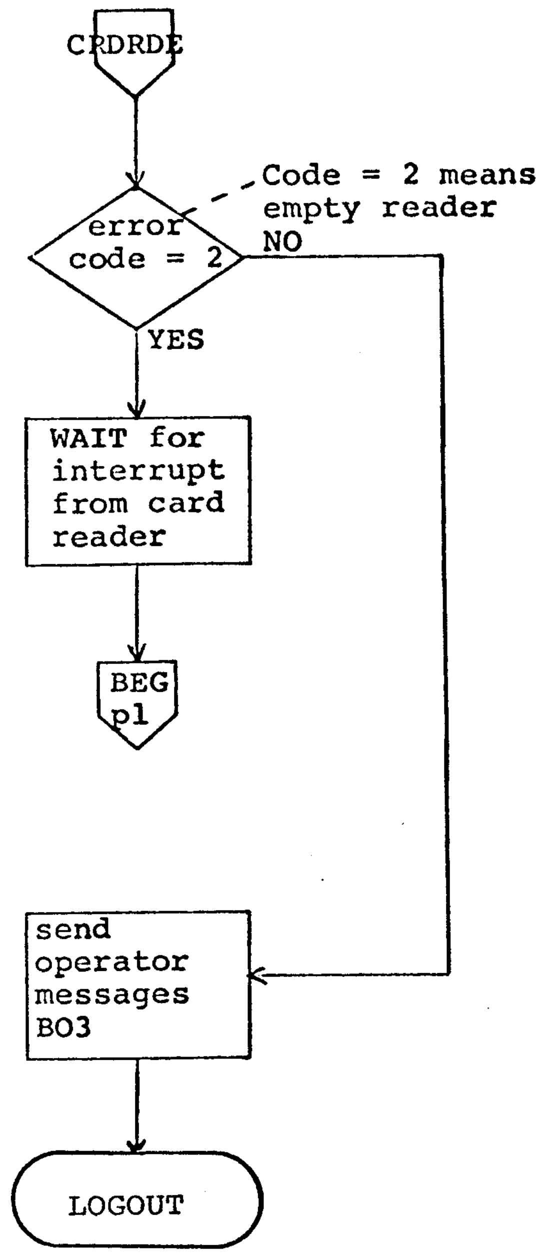
6. Flowchart of the BATRD routine
(page numbers refer to pages of the flowchart)

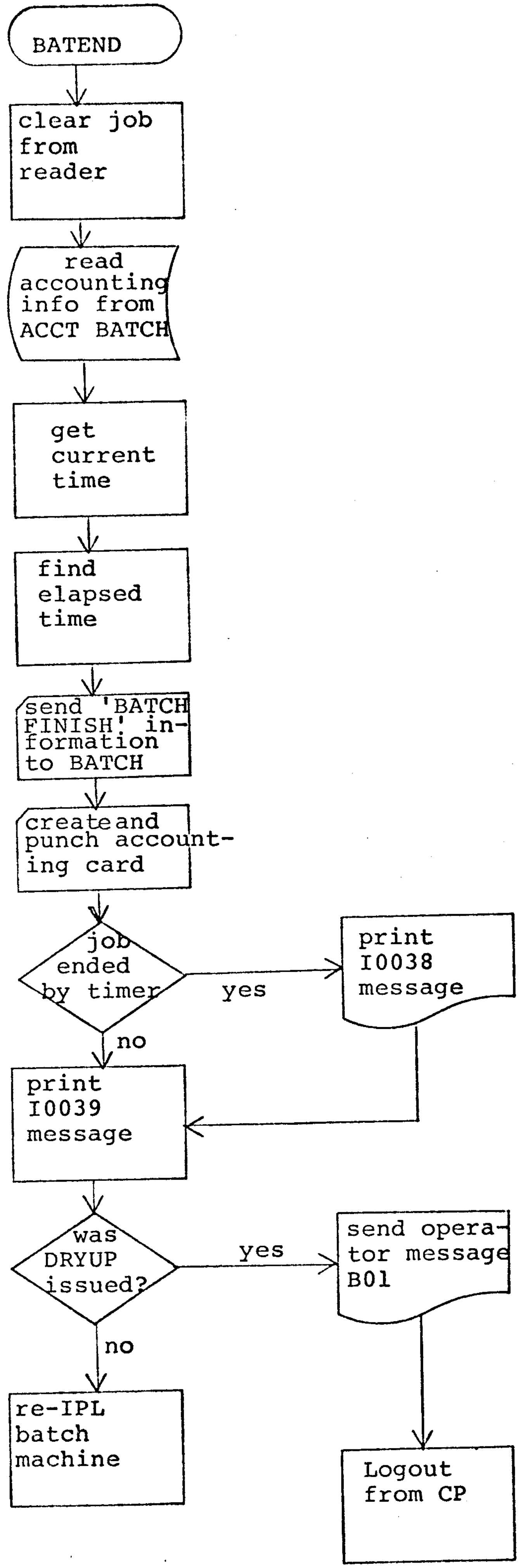


120


RDCRD
SVC to CMS
RDCRD
error reading card
YES
CRDRDE p2
NO
interpret batch header cards
send 'BATCH START' information to BATCH
send operator message BO4
initialize accounting
get system or user time limit if either exists
any time limit
NO
WRACCT
YES
A


A
obtain free storage with SVC FREE
move TIMINT code to free storage
execute TIMINT
WRACCT
save time and account info on file ACCT BATCH
return to BPROFILE


CRDRDE
Code = 2 means
empty reader
error
code = 2
NO
YES
WAIT for
interrupt
from card
reader
BEG
p1
send
operator
messages
BO3
LOGOUT


BATEND
clear job from reader
read accounting info from ACCT BATCH
get current time
find elapsed time
send 'BATCH FINISH' information to BATCH
create and punch accounting card
job ended by timer
yes
print I0038 message
no
print I0039 message
was DRYUP issued?
yes
send operator message B01
no
re-IPL batch machine
Logout from CP

LARS Program Abstract 056

MODULE IDENTIFICATION

Module Name: BPROFILE Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Control EXEC for the batch monitor

System/Language: CMS/EXEC

Author: Earl M. Rodd Date: 08/15/72

Latest Revisor: Ellen McDonald Date: 05/01/74

MODULE ABSTRACT

BPROFILE is the control EXEC for a LARSYS batch machine. For the complete documentation of the batch monitor see the documentation for module BATRD (LARS Program Abstract 055). BPROFILE executes a batch job by calling BATRD to interpret the header cards, running LARSYS and finally calling BATEND to complete accounting information and re-IPL the virtual machine and re-start BPROFILE.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised September 1974

1. Module Usage

See LARS Abstract 55

2. Internal Description

This module examines the parameter received when it was called, for timing and disconnect information. If no time limit was included, message B10 will request a time limit. It FILEDEF's FORTRAN unit 15 to the file BATCH STOP on the C-disk (this will cause termination of any job if LARSYS should issue a read request to the typewriter) and erases old accounting information. It detaches its temporary disk and clears its P-disk. It creates a file (.TEMP. .NAME.) on the P-disk and then alters the name to EXCOMD EXEC (the purpose of this EXCOMD EXEC is to re-call BPROFILE after the batch job is finished, using the previously specified system time limit and disconnect parameters, and the BATEND entry of BATRD re-IPL's LARSYS). BPROFILE then loads and executes BATRD passing the system time limit to it. BATRD will either read and process the batch header cards of the next batch job in its virtual card reader, or (if no cards are there) wait until it receives an interrupt from its virtual card reader (indicating that a job has been read in) and then read and process the batch header cards. It will try to link a temporary disk and will print a message and log out if unsuccessful. After BATRD has run and returned control to BPROFILE, the batch job is run using RUN EXEC. After the batch job returns to BPROFILE, BATRD is loaded and the entry BATEND is executed. BATEND completes the accounting information and then re-IPL's the virtual machine. Control passes through the EXCOMD EXEC created before back to BPROFILE to re-start the control loop.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 057

MODULE IDENTIFICATION

Module Name: EXCOMD

Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Major executive control routine

System/Language: CMS/EXEC

Author: E. M. Rodd   Date: 08/08/72

Latest Revisor: Louis Lang   Date: 08/28/80

MODULE ABSTRACT

EXCOMD is the EXEC routine which interprets and executes all LARSYS Control Commands. Before EXCOMD begins reading and interpreting these commands, it performs certain initialization actions, including acquiring a temporary disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

EXCOMD

This CMS Executive routine interprets and executes all LARSYS Control Commands. An optional parameter, 'SAVEDISK', can be specified to eliminate the clearing of the temporary disk (see internal description below). After the user types the command "i larsys", EXCOMD is executed automatically by the modified CMS nucleus INIT module INITLARS, in the same fashion that a PROFILE EXEC is automatically executed in standard CMS. The routine is divided logically into two parts. The first performs initialization functions and the second operates as a continuous loop that reads and executes LARSYS Control Commands. This operation is described further under paragraph two below.

2. Internal Description

During initialization EXCOMD rewinds and unloads all tapes. It then releases any existing output and clears any partially read card deck. A temporary disk is then attached and logged in. This is done by using the GETDISK PROGRAM to link to one of the temporary system disks and attach it as 192. If a 192 is already attached, EXCOMD releases the disk and then accesses it. The logging of 192 is done with the ERASE option to clear the disk unless the 'SAVEDISK' parameter was specified when EXCOMD was entered. If no statistics deck exists on the temporary disk (file STATS DATA), then the D1 file is created with one record containing the characters 'EOS'. This will be used by the programs to determine that no saved statistics are on the disk. Necessary remote commands are set up for printer and punch output so they go to the proper user's site. This is done by using the alternate call WHERE STACKRES of the CMS system WHERE module. WHERE determines the physical location of a CMS user's terminal and stacks one line which may be retrieved by EXCOMD in the form &REMOTE = location. The WHERE exec is further described in LARS Program Abstract 9132. The final step of initialization is to print a file called PRIORITY NEWS if it exists.

At this point EXCOMD enters the Control Command loop by unlocking the keyboard and accepting commands. If the command is unrecognized, an error message is typed and the keyboard unlocks for another command. If a carriage return is given rather than a command, 'LARSYS' is typed, a blank line, is typed, and the keyboard unlocks for the next command.

If the command is recognized, EXCOMD takes the action described below for each possible command. After any command, control returns to the start of the loop where the next command is read.

BATCH

First interpret the BATCH command. Determine whether the control card deck has been entered via the CCINPUT command or is in the virtual card reader. If the deck resides on disk, COPY the file into BATCH DATA D1. Otherwise READ the user's data deck into the file BATCH DATA D1 and then make certain that the file BATCH DATA exists. PRINT the file BATCH DATA for user verification. Now the batch deck must be sent to the batch virtual machine. To do this, first CLOSE D to release any spooled punch output, then the virtual punch to the batch virtual SPOOL machine, issue SPOOL RUN CONT to hold punched output, punch the batch header cards and the file BATCH DATA D1 release punched output with SPOOL PUN NOCONT CLOSE and finally SPOOL D OFF. The final step is to erase the file BATCH DATA D1.

CCINPUT

CARDS	- clear the FILEDEF to unit 5. Otherwise check to be certain that the file &2 &3 exists on the user's &4-disk. If not, write an error message (E107). If so, issue a FILEDEF for unit 5 to the file using the PERM option.

CLEAR

The Clear command allows the user to clear his virtual reader, or erase a Statistics deck or the Histogram deck from his A-disk.

READER	- EXCOMD issues a CP purge reader, close reader, and ready reader commands.
HISTDECK	- EXCOMD issues the CMS erase command to erase the file from the user's A-disk.
STATDECK	- EXCOMD issues the CMS erase command to erase the file from the user's A-disk.

CMS

Clear all FILEDEFs to units 5 and 6. Then enter the CMS command environment.

MSG userid

Send the message to the CP operator or another user.

NEWS, REFERENCE, LIST

Execute the EXEC's of that name. See the individual module descriptions for details.

PRINT

TYPEWRITER option	- issue a FILEDEF (with the PERM option) of unit 6 to the console. (if any other PRINT command is given after PRINT TYPEWRITER, this console FILEDEF is cleared.)
HOLD option	- issue a SPOOL PRINTER CONT
RELEASE option	- issue a SPOOL PRINTER NOCONT CLOSE
None of the above	- check for a valid site-id and issue the appropriate REMOTE command.

QUIT

Logout from CP.

RUN

The Run command allows the user to execute a LARSYS function or a program residing on the source/text library.

LARSYS or blank	- The RUNLS EXEC file is executed.
PROGNAME ARG1 ARG2...ARGN	- The file PROGNAME EXEC or PROGNAME TEXT, residing on the source/text library, is executed.

If PROGNAME was specified, the source/text library is linked to the user and is searched first for PROGNAME EXEC, then for PROGNAME TEXT. If the EXEC file exists, it is executed using the command

```
EXEC PROGNAME ARG1 ARG2...ARGN
```

and then the source/text library is detached. If a TEXT file is found, it is executed using the commands

```
LOAD PROGNAME
START PROGNAME ARG1 ARG2...ARGN
```

In this case, the source/text library is detached before the START command is issued.

STATDECK & HISTDECK

These two commands have the same implementation except for file names. The description below uses the STATDECK file names. Each employes the three options below:

SAVE [NAME] - First check that a statistics deck (file STATS DATA) exists on the D-disk. If not write an error. Then erase any old statistics deck with that name from the A-disk. (file [name] STATDECK). Then use an assembler program called SPACE to determine if there is sufficient space on the A-disk to copy the statistics deck onto it. If there is not enough space on the A-disk, write a message, otherwise, copy the file STATS DATA D1 into [NAME] STATDECK A1. [NAME] defaults to SAVED.

STATUS - Write a message if there is not statistics deck on the A-disk. If there is a deck, issue a LISTF for the file.

USE [NAME] - First be certain that the file [NAME] STATDECK exists. If not, write a message; otherwise, copy it into the file STATS DATA D1.

3. Input Description

Not Applicable

4. Output Description

The following is a list of the message numbers used in this program. See the User's Guide for a complete description of the messages.

Error Messages		Information Messages
E 105	112	0006
E 106	E 113	0007
107	E 114	
108	115	
109	116	
110	E 129	
111	E 305	
	E 132	

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 058

MODULE IDENTIFICATION

Module Name: LIST Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: EXEC file to Implement LARSYS "LIST" Command

System/Language: CMS/EXEC

Author: Howard L. Grams Date: 10/20/72

Latest Revisor: Randy Alan Culp Date: 12/24/75

MODULE ABSTRACT

This EXEC file is called by the principal control EXEC file (EXCOMD) whenever the latter recognizes the LARSYS command "LIST". If no parameter is specified, the file LARSYS INDEX is printed. This file constitutes a reference list of all the information available from either the NEWS or REFERENCE commands. If a parameter is included, the single line from the file LARSYS INDEX that matches the specified parameter is typed to the terminal (with a suitable heading.)

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LIST is a short CMS EXEC procedure.

It either offline prints the file LARSYS INDEX, or types a single line from that file to the user's terminal. It then returns to the caller (normally EXCOMD).

2. Internal Description

Either no parameters or one parameter is accepted, as described in section 4 below.

That parameter, truncated to 8 characters, is the name of a CMS file managed by either the NEWS command or the REFERENCE command (and hence included in the file LARSYS INDEX).

When LIST EXEC is processing the 'LIST ITEMNAME' form of the command, it calls the "subroutine" module TYPEITEM, passing to it the itemname requested. TYPEITEM finds the required line in the LARSYS INDEX file and types it to the user's terminal.

3. Input Description

There are no input data files or cards.

4. Output Description

If no input parameter is included, the file LARSYS INDEX is printed using the CMS PRINT command with the CC option.

If an input parameter &1 is included, a heading is written to the terminal along with the appropriate line from the LARSYS INDEX file. If the parameter is not found, an error message is written to the terminal.

5. Supplemental Information

RUNTABLE is only a pseudo entry in LARSYS INDEX. LIST adds this entry when it is needed. This is done by use of the file RUNTABLE INDEX, which resides on the system RUNTABLE disk. This file is combined to the LARSYS INDEX file if no parameter is specified. If RUNTABLE is passed as the parameter, the RUNTABLE INDEX itself serves in place of the LARSYS INDEX file in the usage detailed above.

6. Flowchart

Not Applicable.

LARS Program Abstract 059

MODULE IDENTIFICATION

Module Name: NEWS   Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: EXEC file to implement LARSYS "NEWS" Command

System/Language: CMS/EXEC

Author: Howard L. Grams   Date: 10/25/72

Latest Revisor: Date:

MODULE ABSTRACT

The EXEC file is called by the principal control EXEC file (EXCOMD) whenever the latter recognizes the LARSYS command "NEWS". If no parameter is specified, the file SYSTEM NEWS is typed to the terminal. If a parameter is included, the file having that filename and filetype NEWS is printed.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

NEWS is a short CMS EXEC procedure. It checks its parameters, determines if the file requested is available, and if so, off-line prints it, or types it at the terminal. It then returns to the caller (normally EXCOMD).

2. Internal Description

Either no parameters or one parameter is accepted, as described in section 4 below. That parameter, truncated to 8 characters, is the name of a CMS file managed by this exec procedure. The possible parameters currently included are:

SYSTEM
LARSYS
SCHEDULE

Note: It is anticipated that from time to time other news files may be added, either temporarily or permanently. Their existence would be noted in the SYSTEM NEWS FILE and they would be listed by the LIST command. The current version of the NEWS EXEC needs no modifications to handle such additions.

3. Input Description

There are no input data files or cards.

4. Output Description

If no input parameter is included, the file SYSTEM NEWS is typed to the user's terminal using the CMS PRINTF command.

If an input parameter &1 is specified, a STATE is done to see if the file &1 NEWS exists. If so, it is PRINTed. If not, an error message is produced.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 060

MODULE IDENTIFICATION

Module Name: PBID Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Support routine for EXCOMD (punch 1st batch header card)

System/Language: CMS/ASSEMBLER

Author: Howard Grams Date: 11/13/72

Latest Revisor: Date:

MODULE ABSTRACT

PBID is a module which is called by the main LARSYS routine EXCOMD when it is processing the BATCH command. It punches the first batch header card:

BATCH ID 'USERID' 'USERNAME'

It extracts the 'USERID' and 'USERNAME' from the CP UTABLE using subroutine IDNAME and uses the CMS routine CARDIO to punch the card.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

PBID is designed to be called from a CMS EXEC file as a command, e.g.

PBID

There are no arguments.

2. Internal Description

A. Calls IDNAME subroutine to obtain USERID and USERNAME.

B. Calls CMS PUNCHC macro to punch the BATCH ID card.

3. Input Description

Not Applicable

4. Output Description

A single card is punched to the virtual card punch, e.g.

BATCH ID TEST 3. HOWARD GRAMS

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 061

MODULE IDENTIFICATION

Module Name: REFERENC Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: EXEC file to implement LARSYS "REFERENCE" Command

System/Language: CMS/EXEC

Author: Howard L. Grams Date: 09/07/72

Latest Revisor: Howard L. Grams Date: 01/26/73

MODULE ABSTRACT

This EXEC file is called by the principal control EXEC file (EXCOMD) whenever the latter recognizes the LARSYS command "REFERENCE". Depending on the parameter supplied, it produces printed listings of one or more files whose filetype is REFERENC. If the two parameters RUNTABLE NNNNNNNN(NNNNNNNN is an 8 digit run number) are given, the single line describing run NNNNNNNN from the file RUNTABLE REFERENC is typed to the terminal (with a suitable heading).

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

REFERENC is a simple CMS EXEC procedure. It checks its parameters, issues one or more OFFLINE PRINTCC commands, and returns to the caller (normally EXCOMD).

2. Internal Description

One input parameter &1 (and if &1 = RUNTABLE, a second optional parameter &2) is passed to REFERENC EXEC from its caller. That parameter, truncated to 8 characters is the name of a CMS file (filetype=REFERENC) managed by this EXEC procedure. The possible parameters currently included are:

COMMANDS	GRAPHHISTOGRAM	PRINTRESULTS
INITIALIZATION	HISTOGRAM	PUNCHSTATISTICS
CLASSIFYPOINTS	IDPRINT	SAMPLECLASSIFY
CLUSTER	IMAGEDISPLAY	SEPARABILITY
COLUMNGRAPH	LINEGRAPH	STATISTICS
COPYRESULTS	LISTRESULTS	TRANSFERDATA
DUPLICATERUN	PICTUREPRINT	RUNTABLE

When REFERENC EXEC is processing the 'REFERENCE RUNTABLE NNNNNNNN' form of the command, it calls the "subroutine" module TYPEITEM, passing to it the run number requested. TYPEITEM finds the required line in the RUNTABLE REFERENC file and types a portion of it to the user's terminal.

3. Input Description

There are no input data files or cards.

4. Output Description

For the input parameters (see section 2) COMMANDS, INITIALIZATION, "PROCESSORNAME", or RUNTABLE, one CMS file is printed using the OFFLINE PRINTCC command, e.g., the file INITIALI REFERENC is a result of the input parameter INITIALIZATION. If the input is ALL, the file ALL REFERENC is printed. It is a concatenation of all the other REFERENC files. The order is alphabetical except that COMMANDS and INITIALIZATION appear first and RUNTABLE does not appear.

If the first parameter is RUNTABLE and the second is an 8-digit number, no printed output is produced. Instead the one single line is located in the RUNTABLE REFERENC file describing that run, and it is typed at the user's terminal. (If the run is not found in the reference list, an error message is produced).

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

140

LARS Program Abstract 062

MODULE IDENTIFICATION

Module Name: RUNLS   Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Exec to load and start LARSMN

System/Language: CMS/EXEC

Author: E. M. Rodd   Date: 08/08/72

Latest Revisor: E. M. Rodd   Date: 12/12/72

MODULE ABSTRACT

RUNLS EXEC issues all FILEDEF commands and then loads the root module using the LOADMOD command. It then starts execution. After completion of the run, any tapes attached are detached.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

RUNLS

EXEC RUNLS

RUNLS is executed to issue all FILEDEF commands and start LARSMN running. RUNLS cannot be executed from CMS. It must be executed from an EXEC file.

2. Internal Description

RUNLS issues all FILEDEF commands for the LARSYS system. All FILEDEFs except those for permanent data sets are issued with the NOCHANGE option so that an experienced CMS user can have issued his own FILEDEF commands prior to execution of RUNLS. The LARSMN module is loaded via the LOADMOD command. Then the START command is used to start execution of LARSMN. When control returns to RUNLS it determines which tapes are attached by the error code from a TAPE RUNLS command. Those that are attached are detached.

3. Input Description

The LARSMN MODULE disk file is read from the N-disk.

4. Output Description

The message '?CMS RUNLS' is typed if RUNLS is executed from the CMS environment.

5. Supplemental Information

A check is made to be certain that RUNLS is being executed from an EXEC and not CMS environment. This is done because if executed from CMS the D-disk may not have been attached and the file EOSTAT INIT may not be present on the D-disk.

6. Flowchart

Not Applicable

LARS Program Abstract 063

MODULE IDENTIFICATION

Module Name: SPACE Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Determines if there is space on P-disk to copy a file

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd Date: 08/06/72

Latest Revisor: Date:

MODULE ABSTRACT

SPACE is a stand alone module to determine if there is room on the A-disk to copy a file from the D-disk. The filename and filetype are passed as parameters to SPACE.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised October 1980

1. Module Usage

SPACE is executed as a stand alone module and can thus be executed as a command from CMS or an EXEC. The command has two parameters, the filename and filetype of the file on the D-disk. The command is:

SPACE fn ft

If there is space on the A-disk to copy the file from the D-disk, R15 is returned = 0 and if not R15 is returned = 1.

2. Internal Description

SPACE determines if there is sufficient space on the A-disk to copy a file onto it from the D-disk. SPACE checks for enough space plus 15 records to leave some space on the A-disk. This is done so that the disk will not become 99% full and give the CMS disk full error. SPACE uses the CMS STATE routine to find the size of the file on the D-disk. To do this, the address of the FVS area is obtained from the SYSREF table. In the FVS table is the address of the STATE routine. The functioning of the STATE routine is in the CMS PLM. STATE puts the file information in FVS area. From there SPACE obtains the number of records in the file on the D-disk. Then SPACE calls the CMS ADTLKP routine to find out how much space remains on the A-disk. The address of the ADTLKP routine is also in the FVS area. ADTLKP (described in the CMS PLM) returns in R1 the address of the active disk table of the requested disk (in this case the A-disk). The number of records left on the A-disk is obtained from the active disk table. Then the number of records needed for the file and 15 are subtracted from the number of records left. If this is positive, R15 is returned as 0, otherwise R15 is returned as 1.

The FVS and ADT macros are used to generate DSECTs for these CMS areas. The DSECTs contain the labels used to access the desired information. FVSFSTDB contains the number of records used by the file on the D-disk (after a call to STATE has placed the directory entry in FVS). ADTLEFT in ADT contains the number of records left on the disk.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 064

MODULE IDENTIFICATION

Module Name: TYPEITEM Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Routine to find a keyword in a file and type that line to the typewriter

System/Language: CMS/ASSEMBLER

Author: Howard L. Grams Date: 10/24/72

Latest Revisor: Date:

MODULE ABSTRACT

This program module was designed to be called by the REFERENC and LIST EXEC files to enable them to type a portion of the line from a specified CMS file that has a specified keyword in a specified position. If the line is found, the specified portion of it is typed at the user's terminal. If it is not found, an appropriate error code is returned.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

This program module opens the file requested by the parameters (CMS STATE and SETUP macros), reads successive records from it, (CMS RDBUF macro) and searches for the specified keyword to be located beginning in the specified column. If a match is found, the specified number of characters are typed to the terminal (CMS TYPE macro), and the routine returns to the caller (normally LIST EXEC or REFERENC EXEC). Suitable error codes are returned if the keyword is not found, or if other errors occur.

2. Internal Description

Input Parameters:

&1,&2 - CMS filename-filetype of file to be searched. (all filemodes are considered, using the standard order of search).

&3 - A keyword-8 characters or less, assumed to be left justified and blank filled when received by the program.

&4 - The Column number of the first character of the keyword. This parameter must be given as an 8 digit hexidecimal number (i.e. right justified and zero filled).

&5 - The number of characters to be typed. If the keyword is found, the line typed will start at the keyword position (&4) and continue for this many characters. This parameter must also be an 8 digit hexidecimal number.

Possible error codes returned to caller:

0 = Normal return. The line requested was found and typed as requested.

4 = EOF was reached while reading the requested file. The keyword was not found.

8 = An error in the input parameters was detected. Either &4 or &5 could not be properly converted to a binary number.

12 = Error reading input file.

16 = The specified input file could not be found.

3. Input Description

There are not input data cards, or files (other than the one being searched).

4. Output Description

The only output is the single line typed to the terminal (described above) and an error code returned to the caller (see section 2).

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 0068

MODULE IDENTIFICATION

Module Name: TERMTEST Function Name: LARSYS EXECUTIVE

Purpose: Test 2741 Terminal and Phone Line

System/Language: CMS/ASSEMBLER

Author: Howard L. Grams Date: 02/08/73

Latest Revisor: Date:

MODULE ABSTRACT

This program performs the function requested by the LARSYS 'termtest' command. It types a line inviting the user to enter a test input line on his typewriter, then reads that line, echoes it back the number of times the user specified (or ten times if no number was specified), and then exits back to the LARSYS command environment.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

TERMTEST is an assembler main program, called by the EXCOMD routine to support the LARSYS 'termtest' command. Normally it is called with no parameters; however, it may receive one parameter. If that parameter is a one or two character number, that number will be used as the number of times the test line is to be re-typed back to the user. If no parameter is supplied, the default of 10 is used.

2. Internal Description

TERMTEST was adapted from the CMS module ECHO, which supports the CMS 'echo' command. See the description in "CMS Program Logic Manual", Form GY20-0591-1, page 155 for a description of ECHO. This description applies to TERMTEST, except that the following changes were made. The first parameter of ECHO was removed ('X' is always used), and the default for the remaining parameter was changed to 10. Also, only one input test line is accepted, and the user does not (as in ECHO) have to type 'return' to exit from TERMTEST. The start and end messages were altered and brought into accord with LARSYS standards.

3. Input Description

Terminal input only.

4. Output Description

Test output to terminal only.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

150

LARS Program Abstract 069

MODULE IDENTIFICATION

Module Name: GLARSMN Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Generates root load module

System/Language: CMS/EXEC

Author: E. M. Rodd Date: 08/20/72

Latest Revisor: E. M. Rodd Date: 11/20/72

MODULE ABSTRACT

GLARSMN generates the root load module.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GLARSMN MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

151

LARS Program Abstract 070

MODULE IDENTIFICATION

Module Name: LLARSMN

Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Loads program modules in the root

System/Language: CMS/EXEC

Author: E. M. Rodd

Date: 08/20/72

Latest Revisor: J. Etheridge

Date:

MODULE ABSTRACT

LLARSMN uses CMS LOAD and USE Commands to load TEXT files for all programs in the root.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 071

MODULE IDENTIFICATION

Module Name: GCLASUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS EXEC Routine to generate CLASUP load module

System/Language: CMS/EXEC

Author: E. M. Rodd Date: 09/11/72

Latest Revisor: T. Ransom Date:

MODULE ABSTRACT

This CMS Executive routine generates the CLASUP load module, which includes all of the program routines necessary for the Classify-points function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GCLASUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 072

MODULE IDENTIFICATION

Module Name: GCLUSUP   Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS Executive Routine to generate the CLUSUP load module

System/Language: CMS/EXEC

Author: J. B. Etheridge   Date: 11/14/72

Latest Revisor: P. Spencer   Date:

MODULE ABSTRACT

This CMS Executive Routine generates the CLUSUP load module, which includes all of the program routines necessary for the Cluster function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GCLUSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 073

MODULE IDENTIFICATION

Module Name: GGRHSUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS Executive Routine to create the GRHSUP load module

System/Language: CMS/EXEC

Author: B. K. Addessio Date: 11/16/72

Latest Revisor: T. Ransom Date:

MODULE ABSTRACT

This CMS Executive Routine generates the GRHSUP load module, which contains all of the functional program routines for the Graphhistogram, Columngraph, and Linegraph functions.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GGRHSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

155

LARS Program Abstract 074

MODULE IDENTIFICATION

Module Name: GHISSUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Generates the HISSUP load module

System/Language: CMS/EXEC

Author: L. D. England Date: 10/04/72

Latest Revisor: T. Ransom Date:

MODULE ABSTRACT

This CMS Executive generates the HISSUP load module, which includes all of the program routines necessary for the Histogram function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GHISSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

156

LARS Program Abstract 076

MODULE IDENTIFICATION

Module Name: GPICSUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Generates the PICSUP load modules

System/Language: CMS/EXEC

Author: L. D. England Date: 10/16/72

Latest Revisor: P. Alenduff Date:

MODULE ABSTRACT

This CMS Executive routine generates the PICSUP load module, which includes all of the program routines necessary for the Pictureprint function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GPICSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 077

MODULE IDENTIFICATION

Module Name: GPRISUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Generates the load module for the PRISUP load module.

System/Language: CMS/EXEC

Author: E. M. Rodd Date: 10/09/72

Latest Revisor: J. Etheridge Date:

MODULE ABSTRACT

This CMS Executive routine generates the PRISUP load module, which includes all of the program routines necessary for the Printresults function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GPRISUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 078

MODULE IDENTIFICATION

Module Name: GRESSUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Generates RESSUP load module

System/Language: CMS/EXEC

Author: S. K. Hunt Date: 11/21/72

Latest Revisor: S. K. Hunt Date: 11/22/72

MODULE ABSTRACT

This CMS Executive routine generates the RESSUP load module, which includes all of the program routines necessary for the Copyresults, Listresults, and Punchstatistics functions.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GRESSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 079

MODULE IDENTIFICATION

Module Name: GRUNSUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Executive Routine to generate the RUNSUP load module

System/Language: CMS/EXEC

Author: J. B. Etheridge Date: 10/24/72

Latest Revisor: T. Ransom Date:

MODULE ABSTRACT

This CMS executive routine generates the RUNSUP load module containing all of the program routines for the Idprint, Duplicaterun, and Transferdata routines.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GRUNSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

160

LARS Program Abstract 080

MODULE IDENTIFICATION

Module Name: GSAMSUP

Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS Executive Routine to create the SAMSUP load module

System/Language: CMS/EXEC

Author: B. K. Addessio

Date: 12/06/72

Latest Revisor: T. Ransom

Date:

MODULE ABSTRACT

This CMS Executive routine generates the SAMSUP load module, which contains all of the functional program routines necessary for the Sampleclassify function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GSAMSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 081

MODULE IDENTIFICATION

Module Name: GSEPSUP   Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS Executive Routine to generate the SEPSUP load module

System/Language: CMS/EXEC

Author: E. M. Rodd   Date: 09/11/72

Latest Revisor: T. Ransom   Date:

MODULE ABSTRACT

This CMS Executive routine generates the SEPSUP load module, which contains all of the program routines necessary for the Separability function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GSEPSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 082

MODULE IDENTIFICATION

Module Name: GSTASUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Generates the STASUP module

System/Language: CMS/EXEC

Author: E. M. Rodd Date: 11/27/72

Latest Revisor: T. Ransom Date:

MODULE ABSTRACT

This CMS executive routine generates the STASUP load module, which includes all of the program routines necessary for the Statistics function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GSTASUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 083

MODULE IDENTIFICATION

Module Name: GBIPSUP

Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS EXEC Routine to generate BIPSUP load module

System/Language: CMS/EXEC

Author:

Date:

Latest Revisor:

Date:

MODULE ABSTRACT

This CMS Executive routine generates the BIPSUP load module, which includes all of the program routines necessary for the Classify-points function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GBIPSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 084

MODULE IDENTIFICATION

Module Name: GCOMSUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS EXEC Routine to generate COMSUP load module

System/Language: CMS/EXEC

Author: Date:

Latest Revisor: Date:

MODULE ABSTRACT

This CMS Executive routine generates the COMSUP load module, which includes all of the program routines necessary for the Classify-points function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GCOMSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 085

MODULE IDENTIFICATION

Module Name: GMERSUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS EXEC Routine to generate MERSUP load module

System/Language: CMS/EXEC

Author: Date:

Latest Revisor: Date:

MODULE ABSTRACT

This CMS Executive routine generates the MERSUP load module, which includes all of the program routines necessary for the Classify-points function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GMERSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 086

MODULE IDENTIFICATION

Module Name: GRATSUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS EXEC Routine to generate RATSUP load module

System/Language: CMS/EXEC

Author: ______ Date: ______

Latest Revisor: ______ Date: ______

MODULE ABSTRACT

This CMS Executive routine generates the RATSUP load module, which includes all of the program routines necessary for the Classify-points function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GRATSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 087

MODULE IDENTIFICATION

Module Name: GSECSUP
Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS EXEC Routine to generate SECSUP load module

System/Language: CMS/EXEC

Author: Date:

Latest Revisor: Date:

MODULE ABSTRACT

This CMS Executive routine generates the SECSUP load module, which includes all of the program routines necessary for the Classify-points function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GSECSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 088

MODULE IDENTIFICATION

Module Name: GSMOSUP Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: CMS EXEC Routine to generate SMOSUP load module

System/Language: CMS/EXEC

Author: Date:

Latest Revisor: Date:

MODULE ABSTRACT

This CMS Executive routine generates the SMOSUP load module, which includes all of the program routines necessary for the Classify-points function.

The load map will be printed unless the parameter NOMAP is present. In either case a file called GSMOSUP MAP will be created on the A-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 089

MODULE IDENTIFICATION

Module Name: GENALLDV Function Name: LARSYS EXECUTIVE ROUTINES

Purpose: Generates all load modules.

System/Language: CMS/EXEC

Author: E. M. Rodd Date: 01/09/73

Latest Revisor: Paul W. Spencer Date: 02/01/74

MODULE ABSTRACT

GENALLDV generates all LARSYSDV load modules including the root. The load maps will be printed unless the parameter NOMAP is present.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 100

MODULE IDENTIFICATION

Module Name: BLOAD Function Name: LARSYS MONITOR

Purpose: Loads overlay modules

System/Language: CMS/Assembler

Author: E. M. Rodd Date: 08/01/72

Latest Revisor: Date:

MODULE ABSTRACT

BLOAD uses the CMS LOADMOD command to load overlay modules. BLOAD is designed to be called from FORTRAN.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

BLOAD

CALL BLOAD ('name   ')

Input Arguments:

name is the name of the CMS MODULE file to be loaded.

name must appear in quotes and be padded with blanks to 8 characters

2. Internal Description

BLOAD uses the CMS LOADMOD command to load the module. The command is executed via SVC 202.

3. Input Description

The MODULE file.

4. Output Description

Error 100 is written via ERPRNT if the LOADMOD command takes an error return.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 101

MODULE IDENTIFICATION

Module Name: ERMNAM Function Name: LARSYS MONITOR

Purpose: Returns name of subroutine calling ERPRNT

System/Language: CMS/Assembler

Author: E. M. Rodd Date: 07/27/72

Latest Revisor: Date:

MODULE ABSTRACT

ERMNAM is used to get the name of the subroutine calling its caller. It traces back through save areas to find the name of the subroutine calling its caller.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

ERMNAM

CALL ERMNAM (MODNAM)

Output Arguments:

MODNAM - R*8 The name of the caller of the subroutine calling ERMNAM.

2. Internal Description

ERMNAM traces back through save areas to find the name. It assumes standard FORTRAN save areas with frontwards and backwards save areas pointers.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

ERMNAM is designed for calling from ERPRNT to give the name of the subroutine generating the error though it can be used by any program.

6. Flowchart

Not Applicable

LARS Program Abstract 102

MODULE IDENTIFICATION

Module Name: ERPRNT Function Name: LARSYS MONITOR

Purpose: Error Message Printing Routine

System/Language: CMS/FORTRAN

Author: P. A. Legare Date: 07/31/72

Latest Revisor: Date:

MODULE ABSTRACT

ERPRNT is used to read error messages from the disk file, then print them on the typewriter, and also on the line printer, if desired. ERPRNT will then perform a RETURN, a RETURN1, or a STOP, as specified by the "action desired" parameter 'DISPOS'.

Entry ERPRT1 is called for a program interrupt error and includes the PSW address in the calling list.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

ERPRNT

CALL ERPRNT (ERRNUM, DISPOS,*)

Input Arguments:

ERRNUM - I*4 The error number to be written. This error number must be on the error message file (DSRN 8).

DISPOS - I*4 A disposition parameter. It is a literal in the call. If it is 'GO', ERPRNT Returns, if it is 'STOP', ERPRNT Calls RTMAIN after the error is written. If it is 'GOTO', a non-standard return is executed after writing the message. Only if 'GOTO' is specified as the second parameter is the third parameter of the calling list required.

Output Arguments:

Not Applicable

Non-Standard Return:

* - Applicable only if DISPOS = 'GOTO'

CALL ERPRT1 (ERRNUM, DISPOS, ADDRES)

Input Arguments:

ERRNUM - Same

DISPOS - Same except 'GOTO' is not permitted and will cause unpredictable results.

ADDRES - I*4 The address is taken from the PSW when a program interrupt occurred. It will be written with Z8 format.

Output Arguments:

Not Applicable

2. Internal Description

ERPRNT is called when an error has been detected and it is desired to print out an error message. It searches the error message disk file for an error number which matches the input parameter ERRNUM; calls ERMNAM to get the name of the module which has called ERPRNT; and prints the desired message (including the module name), at both the typewriter and line printer if desired. (if the 80th character of the message, PNTCOD, is = N, the message will be typed only). If the desired message number is not found on the disk file, an error message reading "E101 program error; contact programming staff" (MODNAM) is printed to indicate this and execution is stopped.

The action desired to be taken after the message is printed is determined by checking the input parameter DISPOS.

If DISPOS =	'GObb' or 'Go'	a Return is performed
" "	"	'GOTO'	a Return 1 is performed
" "	"	'STOP'	a Call to FRTMN is made

If DISPOS is not one of the above, it is invalid and an error message is printed indicating this after the desired message has been printed. This message reads "E130 program error. Contact programming staff" (MODNAM), and execution is stopped.

The entry point ERPRT1 is called if certain program interrupts were detected. It will perform all of the above functions and will also print out the old PSW address.

3. Input Description

Disk DATA SET - ERROR MESSAGE
Each record is 80 bytes long and will have the following fields

BYTES

1-5 Error number (right justified)

6-77 Text of the error message

78-79 Not used

80 Indicator which is equal to 'N' if message is not to be printed on the line printer. If other than 'N' message is printed at both the typewriter and line printer.

4. Output Description

Each error message will have the following format:

EnnnbbTEXT OF THE MESSAGE (72 CHARACTERS MAX)bbb (MODNAM) where nnn represents the error message number and MODNAM represents the module which detected the error and called ERPRNT.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 103

MODULE IDENTIFICATION

Module Name: ERRINT Function Name: LARSYS MONITOR

Purpose: Program Interrupt Handler

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd Date: 07/27/72

Latest Revisor: Steve Wilkinson Date: 04/26/74

MODULE ABSTRACT

ERRINT is the program interrupt handler. ERRINT intercepts program interrupts and either gives control to FORTRAN error processing or does the error processing. The use of ERRINT avoids CMS DEBUG being entered. Entry ERINT1 is the entry actually used to process interrupts.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised September 1974

1. Module Usage

Neither entry has any parameters; see section 2 below.

2. Internal Description

ERRINT consists of two entry points. ERRINT is called to issue an OS SPIE macro to intercept all program interrupts except 8 (fixed-point overflow), 10 (decimal overflow) and 14 (significance exception) which are masked off. ERRINT saves the address of the FORTRAN program interrupt handler so that FORTRAN can process interrupt 13 (exponent underflow) as it normally does. The SPIE macro specifies entry point ERINT1 as the interrupt handler.

ERINT1 receives control when an interrupt occurs. ERINT1 determines if the interrupt is one normally handled by FORTRAN. If so, the FORTRAN interrupt handler is called. This means that FORTRAN writes the message and counts number of errors, etc. Otherwise, ERINT1 constructs a call to ERPRT1 passing as the error number 900+ program interrupt number. Then ERRTRA is called and then EXIT is called to terminate execution.

ERRINT uses the system generated PIE (Program Interrupt Element) and PICA (Program Interrupt Control Area).

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

See the section "Program Interrupt Processing" of the OS supervisor services manual and the section "Program Interruptions" of the CMS. FORTRAN itself issues an SPIE as execution of a load module begins (in the 1HCFCOMH module). See the "OS 360 FORTRAN IV (G) Compiler Program Logic Manual" for more details. At that call, interrupts 8, 10, and 14 are disabled. It is not possible to enable them (as would be desirable) because some compiler-generated code (subscript maniplation) can cause overflows even though no error has in fact occurred.

6. Flowchart

Not Applicable

180

LARS Program Abstract 104

MODULE IDENTIFICATION

Module Name: GLOCOM Function Name: LARSYS MONITOR

Purpose: Block data for GLOCOM

System/Language: CMS/FORTRAN

Author: E. M. Rodd Date: 08/09/72

Latest Revisor: Date:

MODULE ABSTRACT

This is the BLOCK DATA subroutine for the GLOCOM common block.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

181

LARS Program Abstract 105

MODULE IDENTIFICATION

Module Name: LARSMN Function Name: LARSYS MONITOR

Purpose: FORTRAN Main Program for LARSYS

System/Language: CMS/FORTRAN

Author: E. M. Rodd Date: 08/08/72

Latest Revisor: Jeanne Etheridge Date: 07/14/75

MODULE ABSTRACT

This is the FORTRAN main program for LARSYS. It does some overall initialization and then reads Function Selector Cards and System Initialization Cards and carries out the requests. System Initialization Cards are executed in LARSMN and Function Selector cards cause loading and executing of overlay modules.

1. Module Usage

LARSMN is the monitor control routine. It receives control from EXCOMD when a "run larsys" control command is issued and proceeds to read the user's input deck, and interpret LARSYS Initialization Functions (Cards) and LARSYS Function Selector Cards. It performs all initialization actions itself, and loads LARSYS Functional Load Modules and passes control to them when Function Selector Cards are encountered.

2. Internal Description

The code in LARSMN can be logically divided into six sections:

1) One time only initialization is executed. This consists of a call to ERPRNT and ERRSET to initialize the interception of program interrupts and to cause execution to stop after one interrupt has occurred. A call to the subroutine IDNAME to get the userid and name into the heading is executed.

2) Reset initialization is then executed. This initialization is performed on entry to LARSMN and anytime the -RESET system initialization card is encountered. This consists of putting the time and date into the header, clearing the user runtable, putting default titles into the header and setting the control card checkout flag off.

3) LARSMN goes into a loop (starting at statement 200) of reading control cards and executing the requested function. This consists of using CTLWRD to read and interpret the card. The card may be one of the following three forms: -ABC, where ABC is the name of a standard initialization function; *DEF where DEF is the name of a standard processing function; or *XYZ where XYZ is the name of a non-standard (i.e., experimental) processing function. An array contains all the standard function names and one additional entry of '*bbb' to handle any non-standard processing function. If an initialization function is requested, a branch is made to statement 500 as described in 5). Otherwise, PACCT is called to punch an account card. If a standard processing function is requested, UNMNT is called, followed by a branch to statements 230-500, as described in 4). If a non-standard function is requested, STATE is called to determine whether or not the file XYZSUP MODULE exists on any disk. If not, the user is told that the keyword is invalid and is requested to retype the card. Control passes to statement 200 again. If XYZSUP MODULE exists, a branch is made to statement 330, as described in 4).

4) The code from statement 230 - 500 loads an overlay module and passes control to it. Upon return from the processor any scratch disk data sets used by the module are erased. Also, a check is made on the flag in GLOCOM which tells whether the END card for the function has been read. If it has not, LARSMN reads from the reader until the END card has been read.

5) The code from statement 500 - 800 executes System Initialization Cards. These are all executed in-line except for -RUNTABLE which is executed by a call to the subroutine RTBSUP.

6) The code from statement 800 to the end is executed when EOF is read on the input data set. (LARSMN learned of this occurance when CTLWRD returned ERRCOR =2). Message I004 is printed and typed and then LREND is called to terminate execution.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplementary Information

When processors are loaded and called, this is done by first calling BLOAD which will use CMS LOADMOD to load the module. Then a call is made to PROCES. When the root module was generated, a dummy routine called PROCES was loaded at the very end of the program. But in the GENMOD to actually generate the module, PROCES was not included. Then the processor modules are loaded starting at the address of PROCES. Thus the call to PROCES will pass control to the first byte of the processor module.

6. Flowchart

Not Applicable

LARS Program Abstract 106

MODULE IDENTIFICATION

Module Name: PROCES Function Name: LARSYS MONITOR

Purpose: Dummy to resolve a reference

System/Language: CMS/FORTRAN

Author: E. M. Rodd Date: 08/20/72

Latest Revisor: Date:

MODULE ABSTRACT

For the usage of PROCES, see Section 3.5 of the System Manual under "Generating Functional Load Module".

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

185

LARS Program Abstract 107

MODULE IDENTIFICATION

Module Name: RTBSUP Function Name: LARSYS MONITOR

Purpose: Sets up user runtable

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Paul W. Spencer Date: 02/01/74

MODULE ABSTRACT

RTBSUP is called by LARSMN to execute the RUNTABLE function. RTBSUP reads the data cards for RUNTABLE and generates and prints the user runtable.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

RTBSUP

Input Arguments:

RTBSUP is called to read the - RUNTABLE control cards and interpret them.

The array RUNTAB in GLOCOM is created. IMARK is set to the number of user entries in the runtable. Since IMARK is initialized in the BLOCK DATA for GLOCOM, successive executions of RUNTABLE within the same LARSYS run will add to the runtable. If the user attempts to add an eleventh entry to the runtable an error is written and the entry is ignored. Also if an entry is already in the user runtable (that is the run number), an error is written and the entry ignored.

Output Arguments:

Not Applicable

Non-standard Returns:

Not Applicable

2. Internal Description

RTBSUP reads the RUNTABLE data cards and creates the user runtable. It uses standard LARSYS support routines to read and interpret the data cards and to print error messages for control card errors.

3. Input Description

The RUNTABLE system initialization function data cards are read via CTLWRD.

4. Output Description

I0010 is typed.
Errors 133-140 are written via ERPRNT.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 108

MODULE IDENTIFICATION

Module Name: UNIDAT Function Name: LARSYS MONITOR ROUTINES

Purpose: Common Block for Universal tapes

System/Language: CMS/FORTRAN

Author: J. Buis Date:

Latest Revisor: Date:

MODULE ABSTRACT

Common Block to handle Universal Multispectral Image Storage tapes.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 120

MODULE IDENTIFICATION

Module Name: RUNERR Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Interprets errors from GADRUN and GADLIN

System/Language: CMS/FORTRAN

Author: E. M. Rodd Date: 08/28/72

Latest Revisor: P. D. Alenduff Date: 01/28/75

MODULE ABSTRACT

RUNERR (or entry LINERR) is designed to be called after GADRUN (or GADLIN) have returned with an error condition. RUNERR will interpret the error code and print the appropriate error via call to ERPRNT, print any supplemental information, and stop on serious errors or return on recoverable errors.

1. Module Usage

RUNERR

CALL RUNERR (ERROR, RUN)

Input Arguments:

ERROR - I*4, The error code from GADRUN. See the module document for GADRUN for the interpretation of these codes.

RUN - I*4, The run number used in the call to GADRUN.

Output Arguments:

Not Applicable

Non-standard Returns:

Not Applicable

LINERR

CALL LINERR (ERROR, CURLIN,*,*,*)

Input Arguments:

ERROR - I*4, The error returned from GADLIN. See the module document for GADLIN for the interpretation of these codes.

CURLIN - I*2, The line number used in the call to GADLIN.

Output Arguments:

Not Applicable

Non-standard Returns:

See Section IV.

Neither RUNERR nor LINERR will function correctly for values of ERROR $\leq$ 0. If GADRUN or GADLIN returned error code = 0, RUNERR and LINERR should not be called.

2. Internal Description

Not Applicable

3. Input Description

RUNERR interprets errors from GADRUN and entry LINERR interprets errors from GADLIN. Error messages are written via ERPRNT though after some messages, supplemental information in the form of a number is written. This number is an error code for use by the programming staff or a run number or line number. If this information is printed, the error message indicates that it is to follow the error message.

4. Output Description

RUNERR writes error messages via ERPRNT and for some errors supplemental information. Below are listed the value of ERROR, the entry point, the error number in the call to ERPRNT, any supplemental information, and the disposition.

ERROR	ENTRY	NUMBER	SUPPLEMENTAL INFORMATION	DISP
1-5	RUNERR	310	error code	execution terminated
6	RUNERR	311	none	execution terminated
7	RUNERR	312	run number	execution terminated
8	RUNERR	313	run number	execution terminated
1	LINERR	none	none	RETURN 1
2-5	LINERR	314	error code	RETURN 2
6	LINERR	315	none	execution terminated
7-14	LINERR	316	error code	execution terminated
15,17	LINERR	317	line number	RETURN 3
16	LINERR	318	line number	RETURN 3

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 121

MODULE IDENTIFICATION

Module Name: UNMNT　Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Detaches tapes

System/Language: CMS/FORTRAN

Author: E. M. Rodd　Date: 08/08/72

Latest Revisor: P.D. Alenduff　Date: 02/20/75

MODULE ABSTRACT

UNMNT is used to detach all tapes not needed by the function to be executed.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

192

1. Module Usage

UNMNT

CALL UNMNT (CODE)

Input Arguments:

CODE - I*4, This is a code representing the next function to be executed. The code references a list called LIST in subroutine LARSMN.

Output Arguments:

Not Applicable

UNMNT will

UNMNT will detach any tape drives currently attached which are not needed by the function indicated by CODE. It detaches the tapes and resets the appropriate status words in GLOCOM. The status words latered are:

TPSTAT SUBSCRIPT	Tape	Status word	Value which indicates no tape attached
1	Results	MAPTAP	-1
2	Separability scratch	SEPSCR	-1
3	Duplicaterun Input	SUPIN	-1
4	MIST	DATSAV	0 (DASTAT also set to 0)
5	Copyresults output	COPSER	-1
6	Transferdata output	TRAOUT	-1

2. Internal Description

UNMNT contains a table (TAPUSE) which indicates which functions use which unit numbers (i.e., 180, 181, or 182). TAPUSE (I,J) = .true. if function I uses tape unit J. CMSFNC is used to implement the QREAL command for each of the device numbers and determine if the device is attached. If the tape drive is not attached, all status words (in GLOCOM) associated with the drives are reset.

MAPSAV, SEPSCR, and DUPIN are all used on virtual device address 181. DASTAT and COPSER are both used with device 182, and TRAOUT is used in connection with 183.

If the device is attached, TAPUSE (I,J) is referenced to determine if the drive is needed. The drive is detached if not needed. If the next function requires the drive, the last function used in LARSYS is used to pass the tape number of the currently mounted tape to the status word of the requested function. The current function number is then stored for the next reference.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 122

MODULE IDENTIFICATION

Module Name: BONDSU Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Field boundary storage or deletion on disk

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: L. D. England Date: 10/23/72

MODULE ABSTRACT

BONDSU will erase the FIELD BNDRIES disk file if DELETE is specified. If STORE is specified, existing records of FIELD BNDRIES are read and new records added by Field Description Cards. A summary list of new boundaries is printed.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

BONDSU

CALL BONSU(MODE)

Input Arguments:

MODE - INTEGER*2 MODE(8); A general purpose array for communicating user requests. For BONDSU, the following are used:

MODE (4) = 1 if delete old boundaries
MODE (5) = 1 if store more boundaries

BONDSU decides which of three actions to take based on the values in the MODE array. If neither MODE(4) or MODE(5) is 1, the module simply exits. For MODE(4)= 1 (BOUNDARY DELETE), the FIELD BNDRIES disk file is erased by a call to ERASE. For MODE(5) = 1 (BOUNDARY STORE), all records of FIELD BNDRIES are read and then the Field Description Cards are read and new records are added to FIELD BNDRIES. A printout is produced that itemizes the boundaries added to the file.

2. Internal Description

The internal operation of BONDSU follows closely the description on "Module Usage". Additionally, when MODE(5) = 1, LAREAD is called to actually read the Field Description Cards and GETIME is used to get the time of day to be inserted in the heading of the printout for the new boundary list. The COMMONS GLOCOM and PICCOM are required.

3. Input Description

BONDSU reads data from the disk file FIELD BNDRIES using the symbolic DSRN FLDBND which is usually DSRN 19. LAREAD is called to actually read the Field Description Cards. Layouts for these files are in the LARSYS System Manual.

4. Output Description

BONDSU writes data records to the FIELD BNDRIES disk file. In addition, the "Field Cards added to Boundary Storage" listing is produced on the printer with symbolic DSRN PRNTR. No messages are written by BONDSU. Record layouts are contained in the LARSYS System Manual.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 123

MODULE IDENTIFICATION

Module Name: CLSCHK Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Checks validity of input data

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Paul W. Spencer Date: 02/01/74

MODULE ABSTRACT

CLSCHK checks validity of pooled classes. Entry GRPCHK checks validity of grouping, Entry FETCHK checks for existence of channels in the statistics.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised September 1974

1. Module Usage

CLSCHK

```
CALL CLSCHK (CLSDES, NOCLS, NOPOOL, POLPTR, POLSTK,
             POLNAM, POLNM1, STKPTR, NO)
```

Input Arguments:

CLSDES - I*4 CLSDES(1,I) = first four characters of the name of class I.

CLSDES(2,I) = second four characters.

NOCLS - I*4, Number of classes in the statistics.

NOPOOL - I*4, Number of pooled classes. If no pooling was requested, NOPOOL = 0.

POLPTR - I*2 POLPTR(1,I) = number of classes in pool I.

POLPTR(2,I) = index in POLSTK of first class in pool I.

POLSTK - I*2, List of classes grouped by pool.

POLNAM - I*4 POLNAM(1,I) = first four characters of name of pool I.

POLNAM (2,I) = second four characters.

POLNM1 - L*1 POLNM1 (I) = .TRUE. if POLNAM contains a group name, otherwise .FALSE.

Note: If NOPOOL is input = 0, the contents of POLPTR and POLSTK and POLNAM are irrelevant.

Output Arguments:

POLPTR - Same as input unless POLSTK contained a class not in the statistics (i.e. a class number in POLSTK was greater than NOCLS). In this case the entry in POLPTR concerning the pool containing that class are recomputed. Also if NOPOOL was input as = 0 indicating that no pooling was requested, then POLPTR is computed by CLSCHK.

POLSTK - Same as input. Recomputed under the same conditions as POLPTR.

POLNAM - Same as input. Also altered under the same conditions as POLPTR. Also names for pools of only one class are taken from CLSDES and placed in POLNAM.

NOPOOL - Same as input. If any pools had no valid classes, NOPOOL will be reduced accordingly. If NOPOOL = 0 on entry, NOPOOL is output as NOCLS.

STKPTR - I*4, Same as NO.

NO - I*4, Number of entries in POLSTK.

CLSCHK is called after all pooling of classes has been performed to check that all class numbers are valid. Also if no pooling was requested the arrays are set up correctly for all single class pools.

GRPCHK

CALL GRPCHK (POLNAM, NOPOOL, GRPSTK, GRPNAM, NOGRPS)

Input Arguments:

POLNAM - I*4, Same as in CLSCHK. This is the POLNAM array left from any pooling of classes before grouping. If no class pooling preceeded grouping, POLNAM is just a list of class names.

NOPOOL - I*4, Number of pooled classes before grouping.

GRPSTK - I*2 GRPSTK(I) = group number of class I (pooled class I).

GRPNAM - I*4 GRPNAM(1,I) = first four characters of name of group containing class I. GRPNAM(2,I) = second four characters.

NOGRPS - I*4, Number of groups already noted in GRPSTK and GRPNAM. This does not include single class groups which were not included on a GROUP control card.

Output Arguments:

NOGRPS, GRPSTK and GRPNAM are updated to reflect single class groups. These are identified by GRPSTK(I) = 0 on input for such classes.

GRPCHK is called to update NOGRPS, GRPSTK and GRPNAM as noted. Also, all elements of GRPSTK above NOPOOL should be = 0 and if they are not, I0014 is typed and printed indicating that the pool is not present.

FETCHK

```
CALL FETCHK (MAXCHA, FETVEC, NOFEAT, NOFET3, FETVC3,
             CSEL3, CSEL)
```

Input Arguments:

MAXCHA - I*4, Maximum number of channels permitted. Any channel number greater than this value is invalid.

FETVEC - I*2 FETVEC(I) = channel number of I'th channel in the statistics.

NOFEAT - I*4, Number of channels in the statistics.

NOFET3 - I*4, Number of channels requested and present in FETVC3.

FETVC3 - I*2 FETVC3(I) = channel number of I'th channel requested for this function.

CSEL3 - I*2 CSEL3(I) = calibration code for channel I if channel I was selected, otherwise = 0.

CSEL - I*2, Same as CSEL3 except for channels in the statistics.

Output Arguments:

If any channels were requested which are not in the statistics, a message will be written indicating the channel will be ignored. In this case, CSEL3, FETVC3 and NOFET3 will be recomputed.

FETCHK is called to check that all requested channels are in the statistics. Also FETCHK checks to see that the calibration code specified is the same as in the statistics. If not a message is written.

2. Internal Description

Not Applicable

3. Input Description

Not Applicable

4. Output Description

I0013 if a class was specified in a pool and the class is not in the statistics. Used by CLSCHK.

I0014 if a group is specified as containing a pool not present. Used by GRPCHK.

I0015 if a channel was requested which is not in the statistics. Used by FETCHK.

I0016 if any channel has a different calibration code in the request than in the statistics. Used by FETCHK.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 124

MODULE IDENTIFICATION

Module Name: DECCLS Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Decodes part of the CLASSES or GROUP card.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P. D. Alenduff Date: 08/15/74

MODULE ABSTRACT

DECCLS decodes the portion of CLASSES and GROUP cards which is enclosed in parentheses and follows the class or group name.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

DECCLS

CALL DECCLS (CARD, COL, GRPNUM, IWORK, CNT,*)

Input Arguments:

CARD - L*1, Eighty byte input buffer containing card image of CLASSES or GROUP card.

COL - I*4, Column number on card to start decoding (this should be the column number of the left parenthesis following the group name).

Output Arguments:

COL - The column number of the comma following the right parenthesis or if there are no more entries on the card, returned with a value of 72.

GRPNUM - I*4, The group number (or pooled class number). In the form:

CLASSES WHEAT (2/4, 5)

the group number is 2.

IWORK - I*4, Sixty fullword buffer to be returned with the class numbers in the group. In the above example, IWORK would be returned with IWORK(1) = 4 and IWORK(2) = 5.

CNT - I*4, The number of classes in the group. In the above example CNT would be returned as 2.

Non-standard Return:

A RETURN 1 is executed if an error is detected. There are six possible error conditions:

1. Any non-numeric is found between commas
2. More than one group number is present (i.e. two numbers separated by a comma precede the slash)
3. No group number is present (i.e. a right parenthesis precedes the slash)

4. The first class number is missing (i.e. comma preceded by a slash)

5. The last class number is missing (i.e. slash preceded by a comma)

6. Any class number is missing (i.e. two consecutive commas)

DECCLS must be called once for each group on the card.

2. Internal Description

FORTRAN byte manipulation using L*1 buffers.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 125

MODULE IDENTIFICATION

Module Name: GRHIST Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Graphs histogram data

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. K. Addessio Date: 09/25/72

MODULE ABSTRACT

This subroutine reads histogram information from the program direct access storage set and generates graphs of that data.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GRHIST

CALL GRHIST (NCR,CHAN)

Input Arguments:

NCR - INTEGER*4, number of channels requested as defined in GRHRDR by a call to CHANEL.

CHAN - INTEGER*2 array (30), selected channels to be processed as defined in GRHRDR by a call to CHANEL.

Output Arguments:

Not Applicable

Non-standard Returns:

Not Applicable

For each channel requested, GRHIST reads the stored histogram data from a direct access data set. This data set is previously generated by the HISTD program module as used by the HISTOGRAM processing function.

2. Internal Description

GRHIST can best be described as performing two specific tasks each of which is dependent upon the value of NCR (number of channels requested). Task 1, the data acquisition phase, is the direct input to task 2 or the histogram graphing phase of the program module.

Task 1 is performed by reading histogram information from a direct access data set previously generated by the HISTOGRAM processing function.

Task 2 is performed by first printing the header information including the spectral band definition, run number, calibration code, lines, columns, and total number of samples. At this point the scales are set up, the requested histograms are calculated and graphed and control is returned to GRHSUP.

If there is no histogram information stored for a particular channel, the channel is identified and a message is printed to that effect; processing continues.

CALL TSTREQ checks for operator intervention allowing for termination due to a STOP command. Control is transferred to GRHSUP.

Subroutines called by GRHIST:

GETIME
TSTREQ

Commons used in GRHIST:

GLOCOM

3. Input Description

File name - HISTO DATA
DSRN - HDATA
Device type - Direct access disk data set
Usage - Input
Description - Histogram data

4. Output Description

File name - Header, scale and histogram information
DSRN - PRNTR
Device type - Printer
Usage - Output for any GRAPHHISTOGRAM job.
Description - For each channel requested:
1) header information including channel number, spectral band, run number, calibration code, lines, columns, number of points represented by *, and the total number of samples.
2) graph of the histogram with the appropriate scales.

File name - Information messages
DSRN - PRNTR and TYPEWR
Device type - Printer and typewriter
Usage - Output
Description - Message numbers are listed below, for text of message see User's Manual.

INFORMATIONAL	MESSAGES
023	
260	

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 126

MODULE IDENTIFICATION

Module Name: GRPSCN Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Interprets GROUP and CLASSES cards

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P. D. Alenduff Date: 04/11/75

MODULE ABSTRACT

GRPSCN interprets the GROUP card. Entry POLSCN interprets the CLASSES card.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GRPSCN

CALL GRPSCN (GRPNAM, GRPSTK, NOGRPS, LSTGRP, COL, CARD,*)

Input Arguments:

CARD - I*4, 20 fullword buffer containing card image of GROUP card.

COL - I*4, Column number preceding first character of first group name on the card.

LSTGRP - I*4, Group number of the last group processed, initially 0.

NOGRPS - I*4, Number of groups previously processed (on earlier GROUP cards).

GRPSTK - I*2, GRPSTK (I) = group number of class I. This array must reflect any earlier GROUP cards. If the call to GRPSCN is for the first GROUP card, GRPSTK (I) must = 0 for all I (I=1, 60).

Output Arguments:

COL - I*4, COL = 72 unless RETURN 1 is executed in which case COL points somewhere near the field in error.

GRPNAM - I*4, GRPNAM (1,I) = first four characters of the name of the group containing class I. GRPNAM (2,I) = second four characters.

GRPSTK - Same as input except updated to reflect this GROUP card.

NOGRPS - Same as input except updated to reflect this GROUP card.

Non-standard return:

RETURN 1 is executed if 1) the card contains no left parenthesis or 2) an error return was received from the call to DECCLS or 3) if a class appears which has already been assigned to a group or 4) a class number greater than 60 appears. GRPSCN is called to interpret each GROUP card. The effect is cumulative, so that many GROUP cards can be added together.

POLSCN

CALL POLSCN (POLNAM, POLPTR, POLSTK, POLNM1, NOPOOL,
STKPTR, COL, CARD,*)

Input Arguments:

CARD - I*4, Same as GRPSCN

COL - Same as GRPSCN

POLPTR - I*2, POLPTR(1,I) = the number of classes in pool I. POLPTR(2,I) = index of first element in POLSTK which refers to pool I. The first row of POLPTR must have valid data when entering POLSCN. If this call to POLSCN is for the first CLASSES card, POLPTR(1,I) must +0 for I=1,60)

NOPOOL - I*4, Number of pools on previous CLASSES CARDS.

STKPTR - I*4, Pointer to the next free element in POLSTK.

Output Arguments:

COL - COL = 72 for a normal return. If RETURN1 is executed, COL points somewhere near the error.

POLNAM - I*4, POLNAM(1,I) = First four characters of name of class I. POLNAM (2,I) = second four characters. If the form of the CLASSES card

CLASSES i,j,k,l...

is used, POLNAM will be unchanged.

POLPTR - Same as input except updated to reflect this CLASSES card.

POLSTK - I*2, POLSTK is a list of class numbers grouped by pool. Within the list of class numbers for a given pool, the smallest class number is first.

POLNM1 - L*1 POLNM1(I) = .TRUE. if pool I has a name in POLNAM. Otherwise POLNM1(I) = .FALSE.

NOPOOL - Same as input except undated.

STKPTR - Same as input except undated.

Non-standard return:

RETURN1 is executed if 1) the card is not the simple form CLASSES card and does not contain a left parenthesis or 2) an error return was received from DECCLS or 3) if a class number appears which already has an entry in POLPTR(1,I). POLSCN is called to interpret each CLASSES card. The effect is cumulative so that many CLASSES cards may be used.

2. Internal Description

POLSCN distinguishes between the simple form of CLASSES card and the full form by first calling IVAL. If it is the simple form IVAL returns the class numbers. If the card is the full form, IVAL will take a non-standard return.

3. Input Description

Not Applicable

4. Output Description

Error message 458 is written via ERPRNT.

5. Supplemental Information

See the control card dictionary for the description of GROUP and CLASSES cards.

6. Flowchart

Not Applicable

LARS Program Abstract 127

MODULE IDENTIFICATION

Module Name: HEADER

Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Prints run information header

System/Language: CMS/FORTRAN

Author:

Date:

Latest Revisor: P. D. Alenduff

Date: 04/26/74

MODULE ABSTRACT

Prints run information header from the Multispectral Image Storage Tape.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised September 1974

1. Module Usage

HEADER

CALL HEADER (NOFET3,FETVC3,CSEL3,CSET3,ID,FRQCAL)

Input Arguments:

NOFET3 - INTEGER*4, Number of features used.

FETVC3 - INTEGER*2 (30), Vector containing the classification channels chosen.

CSEL3 - INTEGER*2 (30), Calibration codes for the classification channels.

CSET3 - REAL*4(3,30), Vector calibration values for each classification channel.

ID - INTEGER*4(200), Array for storing identification of aircraft data set.

FRQCAL - REAL*4(5,30), Equivalenced to ID(51,200).

Output Arguments:

Not Applicable

HEADID

CALL HEADID (ID,UNIT)

Input Arguments:

ID - Same as in Header.

UNIT - INTEGER*4, Unit to which output is directed.

Output Arguments:

Not Applicable

2. Internal Description

HEADER first prints the ID information. It then prints the title 'Channels Used' and writes out the associated channels and calibration values. Control returns to the calling program.

HEADID, an optional entry point, prints out only the ID information. Control returns to the calling program.

No COMMON is used in this subroutine thus making it available for general use.

3. Input Description

Not Applicable

4. Output Description

File name	- Run and channel information
DSRN	- PRNTR
Device type	- Printer
Usage	- Output
Description	- Run information identifies the run number, flight line, data storage tape and file, reformatting date, date, time altitude and ground heading. The channel information identifies the channel, spectral band and calibration code.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 128

MODULE IDENTIFICATION

Module Name: HISTD   Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Histograms real data values into a 100 bin array

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: L. D. England   Date: 10/04/72

MODULE ABSTRACT

HISTD calculates a 100 bin histogram for a specified block of data in each selected channel of the multispectral image storage tape. The results for each channel calculated are then stored in the HISTO DATA disk file and, optionally, punched into cards.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

HISTD

CALL HISTD (RDATA, HISTA, BRDATA, NCR, NSD, MODE, CSEL, CSET, CHAN, BLOCK, &XXX)

Input Arguments:

RDATA - REAL*4 RDATA(NSD, NCR); This is a real, variable dimension array that HISTD uses to receive data from GADLIN

HISTA - INTEGER*2 HISTA(100, NCR); This array is used to store the actual results of the histogramming.

BRDATA - INTEGER*4 BRDATA(N); This variable dimension array is used by HISTD to provide a data buffer for GADLIN. N should allow for NCR*ID(6) bytes.

NCR - INTEGER*4; The total number of channels requested.

NSD - INTEGER*4; The number of samples dimensioned in RDATA.

MODE - INTEGER*2 MODE(8); A general purpose array for communicating user requests. For histogramming only the following are defined:

MODE(1) = 1 if accumulating histograms
MODE(2) = 1 if punching histograms

CSEL - INTEGER*2 CSEL(30); The channel select vector as defined for GADLIN and CHANEL.

CSET - REAL*4 CSET(3,30); A thirty by three array containing the calibration values (CØ, C1, C2) for each of the channels as required for GADLIN.

CHAN - INTEGER*2 CHAN(30); An array containing the numbers of the channels selected in ascending order as provided by the module CHANEL.

BLOCK - INTEGER*2 BLOCK(8); An array which defines the multispectral data selected for processing.

BLOCK(1) = Initial line of data
BLOCK(2) = Last line of data
BLOCK(3) = Line interval
BLOCK(4) = Current line requested
BLOCK(5) = Initial sample of data
BLOCK(6) = Last sample of data
BLOCK(7) = Sample interval

Output Arguments:

Not Applicable

Non-standard Returns:

&XXX - Return1 will be used when unrecoverable errors are detected.

HISTD provides its results via the HISTO DATA disk file or a punched card deck rather than with output arguments. RDATA, HISTA, and BRDATA are arrays passed to HISTD for its work space so that the caller can provide the required main storage rather than having the storage allocated within HISTD. The other parameters define the data to be histogrammed and are usually defined by a reader routine such as HISRDR, PICRDR or IMARDR from control card input.

Upon entry, if the caller has MODE(1) = 1 to indicate "accumulate histograms", HISTD will read HISTO DATA and use this data as part of the histogramming process. Otherwise, any previous histograms are ignored. The results are always stored on disk. HISTD will print summary statistics for the histogram for each channel and will, optionally, punch the histogram data deck that can be used in other LARSYS functions.

2. Internal Description

TSTREQ is first called to initialize the TSTREQ parameter. Then a WRITE to an extra record in the HISTO DATA file is performed. This is necessary so as to prevent a Read being issued when no file exists as could happen if "accumulate histograms" was requested, but no previous histograms are available.

If histograms are to be accumulated, the records (2 per channel) for the selected channels are read. Otherwise, the necessary variables are initialized to zeroes. The printer heading, as stored in GLOCOM COMMON, is printed using GETIME to obtain the time of day.

GADLIN (with LINERR to handle any errors) is used to obtain the requested multispectral image data from tape. For channels not previously histogrammed, RLCP (an entry in GADLIN) is called to obtain calibration information. At this point, the requested data for each channel is histogrammed into 100 bins. TSTREQ is called after each line has been processed to determine if the user wants to "STOP".

After the histogramming is complete, the histogram statistics for a channel are calculated, the second of two disk records for that channel is written, and, optionally, 10 cards are punched. This proceeding sequence is repeated for each channel histogrammed. The statistics for all channels histogrammed is now printed and the first disk record for each channel is written. Control is now returned to the caller.

ERPRNT is used to process error messages after which, a RETURN1 is executed.

3. Input Description

The following table defines the types of I/O performed by HISTD:

File Name	Symbolic DSRN	Type	Usage	Description
HISTO DATA	HDATA	Disk	I/O	Histogram data file
-	DATAPE	Tape	Input	Multispectral Image Storage Tape (via GADLIN)
-	TYPEWR	Type-writer	Output	For information messages
-	PRNTR	Printer	Output	Histogram statistics by channel
-	PNCH	Punch	Output	Histogram deck

Detailed layouts of the contents of the above files are described in the LARSYS System Manual.

4. Output Description

The four types of output are described in section 3. The information messages that can be produced are 23, 82, 84, 85 and 86. The following error messages are processed by ERPRINT: 471, 472, and 473. The text for these messages is defined in the LARSYS User's Guide.

5. Supplemental Information

The histogramming algorithm creates a 100 bin histogram for data from the MIST tape. The location of the bin edges and the size of the bins are determined dynamically based on the data itself. This allows smooth histograms with maximum resolution to be obtained as opposed to allocating the entire data range into 100 equal-size bins. The number of bins was originally set at 100 in order to conserve main storage.

The data values on the tape may vary between 0 and 255 and are integers. However, once the data has been calibrated, the irradiance becomes a REAL number and can have any value. (For more information concerning calibration, see LARS Information Note 071069.) The basic equation for calibration is:

$$\text{Rdata} = A*\text{Idata} + B$$

A and B may change slightly from line to line but are assumed constant within a line.

The algorithm initially assumes a bin size of A and a lower limit of A*14.5 + B unless the histograms are to be accumulated with previous histograms. For this case, the bin size and lower limit are set to those of the previous histogram.

There are three major loops for calculating the histogram. The outer loop is executed once for each line of data; the next loop is executed once for each channel selected; and the inner loop is executed once for each data value (sample) selected from the line. Each data value has its bin number calculated as:

$$\text{BIN NUMBER} = \frac{\text{(Real data) - (Lower limit)}}{\text{(Bin Size)}}$$

If the calculated bin number is from 1 to 100, the accumulation for that bin (the array HISTA) is incremented by 1 and then the next sample is processed. After all samples for the line have been processed, the same line is processed for the next channel, etc. Finally, the above process is repeated for each line of data. Since the array HISTA (INTEGER*2) was initialized in the beginning to -32767, a maximum accumulation of over 65,000 is possible. The histogramming process stops if any bin approaches overflowing prior to all lines being histogrammed.

The above sequence must often be modified, however, since the calculated data values may fall outside the range of the histogram (the bin number is less than 1 or greater than 100). For this situation, the histogram must either be shifted and/or the bin size must change.

The first choice is to keep the bin size the same and to shift the range of the histogram. This involves calculating the lowest bin number used and the highest bin number used. The range is then shifted to be centered over the data accumulated so far. Then HISTA is adjusted accordingly.

If, with the current bin size, the data is outside the range, the bin size must be increased. Therefore, the bin size is doubled until 100 bins will cover the range. At this point, the accumulations in the new bin size must be checked for overflow. If no overflow, the range will be centered over the data once again. HISTA will be adjusted and then the whole process is repeated for the next sample in the line.

The overall objective of the above algorithm is to produce smooth histograms that maximize the probability that the 100 bins used represent the actual data as closely as possible.

6. Flowchart

Not Applicable

LARS Program Abstract 129

MODULE IDENTIFICATION

Module Name: LAREAD Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Reads both forms of field description cards.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: E. M. Rodd Date: 12/18/72

MODULE ABSTRACT

LAREAD is designed to be called when reading field description cards when they may or may not be mixed with 'CLAS' or 'TEST' cards. It can interpret either form of field description cards. LAREAD returns field description information if a field description card was read and a code if some other card was read.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LAREAD

CALL LAREAD (INFO,CODE,UNIT,ERRETN,FORMAT)

Input Arguments:

UNIT - INTEGER*4, The data set reference number from which the field description cards are to be read.

ERRETN - INTEGER*4, ERRETN is a flag used to inform LAREAD what to do if it fails to recognize the card (see text below). If ERRETN=0, LAREAD will write an error and request a correct card from the typewriter. If ERRETN=1, LAREAD will return with CODE=5.

FORMAT - INTEGER*1, Flag indicating form of input data set. If FORMAT=0, the input unit UNIT is to read with formatted I/O. If FORMAT=1, the unit is to be read using unformatted I/O.

Output Arguments:

INFO - INTEGER*4, Seventeen fullword vectors used to return field description information. If a field description card is not read, INFO is returned with all the defaults of a field description card. If the free form field description card is read only INFO(1) and INFO(4-9) are non-defaults. For the fixed form card, there are no defaults and all 17 words of INFO are taken from the field description card. Below are listed the entries in INFO and their defaults if no field description card was read or if the free form card was read.

Element	Entry	Default	Columns (Fixed Form Card)
1	Run Number	blank	1-8
2-3	Description	blank	11-18
4	First line of field	1	21-25
5	Last line	30,000	26-30

Element	Entry	Default	Columns (Fixed Form Card)
6	Line interval	2	31-35
7	First column	1	36-40
8	Last column	5000	41-45
9	Column interval	2	46-50
10-17	Other information	blank	51-80

Note that the default of blank is EBCDIC blanks and not zeroes.

CODE - I*4, CODE=0 if a valid field description card is read. CODE = 1,2,3,4, respectively if a TEST, CLAS, DATA or END card is read. CODE=5 if the card is invalid and ERRETN = 1.

LAREAD is called to read either form of field description card. If a field description card is read and it is recognized as a field description card but has an error such as last line less than first line, LAREAD will write an error and request a corrected card typed in. If the card is unrecognized (that is neither a free or fixed form field description card nor a TEST, CLAS, DATA or END card) and ERRETN = 0, then LAREAD will request a correct card from the typewriter. A response of carriage return to a request for a new card will cause LAREAD to read the next card from UNIT. If a response of KILL is typed, LAREAD will terminate execution with a STOP statement.

2. Internal Description

LAREAD uses either formatted or an unformated READ depending upon the value of FORMAT to read the card. It first checks for the first four characters of KILL and if so terminates the function. This check is made for use when reading a corrected card from the terminal (of course, when reading from the typewriter, a formatted READ is used). LAREAD then checks to see if the card is a free form field description card. If the card begins with RUN, LINE or COL, it is considered a free form field description card. If not, LARS12 is called to attempt to interpret the card as a fixed form field description card. If LARS12 makes an error return, LAREAD assumes that the card is not a correct field description card. At this time LAREAD uses DATCRD to see if the card has a keyword of TEST, CLAS, DATA or END. If so LAREAD returns with the appropriate value of CODE. If not, and if ERRETN = 0, LAREAD calls ERPRNT to write a message and requests a correct card. If, however, ERRETN = 1, CODE = 5, and LAREAD returns.

3. Input Description

The cards are read from DSRN UNIT which can be formatted or unformatted 80 character records.

The typewriter is used for input of corrected cards.

4. Output Description

Any erroneous cards are typed before the request for a correct card. E150 is written via ERPRNT for an unrecognized card (if ERRETN = 0). Errors 151-159 are written via ERPRINT for errors in field description cards. E139 is written via ERPRNT if EOF is read on UNIT.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 130

MODULE IDENTIFICATION

Module Name: MMTAPE Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Mounts and positions results tapes

System/Language: CMS/FORTRAN

Author: E. M Rodd Date: 09/05/72

Latest Revisor: P. D. Alenduff Date:

MODULE ABSTRACT

MMTAPE mounts and positions the results tape (or a tape to be used as output for copying results files).

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

MMTAPE

CALL MMTAPE (RQTAPE,RQFILE,MODE)

Input Arguments:

RQTAPE - I*4, Tape number of requested tape. A tape number of 0 is a request for a scratch tape.

RQFILE - I*4, File number of requested file. If RQFILE is = 0, then the tape will be initialized by writing a record type 1 on the results tape with filetype = 0.

MODE - I*4, Flag indicating usage of MMTAPE. MODE = -1 indicates MMTAPE has been called to mount and position a tape to be used for copying results files onto. Mode = 0 indicates that a results tape is being mounted for reading a results file. In this case, the tape is mounted ring out. Also, if MODE = 0, RQFILE = 0 is invalid and will cause an error when an attempt is made to write on the tape. MODE = 1 indicates a tape is being mounted for writing a new results file (or continuing a suspended classification). The difference between MODE = -1 and MODE = +1 is the DSRN used for the tape. For MODE = -1 DSRN is CPYOUT and for MODE = +1, DSRN is MAPTAP. (DSRN is MAPTAP for MODE = 0).

Output Arguments

RQFILE - I*4, Set to -1 if requested output tape file was full and user decided to enter new tape and file request. Otherwise, sends back current file position of tape.

MMTAPE checks the validity of the tape by reading the record type 1 from the tape and verifying the tape and file number as well as checking for the correct type of file. Any attempt to overwrite an existing file causes MMTAPE to ask the user (via the typewriter) if he wishes to overwrite the file, respecify a new results card, or terminate the function. Note, however, that if a request has been made to initialize a tape, no checking is performed on previous contents.

2. Internal Description

See Output Description.

3. Input Description

The record type 1 of the results tape is read for each file up to and including the file needed. That is, if file 4 is requested the record type 1 is read from files 1-4.

4. Output Description

The following information messages are issued under the circumstances listed. The term filetype means the filetype code from record type 1 of a results file (the program uses varialbe CHECK for this number).

I0042 is typed when a tape has been mounted and before MMTAPE positions it. This message is not typed when the tape is being initialized or when the correct type number was already mounted.

I0043 is typed when MODE = ± 1 and filetype of the requested file = 0.

I0044 is typed when MODE = ± 1 and filetype of the requested file = 1 and the restart flag from GLOCOM (RESTRT) is not = 1.

I0045 is typed when the tape is correctly positioned. This is not typed when initializing a tape.

I0050 is typed following E 362 to determine what action should be taken since the requested Output File does not exist on the tape.

After I0043 and I0044, the user is asked whether he wishes to overwrite the file, respecify a new results card with a new tape and/or file or disk option, or terminate the function.

I0100 is typed to allow entry of the new results card. This occurs when the user requests to respecify the results card.

I0101 is typed to allow entry of a new 'TO' card. This occurs at the user's request.

The following error messages are typed under the conditions listed.

E361 is written when the tape is being filed forward and a file is encountered with filetype other than zero before the requested file is reached and MODE = 0.

E362 is written when the circumstance for E361 occurs and MODE = 1. It is also written when MODE = 1 and the filetype of the file requested is = -1.

E363 is written if the RESTRT flag is = 1 and the filetype of the requested file is not = 1.

E364 is written when MODE = 1 and the filetype of the fiel requested = 1.

E365 is written when an EOF is read on the results file. This should never occur with valid results files.

For message texts refer to the User's Manual.

5. Supplemental Information

This section deals with the handling of tapes by MMTAPE.

Input:

If a tape is mounted on the device and it is the incorrect tape number (as noted from the appropriate status words in GLOCOM), TOPRU is called to unload the tape before the correct tape is mounted. If the correct tape is mounted, MMTAPE will check for the ring in if MODE = ±1. If the ring is not in, the tape is unloaded and MOUNT is called to mount the tape with the ring in. If the correct tape is mounted, MMTAPE assumes that the file number (as recorded in GLOCOM) is correct and moves the tape backwards or forwards to find the requested file.

Output:

The tape is mounted with ring in for MODE = ±1 and with ring out for MODE = 0.

The tape is left positioned at the beginning of the requested file. When the tape is initialized a TOPRW is used to do this.

6. Flowchart

Not Applicable

LARS Program Abstract 131

MODULE IDENTIFICATION

Module Name: RDFLDS

Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Reads test and training fields and orders them

System/Language: CMS/FORTRAN

Author:

Date:

Latest Revisor: P.D. Alenduff

Date: 12/20/74

MODULE ABSTRACT

RDFLDS reads a test field deck and stores it on disk. Entry RDTRN reads the training field deck out of the statistics file in the results file. Entry RDCLUS reads only field description cards for the Cluster function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

RDFLDS

```
CALL RDFLDS (FLDARY, NOFLDS, NOGRPS, NOPOOL, UNIT, OUTUNT)
```

Input Arguments:

NOFLDS - I*4, The Maximum number of test fields which can be stored in FLDARY. If more than this number are encountered, E490 is written.

NOGRPS - I*4, Number of groups of classes. This is used to detect an invalid group number on the TEST card. If no TEST cards are expected, NOGRPS = 61.

NOPOOL - I*4, Number of classification pools.

UNIT - DSRN from which fields are to be read. The data set must be fixed length 80 character records.

OUTUNT - DSRN onto which the field information will be written. This data set is written with unformatted WRITE statements.

Output Arguments:

FLDARY - FLDARY(10, J) contains the information describing the J'th field read in.

FLDARY(1,J) = Run number

FLDARY(2,J) = First line

FLDARY(3,J) = Last line

FLDARY(4,J) = Line increment

FLDARY(5,J) = First sample number

FLDARY(6,J) = Last sample number

FLDARY(7,J) = Sample increment

FLDARY(8,J) = Group number of the test class (i.e. the group number of the TEST card preceeding the field).

FLDARY(9,J) = NOPOOL +2.

FLDARY(10,J) = Sequence number of field. These sequence number order fields by first increasing run number and within runs by increasing line number.

FDFLDS reads a test field data deck. It terminates when a DATA or END card is read.

RDTRN

```
CALL RDTRN (FLDARY, NOFLDS, NOGRPS, GRPSTK, POLPTR, POLSTK,
            NOPOOL, UNIT, OUTUNT)
```

Input Arguments:

NOFLDS - Same as RDFLDS

NOGRPS - Same as RDFLDS

Note that RDTRN is a special purpose entry designed for use by Printresults to read the training fields from the statistics present in the results file and read to the end of the statistics. The input argument GRPSTK is the array GRPSTK in PRICOM. POLPTR and POLSTK are the POLPTR and POLSTK arrays from the record type 2 of the results file.

NOPOOL - Same as RDFLDS (this also came from the results file).

UNIT - Same as RDFLDS (in the call from Printresults, UNIT is the results file).

OUTUNT - Same as RDFLDS.

RDCLUS

```
CALL RDCLUS (FLDARY, NOFLDS, NOGRPS, NOPOOL, UNIT,
             OUTUNT)
```

Input Arguments:

Same as RDFLDS

Output Arguments:

Same as RDFLDS

Note that RDCLUS is a special purpose entry designed for use by Cluster to accept only field description cards or an end card. Any DATA, CLASS, or TEST card in the deck will cause an error E373 to be returned.

2. Internal Description

Not Applicable

3. Input Description

RDFLDS reads via LAREAD the entire test field deck. RDTRN reads the training field portion of the statistics on the results file via LAREAD. It is then flushed to the end of the statistics records. RDTRN must first read record 1 of the statistics.

4. Output Description

Both RDFLDS and RDTRN write the field descriptions on unit OUTUNT using unformatted I/O. Each record of OUTUNT contains the INFO array plus the group number and class number. After writing all the fields, a record with nineteen words each containing 'EOF' is written. Error message 373 is written by entry point RDCLUS to the printer and typewriter.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 133

MODULE IDENTIFICATION

Module Name: REDSAV Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Reduces statistics

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: E. M. Rodd Date: 12/12/72

MODULE ABSTRACT

REDSAV reduces the mean vectors and covariance matrices. Reducing means that the mean vectors and covariance matrices are computed to reflect grouping of classes and use of only a subset of the channels from the original statistics.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

REDSAV

```
CALL REDSAV (COVAR, AVAR, COVMTX, AVEMTX, KEPPTS, NOPOOL,
             NOFEAT, VARSIZ, NOFET3, POLPTR, POLSTK, FETVEC,
             FETVC3, VARSZ3)
```

Input Arguments:

COVAR - R*4, Covariance matrices for all classes (taken from the statistics file). COVAR (I,J) = the I'th element of the covariance matrix for class J. The order of elements within a covariance matrix for one class is c_{11}, c_{21}, c_{22}, c_{31}, etc.

AVAR - R*4, The mean vectors for all classes and channels in the statistics. (Taken from the statistics file)

AVAR(I,J) = mean of values for the I'th channel in the statistics for class J.

KEPPTS - I*4, KEPPTS(I) = the number of points in class I. (Class I of the statistics in COVAR and AVAR).

NOPOOL - I*4, Number of pooled classes in the reduced statistics.

NOFEAT - I*4, Number of channels in the original statistics.

VARSIZ - I*4, Number of elements in the covariance matrix for one class in COVAR. VARSIZ =

$\frac{NOFEAT^2+NOFEAT}{2}$. This is the number of elements in the lower triangular portion of the symmetric covariance matrix which is all that is stored in COVAR.

NOFET3 - I*4, The number of channels to be kept in the reduced statistics.

POLPTR - I*2, Array defining pooled classes. POLPTR(1,J) = number of classes in pool J. POLPTR(2,J) = pointer to the first element of POLSTK containing pool J.

POLSTK - I*2, A list of classes grouped by pool. Classes for a given pool are contiguous in POLSTK.

FETVEC - I*2, FETVEC(I) = channel number of the I'th channel used in the original statistics.

FETVC3 - I*2, FETVC3(I) = channel number of the I'th channel used in the reduced statistics.

VARSZ3 - I*4, Number of elements in the reduced covariance matrix for one class.

Output Arguments:

COVMTX - R*4, The reduced covariance matrices for all classes. This is computed by pooling classes and eliminating channels not requested in the reduced statistics. Whereas the dimension of COVAR is (VARSIZ, number of classes), the dimension of COVMTX is (VARSZ3, NOPOOL). The arrangement of elements is similar to COVAR.

AVEMTX - R*4, The reduced mean vectors for all classes. This is also computed by pooling class statistics and eliminating channels not requested in the reduced statistics. Whereas the dimension of AVAR is (NOFEAT, number of classes), the dimension of AVEMTX is (NOFET3, NOPOOL).

If NOPOOL is less than or equal to zero, REDSAV assumes that no pooling of classes is to be performed so the contents of POLPTR and POLSTK and KEPPTS are never used. In this case, reducing the statistics consists only of eliminating channels not requested.

2. Internal Description

COVMTX and AVEMTX are computed such that they reflect the true statistics of the pooled classes as if they were the original classes. To perform this task, the number of points in each class (KEPPTS) is required.

3. Input Description

Not Applicable

4. Output Description

I0032 is typed when REDSAV has completed computing.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 134

MODULE IDENTIFICATION

Module Name: REDSTA Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Reads statistics from the statistics file

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: E. M. Rodd Date: 12/12/72

MODULE ABSTRACT

REDSTA reads all statistics from the statistics file on disk. The statistics are passed back to the caller.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

REDSTA

```
CALL REDSTA (COVAR,AVAR,CLSDES,KEPPTS,BINFLG,NOCNT,
             NOCLS,CSET,CSEL,FETVEC,VARSIZ,NOFEAT)
```

Input Arguments:

BINFLG - I*2 = 1 if means and covariances are in machine form on the statistics file. BINFLG = 0 if they are in character form. (See description of statistics file in the System Manual).

NOCLS - I*4 The number of classes in the statistics. If the number of classes found in the training field portion of the statistics file is not equal to NOCLS, E302 is written.

NOFEAT - I*4 The number of channels in the statistics file. If this is incorrect, unpredictable errors will occur.

VARSIZ - I*4 The number of elements in the lower triangular portion of the symmetric covariance matrix for one class.

Output Arguments:

COVAR and AVAR - R*4 The mean and covariance arrays as written on the statistics file. For the arrangement of these arrays see the System Manual description of the statistics file.

CLSDES - I*4 CLSDES(1,I) = first four characters of name of class I. CLSDES(2,I) = second four characters.

KEPPTS - I*4 KEPPTS(I) = number of points in Class I. (that is in the training class.)

NOCNT - I*4 NOCNT(I) = number of training fields in class I.

CSET and CSEL - CSET and CSEL as defined for subroutine GADLIN.

FETVEC - I*2, FETVEC(I) = channel number of the I'th channel in the statistics.

2. Internal Description

Not Applicable

3. Input Description

The statistics file is read. It is rewound and the first record read over. The training field portion is read to determine the class names and the number of training fields in each class. The count card (record type 3) is detected as an erroneous field description card and this is the signal to read the remainder of the file. The file is positioned just in front of the last record which is not read by REDSTA.

4. Output Description

E302 is written via ERPRNT if NOCLS disagrees with the number of classes in the file.

5. Supplemental Information

REDSTA is designed to be used after a call to STAT.

6. Flowchart

Not Applicable

LARS Program Abstract 135

MODULE IDENTIFICATION

Module Name: STAT Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Reads statistics file

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: E. M. Rodd Date: 12/12/72

MODULE ABSTRACT

STAT reads the statistics file from disk or cards. If read from cards, the data is written to the disk file.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

STAT

CALL STAT (UNIT,NOCLS,NOFLD,NOFEAT,NUM,BINFLG)

Input arguments:

UNIT - I*4, DSRN from which the statistics file is to be read.

Output arguments:

NOCLS - I*4, Number of classes in statistics file read. This and NOFLD and NOFEAT are read from the record type 3 of the statistics file. (See data organization section of the System Manual).

NOFLD - I*4, Number of training fields used in generating the statistics.

NOFEAT - I*4, Number of channels in the statistics.

NUM - I*4, The number of records of the statistics file which precede record type 3.

BINFLG - I*2, Flag from record 1 of the statistics file indicating format of means and covariances. BINFLG = 1 means machine, BINFLG = 0 means character format.

STAT is called to read the statistics file from cards or disk. If the file is read from cards, it is placed on disk. The sequence numbers are checked.

2. Internal Description

Not Applicable

3. Input Description

The statistics file is read from cards or disk depending upon the value of UNIT. The file is read until a record beginning with'EOS' is read (normal completion) or a record beginning with 'END' or 'DATA' is read (error condition). As the file is read, the record number of record type 3 is noted so that the disk file can be rewound and then read over up to record type 3 so that NOCLS, NOFLD and NOFEAT can be retrieved from record type 3. This record is detected in the original pass by the fact that its first two columns are always blank and records preceeding it must have non-blank first 2 columns. If reading the statistics file from cards, the next card after the 'EOS' card is read so that the 'END' or 'DATA' card is read.

4. Output Description

If the file is read from cards (UNIT = the card reader), the file is written on disk on unit SDATA.

5. Supplemental Information

The following error conditions are detected by STAT.

E300 When the first record read from UNIT is neither the normal first record (starts with 'LARS') nor the EOS record.

E301 When the first record is the EOS record.

E198 When an 'END' or 'DATA' record is read before an 'EOS' record is read.

E139 When EOF is read.

E303 For a card out of sequence. Only the first such error is written. The user will not be notified of further sequence errors.

6. Flowchart

Not Applicable

LARS Program Abstract 136

MODULE IDENTIFICATION

Module Name: TSPACE Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Computes Space remaining on D-disk

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd Date: 07/27/72

Latest Revisor: Date:

MODULE ABSTRACT

TSPACE is a FORTRAN callable function subprogram used to find how many bytes of storage remain on the D-disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

TSPACE

J = TSPACE(I)

Input Arguments:

I - The value of I is irrelevant. Its value is not changed by TSPACE. It is provided only to make the compiler recognize TSPACE as a function without the use of an external statement.

Output Arguments:

J - I*4 J = 800 times (number of 800 byte records left on the D-disk minus 50)

2. Internal Description

TSPACE consults the active disk table of the D-disk to find out how much space is left on the disk. TSPACE uses the CMS routine ADTLKP to locate the active disk table (i.e. find the address of the table). The DSECT provided by the CMS ADT macro is used to find the offset of the word ADTLEFT which contains the number of records left on the disk. TSPACE finds the number of 800 byte records remaining, subtracts 50 records to leave some room on the disk, and multiplies the difference by 800 to get the number of bytes on the disk.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 137

MODULE IDENTIFICATION

Module Name: WRTTRN Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Reads training fields from the statistics file and prints them.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: E. M. Rodd Date: 12/12/72

MODULE ABSTRACT

WRTTRN reads the training fields from the statistics file and prints a list of the training fields included in the reduced statistics.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

WRTTRN

CALL WRTTRN (NOGRPS, GRPNAM, GRPPTR, GPRSTK)

Input Arguments:

NOGRPS - INTEGER*4, Number of groups of pools of classes.

GRPNAM - INTEGER*4, GRPNAM(1,I) = First four characters of name of group I. GRPNAM(2,I) = Second four characters.

GRPPTR - INTEGER*2, GRPPTR(1,I) = Number of statistics classes in group I. GRPPTR (2,I) = First element of GRPSTK with a class in group I.

GRPSTK - INTEGER*2, A list of classes ordered by group.

WRTTRN will leave the statistics file positioned at the end of the file just before the EOF.

2. Internal Description

Not Applicable

3. Input Description

The statistics file is read. It is first rewound. The first record is read over with no verification performed. The field description cards are read by LAREAD. When LAREAD informs WRTTRN (via CODE = 5) that the end of the field description cards has been reached, WRTTRN flushes the file until 'EOS' is read.

4. Output Description

The list of all training fields included in the reduced statistics is included. For the definition of reduced statistics, see module documentation of REDSAV.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

MODULE IDENTIFICATION

Module Name: CLSDEC Function Name: LARSYS FUNCTIONAL SUPPORT

Purpose: Decodes part of the POOL card.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Joan S. Buis Date: 03/80

MODULE ABSTRACT

CLSDEC decodes portions of the POOL card or CLASSES card which is enclosed in parentheses. CLSDEC must be called once for each set of classes selected from the indicated stat decks that follow the POOL name or CLASSES parameter INCLUDE or DELETE.

Entry Points

POLDEC (CARD,COL,DCKNUM,IWORK,CNT,PAREN,*)
Entry for decoding POOL card. POLDEC is called by POLSET.

1. Module Usage

CLSDEC

CALL CLSDEC (CARD,COL,DCKNUM,IWORK,CNT,PAREN,*)

Input Arguments:

CARD - L*1. Eighty byte input buffer containing card image of POOL or CLASSES card.

COL - I*4. Column number on card to start decoding (this should be the column number of the left parenthesis following the POOL or CLASSES name, or the column number of a comma preceeding a deck number).

Output Arguments:

COL - The column number of the comma following a slash or a right parenthesis, or if there are no more entries on the card, returned with a value of 72.

DCKNUM - I*4. The statistics deck number. In the form:

POOL WHEAT (2/4,5/)

the deck number is 2.

IWORK - I*4. Sixty fullword buffer to be returned with the class numbers in the stat deck. In the above example, IWORK would be returned with IWORK(1) = 4, and IWORD(2) = 5.

CNT - I*4. The number of classes selected from the indicated stat deck. In the above example CNT would be returned as 2.

PAREN - I*4. Indicate whether or not a right parenthesis has been reached. This information is needed to determine if CLSDEC must be called again. CLSDEC must be called once for each indicated stat deck, in the following example CLSDEC would be called three times.

POOL CORN(1/7-9/,2/3,5/,3/6/)

Return values:

0 = another call is needed
1 = right parenthesis reached

Non-standard Return:

A RETURN 1 is executed if one of the following errors occurs:

1. Any non-numeric is found between commas
2. More than one deck number is present (i.e., two numbers separated by a comma precede the slash)
3. No deck number is present (i.e., a right parenthesis precedes the slash).
4. The first class number is missing (i.e., comma preceded by a slash).
5. The last class number is missing (i.e., slash preceded by a comma).
6. Any class number is missing (i.e., two consecutive commas).

A RETURN 2 is executed if the following error occurs:

1. Any character other than blanks or a comma follows the right parenthesis.

2. Internal Description

FORTRAN byte manipulation using L*1 buffer.

3. Input Description

Not applicable.

4. Output Description

Not applicable.

5. Supplemental Information

Not applicable

6. Flowchart

Not applicable.

LARS Program Abstract 150

MODULE IDENTIFICATION

Module Name: CLASUP   Function Name: CLASSIFYPOINTS

Purpose: Supervisor for CLASSIFYPOINTS

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: E. M. Rodd   Date: 09/06/72

MODULE ABSTRACT

Supervisor for the CLASSIFYPOINTS processor. CLASUP performs no computation, but instead makes calls to the subroutines which really make up the processor.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLASUP

CALL CLASUP

CLASUP is called by LARSMN to execute the Classifypoints function.

2. Internal Description

CLASUP receives control from LARSMN to perform the Classifypoints processing. CLASUP calls CLARDR to read and interpret the function control cards, CLAINT to complete initialization and compute array bases. These bases are returned via the calling sequence and are used to pass storage for the arrays to CLSFY1 and CLSFY2 which perform the classification process. At completion message I0040 is typed.

3. Input Description

CLASUP reads the restart record types 1 and 2 if the restart option is in effect. The read is performed here rather than in CLSFY2 in order to obtain the base addresses in ARRAY of arrays in the call to CLSFY2.

4. Output Description

Standard supervisor information messages.

5. Supplemental Information

See system manual for supervisor requirements.

6. Flowchart

Not Applicable

LARS Program Abstract 151

MODULE IDENTIFICATION

Module Name: CLACOM Function Name: CLASSIFYPOINTS

Purpose: Common for CLASSIFYPOINTS

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: E. M. Rodd Date: 09/01/72

MODULE ABSTRACT

This is the BLOCK DATA subroutine for the CLACOM common area. Refer to the LARSYS System Manual for a list of the program modules that require the common.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 152

MODULE IDENTIFICATION

Module Name: CLAINT Function Name: CLASSIFYPOINTS

Purpose: Initialization Including array base computation

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. J. Davis Date: 09/24/73

MODULE ABSTRACT

CLAINT reads statistics and reduces them. As statistics are available, array bases are computed. The areas to be classified are read and the first two records of the results file are written.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLAINT

CALL CLAINT (COVAR3,AVAR3,DATBS3)

Input Arguments:

Not Applicable

Output Arguments:

COVAR3 - I*4, the base of the covariance array for the reduced statistics. This base is (channels**2-channels)/2 soublewords into the array ARRAY. The covariance array is dimensioned (VARSZ3*NOPOOL+1) fullwords where VARSZ3 is (channels**2-channels)/2 and NOPOOL is the number of classification pools.

AVAR3 - I*4, the base of the average array. It follows the covariance array. It is dimensioned (channels**NOPOOL+1) fullwords.

DATBS3 - I*4, the next free word in the array ARRAY after the reduced covariance and mean (average) arrays.

2. Internal Description

CLAINT performs the following functions:

1. Uses STAT to read the statistics from disk or cards.

2 2. Computes space needed for arrays to hold original statistics.

3. Uses CLSCHK to check class validity and FETCHK to check channel validity (that is check validity of user requests in light of the available statistics).

4. Checks number of weights against number of pools and normalizes weights.

5. Compute array bases for reduced statistics.

6. Print data summary.

7. Reduce statistics using REDSAV.

8. A version of the pool arrays POLSTK, POLNAM and POLPTR are computed in POLST1, POLNA1 and POLPT1 for writing on the results file. These arrays are ordered such that the first pool in POLST1 is the pool containing the lowest numbered class. The second pool in POLST1 is the pool containing the lowest numbered class of any pool other than the first pool.

9. The data cards for the areas to be classified are read and the field description information is written onto the scratch disk data set CLASSIFY SCRATCH. If a disk is to be used for results, the space required for writing the results is computed as each field description is read.

10. If results are to be on disk, then the space required is computed and compared to the space available on the LARSYS disk.

11. The first three record types are written on the results file. The first record has a 1 in the sixth full word to indicate the inclusion of weights in the file.

12. If PRCFLG is greater than 0 (L1 or L2 option requested) then return.

13. The table THRTAB is built of threshold values corresponding to rejection percentages from .1 to 99. This is done by calling THRESC. The rejection percentages used are from .1 to 15% in steps of .1 and from 15 to 99% in steps of 1%.

Subroutines Called by CLAINT

STAT	ERPRNT	REDSAV
REDSTA	CTLWRD	LAREAD
CLSCHK	RTMAIN	THRESC
FETCHK	FVAL	

3. Input Description

Statistics are read via a call to STATS. Field description cards are read via calls to LAREAD.

4. Output Description

Information concerning serial number, classes, fields and channels is printed. Also the informational message 34.

The data set CLASSIFY SCRATCH (DSRN CLASSX) is created containing the array INFO for each area to be classified. INFO(10) is set equal to the area number (area number is 1 for the first card read, etc.). This is done for use by printresults. The first three record types are written onto classification results.

5. Supplemental Information

The space for the results file is computed if results are to be written onto disk. The space is computed by accurately computing the space for actual results (2 bytes per point plus four bytes per line. 3000 bytes are reserved for record types 1 and 2. The number of 80 byte records in the statistics data is counted by reading the data set and multiplied by 80 to get the space for record type 3. The space required for record type 4 is (VARSZ3 + NOFET3) *NOPOOL bytes where NOFET 3 is the number of channels. 20 percent is added to this space to account for FORTRAN control information and record types 5 and 7.

6. Flowchart

Not Applicable

MODULE IDENTIFICATION

Module Name: CLARDR Function Name: CLASSIFYPOINTS

Purpose: Reads function control cards

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: J. B. Etheridge Date: 04/30/73

MODULE ABSTRACT

CLARDR reads and interprets all function control cards for CLASSIFYPOINTS. Checks are made for data validity. Also a results tape is readied if results are to be placed on disk.

Entry RDRESU is called to read a results card and mount the tape.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLARDR

CALL CLARDR

This section lists the actions taken when the following control cards are read:

RESTART	- The flag RESTRT in GLOCOM is set to one. This flag is used in CLARDR to skip checks for complete data since only a RESULTS card is required.
AUTO	- The action taken when the AUTO CHANNELS card is read is intended to set the same variables as would be set by subroutine CHANNEL if a CHANNELS card were read. The channels themselves are taken from the GLOCOM array TEMPAS where they were placed by SEPSUP. TEMPAS contains a value of zero in the location after the last channel unless there are thirty channels in which case all of TEMPAS is non-zero. If neither a CHANNELS or AUTO card is given (then NOFET3, the number of channels, will be zero) the program writes an error and calls CTLWRD with ERROR = 3.
RESULTS-TAPE	- The variable RQTAPE is set to the given tape number.
RESULTS-FILE	- The variable RQFILE is set to the given tape number.
RESULTS-INITIALIZE	- The local flag INITFG is set to one. This flag is used by CLARDR to trigger a call to MMTAPE to initialize a tape.
PROCEDURE-L1	- The flag PRCFLG is set to 1 (classify points using the L1 procedure).
PROCEDURE-L2	- The flag PRCFLG is set to 2 (classify points by Euclidean distance procedure).
RESULTS-DISK	- The variable RESULT in CLACOM is set to CLASSR. Thus the DSRN of results is set to disk.

PRINT-STATS	- The flag STATKY is set to 1. This flag will be used in CLAINT to trigger printing of statistics.
PRINT-MAP	- The flag MAPFLG in CLACOM is set to one.
CLASSES	- Subroutine POLSCN is called to set up the arrays POLNAM, POLPTR, POLSTK and POLNM1 and compute NOPOOL and STKPTR based on the interpretation of the classes card.
CARDS READSTATS	- The flag DECKFG is set to one. This is used in CLAINT to set the DSRN from which statistics will be read.
CHANNELS	- Subroutine channel is called to NCR, CSEL3, CSET3, and FETVC3 based upon interpretation of the channels card.
WEIGHTS	- Values are stored in array PROB and PRBCNT is set to the number of weights; both variables are in CLACOM.

RDRESU

This entry is called by CLSFY2 when the SUSPEND command has been given and results are on disk. It reads a RESULTS card from the typewriter and then branches to the code to check the RESULTS card and call MMTAPE to mount the tape.

2. Internal Description

CLARDR uses standard card reader logic in using CTLWRD, CTLPRM, and IVAL in reading and interpreting the control cards. CHANEL is used to interpret the CHANNELS card and POLSCN is used to interpret the CLASSES card.

CLARDR begins by calling TSTREQ to clear the stop/suspend flag. Then GTSERL is called to return the serial number to be used for this classification. Then flags and arrays which will convey control card information are initialized. From this point the program functions in a loop of reading and interpreting control cards until a DATA or END card is read indicating the end of function control cards. After the control cards have been read, several checks are made on the data. There must be channels specified, and if classes are pooled, the pooling is checked for validity. Then the results tape is initialized if requested and mounted and positioned. If disk is used, no tape is mounted. Checks are made to be certain that both disk and tape were not requested. Finally notice is printed of options selected (for printing of statistics, printing of a map and for use of disk for results).

Subroutines Called by CLARDR

TSTREQ	CTLPRM	FVAL
GRSERL	IVAL	RTMAIN
CTLWRD	CHANEL	MMTAPE
ERPRNT	POLSCN	

3. Input Description

Function control cards for CLASSIFYPOINTS are read via CTLWRD.

4. Output Description

Control card error messages are written via ERPRNT. A list of options selected is printed.

5. Supplemental Information

See System Manual for card reader requirements.

6. Flowchart

Not Applicable

LARS Program Abstract 154

MODULE IDENTIFICATION

Module Name: CLASS Function Name: CLASSIFYPOINTS

Purpose: Guassian Classifier

System/Language: CMS/ASSEMBLER

Author: Date:

Latest Revisor: E. M. Rodd Date: 09/07/72

MODULE ABSTRACT

This assembler program is really two programs executed sequentially. The whole program works on one line of data at a time. The first part is the Guassian classifier and classifies all points in a line and computes likelihood values. The second part searches a chi-square table (THRTAB) to determine the likelihood code for the point.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLASS is made up of 2 parts, the first performs the maximum likelihood classification and the second computes coded threshold values.

CLASS

CALL CLASS (RDATA,NV,NO,NC,AVE,COR,IR,VR,CON,THRTAB)

Input Arguments:

RDATE	- Data vectors arranged as data comes from GADLIN, that is all data for line grouped by channel.
NV	- Number of channels
NO	- Number of points in the line
NC	- Number of classes being considered
AVE	- The vector of mean vectors for all classes
COR	- Array of inverse covariance matrices for all classes
IR	- Storage space for output (NO halfwords)
VR	- Storage space for output (NO halfwords)
CON	- Vector of constant terms in the maximum likelihood equation.
THRTAB	- Vector of Chi-square values for various expectations of percent rejections

Output Arguments:

The output of the first part of the program is as follows:

IR(I)	- Contains the class decision for point I in the line as an integer class number.
VR(I)	- Contains the discriminant value as a floating point number for point I.

Then after the second part of the program:

IR(I) Contains data for the i'th point in the line. The first byte of the halfword contains the index of the element of THRTAB corresponding to the discriminant. The second byte contains the class decision.

VR(I) Contains the class decision as a fullword integer. This will be used for indexing purposes in the calling routine since IR can no longer be used for indexing in a FORTRAN program.

2. Internal Description

The first part of CLASS performs the Gaussian classification. This is documented in the LARS Information Note 121569 entitled "The Implementation of the Maximum Likelihood Classification Rule Assuming a Gaussian Density Function." This information note gives the algorithm for a single point, where as CLASS has a loop over all the point in a line added to the calculation. The second part of CLASS consults the table THRTAB of Chi-square values and determines the percent rejection which would cause each point to be thresholded.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 155

MODULE IDENTIFICATION

Module Name: CLSFY1 Function Name: CLASSIFYPOINTS

Purpose: Prints header and prepares for classification computation

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: J.B. Etheridge Date: 04/30/73

MODULE ABSTRACT

CLSFY1 prints header and if requested statistics information. It writes record type 4 of the results file.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLSFY1

CALL CLSFY1 (COVMTX,AVEMTX,COR)

Input Arguments:

COVMTX - R84, array containing the covariance matrices for all classes. Each column contains the matrix for one class. Each column has VARSZ3 elements meaning that only the lower triangular portion of the symmetric matrix is stored.

AVEMTX - R*4, array of mean vectors for each class. AVEMTX(I,J) contains the mean for the J'th class of the I'th channel.

COR - R*4, this is a work area used by CLSFY1 in the call to COVIN. It is also used as a work matrix to compute the correlation matrices. It dimension is VARSZ3.

Output Arguments:

COVMTX - R*4, this is returned with the inverse covariance matrices of the matrices which were input in COVMTX.

2. Internal Description

CLSFY1 prints the header for classification giving the classes considered, symbols representing classes on the map, weights for pools, channels used, and training fields. The training fields are printed via a call to WRTTRN. Record 4 of the results file is written. If statistics were requested via the PRINT STATS function control card, they are printed. If the L1 or L2 procedure was requested then CLSFY1 returns, otherwise, the array COR is used to compute the correlation matrix for each class from the covariance matrix. Finally, COVIN is called. If COVIN returns DET(1) as -1 indicating that the covariance matrix for some class is singular, the following action is taken: If results are on disk (RESULT = CLASSR), STOP is executed to terminate execution. If results are on tape (RESULT = MAPTAP), TOPRF is called to backup the tape to the beginning of this file. Then a check record is written (record type 1 with word 4 = -1) followed by a tape mark. Then STOP is executed.

Subroutines Called by CLSFY1

GETIME	COVIN
WRTTRN	TOPRF
WRTMTX	TOPEF
	RTMAIN

3. Input Description

Not Applicable

4. Output Description

Record type 4 of the results file. The heading for classification is printed. This includes classes information, symbols, weights, and channel and calibration information. The statistics summary is printed.

5. Supplemental Information

If the L1 or L2 classification procedure was requested then the log of the A PRIORl Probabilities is not computed.

6. Flowchart

Not Applicable

LARS Program Abstract 156

MODULE IDENTIFICATION

Module Name: CLSFY2 Function Name: CLASSIFYPOINTS

Purpose: Main Processing Routine

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P. D. Alenduff Date: 07/15/74

MODULE ABSTRACT

CLSFY2 controls the classification process for the classifypoints function. It controls the mounting of the correct data tape, verifies the sample and channel specifications (and adjusts the former if necessary), sets up the calibration set vector, writes the area identification record (record type 5) calls CONTEX or MCONTEX, for each line in the area to have it classified and the results (record type 6) written to the file, calls CONTEX to write the end of area record (record type7) after the last line in the area. It repeats this process for each area specified in the input deck.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLSFY2

```
CALL CLSFY2 (COVMTX,COVAR3,AVEMTX,AVAR3,RDATA,RDATA2,
             DATBS3)
```

Input Arguments:

COVMTX - R*4, Array of inverse covariance matrices for each class. Dimensioned (VARSZ3,NOPOOL), each column contains the lower triangular portion of the summetric matrix for one class.

COVAR3 - I*4, Base of COVMTX array in ARRAY. (i.e., ARRAY(COVAR3) - first byte of COVMTX). This is required for writing onto the restart file.

AVEMTX - R*4, Array of mean vectors for each class. AVEMTX(I,J) - mean of class J for channelI.

AVAR3 - I*4, Base of AVEMTX array in ARRAY.

RDATA - R*4, Area to be used for dynamic allocation. This must be the first unused element of ARRAY in the calling program.

RDATA2 - I*2, This is normally the same as RDATA in the calling program. It is passed as an argument because it is needed as an I*2 variable and arguments cannot be equivalenced.

DATBS3 - I*4, Number of bytes of ARRAY which are in use (i.e., the number of bytes in ARRAY which precede RDATA).

Output Arguments:

Not Applicable

2. Internal Description

CLSFY2 operates as a loop classifying the areas requested. The only actions taken before the loop begins are to call TSTREQ to clear the STOP/SUSPEND flag and to REWIND the data set CLASSIFY SCRATCH containing the field description information for the areas to be classified. Then the loop begins and follows the following steps for each area classified.

1. Read the field description information from CLASSX. If end of file is detected (END = exit from read statement), control passes to the termination coding at statement 600.

2. Use GETRUN to mount the correct data tape after a check to be certain that the correct run is not already available.

3. Check first sample requested to insure area is within data limit and, if necessary, reduce the last sample number requested to conform to the data available on the tape.

4. Check for existence of all requested channels on the tape. If any are missing, write a message and go to the termination code.

5. Set up the calibration set vector according to the data on the tape.

6. Compute space needed for buffers for results information. If not enough memory is available in ARRAY, write and error and branch to termination coding.

7. Write record type 5 on the results file for this area being classified.

8. Print the heading for this classification.

9. If a map is requested, print the column number header.

10. Go into a loop over all lines in this area. For each line, call GADLIN to get the data for the line and then call CONTEX to do the classification and write the results. (record type 6). PRCFLG is checked to determine whether the L1 or L2 classification procedure has been requested. If so, MCONTX is called in place of CONTEX.

11. Call CONTEX to write record type 7 on the results file.

12. Return to the beginning of the loop to read another area to be classified.

The termination coding consists of writing a record type 8 on the results file and if the file is on tape, writing the needed tape marks and the check record.

Two conditions not previously mentioned may terminate the above loop. If an I/O error (such as end of tape) occurs in CONTEX or MCONTX when they are called to classify and write results for a line, control will pass from the loop to statement 890 where a message (I0047) will be printed and the tape backspaced three records to attempt to back over an end of tape marker. Then a record type 7 (via a CALL to CONTEX) is written and a branch made to the normal termination code. Before each line is classified, a call is made to TSTREQ to test the STOP/SUSPEND flag. If the flag is on, a branch is taken to write the user a message and then a record type 7 is written and then the normal termination code is branched to.

The suspend/restart option is implemented by LCSFY2 in the following manner. If the SUSPEND command is detected (by a value of 2 returned from TSTREQ), CLSFY2 does the following: If results are on disk, the array COVMTX and INFO are stored at the end of the disk file. The disk results file is then copied to a tape requested by the user, and (regardless of where the original results file was located) record types 7 and 8 are written onto the tape. Finally, the special restart file is written (see description of results file in the system manual.)

During restart, the restart file is read (the first part of it is read by CLASUP) to restore variables to their values at the time of suspension. The tape is then positioned at the start of the record type 7 for the area being classified. Note that when the restart is read, the file CLASSIFY SCRATCH is created to contain any descriptions of unclassified areas.

Subroutines Called by CLSFY2

TSTREQ	HEADER	RDRESU
TOPBF	GADLIN	TOPEF
GETRUN	LINERR	TOPRF
ERPRNT	CONTEX	ERASE
GETIME	MCONTX	

3. Input Description

The file CLASSIFY SCRATCH (DSRON CLASSX) is read to get the field fescriptions of areas to be classified.

To restart file (record types 3 and 4) are read if the restart option is in effect.

4. Output Description

Information messages 23, 46, 47, 48, 49. A header describing each classified area is printed preceeding the map (which is optional).
Error messages 241, 258, 259, 376.

Record type 5 and 8 of the results file are written. If the file is on tape, a record type 1 with filetype = 1 is written on the next file.
The entire restart file is created on the results file if the SUSPEND command is issued.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 157

MODULE IDENTIFICATION

Module Name: CONTEX Function Name: CLASSIFYPOINTS

Purpose: Puts results on the results file

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: E. M. Rodd Date: 09/06/72

MODULE ABSTRACT

CONTEX calls CLASS to compute classification results and then writes the results to the file.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CONTEX

CALL CONTEX (RDATA,NV,NO,NC,AVA,COR,IR,IR2,LINE,INC,*)

The arguments RDATA, NV, NC, AVE, and COR are used only in the call to CLASS and never used by CONTEX. See documentation of CLASS.

Input Arguments:

NO - I*4, The number of points in the line being classified.

INC - I*4, The sample increment in the line being classified.
Note that NO is the total number of samples to be classified. INC is used only to make the map fit on one printer page width. NO samples are written to the results file but 1 to NO by INC samples are printed on the map.

LINE - I*4, Line number of line being classified.

Output Arguments:

Neither of these arguments is meant for use by the caller though they do contain valid data. Both arguments are used internally in CONTEX to pass to CLASS and receive information returned from CLASS.

IR - I*2, IR(I) contains for the I'th point in the line the class decision in the second byte as an integer class number and the classification likelihood code in the first byte.

IR2 - I*4, IR2(I) contains the class decision for the I'th point in the line. This is redundant in that the same number is the second byte of each element of IR. It is required for use as an index into the SYM vector for printing the map line.

Non-standard return:

RETURN1 - This return is executed if there was an error in writing the results file. (i.e. the ERR = exit was taken from the WRITE statement). This will occur when the end of tape is encountered.

CONTEX is called once for each line classified. If LINE is = 0, then a record type 7 is written to the results rather than a record type 6. (Also, CLASS is not called when LINE = 0).

2. Internal Description

CONTEX is called to get classification results onto the results file. It is called once for each line. It first calls CLASS to do the actual classification and computation of thresholding indices. The results are then written onto the results file and if the print map flag (MAPFLG) is on (equal to 1), the map for the line is printed. If an error occurs in the write (ERR=exit from FORTRAN write), then a return 1 is executed to so indicate. This will occur when end of tape is reached as well as if an I/O error occurs. If the line number is zero, this means that a record type 7 is being written and no call to class is made and no line printed on the line printer. Only the write to the results file is executed.

Subroutines called by CONTEX

CLASS

3. Input Description

Not Applicable

4. Output Description

A line of results is written on the Classification Results File (record type 6 or 7). A line of the printer map is printed if a map is requested.

5. Supplemental Information

Not applicable

6. Flowchart

Not applicable

LARS Program Abstract 158

MODULE IDENTIFICATION

Module Name: COVIN Function Name: CLASSIFYPOINTS

Purpose: Computes inverses, determinants and constants

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. J. Davis Date: 12/07/73

MODULE ABSTRACT

COVIN computes the inverse of the covariance matrix for each class and the constant terms for the maximum likelihood calculations.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

COVIN

CALL COVIN (NC,NV,VAR,COR,R,DET,CON,PROB)

Input Arguments:

NC - I*4, number of classes.

NV - I*4, number of channels.

VAR - I*4, the number of elements in the covariance matrix for one class (that is, in the lower triangular portion of the symmetric matrix).

This value is $\frac{NV^2+NV}{2}$. It is passed as an argument rather than computed in COVIN so it can be used as a dummy dimension.

COR - R*4, array containing the covariance matrices for all classes (lower triangular portion of symmetric matrices). COR is dimensioned (VAR,NC) so each column contains the matrix for one class.

R - R*4, a work area which is dimensioned (NV,NV).

PROB - R*4, A vector of class weights, one for each class.

Output Arguments:

COR - R*4, returned with the inverse covariance matrices.

DET - R*4, vector of determinants of covariance matrices.

DET(I) = determinant of matrix for class I.

CON - R*4, vector of constant terms. CON(I) = constant term for class I. This is the constant term in the maximum likelihood calculation.

COVIN is called to compute the inverse of the covariance matrix for every class. One call does the computation for all classes. Also the constant term in the maximum likelihood computation is computed for all classes.

2. Internal Description

Not Applicable

3. Input Description

Not Applicable

4. Output Description

Error 375 for singular matrix is written via ERPRNT.

5. Supplemental Information

Uses the SSPLIB routine MINV to invert each matrix.

6. Flowchart

Not Applicable

LARS Program Abstract 159

MODULE IDENTIFICATION

Module Name: GTSERL Function Name: CLASSIFYPOINTS

Purpose: Computes a Serial Number

System/Language: CMS/ASSEMBLER

Author: E. M. Rodd Date: 09/14/72

Latest Revisor: Date:

MODULE ABSTRACT

GTSERL is called from FORTRAN by

CALL GTSERL(SERIAL)

when SERIAL is a fullword. GTSERL will return a 9 digit integer in SERIAL of the form ydddsssss where y is the year, ddd is the day of the year and sssss is the seconds since midnight. GTSERL uses the OS TIME MACRO (with the BIN option) and then manipulates the date and time into the desired format.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 160

MODULE IDENTIFICATION

Module Name: THRESC Function Name: CLASSIFYPOINTS

Purpose: Computer

System/Language: CMS/Fortran

Author: Date:

Latest Revisor: B. J. Davis Date: 08/28/75

MODULE ABSTRACT

THRESC is a mathematical subroutine which computes $x = p^{-1}(y)$ such that $y = P(x) = \text{Prob}(X \leq x)$ where X is a random variable distributed $X^2{}_n$, Chi-square with n degrees of freedom.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

THRESC is a mathematical subroutine used to compute the value of the inverse Chi-square function for a given probability value.

THRESC

CALL THRESC (PI, XP, NDF, IER)

Input Arguments:

PI - Real*4; Probability expressed as 1 minus the fraction of samples expected to be rejected.

NDF - Integer*4; Degrees of freedom.

Output Arguments:

XP - Real*4; The inverse Chi-square value.

IER - Integer*4; Error code. If IER is zero, the calculation succeeded. If it is -1, an input parameter was invalid, such as a value of PI of less than 0 or greater than 1. If it is +1, the output is invalid.

2. Internal Description

THRESC computes $x = p^{-1}(y)$ such that $y = P(x) = \text{Prob}(X \leq x)$ where X is a random variable distributed χ^2_n, Chi-square with n degrees of freedom. To do this, a zero or root of the equation

$$f(x) = \chi_n^2(x) - p = 0$$

(where p is the input probability) is found. The function f(x) is strictly increasing and ranges from -p to 1-p. Thus a unique zero exists.

The algorithm used to find this root is the method of 'regula falsi' or false position (1). The first approximation of the desired root is made from the normal distribution (2). In order to make this approximation, the code of the Scientific Subroutine Package (3) routine NDTRI has been incorporated into THRESC. Code from three other SSP routines has also been incorporated into THRESC. CDTR is needed to evaluate $\chi_n^2(x)$, and NDTR and DLGAM are used by CDTR. See comments in the source programs for references for these subroutines.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

THRESC does not have an IMPLICIT statement like other LARSYS programs because it is all floating point mathematics and such a statement would require many explicit statements.

6. Flowchart

Not applicable

(1). Shan S. Kuo, Numerical Methods and Computers. Addison-Wesley Publishing Company, Inc., Reading, Mass. 1965.

(2). M. Abramowitz and I.A. Stegun, Handbook of Mathematical Functions. U.S. Department of Commerce, National Bureau of Standards, Applied Mathematics Series #55, 1964, equation 26.4.17.

(3). IBM Corporation, System/360 Scientific Subroutine Package (360A-CM-03X), 1970.

LARS Program Abstract 161

MODULE IDENTIFICATION

Module Name: MCONTX Function Name: CLASSIFYPOINTS

Purpose: To classify data points

System/Language: CMS/FORTRAN

Author: John Cain Date:

Latest Revisor: John Cain Date:

MODULE ABSTRACT

MCONTX classifies a line of data according to the procedure option chosen; that is, either the minimum L1 or the minimum L2 distance procedure. MCONTX also prints a map (if requested) and writes the results to a file.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

MCONTX contains two main loops, one for classifying data (one line) using the L1 procedure and the other classifying data using the L2 procedure.

MCONTX

```
CALL MCONTX(RDATA, NV, NO, NC, AVE, COR, IR, IRZ, LINE,
            INC, *)
```

Input Arguments

RDATA - R*4 Data vectors arranged as data comes from GADLIN, that is all data for line grouped by channel.

NV - I*4 Number of channels

NO - I*4 Number of samples (points) in the line

NC - I*4 Number of classes being considered

AVE - R*4 The vector of mean vectors for all classes

COR - R*4 Array of inverse covariance matrices for all classes

LINE - I*4 Line number of line being classified

INC - I*4 The sample increment in the line being classified. Note that NO is the total number of samples to be classified. INC is used only to make the map fit on one printer page width. NO samples are written to the results file but 1 to NO by INC samples are printed on the map.

Output Arguments

IR - I*2 storage vector for output

IR2 - I*4 storage vector for output

2. Internal Description

MCONTX is called to get classification results onto the results file. Upon entry MCONTX transfers the sample data in RDATA into the local array, DATA. PRDFLG is checked to determine which classification procedure loop is to be entered. If PRCFLG is 1, then the L1 loop is entered. Here, the summation from one to the number of channels of the absolute value of the difference between the data sample and the mean is calculated for each class and the minimum value is determined. (see Supplemental Information). If PRCFLG is 2, then the L2 Loop is entered. Instead of summing the absolute value of the difference, Loop 2 sums the square of the difference. Each summation in either loop is multiplied by a weight factor. If no weight factors are given then the summation is multiplied by 1. The class corresponding to the minimum value is then loaded into the IR array. Once all the samples have been classified, the prefix, line number, and classified data values are written to a results file. If a map was requested then these values are also written to the printer. If an error occurs in the write (ERR = exit from FORTRAN write), then a return 1 is executed.

Subroutines Called by MCONTX

Not Applicable

3. Input Description

Not Applicable

4. Output Description

A line of results is written on the Classification Results File (record type b). A line of the printer map is printed if a map was requested.

5. Supplemental Information

L1 distance for class j: $$\left[\sum_{i=1}^{nv} \left| P_{ij} - M_{ij} \right| \right] (1-w_j)$$

L2 distance for class j: $$\left[\sum_{i=1}^{nv} \left(P_{ij} - M_{ij} \right)^2 \right] (1-w_j)$$

Where P = point being classified, M = mean, W = weight, and nv = number of features.

6. Flowchart

Not Applicable.

LARS Program Abstract 170

MODULE IDENTIFICATION

Module Name: CLUSUP Function Name: CLUSTER

Purpose: Supervisor for CLUSTER

System/Language: CMS/FORTRAN

Author: J. B. Etheridge Date: 11/15/72

Latest Revisor: Date:

MODULE ABSTRACT

This supervisor calls three subroutines: One to read in control and data cards, one to read in data from multispectral image storage tapes, and one to do the clustering.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLUSUP

CALL CLUSUP

There are no parameters since only one function, CLUSTER, is involved.

2. Internal Description

CLUSUP first calls the subroutine CLURDR to read in all the control and data cards for the cluster processing function. It then calls the subroutine FIXFLD which prints out the fields to be clustered and obtains the data for these fields from multispectral image storage tapes. Finally, the subroutine CLUPRO is called to cluster the data and print and/or punch the results. CLUPRO returns to CLUSUP which then returns to LARSMN.

List of subroutines called by CLUSUP:

CLURDR
FIXFLD
CLUPRO

Commons in CLUSUP:

GLOCOM
CLUCOM

3. Input Description

Not Applicable

4. Output Description

File name - Informationl messages
DSRN - TYPEWR
Device type - Typewriter
Description - Message numbers are listed below.
For text of messages, see User's Manual.

INFORMATIONAL MESSAGES

165
166

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 171

MODULE IDENTIFICATION

Module Name: CLUCOM Function Name: CLUSTER

Purpose: Block Common for CLUSUP load module

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: J. B. Etheridge Date: 12/21/72

MODULE ABSTRACT

This is a BLOCK DATA subroutine for the Cluster function. Refer to the LARSYS System Manual for a list of the program modules that require the COMMON.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

290

LARS Program Abstract 172

MODULE IDENTIFICATION

Module Name: CLURDR Function Name: CLUSTER

Purpose: Card reader for CLUSTER

System/Language: CMS/FORTRAN

Author: J. B. Etheridge Date: 11/16/72

Latest Revisor: Kathy Mapes Date: 10/05/73

MODULE ABSTRACT

CLURDR reads in user control cards, prints out the user's requests, and calls the subroutine RDFLDS to read in data cards.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLURDR

CALL CLURDR (BEGCOV,BEGAVE,BEGVEC,BEC

Output Arguments

BEGCOV - position in ARRAY where covariances are held.

BEGAVE - position in ARRAY where means begin.

BEGVEC - position in ARRAY where vectors to be clustered begin.

BEGICN - position in ARRAY where number of points per cluster begins.

BEGSUM - position in ARRAY where sums of variances are held.

FLDBAS - position in ARRAY where field boundaries begin.

BEGCVS - position in ARRAY where covariances are stored.

Note: ARRAY is the REAL*8 (12500) array in GLOCOM.

2. Internal Description

CLURDR Is the card reader for the Cluster function. It first reads in all control cards up to and including the 'DATA' card and prints them out on the printer; at this point it checks the values of 'MINCLS' and 'MAXCLS' and checks to see whether or not the required 'CHANNELS' card was read in. The user is given an opportunity to correct any errors encountered in the above checks, and the program continues.

The user's requests are then printed out with the heading 'Cluster Processing Information', the sizes of the arrays to hold the means, covariances, etc., are calculated, and finally the subroutine RDFLDS is called to read in the field description cards and order the fields by run and then line number. Control then returns to the supervisor, CLUSUP.

Subroutines called by CLURDR:

CTLWRD
ERPRNT
CHANEL
CTLPRM
FVAL
IVAL
BCDVAL
RDFLDS

Commons in CLURDR:

GLOCOM
CLUCOM

3. Input Description

Not Applicable

4. Output Description

File name - User requests
DSRN - PRNTR
Device type - Printer
Usage - Output for every CLUSTER job
Description - IDNAME, maximum classes, threshold, convergence, minimum field size, and interval values are printed out.

File name - Informational and error messages
DSRN - PRNTR and TYPEWR
Device type - Printer and typewriter
Uescription - Message numbers are listed below. For text of message, see User's Manual.

MESSAGES

ERROR	INFORMATIONAL
104	* 34
171	*162
432	*163
530	*164
560	245

*means printed on the typewriter only

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 173

MODULE IDENTIFICATION

Module Name: CLUMP Function Name: CLUSTER

Purpose: Calls "clumping" routine and calculates new means and variances.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: NANCY SHEN Date: 03/07/73

MODULE ABSTRACT

CLUMP calls the routine CLUMP1 to cluster the vectors about the cluster centers, calculates new means, and loops back to call CLUMP1. When clustering is successful, the variances are calculated, and means and variances are printed out.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised September 1974

1. Module Usage

CLUMP

CALL CLUMP (VECTP,AVE,COVAR,CAT,ICNT,SUM)

Input Arguments

AVE - REAL*4 (MXCLS,NCHAN), The array containing the means, where MXCLS is the maximum number of clusters requested and NCHAN is the number of channels requested.

VECTP - INTEGER*2 (DIMVEC), The array containing the vectors to be clustered.

Output Arguments

COVAR - REAL*8 (MXCLS,1), The array containing the covariances.

CAT - INTEGER*2 (INTOT), The array containing the clustered vectors; INTOT is the total number of vectors.

ICNT - INTEGER*2 (MXCLS), The array containing the number of vectors per cluster.

SUM - REAL*8 (MXCLS,NCHAN), The array containing the sum of vector values for each channel and cluster.

AVE - (see input argument)

2. Internal Description

The main object of this program is to keep looping through the call to CLUMP1, which does the clustering, until the percent of vectors unchanged by CLUMP1 is greater than or equal to CONV (the user supplied value or the default of 100.0).

First, the arrays ICNT, COVAR, and SUM are initialized to zero and CLUMP1 is called. If clustering was successful, new means are calculated, and the percent of vectors unchanged by CLUMP1 is compared to CONV. If the percent unchanged is greater than or equal to CONV, then clustering for the correct set of cluster points is essentially done. The only job left to be done is clustering the remaining vectors if the user supplied value of INTV was greater than one, calculate the latest means and variances, print them out, and return to CLUPRO.

If the percent unchanged was less than CONV, then the program loops back to the plaed where the arrays ICNT, COVAR and SUM are initialized, and clustering is performed again.

If clustering is not successful, a flag is set to 1 and control returns to CLUPRO which prints out message I0170.

Not that in the loop in which CLUMP1 is called there is a message (I0172) printed out on the typewriter every five minutes to assure the user that the program is still working on clustering. Message I0171 is printed out on the typewriter if clustering was successfully completed.

Subroutines called by CLUMP:

TSTREQ
CLUMP1
TIMER
CPFUNC

Commons in CLUMP:

GLOCOM
CLUCOM

3. Input Description

Not Applicable

4. Output Description

File name - Points/Cluster and Means
DSRN - PRNTR
Device type - Printer
Usage - If clustering is successful, print this out before the cluster maps are printed.
Description - Two lines of headings, and then one line for each cluster containing the number of the cluster, the number of vectors associated with it, and the mean value for each channel for those associated vectors.

File name - Cluster Covariances
DSRN - PRNTR
Device type - Printer
Usage - Used to calculate variances and print them out after Points/Cluster and Means above.
Description - Two header lines, and then one line for each cluster containing the cluster number and variances for each channel.

File name - Informational messages
DSRN - TYPEWR
Device type - Typewriter
Usage - Output
Description - Message numbers are listed below. For text of message, see User's Manual.

INFORMATIONAL MESSAGES

171
172

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 174

MODULE IDENTIFICATION

Module Name: CLUMP1 Function Name: CLUSTER

Purpose: To clump requested data about cluster centers.

System/Language: CMS/ASSEMBLER

Author: Date:

Latest Revisor: B. Davis Date: 07/26/73

MODULE ABSTRACT

CLUMP1 decides which cluster point to associate each vector with, counts the number of vectors per cluster, calculates values for the array SUM, counts the number of vectors unchanged from the previous iteration, and then returns to CLUMP.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLUMP1

```
CALL CLUMP1 (INTOT,NCHAN,NOMOD,ICORR,MXCLS,VECTP,AVE,
             CAT,SUM,ICNT,INTV)
```

Input Arguments:

INTOT - INTEGER*4, Total number of vectors to be clustered.

NCHAN - INTEGER*4, Number of channels requested.

NOMOD - INTEGER*4, Current number of cluster points.

MXCLS - INTEGER*4, Maximum number of clusters requested.

VECTP - INTEGER*2 (DIMVEC), Array containing vectors to be clustered.

AVE - REAL*4, (MXCLS,NCHAN), Array containing means.

CAT - INTEGER*2 (INTOT), Clustered array.

INTV - INTEGER*4, Clustering interval for given fields.

Output Arguments:

ICORR - INTEGER*4, Number of vectors unchanged from previous iteration.

CAT - INTEGER*2 (INTOT), Clustered array.

SUM - REAL*8 (MXCLS,NCHAN), Array containing sum of vector values for each channel and each cluster.

ICNT - INTEGER*2 (MXCLS), Array containing number of vectors per cluster.

2. Internal Description

This program determines the Euclidean distance from each vector to each cluster center and associates each vector to the nearest cluster center by assigning it the number of this cluster center in the corresponding poistion in the CAT array. If a vector is assigned to the same cluster center as it was in the previous iteration (call to CLUMP1), ICORR is incremented by 1.

The arrays SUM and ICNT are initialized to zero before CLUMP1 is entered, so new values are computed for these arrays. Control then returns to CLUMP.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 175

MODULE IDENTIFICATION

Module Name: CLUPRO Function Name: CLUSTER

Purpose: Main processor and output routine for Cluster function.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: KATHY MAPES Date: 10/04/73

MODULE ABSTRACT

CLUPRO is the main processor for the Cluster function. After it is entered, control does not return to the supervisor until clustering and all printed and/or punched output are completed for the function. It establishes the initial cluster points and then calls CLUMP to cluster the data, MDIST to compute separability information and cluster grouping information, and PCHFLD to punch field cards, if requested. CLUPRO also computes the statistics information for the statistics file and punches the file on cards if requested.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised September 1974

1. Module Usage

CLUPRO

CALL CLUPRO (VECTP,COVAR,AVE,CAT,ICNT,SUM,COVRS)

Input Arguments

VECTP - INTEGER*2 (DIMVEC), array containing vectors to be clustered; DIMVEC is in CLUCOM.

Output Arguments

COVAR - REAL*8 (MXCLS,1), array containing covariances; MXCLS in CLUCOM.

AVE - REAL*4 (MXCLS,NCHAN), array containing means; MXCLS and NCHAN in CLUCOM.

CAT - INTEGER*2 (INTOT), array containing clustered vectors; INTOT in CLUCOM.

ICNT - INTEGER*2 (MXCLS), array containing number of points per cluster; MXCLS in CLUCOM.

SUM - REAL*8 (MXCLS,NCHAN), array containing sums of vector values for each channel and cluster; MXCLS and NCHAN in CLUCOM.

COVRS - REAL*8 (MXCLS,1), array containing covariances for punched output; MXCLS in CLUCOM.

2. Internal Description

This program is the main processor and output routine for the Cluster function. The following operations are repeated for each set of cluster points.

First, "Clustering Information", including channel information, is printed out. Then MXCLS initial cluster centers are determined and subroutine CLUMP is called to "cluster" the vectors about these centers and print out the means and variances. If clustering was successful, processing continues; if not, the number of cluster centers is reduced by one, message I0170 is printed out, and we begin again to establish the initial cluster centers.

Next, CLUPRO will print out "Field Information", a map of the field using the symbols requested or the default grey scale symbols, and a chart showing the number of points per cluster. It prints this information for each field, prints out histograms if requested, and calls PCHFLD to punch field description cards if requested. It then calls MDIST to print out "Separability Information" and the "Cluster Grouping Table", calculates statistics, and punches a statistics deck if requested. If NOMOD (number of clusters) is greater than the minimum number requested (MINCLS), control passes back the establishment of initial centers.

Subroutines called by CLUPRO:

CLUMP
MDIST
PCHFLD

Commons in CLUPRO:

GLOCOM
CLUCOM

3. Input Description

File name - Field Information
DSRN - CLUSTX
Device type - Disk
Usage - To obtain information for each field
Description - Information given by user on field description cards.

4. Output Description

File name - Clustering Information
DSRN - PRNTR
Device type - Printer
Usage - Printed out for each new set of cluster centers
Description - Number of clusters, clustering unit size, clustering interval, threshold, and one line of channel information for each channel requested.

File name - Field Information
DSRN - PRNTR
Device type - Printer
Usage - Printed out for each field clustered
Description - Information from field description card, a map of the clustered field depicting different clusters with different symbols, and a chart with the number of points per cluster.

```
File name    - CLASS card
DSRN         - PNCH
Device type  - Punch
Usage        - Punched at the beginning of each set of
               field description cards that constitute
               a class.
Description  - First word is 'CLASS'; class (or cluster)
               number and total number of clusters are
               also punched out.

File name    - Field Card
DSRN         - PNCH
Device type  - Punch
Usage        - Punched for each field
Description  - Field cards in LARS 12 format are punched
               for each sequence of points in a single
               class greater than or equal to the value
               of MINPOINTS.

File name    - Statistics file
DSRN         - SDATA
Device type  - Temporary disk
Usage        - Saves statistics information for input to
               Classifypoints, Separability, and Sampleclassify.
Description  - The statistics file consists of the following
               information
               1) statistics file identifier
               2) class card and dummy field card for each
                  class
               3) number of classes, fields and channels
               4) spectral band and calibration information
                  for each channel
               5) number of points in each cluster
               6) means for each class
               7) covariance matrix for each class
               8) end of file indicator.

File name    - Statistics file
DSRN         - PNCH
Device type  - Punch
Usage        - Punched deck of statistics file
Description  - The statistics file consists of the following
               information
               1) statistics file identifier
               2) class card and dummy field card for each
                  class
               3) number of classes, fields and channels
               4) spectral band and calibration information
                  for each channel
               5) number of points in each cluster
               6) means for each class
               7) covariance matrix for each class
               8) end of file indicator
```

File name - Informational Messages
DSRN - TYPEWR and PRNTR
Device type - Typewriter and printer
Description - Message numbers are listed below. For text of message, see User's Manual.

INFORMATIONAL MESSAGES

*168
*169
170

*means printed on the typewriter only

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 176

MODULE IDENTIFICATION

Module Name: FIXFLD Function Name: CLUSTER

Purpose: Print fields and read data from tapes

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: PAUL SPENCER Date: 02/01/74

MODULE ABSTRACT

FIXFLD prints out the fields to be clustered, first in the order supplied, then in the order processed. It also prints out run information for each run and reads the requested values from the multispectral image storage tapes.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

FIXFLD

CALL FIXFLD (BEGVEC,FLDARY,DATBF,DAT2,BUFARY)

Input Arguments:

BEGVEC - INTEGER*4, position in ARRAY where vectors to be clustered will be held.

FLDARY - INTEGER*4 (10,NOFLD), where NOFLD is the number of fields; FLDARY is the array which contains the field information for the vectors to be clustered.

Output Arguments:

DATBF - REAL*4 (DIMDTB), where DIMDTB is 18*NOFLD; same as BUFARY.

DAT2 - INTEGER*2 (DIMDT2), where DIMDT2 is DIMDTB*2; same as BUFARY.

BUFARY - INTEGER*4 (18,NOFLD), array containing vectors to be clustered.

Note: - ARRAY is the REAL*8 array in GLOCOM.

2. Internal Description

FIXFLD is the routine that prints the fields to be clustered and obtains the data from the multispectral image storage tapes. It first prints out the fields from the disk unit CLUSTX in the order supplied by the user and calculates the total number of locations in ARRAY that will be used by the Cluster function. If ARRAY is not large enough, the number of vectors is automatically reduced by first increasing the line interval and then the column interval, if needed. Message I0167 is printed out if the number of vectors are reduced.

Next, the fields stored on CLUSTX are reordered according to run number and then line number. To do this reordering, FLDARY(10,I), is used. FLDARY(10,I) is the pointer set up by the subroutine RDFLDS (called by CLURDR) to show order of increasing run and line number.

Then the data for each field is read in from the multispectral image storage tapes by calls to GETRUN and GADLIN and ID information for each run is printed out under the heading "RUNS PROCESSED". After the data from the last tape has been read in, the tape unit is detached.

The fields to be clustered are printed out from CLUSTX, for the second time, in the order that the data has been read in, and thus in the order the data will be processed. Control now returns to CLUSUP.

List of subroutines called by FIXFLD:

GETRUN
RUNERR
ERPRNT
GADLIN
LINERR
TOPRU
CPFUNC

Commons in FIXFLD:

GLOCOM
CLUCOM

3. Input Description

File name	-	Fields to be clustered
DSRN	-	CLUSTX
Device type	-	Disk
Usage	-	Used to print out the fields to be clustered in the order supplied by the user
Description	-	Each record contains all the information given on a field description card

File name - Multispectral image storage tape
DSRN - DATAPE
Device type - Tape
Usage - Used to obtain ID information and data for fields requested
Description - Master input tape containing digitized scanner data

4. Output Description

File name - Fields to be clustered
DSRN - PRNTR
Device type - Printer
Usage - All fields supplied by user are printed out from CLUSTX
Description - One line printed out for each field description card supplied; each line contains all the information given on the card; these lines are printed out twice, first in the order supplied, then in the order processed.

File name - Runs processed
DSRN - PRNTR
Device type - Printer
Usage - To print some ID information for each run
Description - Four lines for each run are printed out and these lines contain the 'IDNUMBER' supplied by the user, run number, date of run, flight line, time, tape number, and altitude.

File name - Fields to be clustered
DSRN - CLUSTX
Device type - Disk
Usage - Used to print out the fields to be clustered in the order to be processed; also used by the subroutine CLUPRO.
Description - Each record contains all the information given on a field description card. There is one record per card.

File name - Informational and error messages
DSRN - TYPEWR and PRNTR
Device type - Typewriter and printer
Description - Message numbers are listed below. For text of message, see User's Manual

MESSAGES

ERROR	INFORMATIONAL
561	*167

* means printed on typewriter only

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 177

MODULE IDENTIFICATION

Module Name: MDIST Function Name: CLUSTER

Purpose: Computes separability information and merges modes if necessary.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P.D. Alenduff Date: 12/20/74

MODULE ABSTRACT

MDIST is entered after the data has been clustered about the current points. It computes separability information and cluster grouping information and prints them out.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

MDIST

CALL MDIST (AVE, COVAR, ICNT, SQUOT, CAT, RCAT)

Input Arguments

AVE - REAL*4 (MXCLS,NCHAN), the array containing the means; MXCLS is the number of clusters, NCHAN the number of channels.

COVAR - REAL*8 (MXCLS,1), the array containing the covariances.

ICNT - INTEGER*2 (MXCLS), the array containing the number of points per cluster.

CAT - INTERGER*2 (25000), array containing output group assignment, order value, and quotient values.

RCAT - REAL*4 (12500), equivalenced to CAT array.

Output Argument

SQUOT - REAL*4, smallest separability quotient between two clusters.

2. Internal Description

MDIST computes separability information and recommended grouping of clusters and prints them out.

The variables D(I,J), D(I), D(J), and QUOT that are printed out under "Separability Information" are explained below. D(I,J) is D1 in the coding, D(I) is D2, D(J) is D3, and QUOT is D5. D1 is computed first and is the Euclidean distance between the means of clusters I and J. D2 is a measure of the spread of cluster I in the direction of the mean of cluster J; D3 is a measure of the spread of cluster J in the direction of the mean of cluster I; finally, D5 = D1/(D2 + D3) is the measure of separability between clusters I and J. A separability of 1 implies that the two clusters are fairly separable. The higher the value of SQUOT, the better the separability.

MDIST then calls CGROUP to calculate the recommended grouping of the original clusters and prints this information out in the "Cluster Grouping Table".

commons in MDIST:

GLOCOM
CLUCOM

Subroutines called by MDIST:

CGROUP

3. Input Description

Not Applicable

4. Output Description

File name	- Separability Information
DSRN	- PRNTR
Device type	- Printer
Usage	- Printed out after cluster maps for all fields have been printed.
Description	- Headings are I, J (a pair of clusters), D(I,J), D(I), D(J), D(I) + D(J), QUOT, a line is printed out for each combination of cluster points. See internal description for definitions of D(I,J), etc. The last line gives the average of all the quotient values.

File name	- Cluster Grouping Table
DSRN	- PRNTR
Device type	- Printer
Usage	- Printed out after the Separability Information.
Description	- The column headed "GROUP" gives the recommended group number. The column "CLUSTERS" gives the original cluster numbers comprising the group and the "No. Pts." column gives the number of samples in each of those clusters. The threshold used to compute the groupings is given in the heading.

Information message I0174 is listed on the printer whenever the separability of cluster combinations cannot be computed. This action prevents taking the square root of a negative number.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 178

MODULE IDENTIFICATION

Module Name: PCHFLD Function Name: CLUSTER

Purpose: Punch field description cards and/or clustered array deck

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: J. B. Etheridge Date: 12/12/72

MODULE ABSTRACT

A deck of field description cards is punched for each cluster.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

PCHFLD

CALL PCHFLD (CLASS, CAT)

Input Arguments

CLASS - INTEGER*4, Class (or cluster center) number.

CAT - INTEGER*2(1), Clustered array.

PCHFLD is entered if field description cards are to be punched for a particular class and field.

2. Internal Description

Initialization consists mainly of obtaining field boundary information from the array INFO in CLUCOM and computing the number of lines, columns, and samples in the field. Then control is transferred to punching field description cards.

If field description cards are desired, the program checks the value of MINPTS given by the user (or else the default value) and searches the first line of the field to see if there is a set of consecutive samples of the CLASS requested such that the number of samples in this set is greater than or equal to MINPTS. The entire line is searched for all such sets, and, if there is at least one found, a field description card is punched for this line, as described in the output description section below. The program then loops until all lines in the field have been searched and then control returns to CLUPRO.

Commons in PCHFLD:

GLOCOM
CLUCOM

3. Input Description

Not Applicable

4. Output Description

File name - Field description cards
DSRN - PNCH
Device type - Punch

Usage - Cards are punched if PCHFLD is entered.
Description - Besides the run number and line and column information, field type is punched as NS-I/J, where I is CLASS and J is the total number of classes, and the related field number is punched in columns 15 - 18.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 179

MODULE IDENTIFICATION

Module Name: CGROUP Function Name: CLUSTER

Purpose: Calculates recommended cluster grouping.

System/Language: CMS/FORTRAN

Author: P. H. SWAIN Date: 09/03/73

Latest Revisor: Date:

MODULE ABSTRACT

CGROUP computes the optimal grouping of cluster classes to minimize the total number of classes while ensuring that multimodal class distributions are avoided.

1. Module Usage

CGROUP

CALL CGROUP (LST,N,D,THRESH,ORDR,NG,ND)

Input Arguments:

N - INTEGER*4, number of cluster classes.

D - REAL*4 (ND), array containing quotient values.

THRES - REAL*4, threshold value used in computing groups.

ND - INTEGER*4, number of quotient values.

Output Arguments:

LST - INTEGER*2 (N), output array containing groupings.

ORDR - INTEGER*2 (ND), pointer to order in D.

NG - INTEGER*4, number of group.

2. Internal Description

The computation of the optimal cluster grouping is performed using the following algorithm. The quotient values are denoted by D_{ij}, i, j=1, ..., ND, and the cluster groups are denoted by C_i, i=1, ..., NG, NG<ND.

(1) Assign each cluster to its own group, C_1, ..., C_n.

(2) Order the D_{ij}'s from smallest to largest using the ORDR array and work through the list beginning with the smallest as follows.

(3) If D_{xy} > THRESH, stop (merging is complete).

(4) If $C_x = C_y$ (i.e., belong to the same group) get next value of D_{ij} and go to step 3.

(5) Compute the average distance between C_x and each other group C_u for which $D_{ab} \leq$ THRESH for all a in C_x and all b in C_u. Similarly, compute the average distance between C_y and each other group C_u. If the distance between C_x and C_y is less than all of the other inter-group distances, assign C_x and C_y to the same group.

(6) Select the next D_{ij} and return to step 3.

Subroutines called by CGROUP:

INVPNT

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 180

MODULE IDENTIFICATION

Module Name: INVPNT Function Name: CLUSTER

Purpose:

System/Language: CMS/FORTRAN

Author: P. H. SWAIN Date: 09/03/73

Latest Revisor: Date:

MODULE ABSTRACT

INVPNT is called by CGROUP to derive the i,j pair coordinates corresponding to a given point.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

INVPNT

```
CALL INVPNT (I,J,N,POINT)
```

Input Arguments

N - number of points in array.

POINT - input point whose coordinates are desired.

Output Arguments

I,J - coordinates associated with POINT.

2. Internal Description

Given a point representing a position in the quotient array, INVPNT will derive the i,j pair coordinates of the point.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 181

MODULE IDENTIFICATION

Module Name: CCHIS Function Name: CLUSTER

Purpose: Prints histograms for CLUSTER classes.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: G. F. SANTNER Date: 03/06/73

MODULE ABSTRACT

CCHIS is called from CLUPRO to print out each histogram of every class for every channel.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CCHIS

```
CALL CCHIS (CCI,JFEAT,JCHAN,TALLY,CINSZJ,LLT,PTS,LLBND,
            ULBND,HEAD,HED1,HED2,TIME,DATE,PAGSIZ)
```

Input Arguments

CCI - INTEGER*4, cluster class being histogrammed.

JFEAT - INTEGER*4, index of channel being histogrammed.

JCHAN - INTEGER*2, channel being histogrammed.

TALLY - INTEGER*2, (100), buffer for histogram data.

BINSZJ- REAL*4, value of low edge of first bin.

LLJ - REAL*4, value of low edge of first bin.

PTS - INTEGER*2, number of points in the class.

LLBND - REAL*4, lower limit of spectral band on tape.

ULBND - REAL*4, upper limit of spectral band on tape.

HEAD - INTEGER*4 (79), variable format for printing page header.

HED1 - INTEGER*4 (16), array for storing a 64-character first line header.

HED2 - INTEGER*4 (16), array for storing a 64-character second line header.

TIME - INTEGER*4 (5), array for storing the time of program execution.

DATE - INTEGER*4 (5), array for storing the date of program execution.

PAGSIZ- INTEGER*4, lines per printer page.

2. Internal Description

CCHIS takes the histogram data calculated in CLUPRO and scales and prints the histogram.

3. Input Description

Not Applicable

4. Output Description

File Name - Histograms & related information
DSRN - PRNTR
Device Type - Printer
Usage - Printed out for each histogram
Description - For each new cluster class or top of new page prints header information and cluster class number and number of samples in that class.

- For each histogram prints a subhead including channel number; spectral band; and number of points represented by * and a graph of the histogram with appropriate scales.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 190

MODULE IDENTIFICATION

Module Name: GRHSUP   Function Name: LINEGRAPH COLUMNGRAPH GRAPHHISTOGRAM

Purpose: Supervisor for GLIN, GCOL and GRHIST

System/Language: CMS/FORTRAN

Author: B. K. Addessio   Date: 09/25/72

Latest Revisor: Date:

MODULE ABSTRACT

This module is the supervisor for three graphing functions. GRHSUP determines which subroutines will be called for each processor function; it does not perform any computations.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GRHSUP

CALL GRHSUP(FUNC)

Input Arguments:

FUNC - INTEGER*4, identifies which function will be processed.

Output Arguments:

Not Applicable

Non-standard Returns:

Not Applicable

GRHSUP receives control from LARSMN. Depending on the value of FUNC control is directed to one of three sets of calling sequences for the GRHIST, GLIN or GCOL functions.

2. Internal Description

GRHSUP can be divided into three sets of calling sequences, one for each function. The first call in each of these sequence of calls (CALL COLRDR, CALL GRHRDR, CALL LINRDR) directs control to a function reader subroutine. This reader subroutine reads and interprets the control cards and transfers control back to GRHSUP. Note that GLIN and GCOL share the same reader subroutine with CALL COLRDR being a second entry point to the LINRDR subroutine. The next set of calls (CALL GCOL, CALL GRHIST, CALL GLIN) transfers control to the related processor subroutines for execution. Upon receiving control from the processor subroutines, GRHSUP then transfers control back to LARSMN.

Complete list of subroutines called by GRHSUP:

TSTREQ
COLRDR
GCOL
GRHRDR
GRHIST
LINRDR
GLIN

Commons in GRHSUP:

GLOCOM

3. Input Description

Not Applicable

4. Output Description

File name - Informational messages
DSRN - PRNTR and TYPEWR
Device type - Printer and typewriter
Usage - Output
Description - Message numbers are listed below, for text of message see User's Manual.

MESSAGES

INFORMATIONAL

123
125
126
128
129
131

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 191

MODULE IDENTIFICATION

Module Name: GCOL Function Name: COLUMNGRAPH

Purpose: Processor routine

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. K. Addessio Date: 09/25/72

MODULE ABSTRACT

This subroutine reads and graphs spectral responses for requested columns of data from the Multispectral Image Storage Tape.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GCOL

```
CALL GCOL(RDATA,BRDATA,NCR,NCD,NSD,MODE,CSEL,CSET,
          CHAN,PRINT,SWITCH)
```

Input Arguments:

RDATA - REAL*4, array(NSD,NCD), MIST data read by GADLIN

BRDATA - INTEGER*4, buffer array for reading MIST data as used by GADLIN

NCR - INTEGER*4, number of channels requested as defined by CHANEL

NCD - INTEGER*4, number of channels dimensioned in RDATA used by GADLIN

NSD - INTEGER*4, number of samples dimensioned in RDATA used by GADLIN

MODE - INTEGER*2, array(8), general purpose array for subroutine communication indicating types of requests

CSEL - INTEGER*2, array(30), channel select vector defined by CHANEL

CSET - REAL*4, array(3,30), channel calibration vector defined by CHANEL

CHAN - INTEGER*2, array(30), selected channel, defined by CHANEL

PRINT - INTEGER*2, array(8), data block parameter selected

Print(1) = initial line of data
Print(2) = last line of data
Print(3) = line interval
Print(4) = current line requested
Print(5) = initial sample of data
Print(6) = last sample of data
Print(7) = sample interval
Print(8) = not used

SWITCH - REAL*4, array(3), general purpose communication array

Output Arguments:

Not Applicable

Non-standard Returns:

Not Applicable

GCOL is the processor routine for the columngraph function.

2. Internal Description

Axis labels are set. A graph is plotted for each requested column of data. Header information and graph axes are printed. A call to GADLIN reads the requested columns of data from the multispectral image storage tape. A line of the graph is calculated and printed on the line printer with the appropriate characters for each of the requested lines. CALL TSTREQ periodically checks for operator intervention allowing for processor termination due to a STOP command. Additional checks are made for indications that calibration graphs are requested. If so, appropriate graphs are produced. Control returns to GRHSUP.

Subroutines called by GCOL:

GADLIN
GETIME
LINERR
TSTREQ

Commons used in GCOL:

GLOCOM

3. Input Description

File name - Multispectral Image Storage Tape
DSRN - DATAPE
Device type - TAPE
Usage - Input
Description - Master input containing digitized scanner data

4. Output Description

File name - Header and graph of column
DSRN - PRNTR
Device type - Printer
Usage - Output for any column graph job
Description - For each column requested:

1) header information including run number, flight line, data tape, reformatting date, date, time, altitude, ground heading, spectral band widths and calibration information for each channel requested.
2) one or more of the following graphs: See the PRINT control card in the Control Card Dictionary for specific usage and restrictions).

    a. graph of column for requested lines
    b. graph of ROLL parameter
    c. graph of C0 calibration value
    d. graph of C1 calibration value
    e. graph of C2 calibration value

File name - Informational messages
DSRN - PRNTR
Device type - Printer
Usage - Output
Description - Message numbers are listed below, see User's Manual for text of message.

MESSAGES

INFORMATIONAL

023
057
058

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 192

MODULE IDENTIFICATION

Module Name: GLIN Function Name: LINEGRAPH

Purpose: Processor Module

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. K. Addessio Date: 09/25/72

MODULE ABSTRACT

This subroutine reads and graphs spectral responses for requested lines of data from the Multispectral Image Storage Tape.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GLIN

```
CALL GLIN(RDATA,BRDATA,NCR,NCD,NSD,MODE,CSEL,CSET,
          CHAN,PRINT,SWITCH)
```

Input Arguments:

RDATA - REAL*4, array (NSD,NCD), MIST data read by GADLIN

BRDATA - INTEGER*4, buffer array for reading MIST data as used by GADLIN

NCR - INTEGER*4, number of channels requested as defined by CHANEL

NCD - INTEGER*4, number of channels dimensioned in RDATA used by GADLIN

NSD - INTEGER*4, number of samples dimensioned in RDATA used by GADLIN

MODE - INTEGER*2, array(8), general purpose array for subroutine communication indicating types of requests

CSEL - INTEGER*2, array(30), channel select vector defined by CHANEL

CSET - REAL*4, array(3,30), channel calibration vector defined by CHANEL

CHAN - INTEGER*2, array(30), selected channel, defined by CHANEL

PRINT - INTEGER*2, array(8), data block parameter selected

Print(1) = initial line of data
Print(2) = last line of data
Print(3) = line interval
Print(4) = current line requested
Print(5) = initial sample of data
Print(6) = last sample of data
Print(7) = sample interval
Print(8) = not used

SWITCH - REAL*4, array(3), general purpose
communication array

Output Arguments:

Not Applicable

Non-standard Returns:

Not Applicable

GLIN is the processor routine for the linegraph function.

2. Internal Description

Header information and graph axes are printed. A call to GADLIN reads the requested line of data from the multispectral image storage tape. A line of the graph is calculated and printed with the appropriate characters for each of the requested columns. CALL TSTREQ periodically checks for operator intervention allowing for processor termination due to a STOP command. Control is returned to GRHSUP.

Subroutines called by GLIN:

GADLIN
GETIME
LINERR
TSTREQ

Commons used in GLIN:

GLOCOM

3. Input Description

File name - Multispectral Image Storage Tape
DSRN - DATAPE
Device type - Tape
Usage - Input
Description - Master input containing digitized scanner data

4. Output Description

File name - Header and graph of line
DSRN - PRNTR
Device type - Printer
Usage - Output for any linegraph job
Description - For each line requested:
1) header information including run number, flight line, data tape, time, altitude, ground heading, spectral band widths and calibration information for each channel requested.
2) graph of line for requested columns.
3) means and standard deviations of C0,C1,C2

File name - Informational messages
DSRN - PRNTR and TYPEWR
Device type - Printer and typewriter
Usage - Output
Description - Message numbers are listed below, see User's Manual for text of message

MESSAGES

INFORMATIONAL

023
053

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 193

MODULE IDENTIFICATION

Module Name: GRHRDR Function Name: GRAPHHISTOGRAM

Purpose: Reader Module for GRHIST

System/Language: CMS/FORTRAN

Author: B. K. Addessio Date: 09/25/72

Latest Revisor: Date:

MODULE ABSTRACT

This subroutine reads and interprets the control cards necessary to run the Graphhistogram function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GRHRDR

CALL GRHRDR (NCR,CHAN)

Input Arguments:

Not Applicable

Output Arguments:

NCR - INTEGER*4, number of channels requested

CHAN - INTEGER*2, Array(30), identifies channels requested

Non-standard Returns:

Not Applicable

This subroutine reads and checks the control cards necessary to run the Graphhistogram function. The NCR and CHAN parameters are defined by the CHANEL subroutine.

2. Internal Description

The control cards necessary to run GRAPHHISTOGRAM are interpreted by a call to CTLWRD. If an end of file is reached ERRCOR is set to 2 and the job is terminated by a call to ERPRNT. Otherwise, control is passed on the value of CODE.

The channels card is interpreted by a call to CHANEL where the values of NCR and CHAN are defined. If an error appears on the card, a message is printed at the terminal and the user is given an opportunity to type in a corrected card. When an END card is reached control is transferred back to GRHSUP.

Subroutines called by GRHRDR:

CTLWRD
ERPRNT
CHANEL

Commons in GRHRDR:

GLOCOM

3. Input Description

Not Applicable

4. Output Description

File Name	- Informational and Error Messages
DSRN	- PRNTR and TYPEWR
Device type	- Printer and typewriter
Usage	- Output
Description	- Message numbers are listed below, for text of message see User's Manual.

MESSAGES

ERROR	INFORMATIONAL
139	059
171	

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 194

MODULE IDENTIFICATION

Module Name: LINRDR
Function Name: LINEGRAPH COLUMNGRAPH

Purpose: Reader module for GLIN and GCOL

System/Language: CMS/FORTRAN

Author:
Date:

Latest Revisor: B. K. Addessio
Date: 09/25/72

MODULE ABSTRACT

This subroutine reads and interprets the function control cards necessary to run the GLIN and GCOL functions, the related entry points being CALL LINRDR and CALL COLRDR. It also performs checks for complete and accurate control information.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LINRDR

```
CALL LINRDR(BRDATS,NCR,NCD,NSD,MODE,CSEL,CSET,CHAN,
            PRINT,SWITCH)
```

Input Arguments:

Not Applicable

Output Arguments:

BRDATS - INTEGER*4, buffer array for reading MIST data as used by GADLIN

NCR - INTEGER*4, number of channels requested as defined by CHANEL

NCD - INTEGER*4, number of channels dimensioned in RDATA used by GADLIN

NSD - INTEGER*4, number of samples dimensioned in RDATA used by GADLIN

MODE - INTEGER*2, array(8), general purpose array for subroutine communication indicating types of requests

CSEL - INTEGER*2, array(30), channel select vector defined by CHANEL

CSET - REAL*4, array(3,30), channel calibration vector defined by CHANEL

CHAN - INTEGER*2, array(30), selected channel, defined by CHANEL

PRINT - INTEGER*2, array(8), data block parameter selected

Print(1) = initial line of data
Print(2) = last line of data
Print(3) = line interval
Print(4) = current line requested
Print(5) = initial sample of data
Print(6) = last sample of data
Print(7) = sample interval
Print(8) = not used

SWITCH - REAL*4, array (3), general purpose communication array

COLRDR

CALL COLRDR (BRDATS, NCR, NCD, NSD, MODE, CSEL, CSET, CHAN, PRINT, SWITCH)

Input Arguments:

Not Applicable

Output Arguments:

See CALL LINRDR above

LINRDR is the card reader subroutine to read and interpret the function control cards necessary to run both the line-graph and columngraph functions. The parameter lists are the same for each entry point. With the exception of BRDATS, NCD, and NSD, the parameter values are set while reading and interpreting the function control cards which will further be explained below in the section titled Internal Description.

Upon reaching an END card indicating that all control cards have been read and accepted on accuracy, the LINRDR subroutine then checks for completeness and sets up for execution of the linegraph or columngraph functions. It is here that the values of BRDATS, NCD and NSD are set.

2. Internal Description

The control cards are interpreted by a call to CTLWRD. If an end of file is reached ERRCOR is set to 2 and the job is terminated by a call to ERPRNT, otherwise control is passed on the value of CODE.

When interpreting a PRINT card, a call to CTLPPM decodes the parameters and passes control on the value of CODE. CALL IVAL is used to interpret single BCD, integer or real values also passing control on the value of CODE. If interpretation errors occur, a related message is printed to both the printer and typewriter and the user is allowed to type in a corrected card. Depending on requests made the values of PRINT(1) through PRINT(6) and MODE (1,2,3,6,7,8) are set accordingly. MODE is used as a flag set equal to 1 for a requested parameter; MODE(1) indicating C0, MODE (2) - C1, MODE(3) - C2, MODE(4) - not used, MODE (5) - not used, MODE (6) - ROLL, MODE(7) - lines, MODE(8) - columns. When column 72 is reached, the next control card is read and so on. The SCALE card is decoded similarly only FVAL is used rather than IVAL. The values of SWITCH are set here.

The channels card is interpreted by a call to CHANEL where the values of NCR, CSEL, CSET and CHAN are defined.

After all control cards have been read various checks are made once again allowing for additional cards to be typed in by the user. The requested data run is located by a call to GETRUN, the BRDATS, NCR and NSD parameters are set up for requested execution, and control is returned to GRHSUP.

Subroutines called by LINRDR:

CHANEL
CTLPRM
CTLWRD
ERPRNT
FVAL
GETRUN
IVAL
RUNERR

Commons in LINRDR:

GLOCOM

3. Input Description

File name - Multispectral image storage tape
DSRN - DATAPE
Device type - Tape
Usage - Input
Description - Master input tape containing digitized scanner data

4. Output Description

File name - Informational and error messages
DSRN - PRNTR and TYPEWR
Device type - Printer and typewriter
Usage - Output
Description - Message numbers are listed below, for text of message see User's Manual

MESSAGES

INFORMATIONAL	ERROR
051	411-421
052	
054	

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 200

MODULE IDENTIFICATION

Module Name: HISSUP Function Name: HISTOGRAM

Purpose: Supervisor for histogramming routines

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: L. D. England Date: 10/04/72

MODULE ABSTRACT

This module supervises the execution of the histogramming function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

HISSUP

CALL HISSUP

There are no arguments for HISSUP. This routine is called within LARSYS by the main module, LARSMN, when a *HISTOGRAM card is detected within the input control cards. Control is returned to the caller after the completion of the histogramming function.

2. Internal Description

HISSUP first informs the user that the histogram function has been requested and then calls HISRDR to read and interpret the function control cards. If HISRDR detects an uncorrectable error, it will RETURN1 which will cause a return to the caller. When the normal return is taken, HISSUP allocates ARRAY (in GLOCOM) for the three arrays required by HISTD and then calls HISTD to calculate the histogram. An error return from HISTD will cause a return to the caller. The normal return from HISTD will cause a function completed message to be written and then control is returned to the caller.

3. Input Description

Not Applicable

4. Output Description

Two informational messages, I0087 and I0090, are written at the typewriter (symbolic device TYPEWR which is usually DSRN 16). In addition, the printer (symbolic device PRNTR which is usually DSRN 6) is spaced to the beginning of a new page just prior to returning to the caller.

5. Supplemental Information

See the LARSYS System Manual for a description of supervisor module requirements.

6. Flowchart

Not Applicable

LARS Program Abstract 201

MODULE IDENTIFICATION

Module Name: HISRDR Function Name: HISTOGRAM

Purpose: Control card reader and initialization for histogramming

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Ellen McDonald Date: 12/04/73

MODULE ABSTRACT

HISRDR causes histogram function control cards to be read and then interprets them. If the cards are successfully interpreted, the required data tape is mounted and positioned to the correct file (run), any needed default values are set, the user is informed of his selections, and then control returns to the caller.

1. Module Usage

HISRDR

```
CALL HISRDR(NCR,NCD,NSD,MODE,CSEL,CSET,CHAN,
            BLOCK, &XXX)
```

Output Arguments:

NCR - INTEGER*4; The total number of channels requested

NCD - INTEGER*4; The number of channels dimensioned in RDATA (an array used for storing multispectral image data as required for using GADLIN)

NSD - INTEGER*4; The number of samples dimensioned in RDATA

MODE - INTEGER*2 MODE(8); A general purpose array for communicating user requests. For histogramming only the following are defined:

MODE(1) = 1 if accumulating histograms

MODE(2) = 1 if punching histograms

CSEL - INTEGER*2 CSEL(30); The channel select vector as defined for GADLIN and CHANEL

CSET - REAL*4 CSET(3,30); A thirty by three array containing the calibration values (C0, C1, C2) for each of the channels as required for GADLIN

CHAN - INTEGER*2 CHAN(30); An array containing the numbers of the channels selected in ascending order as provided by the module CHANEL

BLOCK - INTEGER*2 BLOCK(8); An array which defines the multispectral data selected for processing.

BLOCK(1) = Initial line of data
BLOCK(2) = Last line of data
BLOCK(3) = Line interval
BLOCK(4) = Current line requested
BLOCK(5) = Initial sample of data
BLOCK(6) = Last sample of data
BLOCK(7) = Sample interval

Non-standard Returns:

&XXX - Return 1 will be used when unrecoverable errors are detected.

HISRDR interprets the function control cards and returns the results in the output arguments. The MODE array will indicate if a PUNCH and/or OPTIONS card(s) were used; the BLOCK array stores the contents of the BLOCK card; and CHAN, CSEL, CSET, and NCR reflect the CHANNELS card. NCD and NSD may be used to allocate storage required when HISTD is called to actually perform the histogramming.

2. Internal Description

After some initialization of variables, CTLWRD is called to read a card and interpret the key word. An unexpected end of file for the control card input results in ERPRNT being called to terminate execution. After CTLWRD has determined the key word, a branch is made to sections of code to further interpret each of the possible cards. CTLPRM, IVAL, and CHANEL are used to assist with this interpretation.

After the END card has been detected, the correct data run must be available. A call to GETRUN will be made if the requested run is different from the current run. RUNERR is used to handle any errors from GETRUN. The BLOCK values are now checked and the defaults set if required. The users interpreted requests are then printed and control is returned to the caller. ERPRNT is used to handle most error messages. The Return 1 is used for errors detected after all control cards have been read (and possibly corrected). The global COMMON, GLOCOM, is the only COMMON required.

3. Input Description

HISRDR does not actually perform any reading operations. It does invoke CTLWRD which performs reads to the control card input stream (card reader, disk or typewriter). In addition, GETRUN is called which requires the RUNTABLE disk file and a multispectral image storage tape.

4. Output Description

Information messages 34 and 76 are printed at the typewriter (TYPEWR/DSRN16) while information messages 75 and 77 are printed at both the typewriter and the printer (PRNTR/DSRN 6). The following error messages are processed by ERPRNT: 139, 151, 152, 153, 171, 417, 464, 465, 467, 468, 469, 470, 511, & 512. HISRDR also prints a summary of the user's requests on the printer.

5. Supplemental Information

See the LARSYS System Manual for a description of how to create control card reading routines.

6. Flowchart

Not Applicable

LARS Program Abstract 230

MODULE IDENTIFICATION

Module Name: PICSUP Function Name: PICTUREPRINT

Purpose: Supervisor for the Pictureprint Function.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P. D. Alenduff Date: 07/09/75

MODULE ABSTRACT

This module supervises the execution of the Pictureprint function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

PICSUP

CALL PICSUP

There are no arguments for PICSUP. This routine is called within LARSYS by the main module, LARSMN, when a *PICTURE-PRINT card is detected within the input control cards. Control is returned to the caller after the completion of the Pictureprint function.

2. Internal Description

PICSUP first informs the user that the Pictureprint function has been requested and then calls PICRDR to read and interpret the function control cards. PICSUP calls PICINT to perform the required initialization operations. PICSUP then calls PIC1 to produce the pictorial printout. After return from PIC1, a function completed message is written and control is returned to the caller. GLOCOM and PICCOM are required COMMONS for this routine.

3. Input Description

Not Applicable

4. Output Description

Two informational messages, I0092 and I0093, are written at the typewriter (symbolic device TYPEWR which is usually DSRN 16).

5. Supplemental Information

See the LARSYS System Manual for a description of supervisor module requirements.

6. Flowchart

Not Applicable

LARS Program Abstract 231

MODULE IDENTIFICATION

Module Name: FLDBOR Function Name: PICTUREPRINT

Purpose: Outline field boundaries to be displayed

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: L. D. England Date: 10/19/72

MODULE ABSTRACT

FLDBOR will take a line of symbols representing multispectral image data and insert the specified character to represent boundaries that affect that given line.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

FLDBOR

```
CALL FLDBOR (SYMBOL,NOUT,LIN,LINESW,RUNSAV,FIELD,NOFLDS,
             PT,BLOCK)
```

Input arguments:

SYMBOL - LOGICAL*1 SYMBOL(3), This array must contain a character that will be used to represent each type of boundary, training and test; plus, a symbol to be used by common boundaries must be in SYMBOL(3).

NOUT - INTEGER*4, The number of points in a gray-scale line.

LIN - LOGICAL*1 LIN(X), The gray scale line to be printed. X is typically 111.

LINESW - LOGICAL*1 LINESW(X), This logical array is used to record where boundaries have been written; a value of .TRUE. means a boundary symbol has been assigned to the corresponding element in LIN.

RUNSAV - INTEGER*4 RUNSAV(X), This array contains all the run numbers that correspond to the boundaries stored in FIELD.

FIELD - INTEGER*2 FIELD(4,X), This array contains the 2 line numbers and the 2 column numbers for each stored boundary.

NOFLDS - INTEGER*4, This variable contains the total number of fields represented in FIELD.

PT - INTEGER*4, This is a pointer to indicate which type of boundary (i.e., which character in SYMBOL, 1 or 2) to use for the boundary.

BLOCK - INTEGER*2 BLOCK(8), An array which defines the multispectral data selected for display.

BLOCK(1) = Initial line of data
BLOCK(2) = Last line of data
BLOCK(3) = Line interval
BLOCK(4) = Current line requested
BLOCK(5) = Initial sample of data
BLOCK(6) = Last sample of data
BLOCK(7) = Sample interval

Output argument:

LIN - Upon return, this input line will have the boundary symbols inserted.

FLDBOR must be called for each line to be considered for having a boundary displayed. It must be called once for each type of boundary to be displayed per line. For example, if both training and test fields are present, this routine must be called twice with the proper arguments set. The output symbols in LIN are then ready to be printed.

2. Internal Description

FLDBOR first uses RUNSAV to determine which boundaries in FIELD are for the current run. It then checks to see if the current line (BLOCK(4)) is contained within each of the stored boundaries for this data run. For those boundaries which affect this line, the proper symbol is stored and the switch in LINESW is turned on. Shared boundaries are determined by the use of LINESW to see if a previous boundary has been inserted. If it has, SYMBOL(3) is used as the character for the boundary. Control is returned to the caller after all boundaries have been checked and the proper characters inserted in LIN.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 232

MODULE IDENTIFICATION

Module Name: PICCOM Function Name: PICTUREPRINT

Purpose: Block Data for PICCOM

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: L. England Date: 10/06/72

MODULE ABSTRACT

This is the BLOCK DATA subroutine for the PICTUREPRINT common block PICCOM.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 233

MODULE IDENTIFICATION

Module Name: PICRDR Function Name: PICTUREPRINT

Purpose: Control card reader for PICTUREPRINT

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P.D.Alenduff Date: 07/09/75

MODULE ABSTRACT

PICRDR causes PICTUREPRINT function control cards to be read and interpreted. The user is informed of his selections and control returns back to the caller.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

PICRDR

CALL PICRDR

PICRDR interprets the function control cards and returns the results in PICCOM, the PICTUREPRINT common block. The MODE array will indicate requests specified by HISTOGRAM, PRINT, PUNCH, and BOUNDARY cards if used; the BLOCK array stores the contents of the BLOCK card; the DSPBLK array stores the contents of the DISPLAY card; and CHAN, CSEL, CSET, and NCR reflect the CHANNELS card.

2. Internal Description

After initialization of some variables, CTLWRD is called to read a card and interpret the keyword. An unexpected end of file for the control card input results in ERPRNT being called to terminate execution. After CTLWRD has determined the keyword, a branch is made to sections of code to further interpret each of the possible options. CTLPRM, IVAL, BCDVAL, CHANEL, and DATCRD are used to assist with this interpretation. When a DATA or END card has been detected, the users requests are displayed on the printer and control is returned to the caller. PICRDR uses global common, GLOCOM, and the PICTURE-PRINT common, PICCOM.

3. Input Description

PICRDR uses CTLWRD for all reading operations from the control card input stream (card reader, disk, or typewriter).

4. Output Description

PICRDR performs output to two devices. Information and error messages appear on the symbolic DSRN TYPEWR, and the summary of the User's requests, in addition to most error messages, appear on the SYMBOLIC DSRN PRNTR. A list of information and error messages produced by PICRDR follows.

Information Messages	Error Messages	
83	139	504
94	151	505
104	152	506
237	153	507
	171	508
	466	509
	500	510
	501	511
	502	512
	503	520

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

CHAN - INTEGER*2 CHAN(30), An array containing the numbers of the channels selected in ascending order as provided by the module CHANEL.

BLOCK - INTEGER*2 BLOCK(8), An array which defines the multispectral data selected for histogramming.

BLOCK(1) = Initial line of data
BLOCK(2) = Last line of data
BLOCK(3) = Line interval
BLOCK(4) = Current line requested
BLOCK(5) = Initial sample of data
BLOCK(6) = Last sample of data
BLOCK(7) = Sample interval

DSPBLK - INTEGER*2 DSPBLK(7), An array which defines the multispectral data selected to be displayed.

DSPBLK(1) - DSPBLK(7) have the same definitions as BLOCK(1) - BLOCK(7)

Non-standard Returns:

&XXX - Return 1 will be used when unrecoverable errors are detected.

PICRDR interprets the function control cards and returns the results in the output arguments. The MODE array will indicate results of the HISTOGRAM, PRINT, PUNCH, and BOUNDARY cards if used; the BLOCK array stores the contents of the BLOCK card; the DSPBLK array stores the contents of the DISPLAY card; and, CHAN, CSEL, CSET, and NCR reflect the CHANNELS card. NCD and NSD may be used to allocate storage arrays if the caller has this requirement.

2. Internal Description

After initialization of some variables, CTLWRD is called to read a card and interpret the key word. An unexpected end of file for the control card input results in ERPRNT being called to terminate execution. After CTLWRD has determined the key word, a branch is made to sections of code to further interpret each of the possible cards. CTLPRM, IVAL, BCDVAL, CHANEL, and DATCRD are used to assist with this interpretation.

After the DATA or END card has been detected, the user's interpreted requests are printed. If a histogram calculation is required, the following is performed: a call to GETRUN will be made if the BLOCK run is different from the current run; RUNERR is used to handle any errors from GETRUN; the BLOCK values are checked and defaults set if required; and, finally, HISTD is called to calculate the histograms.

Now, the set-up for the display routines is as follows: call GETRUN if the DSPBLK run is different from the current data run; use RUNERR as required; the DSPBLK values are checked and defaults supplied as required; and the allocation of ARRAY in GLOCOM is determined.

The next step is to insure that the direct access files FIELD BNDRIES and HISTO DATA exist. This is accomplished by writing a dummy record to each file, thus forcing FORTRAN to format the files if they did not exist previously. If BOUNDARY DELETE or STORE was specified, BONDSU is called to process those requests.

Control is now returned back to the caller. In addition to global COMMON, GLOCOM, the Pictureprint COMMON, PICCOM, is required. ERPRNT is used to handle any error messages and, then, a RETURN1 is used for errors from which there was no recovery.

3. Input Description

PICRDR does not actually perform any reading operations. It does invoke CTLWRD which performs reads to the control card input stream (card reader, disk or typewriter). In addition, GETRUN is called which requires the RUNTABLE disk file and a multispectral image storage tape.

4. Output Description

File Name	Symbolic DSRN	Type	Usage	Description
HISTO DATA	HDATA	Disk	Output	Histogram data file
FIELD BNDRIES	FLDBND	Disk	Output	Field boundary storage
-	TYPEWR	Typewriter	Output	For information messages
-	PRNTR	Printer	Output	User's Request Selection Summary

Detailed layouts of the contents of the above files are contained in the LARSYS System Manual. The information messages that can be produced are numbers 34, 75, 77, 92, 93, 94, and 95. The following error messages are processed via ERPRNT: 139, 151, 152, 153, 171, 464, 465, 466, 467, 468, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, and 512.

5. Supplemental Information

See the LARSYS System Manual for a description of how to create control card reading routines.

6. Flowcharts

Not Applicable

LARS Program Abstract 234

MODULE IDENTIFICATION

Module Name: PIC1 Function Name: PICTUREPRINT

Purpose: Generates gray scale prints

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P.D. Alenduff Date: 07/11/75

MODULE ABSTRACT

PIC1 produces the gray scale printes for PUCTUREPRINT. PIC1 insures that the data has been histogrammed. The specified section of multispectral image data is then assigned symbols based on the reflectance of each point while boundaries may be represented by predefined symbols.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

PIC1

CALL PIC1 (RDATA,BRDATA,TRRSAV,TRNFLD,TSRSAV,TSTFLD)

Input arguments:

RDATA - REAL*4 RDATA (NSD,NCR), This is a real, variable dimension array that PIC1 uses to receive data from GADLIN.

BRDATA - INTEGER*4 BRDATA (N), This variable dimension array is used by PIC1 to provide a data buffer for GADLIN. N should allow for NCR*ID(6) bytes.

TRRSAV - INTEGER*4 TRRSAV(200), This array, will be used to store training field run numbers and must be dimensioned 200 which is the upper limit.

TRNFLD - INTEGER*2 TRNFLD (4,200), This array is used to store training field coordinates, thus the second dimension must be as large as the maximum number of test field boundaries.

TSRSAV - INTEGER*4 TSRSAV (600), This array will be used to store test field run numbers and must be dimensioned 600 which is the upper limit.

TSTFLD - INTEGER*2 TSTFLD (4,600), This array is used to store test field coordinates, thus the second dimension must be as large as the maximum number of test field boundaries which is 600.

The data supplied by the input arguments allows PIC1 to generate the gray scale, pictorial map of PICTUREPRINT. The outputs of this module are the printer output and, optionally, a histogram deck.

2. Internal Description

PIC1 reads all training and test field boundaries into the appropriate storage arrays. PIC1 then begins a loop, once for each channel and once for each width of the display map for each channel.

The number of levels requested is used to assign the pre-programmed symbols from SYMINT in PICCOM unless user symbols were supplied.

The next step is to assign a gray level to all possible data values. This first requires that the data be grouped into bins or levels. This can be accomplished in one of three ways: 1) if MODE(1)=1, then the histograms for the data are read from the HISTO DATA disk file, 2) if MODE(1)=2, user-supplied levels are available in COMMON PICCOM as read by PICINT, and 3) if MODE(1)=3, a user-supplied histogram deck has been read and stored on disk by PICINT and the histogram again comes from the HISTO DATA disk file. Based on the number of levels the user wants displayed, the bias must be recalculated to find equal distribution amoung the levels selected. At this point, if MODE(8)=1, a call to GRHIST will produce a graph of the histogram. Likewise, if MODE(2)=2, the histograms are punched onto cards.

The header for the map is printed utilizing GETIME for the time of day and GLOCOM COMMON for the LARSYS header. GADLIN is called to read a line from the multispectral image storage tape (LINERR processes any errors from GADLIN). A symbol is then assigned to each data point and then, if there are boundaries to be outlined, FLDBOR is called. The line is then printed and TSTREQ is called to determine if the user has issued a STOP command. If he has not, control loops back to get another line via GADLIN. The STOP command will cause control to return to the caller. After all lines for a given channel have been mapped, almost the entire module is executed again for the next channel, if there is one. ERPRNT handles error messages as required.

3. Input Description

Input to PIC1 comsists of the histogram data from the disk file HISTO DATA with the symbolic DSRN HDATA. The Multispectral Image Storage tape (via GADLIN) serves as the only other input from the symbolic DSRN DATAPE. Formats of these input files are contained in the LARSYS System Manual.

4. Output Description

The output from PIC1 consists of printer output, terminal output, and optionally, a punched histogram deck. Printer output on the symbolic DSRN PRNTR includes the pictorial map, all error messages, and most information messages. Terminal output to the symbolic DSRN TYPEWR includes all error and information messages. Optionally a histogram deck of the format listed in the LARSYS System Manual is punched on the symbolic DSRN PNCH. A complete list of information and error messages (produced through ERPRNT) follows.

Information Messages	Error Messages
23	139
84	267
97	473
98	516
99	

5. Supplemental Information

The algorithm for generating the gray scale pictorial map involves the assignment of data values to the proper symbol. This requires that the data be grouped into bins or levels. There are two techniques for doing this: (a) histogram this or related data, or (b) allow the user to input levels cards.

(a) Assuming that a histogram exists, the histogram is integrated to find equal distribution cut-offs for the number of symbols (gray levels) that the user has requested via the SYMBOLS card or its default. The LARSYS histogramming routine reduces the date down to 100 bins with a range of -32767 to +32767 for the values in each bin (a half-word as in array HISTA). This technique allows values of 0 to 65,534 to be stored in a half-word array; but of course, required that 32767 always be added to the accumulation in HISTA before using it. The array LSET contains the histogram bin numbers for the upper limit of each symbol level (maximum of 16 symbols).

The GLEV array can now be calculated to contain the actual data values for the edge of each symbol level. Also, the array LCHAR is now given a symbol which corresponds to each of the 100 bins.

After each data line is read, each data value within the line must be assigned a symbol. This is accomplished by calculating an index into the LCHAR array to get the symbol which corresponds to the bin number that the data value would be assigned if it were histogrammed. The index calculation (the real value calculated will be truncated to an integer) is:

$$\frac{\text{(Real data) - (Lower Limit of Histogram)}}{\text{(Histogram bin size}} + 1$$

In this way, each line of the map is generated where each symbol has an equal chance of appearing based on the data histogrammed.

(b) If the user specifies his own levels with input cards, the above procedure is altered. The array LCHAR is assigned a symbol for each level (16 or less), not for the 100 bins as when histogrammed. Then, after each data line is read, each data value is compared to the input levels as stored in GLEV. When the correct level has been determined, the level number serves as the index into LCHAR to get the proper symbol.

6. Flowchart

Not Applicable

LARS Program Abstract 235

MODULE IDENTIFICATION

Module Name: PICINT   Function Name: PICTUREPRINT

Purpose: Initialization for the Pictureprint function

System/Language: CMS/FORTRAN

Author: P.D. Alenduff   Date: 07/10/75

Latest Revisor: Date:

MODULE ABSTRACT

PICINT supports the Pictureprint function by reading the histogram, levelscards, and field description cards if requested. If histogram calculation is required, HISTD is called. The display values are set, storage allocation is calculated, and the required date tape is mounted and positioned.

1. Module Usage

PICINT

CALL PICINT (PDATAS,PBDATS,TRRSS,TRFLDS,TSRSS,TSFLDS)

Output Arguments:

PDATAS-INTEGER*4-Base address into ARRAY where the RDATA array will begin.

PBDATS-INTEGER*4-Base address into ARRAY where the BRDATA array will begin.

TRFLDS-INTEGER*4-Base address into ARRAY where the training field coordinates will be stored.

TRRSS-INTEGER*4-Base address into ARRAY where all training field run numbers will be stored.

TRFLDS-INTEGER*4-Base address into ARRAY where all test field coordinates will be stored.

TSRSS-INTEGER*4-Base address into ARRAY where all test field run numbers will be stored.

PICINT carries out the required initialization for the PICTUREPRINT function. PICINT reads the histogram deck, field description cards, and levelscards if each option is selected. If histogram calculation is required, PICINT calls HISTD. BONDSU reads the boundary storage cards and writed them to the FILED BNDRIES disk file. The display parameters are checked and all appropriate defaults are invoked. The total number of field boundaries is calculated and sotrage is allocated for the boundary storage arrays.

2. Internal Description

PICINT performs some initialization which includes a call to TSTREQ to initialize its variable. The number of training and test fields is set to zero. Unless HISTOGRAM LEVELSCARDS is used, the number of levels for each channel is set equal to the number of symbols used. If no symbols were specified, the default of 10 levels per channel is applied. If the calculation of the histogram was requested, the channels and parameters are checked against the 10 record of the requested histogram run, proper values are established and a call to HISTD is made. The HISTD DATA file is initialized and boundary maintanence is performed. If HISTOGRAM HISTOCARDS was requested, the histogram is read from cards and written unto disk.

If HISTOGRAM LEVELSCARDS was requested, the NLEV array is zerroed since no levels have been read for any channel. Each card is read and decaded using DATCRD. An error return from DATCRD indicates when the levelscard is in the second of two forms. FVAL is called to find the channel number, calibration code, and levels. The calibration code is checked for legality and the number of levels previously stored is added to the current requested levels to insure that the maximum number of levels has not been exceeded. The levels are checked for increasing values and are then stored in the GLEV array. The channel number and calibration code are checked to compare to previous requests. After all cards are read, the channel array is sorted into ascending order and a check is made to insure that a levels-card has been read for each channel and that an appropriate number of levels has been read for each channel. Any errors must be corrected together and PICINT reads from the symbolic DSRD KEYBD until an end card is encountered.

The data tape is then mounted by GETRUN and the proper file found. Any errors are handled by RUNERR. The channels to be displayed are checked for validity and the requests are checked and defaulted. Any width request is then used to set up column intervals and the number of printer widths is calculated. The number of samples dimensioned for RDATA is calculated and the base addresses for all arrays are calculated after the number of training and test fields is counted. After to the calling program (PICSUP).

3. Input Description

The histogram data deck, boundary cards, and levels cards are all read from the symbolic DSRN CRDRDR. Any corrections to the levels cards are read from the symbolic DSRN KEYBD. The ID record from the Multispectral Image Storage Tape is read by GETRUN from symbolic unit DATAPE.

4. Output Description

All information and error messages appear on the symbolic unit TYPEWR, with all error and most information messages appearing on symbolic DSRN PRNTR. The disk file HISTO DATA will be written by HISTD if calculation of the histogram was requested or by PICINT if the histogram was read from cards. A complete list of in formation and error messages follows.

Information Messages	Error Messages	
75	139	514
77	463	516
95	464	517
98	467	518
100	468	519
151	473	521

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

MODULE IDENTIFICATION

Module Name: PRISUP Function Name: PRINTRESULTS

Purpose: Supervisor for PRINTRESULTS

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S.K. Schwingendorf Date: 01/17/79

MODULE ABSTRACT

Supervisor for the Printresults function.

1. Module Usage

PRISUP

CALL PRISUP

There are no arguments to PRISUP. It is called from LARSMN when the PRINTRESULTS function is requested. Control returns to LARSMN upon completion of the function.

2. Internal Description

PRISUP calls the card reader and initiator and then checks to see if probability maps or tables were requested. If so, PRISUP then checks if a normal display routine was also requested. The normal display routine (if requested) is run first. PRISUP goes into a loop of calls to display the results of the different areas on the results file. The loop is composed of calls to three subroutines, DISPY1, DISPLY, and DISPY2. DISPY1 is called to find the next area on the classification tape to be displayed. If there are no more areas to be displayed, DISPY1 will RETURN1 which will cause PRISUP to call DISPY2. If there is another area DISPLY is called to perform the display and performance tally function. Normally, after a call to DISPLY, PRISUP will call DISPY1 again. The exception is if a user issued the 'STOP' command while executing DISPLY in which case a RETURN1 is passed back to PRISUP causing DISPY2 to be called. DISPY2 prints up the performance tables and completes the function, if the probability option was not requested. If it was, tapes are rewound and repositioned by entering the initiator at the entry point PRIIN1. This simulates the reading of the proper tapes and positions them for the probability run. Several flags are set appropriately and the display loop is entered and runs until completion.

Subroutines Called by PRISUP

PRIRDR
PRIINT
DISPY1
DISPY2
PRIIN1

3. Input Description

Not Applicable

4. Output Description

Standard supervisor information messages (112 and 71).

5. Supplemental Information

Refer to the LARSYS System Manual for supervisor requirements.

6. Flowchart

Not Applicable.

MODULE IDENTIFICATION

Module Name: DISPLY Function Name: PRINTRESULTS

Purpose: Creates the display map

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S.K. Schwingendorf Date: 01/17/79

MODULE ABSTRACT

DISPLY creates the display map and/or the probability map for one area of the results file. Thus DISPLY is called once for each area of the results file displayed and once more for each area displayed as a probability map.

1. Module Usage

DISPLY

CALL DISPLY (TRNFLD,TSTFLD,TRNTAB,TSTTAB,IR,IVR,IPTS, *)

Input Arguments:

TRNFLD - I*4, An array of training field definitions dimensioned (10, NOFLD) where NOFLD is the number of training fields. The format is that of the array FLDARY in subroutine RDFLDS (which creates the array).

TSTFLD - I*4, An array of test field definitions dimensioned (10, NOTST) where NOTST is the number of test fields. The format is the same as TRNFLD.

IR - I*2, A buffer area used to read in a line of the classification file. It is dimensioned the number of points in a classified line. After unpacking, IR contains either class numbers as valid halfword integers or the coded discriminant value if the probability option is being used.

IVR - I*2, A buffer area used to place the coded discriminant value into when it is unpacked from IR. IVR is dimensioned the same as IR.

IPTS - I*4, The number of points in a line. This is equal in value to PTS but is I*4 rather than I*2. This is required because IPTS is used as a dummy dimension for IR and IVR and dummy dimensions must be I*4.

Output Arguments:

TRNTAB - I*4, An array of performance tallies for training fields. It is dimensioned (NOCLS5, NOFLD). TRNTAB(I,J) = the number of points in training field J classified into class I.

TSTTAB - I*4, The same as TRNTAB except for test fields.

Non-standard Return:

RETURN1 - If DISPLY has terminated because the user has issued a STOP System Support Command, RETURN1 is executed.

DISPLY is called after DISPY1 has located an area to be displayed.

DISPLY does the creation of the display map and tallying of performance statistics for the area.

2. Internal Description

See flowchart

Subroutines Called by DISPLY

TSTREQ	PCTTAL
MOVBYT	FLDBN
ERPRNT	

3. Input Description

Record type 6's are read from the results file until a record type 7 is read. The file PRESULT SCRATCH (DSRN PRESUX) is read if more than one copy of the map is requested. In that case the file has been created earlier in DISPLY and in DISPY1. See Data Organization.

4. Output Description

The file PRESULT SCRATCH is completed. The display map is printed.

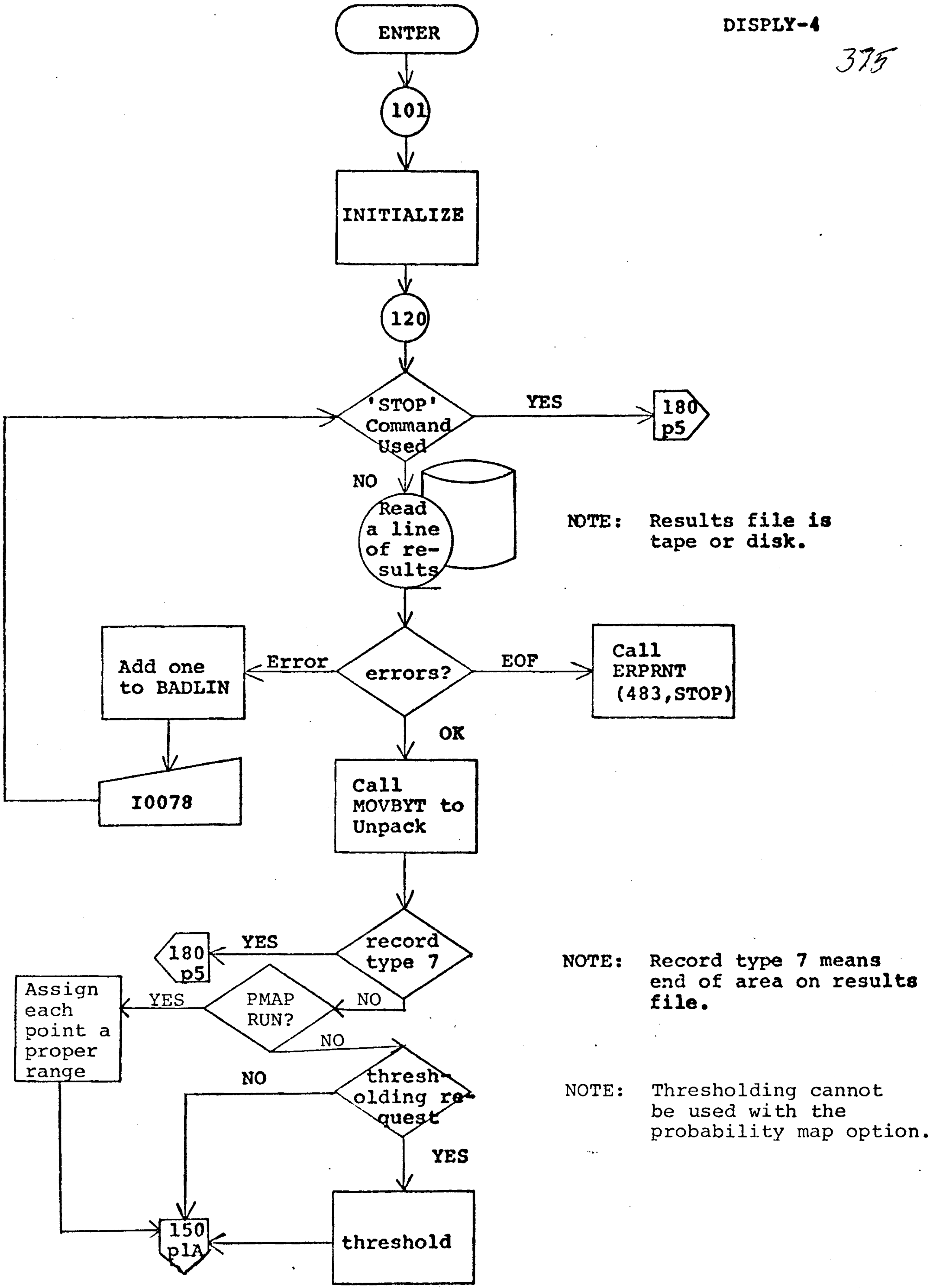
Error 483 (end of file on results file) is written via ERPRNT. Information Messages I0078 (bad line on results file, line ignored), I0079(typed after each 100 lines displayed), I008 (indicating which copy of the map is being printed) and I0023 (when STOP command is issued).

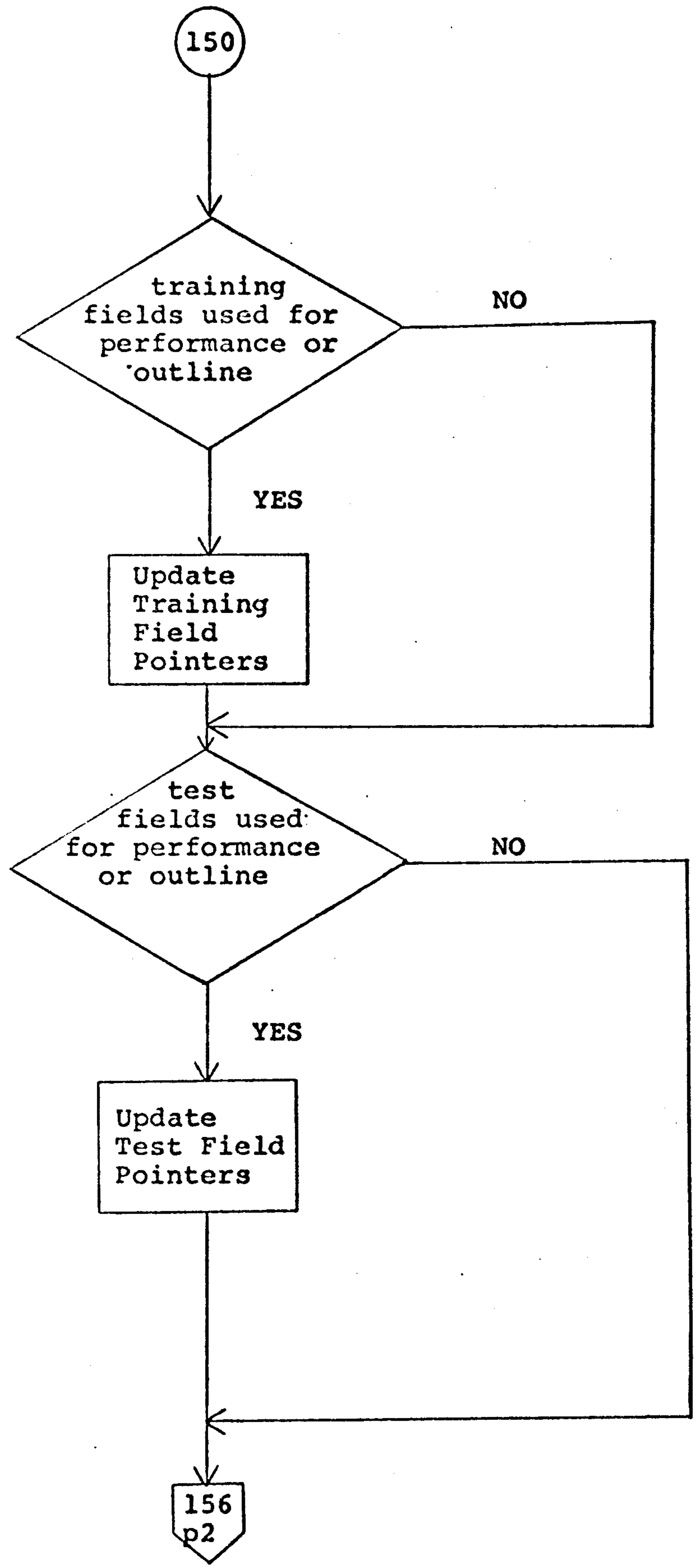
5. Supplemental Information

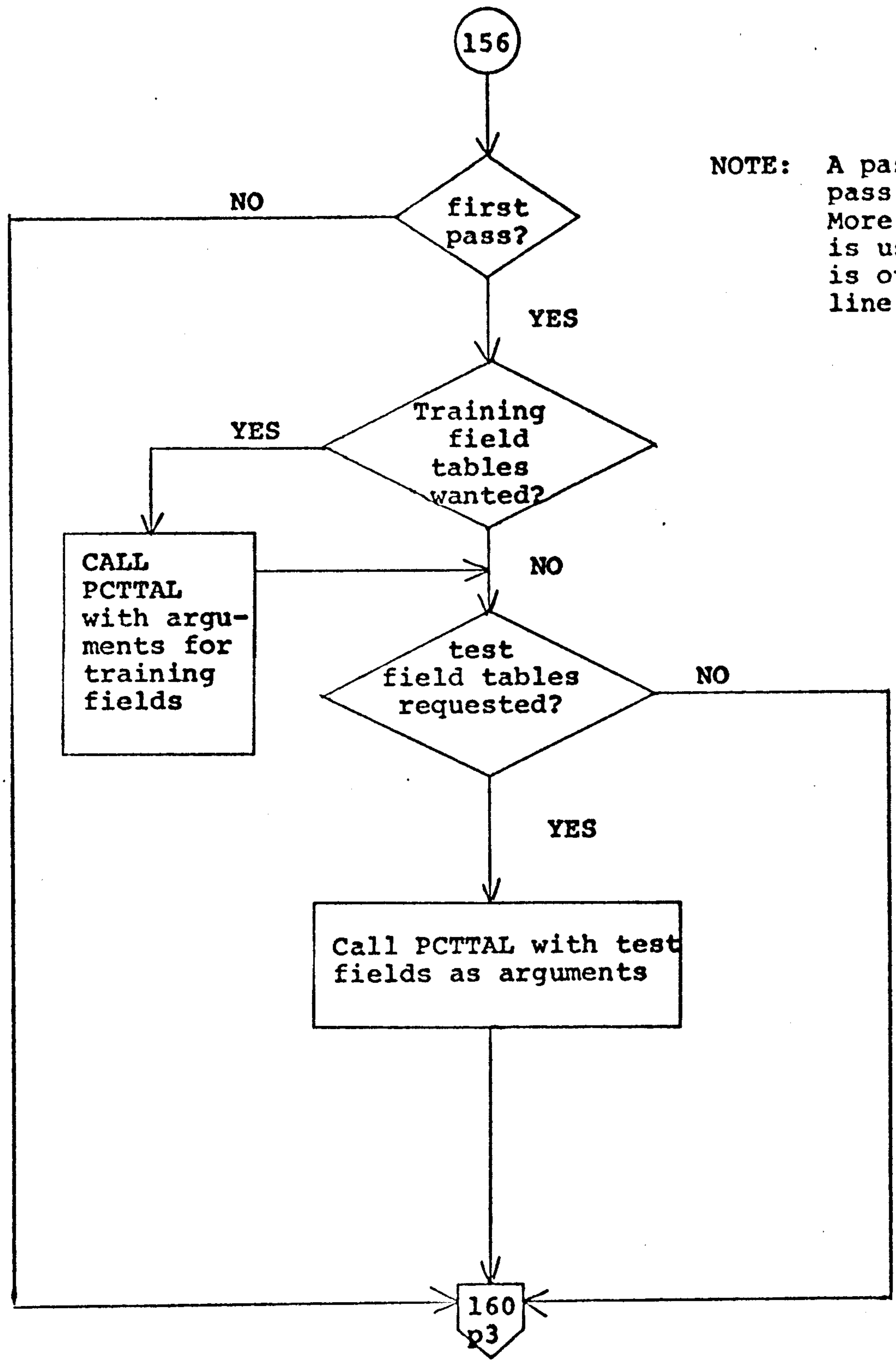
Not Applicable.

6. Flowchart

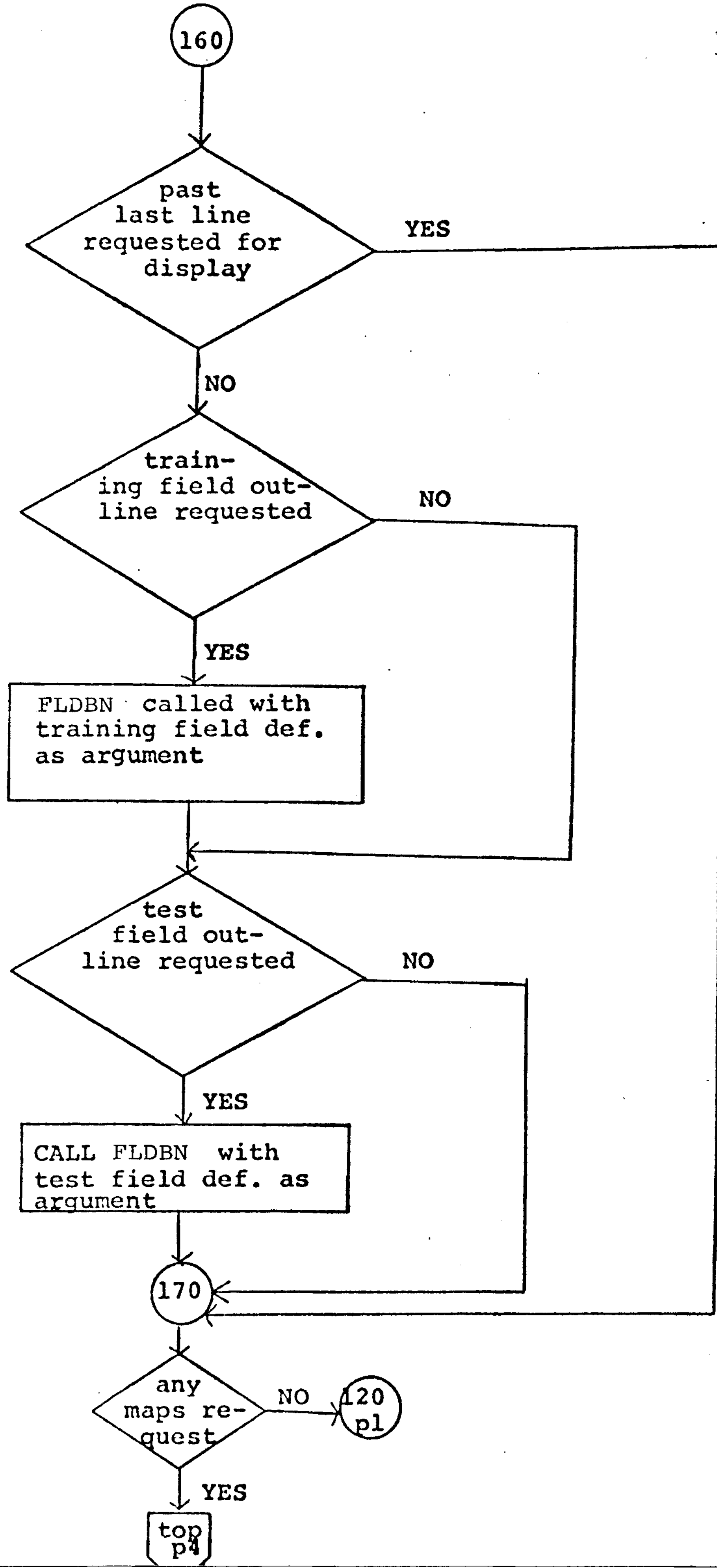
(Page numbers refer to the pages of the flowchart.)

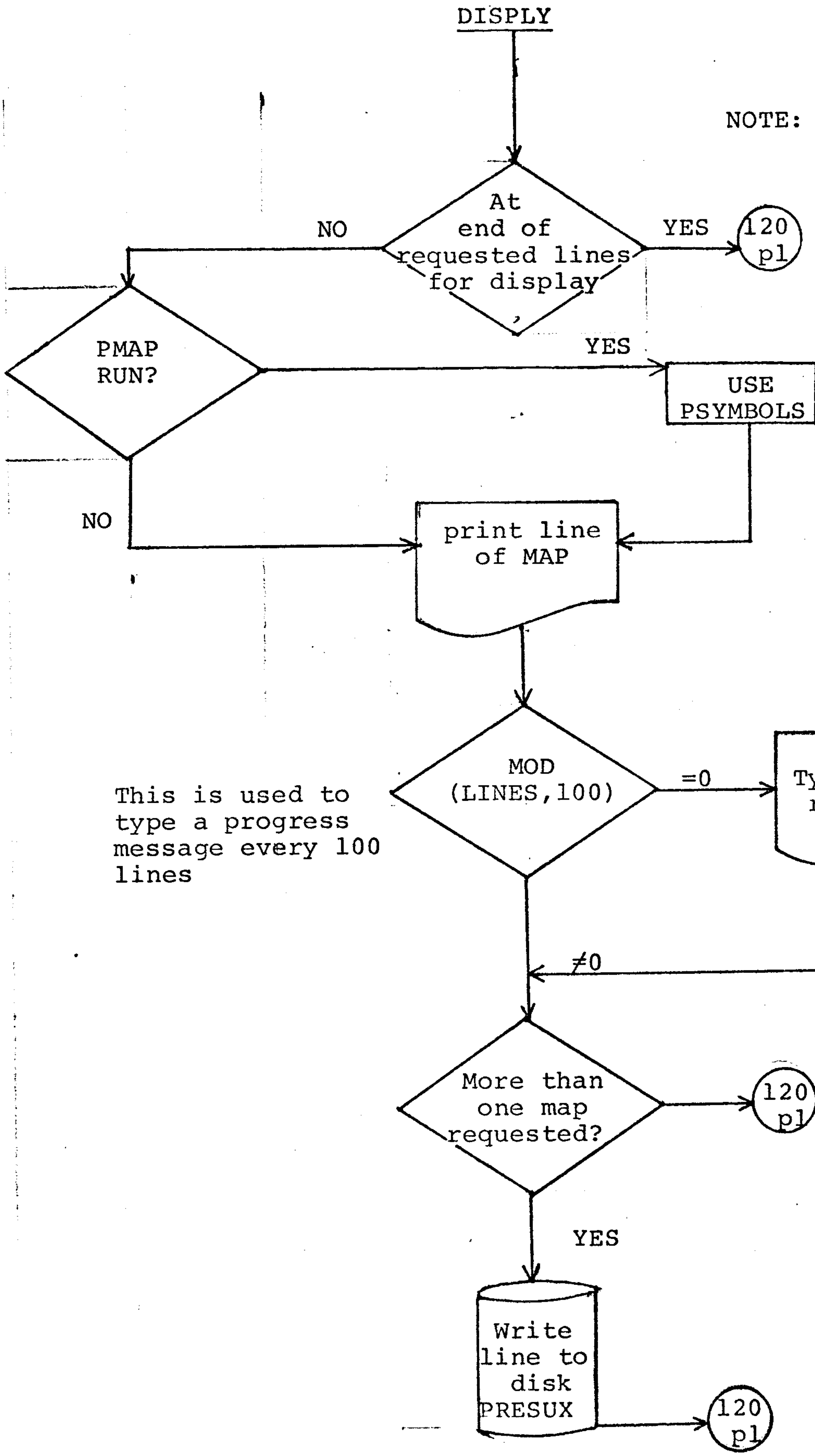

ENTER
101
INITIALIZE
120
'STOP' Command Used
YES
180 p5
NO
Read a line of results
NOTE: Results file is tape or disk.
Error
errors?
EOF
Add one to BADLIN
Call ERPRNT (483,STOP)
OK
I0078
Call MOVBYT to Unpack
record type 7
YES
180 p5
NOTE: Record type 7 means end of area on results file.
Assign each point a proper range
YES
PMAP RUN?
NO
NO
NO
thresholding request
NOTE: Thresholding cannot be used with the probability map option.
YES
150 p1A
threshold


150
training fields used for performance or outline
NO
YES
Update Training Field Pointers
test fields used for performance or outline
NO
YES
Update Test Field Pointers
156 p2



NOTE: A pass here is a pass over the MAPTAP. More than one 'pass' is used if the map is over one printer line wide.

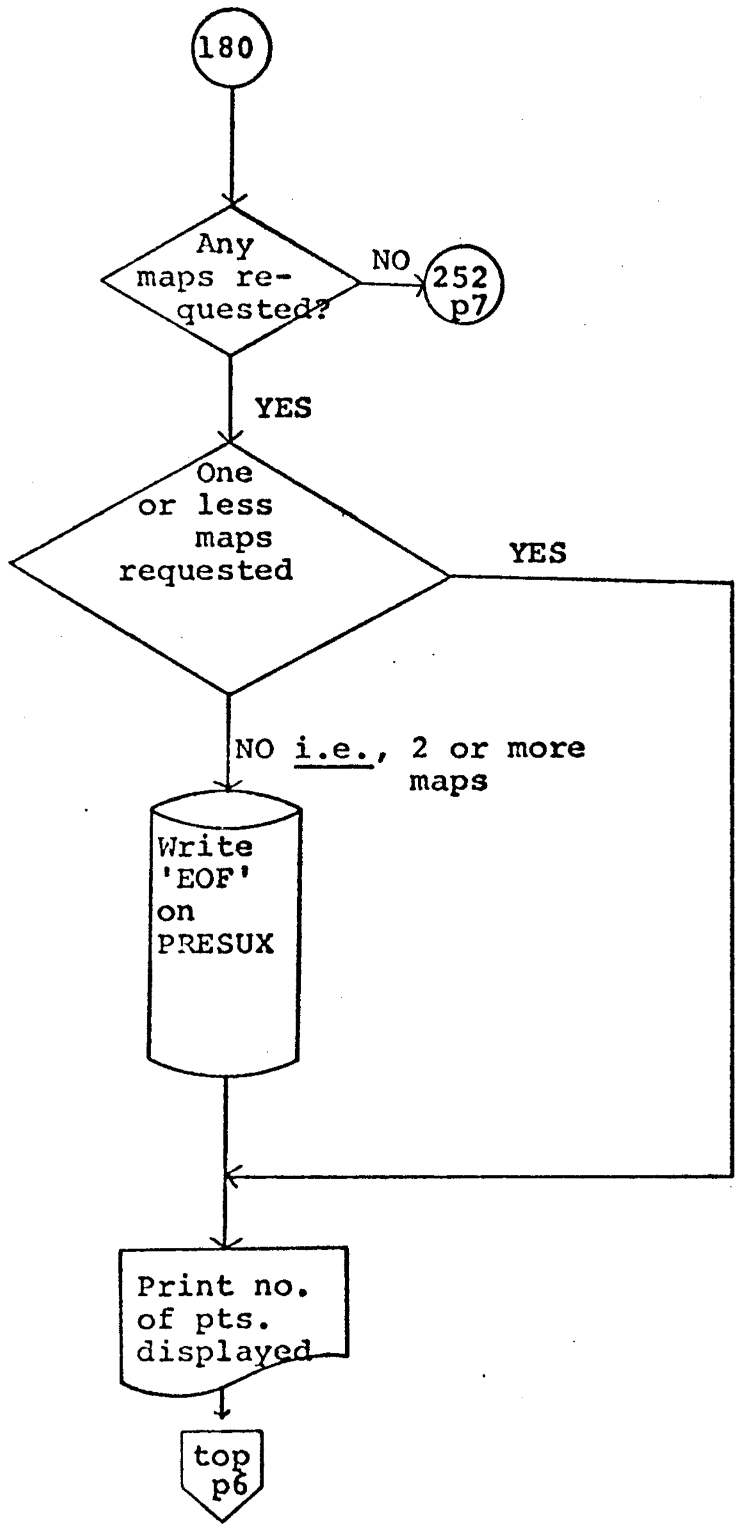

160
past last line requested for display
YES
NO
training field outline requested
NO
YES
FLDBN called with training field def. as argument
test field outline requested
NO
YES
CALL FLDBN with test field def. as argument
170
any maps request
NO
120 p1
YES
top p4



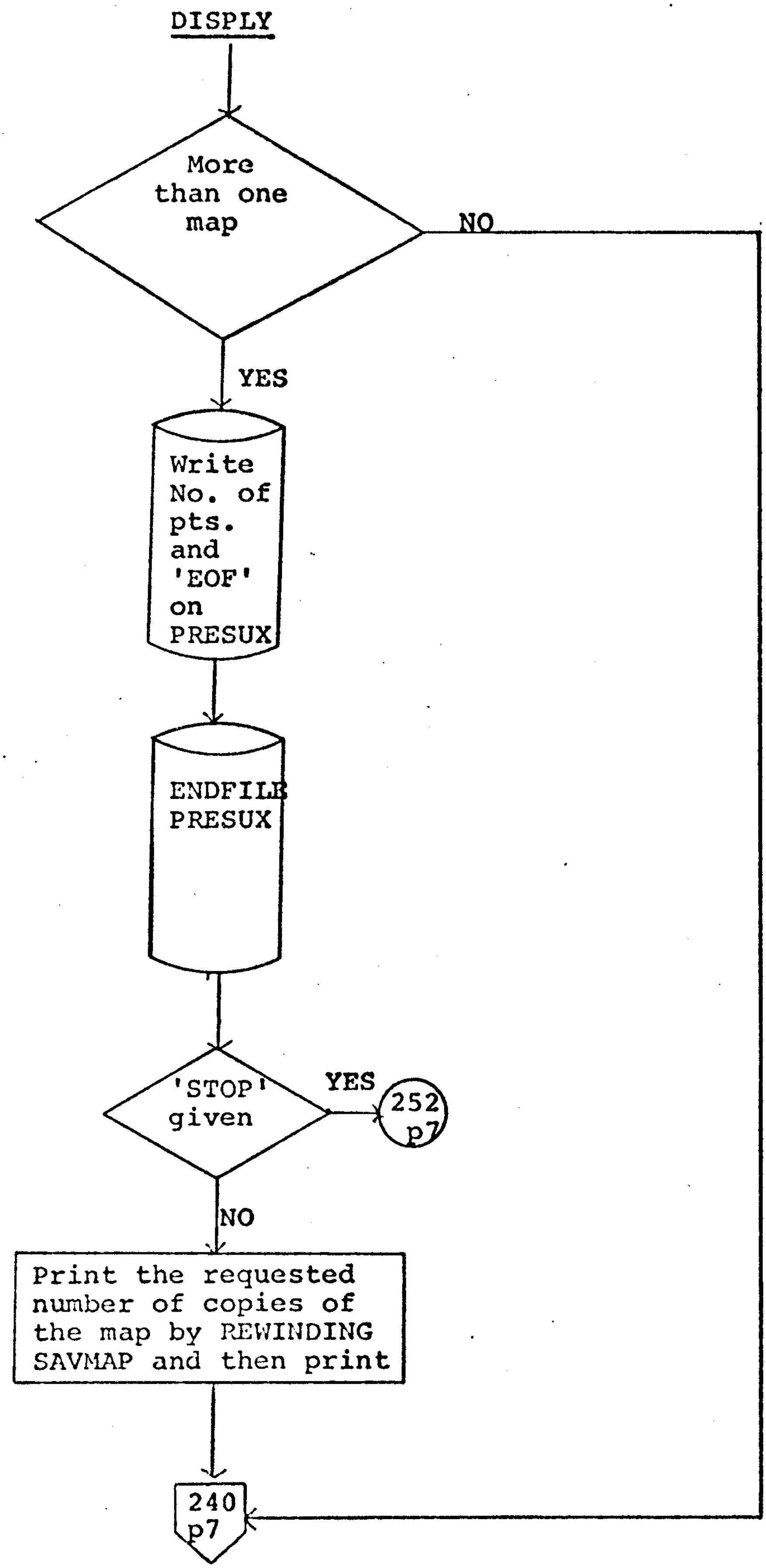
NOTE: This branch says: Since we have read all lines to be displayed, keep reading over lines till EOF is reached to compute performance for those lines.

This is used to type a progress message every 100 lines

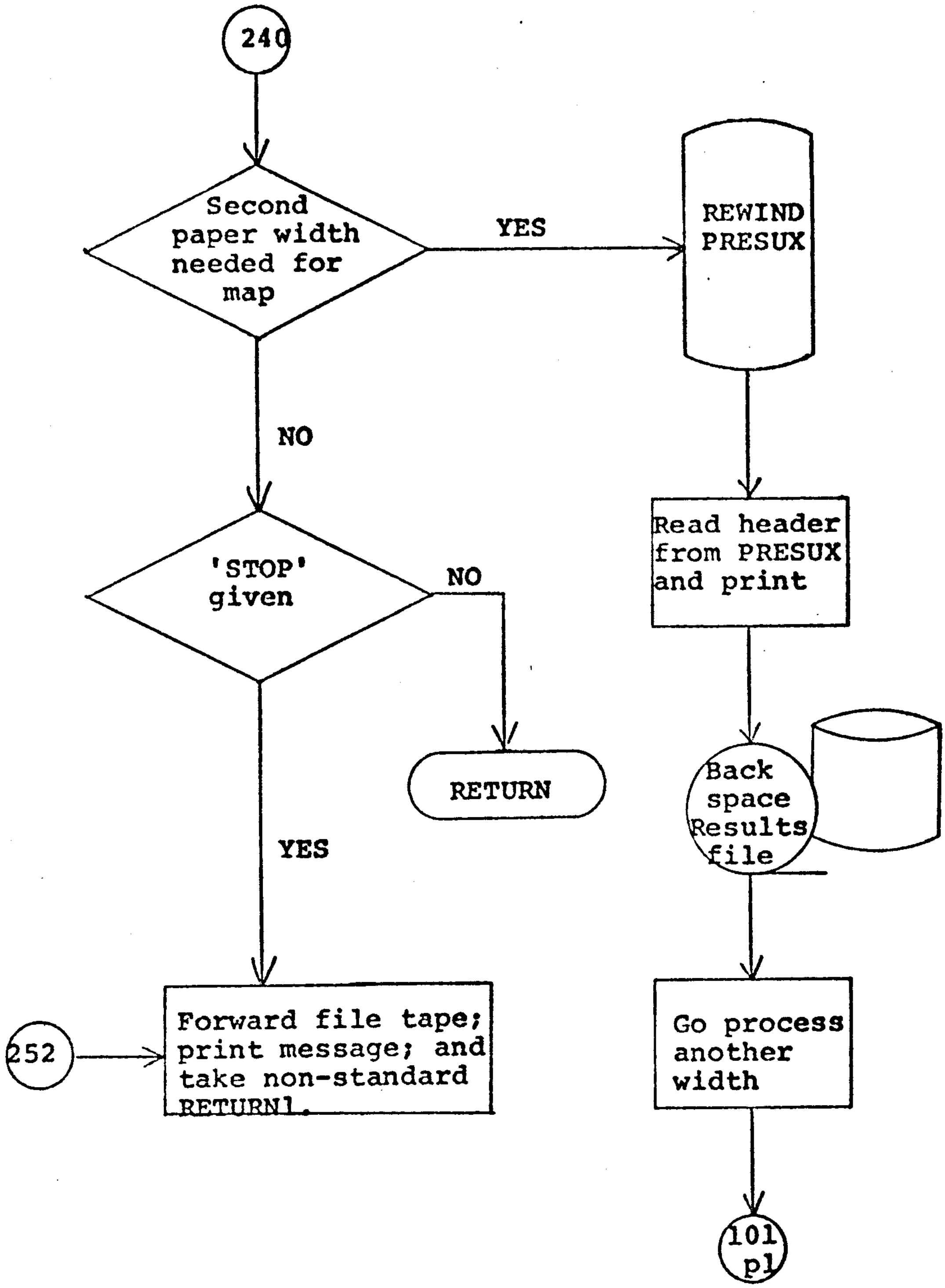
380


180
Any maps re- quested?
NO
252 p7
YES
One or less maps requested
YES
NO i.e., 2 or more maps
Write 'EOF' on PRESUX
Print no. of pts. displayed
top p6

381


DISPLY
More than one map
NO
YES
Write No. of pts. and 'EOF' on PRESUX
ENDFILE PRESUX
'STOP' given
YES
252 p7
NO
Print the requested number of copies of the map by REWINDING SAVMAP and then print
240 p7

382


240
Second paper width needed for map
YES
REWIND PRESUX
NO
Read header from PRESUX and print
'STOP' given
NO
RETURN
Back space Results file
YES
252
Forward file tape; print message; and take non-standard RETURN1.
Go process another width
101 p1

LARS Program Abstract 242

MODULE IDENTIFICATION

Module Name: DISPY1 Function Name: PRINTRESULTS

Purpose: Set up to display one classified area

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P. D. Alenduff Date: 01/10/74

MODULE ABSTRACT

DISPY1 finds the next area on the results file which is to be displayed and then sets up the area definition for DISPLY and prints the header.

1. Module Usage

DISPY1

CALL DISPY1 (DISTOP, VRB, *)

Input Arguments:

DISTOP - I*4, number of bytes remaining unused in ARRAY.

Output Arguments:

VRB - I*4, the number of double words needed for the buffer used to read a line of the results file.

Non-standard Return:

Return 1 - This return means that there are no more areas on the results file. It is generally not an error, just an indication (see PRISUP).

2. Internal Description

DISPY1 first finds the next area of the results file which is to be displayed. An area will be displayed if no BLOCK card was used or if some part of the area defined in the BLOCK card is a subset of the area on the results file. If an area is not to be displayed, the results file is read until the next area is reached, it is checked for display, and so on to the end of the file. If no more areas are to be displayed a RETURN1 is executed. When an area to be displayed is found, the correct first line, last line and interval; and first sample, last sample and interval are computed, taking into account any difference in intervals between the results file and the specification on the BLOCK card.

Subroutines Called by DISPY1

TOPRF
RTMAIN
ERPRNT
PRTHED

3. Input Description

Record type 5 is read from the results file to examine the area definition and determine if the area is to be displayed.

If not, then more records (type 6) are read until record type 7 (indicating the end of the area) is read. If a record type 8 is read from the results file when record type 5 is expected, this indicates that there are no more areas on the file. (See Data Organization).

4. Output Description

Error messages E491 (no areas displayed), and E484 (insufficient storage for arrays) are written via ERPRNT.

The header for an area to be displayed is written to the printer and if more than one map was requested, this header is written on file PRESUX (PRESULT SCRATCH).

5. Supplemental Information

Not Applicable.

6. Flowchart

Not Applicable.

MODULE IDENTIFICATION

Module Name: DISPY2 Function Name: PRINTRESULTS

Purpose: Controls printing of performance tables.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S.K. Schwingendorf Date: 01/17/79

MODULE ABSTRACT

DISPY2 controls the printing of all performance tables including printing multiple copies. It is called after all areas to be displayed have been processed.

1. Module Usage

DISPY2

CALL DISPY2 (TRNFLD, TSTFLD, TRNTAB, TSTTAB)

Input Arguments:

TRNFLD - I*4, the array of training field definitions dimensioned (10,NOFLD) where NOFLD is the number of training fields. The format is the same as the array FLDARY in REFLDS (which creates TRNFLD).

TSTFLD - I*4, the array of test field definitions dimensioned (10,NOTST) where NOTST is the number of test fields. The format is the same as TRNFLD.

TRNTAB - I*4, table of training field performance. Used only to pass on to subroutine PRTPCT.

TSTTAB - I*4, table of test field performance used only to pass on to subroutine PRTPCT.

Note that TRNFLD and TSTFLD are modified in DISPY2. This modification of rows 4 and 7 is used by subroutine PRTPCT when it is passed TRNFLD and TSTFLD but the modification is not considered an output back to the caller.

The list of training test fields and all performance and percentage tables have been printed when DISPY2 has completed execution.

2. Internal Description

DISPY2 first moves group names into a vector for printing. If the probability option is being run, the vector is reloaded with generated range names so that the probability performance tables will be labelled correctly. If LITSKY is 1, the list of training and test fields is printed, otherwise this code is skipped. Then all tables are printed by calls to PRTPCT. Before a table is printed by calling PRTPCT, a branch is made to an internal subroutine which calls TSTREQ to check for the STOP command and prints the header via a call to PRTHED.

If several copies of tables were requested, PRTPCT is passed the DSRN of the file PRESULT SCRATCH rather than the DSRN of the printer. Then in the case of multiple copies, the file PRESULT SCRATCH is rewound and read and printed the desired number of times.

Subroutines called by DISPY2

PRTHED
TSTREQ
PRTPCT

3. Input Description

The training and test field definitions are read from file TRNTEST FIELDS (DSRN TTFLDX). See Data Organization. The file PRESULT SCRATCH may be read. If so, it was created by DISPY2.

4. Output Description

Information message I0023 indicating the user used the STOP command. If multiple copies of tables were requested, the tables are written onto the file PRESULT SCRATCH (DSRN PRESUX). Note that the file is rewound at the beginning of DISPY2 so that any earlier information is overwritten.

The list of training and test fields is printed and all copies of performance, acreage, and training tables are printed. If only one copy of tables is requested, it is printed by subroutine PRTPCT. If multiple copies were requested, DISPY2 prints them (see Section 2 above).

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 244

MODULE IDENTIFICATION

Module Name: FLDBN Function Name: PRINTRESULTS

Purpose: Marks points to be printed as outline symbols

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: E. M. Rodd Date: 09/26/72

MODULE ABSTRACT

For each line to be displayed FLDBN is called to mark points which will be printed as outline symbols.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

FLDBN

CALL FLDBN (FIELD,START,STOP,IR,IMARK)

Input Arguments:

FIELD - I*4, Field definitions of fields (training or test fields depending upon the call) to be outlined. This array has the format of the FLDARY array in RDFLDS (which actually created FIELD).

START - I*4, The index in FIELD of the first field to check for necessity of outline symbols. The caller has determined which fields have been completely displayed and has set START accordingly.

STOP - I*4, The index in FIELD of the last field to check for necessity of outline symbols. The caller has determined which fields are further down the area being displayed and set STOP accordingly.

IMARK - I*4, FLDBN works by placing in IR a class number higher than the number of classes in points which are outline points. IMARK is how much higher a number is to be used. To outline training fields it is 1 and to outline test fields it is 2.

Output Arguments:

IR - I*2, This is the vector containing the class decision for each point in the line. FLDBN modifies the value in IR for points which are to be displayed as outline points. NOCLS5 is the number of classes + 1 for the threshold class. For a point outlining a training field, IR is returned with NOCLS5+1 and for a test field NOCLS5+2. If a point is a common boundary for both kinds of fields, IR is returned with NOCLS5+3.

The outlines created by FLDBN are around the field; that is, the field is completely contained inside of the outline.

2. Internal Description

FLDBN modifies the appropriate points in IR to indicate the boundaries. If a point is found which was already noted as say a training field boundary and then is found also to be a test field boundary, the entry of IR corresponding to that point is set to NOCLS5+3. The loop indexing is complicated because of the necessity of handling the case where the sample increment on the results file (and consequently in IR) is different from the sample interval being displayed.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 245

MODULE IDENTIFICATION

Module Name: PCTTAL Function Name: PRINTRESULTS

Purpose: Tallies performance data

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P. D. Alenduff Date: 01/10/74

MODULE ABSTRACT

PCTTAL is called to tally performance data for one line. The arguments determine whether the tally is of training or test fields.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised September 1974

1. Module Usage

PCTTAL

CALL PCTTAL (FIELD,PCTTAB,START,STOP,NOFLDS,IR)

Input Arguments:

FIELD - I*4, array of field coordinates dimensioned (10,NOFLDS) (see below for NOFLDS). The format is that of FLDARY in subroutine RDFLDS.

PCTTAB - I*4, table of performance data. Though it is a vector in this subroutine the organization is actually two-dimensional. PCTTAB(I,J) - the number of points in field J (that is the field described by the J'th column of FIELD) which were classified into class I.

START - I*4, the index in FIELD of the first field to tally performance for.

STOP - I*4, the index in FIELD of the last field to tally performance for. STOP is greater than or equal to START.

NOFLDS - I*4, the number of columns in FIELD (i.e., the number of fields described).

IR - I*2, vector of class decisions for a line.

Output Arguments:

PCTTAB is updated by PCTTAL to include the line in IR. PCTTAL is called for one line to tally either training field performance or test field performance (depending on which are described by FIELD).

2. Internal Description

For each field in the range of START to STOP, PCTTAL first checks to be certain that the field includes the current line of the current display area. If so, all points in this line in the field are tallied into PCTTAB.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 246

MODULE IDENTIFICATION

Module Name: PRICOM Function Name: PRINTRESULTS

Purpose: Block Data for PRICOM

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 01/17/79

MODULE ABSTRACT

This is the BLOCK DATA subroutine for the PRINTRESULTS common block PRICOM.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

MODULE IDENTIFICATION

Module Name: PRIINT Function Name: PRINTRESULTS

Purpose: PRINTRESULTS initiator

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S.K. Schwingendorf Date: 01/17/79

MODULE ABSTRACT

PRIINT reads the first part of the results file, checks grouping and symbols and allocates array space. It also reads the test fields.

1. Module Usage

PRIINT

CALL PRIINT (FLDBAS,TSTST,TRNST,TSTBAS,UNUSED,DISTOP)

Output Arguments:

FLDBAS - I*4, returned with the base address in ARRAY of TRNFLD (see DISPLY)

TSTST - I*4, returned with base address in ARRAY of the array containing test field calucaltions (TSTTAB in DISPLY).

TRNST - I*4, returned with base address in ARRAY of array containing training field calculation (TRNTAB in DISPLY).

TSTBAS - I*4, returned with base address in ARRAY of the array containing test field coordinates (TSTFLD in DISPLY).

UNUSED - I*4, returned with the base address in ARRAY of the array to be used for the buffer for reading the results file.

DISTOP - I*4, returned with the number of bytes remaining unused in ARRAY.

PRIINT performs the initiation function for the PRINTRESULTS load module.

2. Internal Description

The first two records of the results tape are read. The sixth word of the first record is checked for a flag indicating a results tape produced by the modified *CLASSIFYPOINTS processor. If the flag=1, the weights are read from record 2 of the tape. Otherwise, no weights are included in the READ statement, and the weights are set to zero. The only other information used off record type 1 is the serial number. The users threshold or range values are coded into the same coding scheme used on the results file. Grouping is checked and a check is made for sufficient symbols. If insufficient symbols are available, more are requested. The training fields are read from the results file (record type 3) via RDTRN. The test field data cards are read via RDFLDS. Then the remainder of the array base addresses are computed and a check made for sufficient space in ARRAY. Record type 4 is read from the results file via STATS. During a probability option run the initiator may be re-entered (if run with normal display option) at PRIIN1. Upon entry an entryflag, ENTFG, is set. Records 1 and 2 are read from MAPTAP and then a jump is made to a loop that advances the results file to an 'EOS'. STATKY is set equal to 0 and storage allocation continues until completion. A jump is now made to the STATS call. Upon returning, TRNTAB and TSTTAB are initialized, and the routine returns to the supervisor. This entry is necessary only if a standard display run is used with the probability option. Entering the initiator at PRIIN1 repositions the tapes and reallocates storage area for the probability run.

Subroutines Called by PRIINT

TSTREQ	CTLWRD	RDFLDS
ERPRNT	BCDVAL	RTMAIN
GRPCHK	RDTRN	STATS

3. Input Description

The first four records of the results file have been read by the end of PRIINT (1 and 2 are read in PRIINT and 3 and 4 via calls to other subroutines). If required, an additional symbols card is read from the typewriter. The flags and switches in PRICOM which were set in PRIRDR are used extensively in PRIINT.

4. Output Description

Information messages that are issued are I0034 and I0081.

A list of supervisor options is printed including the serial number of the results, the number of maps and copies of tables requested and the number of training and test fields stored in memory. If a printout of statistics was requested, they will be printed in STATS which is called by PRIINT. Disk file TRNTEST FIELDS is created in the call to RDTRN and RDFLDS.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 247

MODULE IDENTIFICATION

Module Name: PRIINT Function Name: PRINTRESULTS

Purpose: PRINTRESULTS initiator

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. J. Davis Date: 05/08/74

MODULE ABSTRACT

PRIINT reads the first part of the results file, checks grouping and symbols and allocates array space. It also reads the test fields.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised September 1974

1. Module Usage

PRIINT

CALL PRIINT (FLDBAS,TSTST,TRNST,TSTBAS,UNUSED,DISTOP)

Output Arguments:

FLDBAS - I*4, returned with the base address in ARRAY of TRNFLD (see DISPLY)

TSTST - I*4, returned with base address in ARRAY of the array containing test field calculations (TSTTAB in DISPLY).

TRNST - I*4, returned with base address in ARRAY of array containing training field calculations (TRNTAB in DISPLY).

TSTBAS - I*4, returned with base address in ARRAY of the array containing test field coordinates (TSTFLD in DISPLY).

UNUSED - I*4, returned with the base address in ARRAY of the array to be used for the buffer for reading the results file.

DISTOP - I*4, returned with the number of bytes remaining unused in ARRAY.

PRIINT performs the initiation function for the PRINTRESULTS load module.

2. Internal Description

The first two records of the results tape are read. The sixth word of the first record is checked for a flag indicating a results tape produced by the modified *CLASSIFYPOINTS processor. If the flag=1, the weights are read from record 2 of the tape. Otherwise, no weights are included in the READ statement, and the weights are set to zero. The only other information used off record type 1 is the serial number. The user's threshold vlaues are coded into the same coding scheme used on the results file. Grouping is checked and a check is made for sufficient symbols. If insufficient symbols are available, more are requested. The training fields are read from the results file (record type 3) via RDTRN. The test field data cards are read via RDFLDS. Then the remainder of the array base addresses are computed and a check made for sufficient space in ARRAY. Record type 4 is read from the results file via STATS.

3. Input Description

The first four records of the results file have been read by the end of PRIINT (1 and 2 are read in PRIINT and 3 and 4 via calls to other subroutines). If required, an additional symbols card is read from the typewriter. The flags and switches in PRICOM which were set in PRIRDR are used extensively in PRIINT.

4. Output Description

Information messages that are issued are I0034 and I0081.

A list of supervisor options is printed including the serial number of the results, the number of maps and copies of tables requested and the number of training and test fields stored in memory. If a printout of statistics was requested, they will be printed in STATS which is called by PRIINT. Disk file TRNTEST FIELDS is created in the call to RDTRN and RDFLDS.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 248

402

MODULE IDENTIFICATION

Module Name: PRIRDR Function Name: PRINTRESULTS

Purpose: Interprets the PRINTRESULTS control cards.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S.K. Schwingendorf Date: 01/17/79

MODULE ABSTRACT

PRIRDR interprets all function control cards for PRINTRESULTS. Checks are made for complete and valid specifications.

1. Module Usage

PRIRDR

PRIRDR has no calling parameters. Various flags in PRICOM reflect the control cards interpreted. These are:

Control Card	Action
RESULTS	RESULT will be set to the correct DSRN for tape or disk as requested. If tape is requested, MMTAPE is called to mount the tape and position it.
PRINT	STATS — STATKY = 1 NOLIST — LISTKY = 1 MAPS — NOMAPS is set to the number of maps. OUTLINE — OTRKEY and OTSKEY are set to 1 accordingly. TRAIN and TEST — TRFLD, TRCLS, TRACR TSFLD, TSCLS and/or TSACR are set to = 1 accordingly. TABLES — COPIES is set to the number of copies requested.
ACRES	Sets ACRFLG = 1 and stores the number of acres in ACRES.
SCALE	Sets SCLFLG = 1 and stores the user defined scale in SCALE.
DELETE	Sets DELFLG = 1, check to be sure the number of classes does not exceed 60 and stores the class numbers in array DCLSTK.
SYMBOLS	The symbols are stored in SYMMTX.
PROBABILITY	User defined ranges are stored in PRBRNG.
PSYMBOLS	These symbols are stored in PSYMTX and are used for the probability map.
THRESHOLD	The threshold values are stored in THRES.
GROUP	GRPNAM and GRPSTK are computed by a call to GRPSCN.
BLOCK	The first 6 words of BLOCK are used to contain this field boundary definition and the run number is in RUNNUM.

2. Internal Description

Internals are standard for card readers. A check is made to be certain that a results specification was made. If display maps were requested, a check is made that symbols were given. If either is missing, the user is asked to type in the needed card. If a probability card is read with no ranges given, then the reader automatically defaults to 8 preset ranges.

If the last range is not zero another symbol and range (zero) is added. A check is made to be sure that enough probability symbols were given. The user is given a list of ranges and is asked for more symbols if necessary. If no probability symbols were given and there are fewer than nine ranges, then the reader will default to preset symbols. If the user requests both the ACRES option (Acreage of area is known and a scale is calculated) and the SCALE option (A user defined scale is supplied) then the scale option is ignored. If grouping and the DELETE option are requested then the DELETE option is ignored. Requesting TRAINING and/or TEST field acreage tables without supplying acres or scale parameters causes SCALE to default to the standard 1.15 acres per pixel. If, however, the SCALE or ACRES option is used without training or test field parameters, then the request is ignored. If results are on disk, a read is made of the disk file to be certain the file exists. If results are on tape, MMTAPE is called with mode 0 indicating the tape is read only.

Subroutines Called by PRIRDR

TSTREQ	CTLPRM	BCDVAL	MMTAPE
CTLWRD	IVAL	FVAL	
ERPRNT	LOCATE	GRPSCN	

3. Input Description

The control cards are read via call to CTLWRD. If results are on disk, the first record is read to be certain the file exists.

4. Output Description

Information messages 72, 73, and 74 are typed. Error message 441-457 and 459 are written via ERPRNT. A list of options selected is printed. Unnumbered informational and error messages have been added concerning the acreage modifications.

5. Supplemental Information

See the LARSYS Systems Manual for card reader requirements.

6. Flowchart

Not Applicable.

MODULE IDENTIFICATION

Module Name: PRTHED Function Name: PRINTRESULTS

Purpose: Prints headers

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S.K. Schwingendorf Date: 01/17/79

MODULE ABSTRACT

Prints the headers for PRINTRESULTS containing run identification and channels and classes information or ranges information.

1. Module Usage

PRTHED

CALL PRTHED (RUNKEY,CHNKEY,CLSKEY,UNIT)

Input Arguments:

RUNKEY - I*4, flag for writing run identification. RUNKEY = 0 means do not write run identification and RUNKEY = 1 means do write it.

CHNKEY - I*4, flag for writing channel (and calibration) information. CHNKEY = 0 means do not write and = 1 means do write.

CLSKEY - I*4, flag for writing classes information. CLSKEY = 0 means do not write any classes information. CLSKEY = 1 means list class name, group name (if any), threshold percent (if any), and symbol for printing. CLSKEY = 2 means the same as = 1 excpet that symbols are not written. During a probability run, CLSKEY = 1 means list symbol for printing and range interval. CLSKEY = 2 means the same as CLSKEY = 1 except the symbols are not written.

UNIT - I*4, DSRN to write header on.

PRTHED is used to write headers for PRINTRESULTS. The unit number is variable because in the call, the write will be to a scratch data set if several copies are to be printed.

2. Internal Description

The writing of channel and calibration information is done with a variable format statement depending upon the calibration codes. The writing of classes information is done using the FORTRAN carriage control character '+' in order to print the group name and threshold percent only if they exist and to write the heading for symbol group name, threshold percent, and weights only if they are to be written. Note that in CMS the '+' is executed as no space after print rather than no space before print (which is specified in FORTRAN language specifications).

PRTHED is programmed such that the output will have data items aligned correctly independent of whether the '+' control is executed as no space after or before print. The only difference will be the number of spaced between the heading and the data items.

Subroutines Called by PRTHED

GETIME
HEADID

3. Input Description

Not Applicable

4. Output Description

The following outputs are written to unit UNIT:

The standard LARSYS header.

A line giving the classification serial number and date of classification.

Run identification if requested.

Channel and calibration information if required.

Class and weight information if requested.

Probability symbols and range intervals if a probability run.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 250

MODULE IDENTIFICATION

Module Name: PRTPCT Function Name: PRINTRESULTS

Purpose: Writes performance tables

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P. D. Alenduff Date: 01/10/74

MODULE ABSTRACT

Takes as input the arrays of performance data and generates and writes performance and percentage tables. One call generates a single table.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised September 1974

1. Module Usage

PRTPCT

CALL PRTPCT (PCD, TSTKEY, NOFLDS, PRTNAM, PCTTAB, FLDARY, OUT)

Input Arguments

PCD - I*4, flag indicating type of performance tables desired. PCD = 1 means table by field, PCD = -1 means table by class and PCD = 0 means percentage table rather than straight tally of performance.

TSTKEY - I*4, flag indicating training or test fields or classes (field or class determined by PCD). TSTKEY = 0 means training fields or classes and TSTKEY = 1 means test fields or classes.

NOFLDS - I*4, number of fields considered (training or test fields according to TSTKEY).

PRTNAM - R*8, vector of group names dimensioned (NOGRPS+1). PRTNAM(1) = name of the group with the lowest number class. PRTNAM(2) = name of group with lowest numbered class of the remaining class etc.

PCTTAB - I*4, array of performance data dimensioned (NOCLS5, NOFLDS). PCTTAB(I,J) = number of points from field J classified into Class I.

FLDARY - I*4, field descriptions for fields (training or test). Dimensioned (10, NOFLDS) and in the format of FLDARY in RDFLDS.

OUT - I*4, DSRN to which the output of PRTPCT is to be written.

PRTPCT is called to generate a given table such as a training field performance table etc. The header that is printed in front of each table is printed by the caller.

2. Internal Description

PCTTAB contains performance data by classification class PRTPCT sums the data into the groups specified by the user and then the tables themselves are by group.

Subroutines Called by PRTPCT

GETIME
PRACRE

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

The output goes to DSRN OUT in printer format. The output contains a title (e.g., TRAINING CLASS PERFORMANCE), headings for the performance data and lines of performance data for each field of class depending on the table.

If a second page is required for the table (9 groups per page) the additional page(s) start with the standard LARSYS header followed by a line indicating this page is a continuation.

6. Flowchart

Not Applicable

LARS Program Abstract 251

MODULE IDENTIFICATION

Module Name: STATS Function Name: PRINTRESULTS

Purpose: Read statistics and print them if requested

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: E. M. Rodd Date: 09/25/72

MODULE ABSTRACT

STATS reads record type 4 from the results file. If the user requested statistics printed, STATS prints them.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

STATS

CALL STATS (COVMTX, AVEMTX, VARSIZ, COR)

Input Arguments:

COVMTX - R*4, This is an area to be used by STAT for reading in the covariance matrices. They will be returned in normal storage form (that is each column of COVMTX contains a covariance matrix for one class as a vector and only the upper half of the symmetric matrix is stored.)

AVEMTX - R*4, This is an area provided for use by STAT for reading in the means. The means will be returned in this array. Each column will contain the means for one class.

VARSIZ - I*4, Provided to STATS by caller. It is equal to:

$$\frac{\text{Channels}^2+\text{Channels}}{2}$$

which is the number of elements in the covariance matrix for one class.

COR - R*4, A work area provided to STATS to compute the correlation matrix from the covariance matrix.

STATS is used to read the covariance and average values from the results file and if STATKY = 1 (i.e., user requested a statistics printout), the statistics summary will be printed via WRTMTX. Note that if the user does not request a printout, the only function served by STATS is to space over the record of the results file since PRIINT (which calls STATS) makes no other use of the COVMTX and AVEMTX arrays.

2. Internal Description

The covariance and mean arrays are read from the results file. If no statistics summary was requested, STATS returns to caller. If the summary was requested, STATS does for each class a calculation of the correlation matrix from the covariance matrix and prints the summary.

Subroutines Called by STATS

ERPRNT
GETIME
WRTMTX

3. Input Description

Record type 4 is read from the results file.

Refer to the data organization section of the LARSYS System Manual.

4. Output Description

Error 483 is written via ERPRNT if end of file is detected on the results file.

If the user has requested a statistics summary to be printed, WRTMTX is called for each class to print the summary.

5. Supplemental Information

Not Applicable.

6. Flowchart

Not Applicable.

MODULE IDENTIFICATION

Module Name: PRACRE Function Name: PRINTRESULTS

Purpose: calculate and print acreage tables

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: J.P. Cain Date: 6/12/80

MODULE ABSTRACT

PRACRE calculates acreage and hectare tables for training and/or test fields.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

PRACRE

CALL PRACRE (FLAG, BUF, GTOTAL, PRTNAM, OUT, NAME)

Input Arguments:

FLAG	- I*4, FLAG is used to indicate what type of acreage table is desired. FLAG = 1 means a training acreage table and FLAG = 2 means a test acreage table.
BUF	- I*4, BUF(I) contains the total number of points of the I'th group in the field being considered.
GTOTAL	- I*4, The total number of points in the field being considered.
PRTNAM	- R*8, Vector of group names dimensioned (NOGRPS + 1) PRTNAM(1) = Name of the group with the lowest numbered class PRTNAM(2) = The name of the group with the lowest numbered class of the remaining classes, etc.
OUT	- I*4, DSRN to which the output of PRACRE is to be written.
NAME	- R*8, Name of the group for which the training or test field was requested.

PRACRE is called to calculate acre and hectare tables for a training or test field. The tables include the group names, the total number of points in the field from each group, the acreage (acres and hectares) make-up of each group, and each group's percentage of the field. The acre and hectare conversion factor and the total number of points in the field are also displayed. If no test or training options were supplied then the table merely consists of the group names, number of points, and the group percentage. If the 'DELETE' option is used then the classes not considered are listed.

2. Internal Description

Initially, PRACRE checks to see if 'ACRES' = 0. If so, it does not include the acreage columns in the table. If not, it calculates an acre size factor by dividing the total number of points in the field into the given number of acres. The hectare size factor, hecsize, is derived from the acre size by dividing by 2.47. Before printing the table, a call is made to 'PRTHED' to print a run identification header. Next, the header that states to which group the table applies and whether the field being considered is a training or test field, is printed. PRACRE then enters a loop which calculates, tallies, and prints each of the different table values - points, acres, hectares, and percent. After exiting this loop, the totals are printed along with the total number of points in the classification. Finally, 'PRACRE' returns control to 'PRTPCT'. If the 'DELETE' option is requested, then these deleted classes are listed along with their corresponding number of points. These points are subtracted from the total so that the percent figures of the classes being considered will not be affected.

Subroutines Called by PRACRE

PRTHED

3. Input Description

Not Applicable

4. Output Description

'PRACRE' writes to DSRN 'OUT'. The unit number is variable because in the call the write will be to a scratch data set if several copies are to be printed.

5. Supplemental Information

Not Applicable.

6. Flowchart

Not Applicable.

LARS Program Abstract 260

MODULE IDENTIFICATION

Module Name: RESSUP

Function Name: COPYRESULTS
LISTRESULTS
PUNCHSTATISTICS

Purpose: Supervisor for RESSUP load module

System/Language: CMS/FORTRAN

Author: S. K. Hunt   Date: 11/07/72

Latest Revisor: S. K. Hunt   Date: 11/08/72

MODULE ABSTRACT

This module supervises the execution of the above three functions. RESSUP determines which function has been requested and calls the processing subroutines. It does not perform any calculations.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

RESSUP

CALL RESSUP (FUNC)

Input Arguments:

FUNC - INTEGER*4, Contains either *COP, *LIS, or *PUN

This routine is called within LARSYS by the main module, LARSMN, when a *COPYRESULTS, *LISTRESULTS, or *PUNCHSTATISTICS card is detected within the input control cards. Control is returned to the caller after the completion of the requested function.

2. Internal Description

RESSUP determines which function has been requested and sets a flag in RESCOM as follows:

FUNCTION	Value of PROCES
COPYRESULTS	1
LISTRESULTS	2
PUNCHSTATISTICS	3

The user is informed which function was requested and then calls RESRDR to read and interpret the function control cards and perform the required initialization operations. RESCOP is called to set-up the dynamic core allocation and to process the given requests. A function completed message is written and control is returned to the caller.

Complete list of subroutines called by RESSUP

RESRDR
RESCOP
ERPRNT

COMMONS used in RESSUP

RESCOM
GLOCOM

3. Input Description

Not Applicable

4. Output Description

File name - Information and Error Messages
DSRN - PRNTR and TYPEWR
Device type - Printer and typewriter
Usage - Output
Description - Message numbers are listed below, for text see User's Manual.

MESSAGES

INFORMATIONAL	ERROR
182	580
183	
184	
185	
186	
187	

5. Supplemental Information

See the LARSYS System Manual for a description of supervisor module requirements.

6. Flowchart

Not Applicable

LARS Program Abstract 261

MODULE IDENTIFICATION

Module Name: COPY

Function Name: COPYRESULTS
LISTRESULTS
PUNCHSTATISTICS

Purpose: Performs results utility function (copy, list, punch)

System/Language: CMS/FORTRAN

Author: S. K. Hunt Date: 11/20/72

Latest Revisor: P. D. Alenduff Date: 03/04/75

MODULE ABSTRACT

COPY fulfills all of the requests for one file for the Copyresults, Listresults, and Punchstatistics functions. It reads the classification results file then copies, lists, or punches statistics according to the flags set iN RESRDR. Upon completion of the task control is returned to RESCOP.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised October 1980

1. Module Usage

COPY

```
CALL COPY(CHAN,CSEL,FRQUP,FRQLW,POLNAM,POLPTR,POLSTK,
          COVMTX,AVEMTX,BUF)
```

Input Arguments:

CHAN - INTEGER*2, Array of channel numbers read from the results file.

CSEL - INTEGER*2, Array of calibration codes used for each channel.

FRQUP - REAL*4, Array of upper wavelength band values for each channel.

FRQLW - REAL*4, Array of lower wavelength band values for each channel.

POLNAM - REAL*8, Array of names assigned to each pool used in classification.

POLPTR - INTEGER*2, A 2 by i matrix where i = the number of pools. POLPTR(1,i) = the number of classes in pool i, and POLPTR(2,i) = the location of the first class for the pool in POLSTK.

POLSTK - INTEGER*2, Array of class numbers of all classes in the statistics deck grouped by classification pool.

COVMTX - REAL*4, Lower half covariance matrices for each pool.

AVEMTX - REAL*4, Mean vector for each pool.

BUF - INTEGER*2, Buffer array to read in each line classified.

COPY is the main processor for the utility (results tape) load module. It either copies results tapes, lists results tapes, or punches statistics decks from results tapes.

2. Internal Description

COPY reads the first two records from the classification results file and prints header information on the line printer concerning the tapes and files used. If the user didn't request NOLIST then channel, calibration, and class information is printed on the line printer. Class weight information is also printed if weights were used in the classification. If copying is requested, the first two records are written on the output tape. The stat deck is either read, read and copied, or read and punched. The covariance and mean matrices are then read and copied if requested. Each area classified is then either read or read and copied. After the last area is processed the copy output tape is terminated by a file mark. When COPY is called with the common variable COPNUM equal to zero, the signal to write the dummy record on the output tape followed by two end-of-file marks jas been given from RESCOP. The input file will be positioned at the beginning of the next file if the results were on tape. Return is then made to the caller.

Complete list of subroutines called by COPY:

ERPRNT
TOPRF
TOPEF
TOPBF

Commons used in COPY:

RESCOM
GLOCOM

3. Input Description

File name - Classification Results File
DSRN - CLASSR or MAPTAP
Device type - Disk or tape
Usage - Input
Description - See LARSYS System Manual for detailed description

4. Output Description

File name - Information and Error Messages
DSRN - PRNTR and TYPEWR
Device type - Printer and Typewriter
Usage - Output
Description - Message numbers are listed below, for text, see User's Manual.

MESSAGES

INFORMATIONAL	ERROR
193	
194	592
	593
	594

File name - Results file printer listing
DSRN - PRNTR
Device type - Printer
Usage - Output
Description - Classification results file information. For examples see User's Manual.

File name - Classification Results File
DSRN - CPYOUT
Device type - Tape
Usage - Output
Description - See LARSYS System Manual for detailed description.

File name - Statistics Deck
DSRN - PNCH
Device type - Cards
Usage - Output
Description - See LARSYS System Manual for detailed description.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 262

MODULE IDENTIFICATION

Module Name: RESCOM

Function Name: COPYRESULTS
LISTRESULTS
PUNCHSTATISTICS

Purpose: Block Common for RESSUP load module

System/Language: CMS/FORTRAN

Author: S. K. Hunt Date: 11/21/72

Latest Revisor: S. K. Hunt Date: 11/22/72

MODULE ABSTRACT

This is a BLOCK DATA subroutine for the above three functions. Refer to the LARSYS System Manual for a list of the program modules that require the COMMON.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 263

MODULE IDENTIFICATION

Module Name: RESRDR

Function Name: COPYRESULTS
LISTRESULTS
PUNCHSTATISTICS

Purpose: Reads function control cards for RESSUP load module

System/Language: CMS/FORTRAN

Author: S. K. Hunt Date: 11/07/72

Latest Revisor: P. D. Alenduff Date: 03/04/75

MODULE ABSTRACT

RESRDR causes COPYRESULTS, LISTRESULTS, and PUNCHSTATISTICS cards to be read and then interprets them. The cards are error checked for completeness and validity. Then the required classification results tape is mounted and positioned to the correct file. The user is informed of his selections and control returns back to the caller.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

RESRDR

CALL RESRDR

This routine interprets the function control cards and puts the results into variables located in RESCOM. All results are validated and any required tapes are mounted.

2. Internal Description

After initialization of variables, CTLWRD is called to read and interpret the key words. An unexpected end of file for the control card input results in ERPRNT being called to terminate execution. After CTLWRD has determined the key word, a branch is made to sections of code to further interpret each of the possible cards. CTLPRM and IVAL are used to assist with this interpretation. After the END card is detected the user's requests are checked for completeness and validity.

Once all inputs are complete MMTAPE is called to mount any required classification results tapes. The user's requests are written on the line printer and control is returned to the caller.

Complete list of subroutines called by RESRDR:

TSTREQ
CTLWRD
ERPRNT
CTLPRM
IVAL
MMTAPE

Commons used in RESRDR:

GLOCOM
RESCOM

3. Input Description

RESRDR does not actually perform any reading operations. It does invoke CTLWRD which performs reads to the control card input stream (card reader or typewriter). In addition MMTAPE performs the mounting, reading, and initializing of the classification results tape.

4. Output Description

File name - Information and Error Messages
DSRN - PRNTR and TYPEWR
Device type - Printer and Typewriter
Usage - Output
Description - Message numbers are listed below, for text see User's Manual.

MESSAGES

INFORMATIONAL	ERROR
188	139
189	459
190	581
191	582
192	583
	584
	585
	586
	587
	588
	589
	590
	595
	596
	597
	598
	599
	600

File name - Request Selection Summary
DSRN - PRNTR
Device type - Printer
Usage - Output
Description - List of all control cards input and options requested.

5. Supplemental Information

See LARSYS System Manual for a description of how to create control card reading routines.

6. Flowchart

Not Applicable

LARS Program Abstract 264

MODULE IDENTIFICATION

Module Name: RESCOP

Function Name: COPYRESULTS
LISTRESULTS
PUNCHSTATISTICS

Purpose: Sets up dynamic core for RESSUP load module

System/Language: CMS/FORTRAN

Author: S. K. Hunt Date: 11/19/72

Latest Revisor: P. D. Alenduff Date: 03/04/75

MODULE ABSTRACT

RESCOP reads the first two records from the input classification results file. From this information, array bases are calculated for dynamic core allocation. A check is made for core overflow, the results file is repositioned to the beginning of the file and the processing routine is called once for each file to be processed. Upon return from the processing routine the final time, file pointers are set and return is made to the caller.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised October 1980

1. Module Usage

RESCOP

CALL RESCOP

This routine is called by RESSUP after all function control cards have been read. RESCOP determines the number of files to be handled and loops through calculating the base addresses of the arrays and calling the processing routine COPY once for each file. Upon completing all requested files in COPYRESULTS, RESCOP calls COPY with the common variable COPNUM equal to zero as a signal for COPY to write the dummy record and two successive end-of-file marks on the output tape. COPNUM is used to count the number of successive files to be handled, but is ignored when the ALL option is being used.

2. Internal Description

RESCOP reads the first two records from the classification results tape following the calculation of the number of files to be used from the input tape. Using the values read from these two records, the base addresses of arrays in ARRAY are calculated to set up dynamic core allocations. Subroutine COPY is called once for each file request to carry out the processing of the data. When COPYRESULTS is requested and all files have been processed, RESCOP calls COPY with COPNUM, the common variable used to keep the number of successive files to be processed, set equal to zero. COPY then writes the dummy record and two file marks onto the output tape. Control is then returned to the calling program.

Complete list of subroutines called by RESCOP:

ERPRNT
TOPRF
COPY

Commons used in RESCOP:

RESCOM
GLOCOM

3. Input Description

File name - Classification Results File
DSRN - CLASSR or MAPTAP
Device type - Disk or Tape
Usage - Input
Description - See LARSYS System Manual for a complete description of the results file.

4. Output Description

File name - Error Messages
DSRN - PRNTR and TYPEWR
Device type - Printer and Typewriter
Usage - Output
Description - Message numbers are listed below, for text, see User's Manual.

Error Messages

591
593
594

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 270

MODULE IDENTIFICATION

Module Name: RUNSUP Function Name: IDPRINT
CHANNELTRANSFORM
TRANSFERDATA

Purpose: Supervisor for IDPRINT, CHANNELTRANSFORM & TRANSFERDATA. Contains IDPRINT processor.

System/Language: CMS/FORTRAN

Author: J. B. Etheridge Date: 10/24/72

Latest Revisor: Date:

MODULE ABSTRACT

This supervisor determines which of three functions has been requested: IDPRINT, CHANNELTRANSFORM, or TRANSFERDATA. If TRANSFERDATA or CHANNELTRANSFORM was requested, it calls in the appropriate function. It IDPRINT was requested, it branches to the coding in RUNSUP which then processes the IDPRINT function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised October 1980

1. Module Usage

RUNSUP

CALL RUNSUP(FUNC)

Input Argument:

FUNC - INTEGER*4, contains either *IDP, *CHA, or *TRA

2. Internal Description

RUNSUP determines which function has been requested and then branches to one of the three places described below.

a. CALL CHASUP, the CHANNELTRANSFORM function. CHASUP returns to RUNSUP which then simply returns to LARSMN.

b. CALL TRARDR, the Transferdata function. Having gotten to the point where it has read in one data card, TRARDR returns to RUNSUP. Then the statement CALL TRAXEQ initiates the printing, recording on tape, or punching. When control returns to RUNSUP, either (i) an error has occurred in a call to GADLIN (in TRAXEQ), or (ii) the requested output was completed for the run specified on the data card.

   If (i) has occurred, LINERR is called, a message is printed out, and output is terminated for this run. Incidentally, if tape output is requested, the output tape is backspaced to the beginning of this run. Then the next data card is read in (See description of (ii) in next paragraph).

   If (ii) has occurred, it is time to read in a new data card by using CALL TRADAT, the second entry point in TRARDR. If this data card is an END card, control returns to LARSMN. If it contains another run, RUNSUP loops back to CALL TRAXEQ.

c. The Idprint function, contained in RUNSUP. This routine reads in the IDPRINT control cards and then branches to one of three separate routine: one for PRINT RUN(XXXXXX), one for PRINT TAPE(XXX), and one for PRINT ALL. Each of these routines will return to LARSMN. The PRINT RUN and PRINT ALL routines use the system runtable and the PRINT TAPE routine mounts and uses the requested tape. (Note - This tape (on DATAPE) is rewound). Thus, the user's runtable is not used at all.

Complete list of subroutines called by RUNSUP:

CHASUP
TRARDR
TRAXEQ
TRADAT
LINERR
ERPRNT
TOPBS
(The following are used by the IDPRINT processor)
TSTREQ
CTLWRD
ERPRNT
CTLPRM
IVAL
GETIME
TOPRW
TOPRU
MOUNT
TOPRD
TOPFF

Commons in RUNSUP:

GLOCOM
RUNCOM

3. Input Description

File name	- Multispectral Image Storage Data
DSRN	- DATAPE
Device type	- Tape
Usage	- Input
Description	- Master input tape containing digitized scanner data

4. Output Description

File name	- Multispectral Image Storage Tape ID listing
DSRN	- PRNTR
Device type	- Printer
Usage	- Output
Description	- One-page table of ID information for the run. See sample in User's Manual.

RUNSUP generates this file only if IDPRINT is requested.

File name	- Informational and error messages
DSRN	- PRNTR and TYPEWR
Device type	- Printer and typewriter
Description	- Message numbers are listed below. For text of messages, see User's Manual.

MESSAGES

ERROR	INFORMATIONAL	
104	23	*118
139	* 62	*121
430	63	
431	* 65	
433	*114	
436	116	
533	*115	
	*117	

* means only printed on typewriter

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 271

MODULE IDENTIFICATION

Module Name: CHASUP Function Name: CHANNELTRANSFORMATION

Purpose: Duplicates a run from Multispectral Image Storage Tape onto another tape with data conversion.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Paul Spencer Date: 8/24/76

MODULE ABSTRACT

This supervisor reads the control cards for the CHANNELTRANSFORMATION function and duplicates a run from a Multispectral Image Storage Tape onto another tape. It may or may not perform data conversion, depending on the user's control cards. The output tape is left mounted (and rewound) on DATAPE so that it may be used as an input tape in the next program.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CHASUP

CALL CHASUP

2. Internal Description

CHASUP reads in the control cards for the CHANNELTRANSFORMATION function, mounts the input (Multispectral Image Storage Tape) tape on DUPRUN and the output tape on DATAPE, and checks to see if the requested file on the output tape is empty. If it is empty, processing continues. If not, the user is asked whether or not he wishes to destroy the file: if the answer is yes, processing continues; if no, the user may type in a new file number (and this file is checked to see whether it is empty, etc.) or he may type in 'KILL' to stop execution altogether.

If NEWCHANNELS were specified, calls are made to SUBGEN to read the data transformation cards and generate the appropriate code. The new data values are then output to the output tape.

Next, a call to TOPWR (tape write) writes the ID record of the run on the output tape, and then calls to TOPRD and TOPWR copy the run from the input tape onto the output tape. The following is then written on the output tape: end of file mark, a dummy ID record with ID(3) = 0, and end of file and end of tape marks. Both the input and output tapes are rewound.

Complete list of subroutines called by CHASUP:

CTLWRD	TOPWR
ERPRNT	TOPRD
CTLPRM	TOPEF
IVAL	TOPRW
GETRUN	SUBGEN
MOUNT	SUB
TOPFF	
TOPRF	

Commons in CHASUP:

GLOCOM
RUNCOM

3. Input Description

File Name	- Multispectral Image Storage Tape
DSRN	- DUPRUN
Device Type	- Tape
Usage	- Input
Description	- Master input tape containing digitized scanner data

4. Output Description

File Name	-	Transform run from the Multispectral Image Storage
DSRN	-	DATAPE
Device type	-	Tape
Usage	-	Output
Description	-	Tape containing transformed run

File Name	-	Informational and error messages
DSRN	-	PRNTR and TYPEWR
Device type	-	Printer and typewriter
Description	-	Message numbers are listed below. For text of message, see User's Manual.

MESSAGES

INFORMATIONAL	ERROR
* 61	104
* 66	139
* 67	426
68	427
69	434
70	435
**119	534
*120	535
*122	
*160	
161	

*means printed on typewriter only
**means printed on printer only

5. Supplemental Information

Not Applicable.

6. Flowchart

Not Applicable.

LARS Program Abstract 272

MODULE IDENTIFICATION

Module Name: RUNCOM

Function Name: CHANNELTRANSFORM
IDPRINT
TRANSFERDATA

Purpose: Block Data for RUNCOM

System/Language: CMS/FORTRAN

Author: J. Etheridge Date: 10/24/72

Latest Revisor: Date:

MODULE ABSTRACT

This is the BLOCK DATA subroutine for the RUNSUP load module, which contains the Channeltransform, Idprint, and Transferdata functions.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised October 1980

LARS Program Abstract 273

MODULE IDENTIFICATION

Module Name: TRARDR Function Name: TRANSFERDATA

Purpose: Reader for TRANSFERDATA function

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P. D. Alenduff Date: 03/04/75

MODULE ABSTRACT

This program reads the control cards for the Transferdata function and calls LAREAD to read in the data cards. TRARDR mounts a tape or tapes if they are needed and writes header and channel information on the requested output (printer, tape, or punch).

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

TRARDR

CALL TRARDR

TRARDR reads in the control cards for TRANSFERDATA, calls LAREAD to read in the first data card, prints, records on tape, and/or punches header and channel information, and then returns to RUNSUP.

TRADAT

CALL TRADAT (*)

* - 'RETURN 1' occurs when the END card is read by LAREAD

TRADAT is the entry point before the statement CALL LAREAD mentioned above, so it reads in one data card and sets up header and channel information. Unless an END card is read, it then returns to RUNSUP.

2. Internal Description

TRARDR essentially operates as described above. After the control cards have been read in, the program checks to see that the required CHANNELS card was read in, and, if not, gives the user an opportunity to type one in. If PRINT ROLL was the only output requested, the CHANNELS card is not required and TRARDR does not check for it. If no output was requested, the default PRINT DATA is set up.

The input tape (Multispectral Image Storage Tape) is mounted on DATAPE by a call to GETRUN, and the channels and areas requested are checked to see that they exist for the requested run. If a channel does not exist, it is omitted, a message is printed out, and processing continues. If an area does not exist, it is ignored and the next card is read. Then header and channel information are printed out on the printer. Header information also includes the module number, number of channels requested, and number of sample vectors in the module if TAPE or PUNCH output is requested.

The output tape, if requested, is mounted on DUPLTP. Next, header and channel information is punched and/or recorded on the output tape TRAOUT, if card and/or tape output was requested. Finally, control returns to RUNSUP.

If the END card is read in, TRARDR puts an end of file mark on TRAOUT and EXECUTES a 'RETURN 1'.

Complete list of subroutines called by TRARDR:

CTLWRD
ERPRNT
CHANEL
CTLPRM
IVAL
GETIME
GETRUN
TOPRU
MOUNT
TOPBF
TOPFF
TOPEF

Commons in TRARDR:

GLOCOM
RUNCOM

3. Input Description

File name - Multispectral Image Storage Tape
DSRN - DATAPE
Device type - Tape
Usage - Input
Description - Master input tape containing digitized scanner data

4. Output Description

File name - Header and channel information for Multispectral Image Data Listing
DSRN - PRNTR
Device type - Printer
Usage - Output for any Transferdata job
Description - Header: run number, lines requested, and columns requested.

Channel: for each channel requested, band width, calibration code, and CO are printed out. See User's Manual for sample output.

File name - Header and channel information for Multispectral
- Image Data Tape or Deck
DSRN - DUPLTP and/or PUNCH
Device type - Tape and/or punch
Usage - Output for TAPE and/or PUNCH requests
Description - Record size = 80

Header: one record containing run, lines, columns requested, number of channels requested, sample vectors in module, module number, and number of records for the module.

Channel: one record for each channel requested containing channel, calibration code, band width, and C0, C1, & C2.

File name - Informational and error messages
DSRN - PRNTR and TYPEWR
Device type - Printer and typewriter
Description - Message numbers are listed below. For text of message, see User's Manual.

MESSAGES

INFORMATIONAL	ERROR	
*64	104	440
75	139	530
	171	
	255	
	432	
	439	

*means printed on typewriter only

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 274

MODULE IDENTIFICATION

Module Name: TRAXEQ Function Name: TRANSFERDATA

Purpose: Prints, Punches, and/or records on tape the requested Multispectral Image Storage Tape

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: J. B. Etheridge Date: 10/24/72

MODULE ABSTRACT

TRAXEQ outputs the requested values from a Multispectral Image Storage Tape onto the printer, punch, and/or tape. It is called by RUNSUP after TRARDR has been called to read cards and output the header and channel information.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

TRAXEQ

CALL TRAXEQ (BDATA,RDATA)

BDATA - INTEGER*2 BDATA(N), where N is at least (ID(G)*NCR+1)/2. This array is used by GADLIN for reading from the Multispectral Image Storage Tape.

RDATA - REAL*4 RDATA(NSD,NCR), used by GADLIN to store converted BDATA.

2. Internal Description

TRAXEQ calls GADLIN to read in a line of data from the Multispectral Image Storage Tape. If ERROR, set by GADLIN, is greater than 0, TRAXEQ returns to RUNSUP. Otherwise, according to what the user requested, it first prints out the roll parameter, then prints the Multispectral Image Storage Tape values, records them on tape, and punches them, all for the first requested line of data. It keeps looping back to the call to GADLIN until the requested number of lines is reached, and then it returns to RUNSUP.

Note that the printed Multispectral Image Storage Tape values are FORTRAN real numbers with one decimal place and punched and taped Multispectral Image Storage Tape values are integers rounded off to the nearest integer.

Subroutines called by TRAXEQ:

GADLIN

Commons in TRAXEQ:

GLOCOM
RUNCOM

3. Input Description

File name - Multispectral Image Storage Tape
DSRN - DATAPE
Device type - Tape
Usage - Input
Description - Master input tape containing digitized scanner data

4. Output Description

File name - Roll Parameter
DSRN - PRNTR
Device type - Printer
Usage - Output for PRINT ROLL
Description - Either the roll parameter is printed out or a message to the effect that the roll was not calculated.

File name - Multispectral Image Data Listing
DSRN - PRNTR
Device type - Printer
Usage - Output if PRINT DATA requested
Description - Each printed line contains the line number, column number, and Multispectral Image Storage Tape values for each requested channel. See User's Manual for sample output.

File name - Multispectral Image Storage Tape values
DSRN - DUPLTP and/or PNCH
Device type - Tape and/or punch
Usage - Output if TAPE and/or PUNCH requested
Description - Record size = 80
Each record contains 24 Multispectral Image Storage Tape values (I3 format), the module number, and the record number. The Multispectral Image Storage Tape values are in the same order, line by line, as in the printed output (b). (See the User's Manual for printed output).

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 275

MODULE IDENTIFICATION

Module Name: SUBGEN Function Name: CHANNELTRANSFORMATION

Purpose: Run-Time compiler for data conversion cards

System/Language: CMS/OS/VS Assembler

Author: Paul Spencer Date: 1973

Latest Revisor: Louis Lang Date: 6/30/80

MODULE ABSTRACT

SUBGEN reads an input card image, and "compiles" it as a function at execution time.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL SUBGEN (CARD)

CARD - I*4 array of 20 fullwords. Contains the equation the user wants interpreted.

2. Internal Description

The card input is an 80 byte card image. It has the form of an equation using constants, function calls and variables of the form 'CNN' where NN is an integer between 0 and 30. CNN will be a new data channel on the output tape. These quantities may have the operations of additional subtraction, multiplication, division, and exponentiation performed on them. It translates the input algebraic form to an internal polish "stack" notation for faster execution time.

1a. Module Usage

CALL SUB(VALUES)

VALUES - R*4 (30) where the data values are stored for a channel.

2a. Internal Description

SUB performs the required calculations (provided by SUBGEN) on the values stored in VALUES and returns the new values to the program.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 280

MODULE IDENTIFICATION

Module Name: SAMSUP Function Name: SAMPLECLASSIFY

Purpose: Supervisor for the Sampleclassify function.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. K. Addessio Date: 11/20/72

MODULE ABSTRACT

This module is the supervisor for the Sampleclassify function which classifies points by incorporating spatial information; that is, given that a particular point is a member of a particular class, there is considerable likelihood that its neighbor is also. The results of the classification of each field is written on the printer.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SAMSUP

CALL SAMSUP

SAMSUP receives control from LARSMN and regulates the chain of events required to classify points by incorporating spatial information. In other words, classifications are made on a field basis rather than a point basis.

2. Internal Description

The chain of events necessary to complete the Sampleclassify function begins with a call to SAMRDR which reads, interprets and makes some checks on the control cards required by SAMPLECLASSIFY. In addition, SAMRDR initializes values and sets the value of DECK to indicate whether statistics will be read from cards or disk. Once a DATA card is encountered, control is returned to SAMSUP at which time a call is made to SAMINT to obtain statistics, allocate core, reduce and save statistics, and read and sort test field description cards. SAMSUP then calls SMCLS1 to print classification information and invert the covariance matrix. Finally SAMSUP calls SMCLS2 to actually perform the field classification. In addition, SAMSUP prints appropriate messages to the user typewriter indicating progress through the function.

List of subroutines called by SAMSUP:

SAMRDR
SAMINT
SMCLS1
SMCLS2

Common in SAMSUP:

GLOCOM

3. Input Description

Not Applicable

4. Output Description

File Name - Informational messages
DSRN - PRNTR and TYPEWR
Device type - Printer and typewriter
Usage - Output
Description - Message numbers are listed below.
For text of message see User's Manual.

MESSAGES

INFORMATIONAL

034
217
218

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 281

MODULE IDENTIFICATION

Module Name: SAMCOM Function Name: SAMPLECLASSIFY

Purpose: Block common for SAMSUP load module

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. K. Addessio Date: 11/20/72

MODULE ABSTRACT

This is a block data subroutine for the Sampleclassify function. Refer to the LARSYS System Manual for a list of the program modules that require the common.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 282

MODULE IDENTIFICATION

Module Name: SAMINT Function Name: SAMPLECLASSIFY

Purpose: Initiator for the Sampleclassify function.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. K. Addessio Date: 11/28/72

MODULE ABSTRACT

This initiator module sets up object time dimensions and calculates base addresses for arrays to be stored in ARRAY.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SAMINT

CALL SAMINT (COVAR3,AVAR3,FLDBAS,DATBS3,DECK)

Input arguments:

DECK - INTEGER*4, flag to indicate whether statistics will be read from cards or disk.

Output arguments:

COVAR3 - INTEGER*4, base address of the reduced covariance matrix

AVAR3 - INTEGER*4, base address of the reduced mean matrix

FLDBAS - INTEGER*4, base address of the test field array

DATBS3 - INTEGER*4, number of double words used in reduced ARRAY; also the base address of RDATA which is used later as a data buffer for GADLIN.

2. Internal Description

SAMINT first reads the statistics data from the card reader or card set on disk by a call to STAT. If read from card reader as indicated by input parameter DECK, statistics are placed on disk for future use by another Sampleclassify request or by some other function, within a particular terminal session. A call to REDSTA then reads the statistics deck into CORE. Validity checks, particularly related to pooled information, are then made on the CLASSES, GROUP and CHANNELS cards by calls to CLSCHK, GRPCHK, and FETCHK. Irrecoverable errors are so stated and handled by a call to ERPRNT after which the job is terminated. If no errors are encountered, SAMINT will then calculate the base addresses for the reduced arrays and will call REDSAV to reduce the appropriate statistics. A call is then made to RDFLDS to read and sort test field description cards and store them in CORE and on disk. Finally DATBS3 is calculated and a list of classes and channels to be processed is written on the printer under the title of 'Sampleclassify supervisor information'. Control returns to SAMSUP.

List of subroutines called by SAMINT:

STAT
REDSTA
CLSCHK
GRPCHK
FETCHK
REDSAV
RDFLDS

Commons in SAMINT:

GLOCOM
SAMCOM

3. Input Description

Statistics information are input via REDSTA. Field description information is input via RDFLDS.

4. Output Description

Statistics deck is placed on disk via STAT remaining available for use within a particular terminal session.

File name - Sampleclassify Supervisor Information
DSRN - PRNTR
Device type - Printer
Usage - Output
Description - List of classes and channels to be processed

File name - Error messages
DSRN - PRNTR and TYPEWR
Device type - Printer and typewriter
Usage - Output
Description - Message numbers are listed below, for text of messages see User's Manual.

MESSAGES

ERROR

371
372

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 283

MODULE IDENTIFICATION

Module Name: SAMINV Function Name: SAMPLECLASSIFY

Purpose: Inverts a matrix

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: PAUL W. SPENCER Date: 01/30/74

MODULE ABSTRACT

This subroutine finds the inverse and determinant of a matrix.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

Revised September 1974

1. Module Usage

SAMINV

CALL SAMINV (A,NOFEAT,DET,WORK)

Input Arguments:

A - REAL*4, the unique elements (lower triangular portion) of a symmetric matrix which is NOFEAT by NOFEAT. The elements are stored as follows: A_{11} A_{21} A_{22} A_{31} A_{32} A_{33}....

NOFEAT - INTEGER*4, number of elements in one row of the matrix represented by A above.

WORK - REAL*4, allotted working area with NOFEAT times (NOFEAT + 2) elements in it.

Output Arguments:

A - REAL*4, the inverted matrix in the same form as the input A above.

DET - REAL*4, determinant of the input matrix.

2. Internal Description

SAMINV first sets up the array to be inverted into the work array. The matrix is then inverted via a call to the SSP routine MINV after which it is loaded back into its original input array thus being input and output in the same array. Control returns to the calling program.

The Scientific Subroutine Package MINV is the only subroutine called by SAMINV.

No COMMON is used in this subroutine thus making it available for general use.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 284

MODULE IDENTIFICATION

Module Name: SAMRDR Function Name: SAMPLECLASSIFY

Purpose: Reader Module for SAMPLECLASSIFY

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: P. D. Alenduff Date: 08/16/74

MODULE ABSTRACT

This subroutine reads and interprets the control cards necessary to run the Sampleclassify function. It also checks for the presence and validity of that data.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SAMRDR

CALL SAMRDR (DECK)

Output Arguments:

DECK - INTEGER*4, flag to indicate whether statistics will be read from cards or disk.

This reader subroutine reads and interprets the necessary control cards to run SAMPLECLASSIFY.

2. Internal Description

After initializing values, control cards are interpreted by a call to CTLWRD. If an end of file is reached, ERRCOR is set to 2 and the job is terminated by a call to ERPRNT, otherwise control is passed on the value of CODE for detailed interpretation. When a data card is encountered, SAMRDR will check for the presence and validity of required control card information. If the PRINT card was used, a message will be printed stating that the user has selected the option of printing multispectral statistics. Control is then returned to SAMSUP. If interpretation errors occur, a related message is printed to both the typewriter and printer and the user is allowed to type in a corrected card. If, however, an irrecoverable error occurs, a message is printed to that affect and the job is terminated at that point.

List of subroutines called by SAMRDR:

CTLWRD
ERPRNT
CTLPRM
POLSCN
GRPSCN
CHANEL

Commons in SAMRDR:

GLOCOM
SAMCOM

3. Input Description

Not Applicable

4. Output Description

File name - Informational and error messages
DSRN - PRNTR and TYPEWR
Device type- Printer and typewriter
Usage - Output
Description- Message numbers are listed below, for text of message see User's Manual.

MESSAGES

INFORMATIONAL	ERROR
211	341
	146
	171
	124
	344
	254

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 285

MODULE IDENTIFICATION

Module Name: SMCLS1 Function Name: SAMPLECLASSIFY

Purpose: Prints classification and training field information

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. K. Addessio Date: 11/28/72

MODULE ABSTRACT

The SMCLS1 module prints a statistics summary of the selected classes, channels and training fields. It also inverts the covariance matrix and calculates the determinant of that matrix via SAMINV.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SMCLS1

CALL SMCLS1 (COVMTX,AVEMTX,DET,COR)

Input arguments:

COVMTX - REAL*4, (VARSZ3,NOPOOL), reduced statistics covariance matrix where VARSZ3 identifies the size of the reduced covariance matrix and NOPOOL indicates the number of pooled classes.

AVEMTX - REAL*4, (NOFET3,NOPOOL), reduced statistics average matrix where NOFET3 indicates the number of channels requested and NOPOOL indicates the number of pooled classes.

Output arguments:

DET - REAL*4(60), determinant vector (DET(I)= the determinant of the I'th covariance matrix)

COR - REAL*4, correlation matrix

COVMTX - REAL*4, inverse of input matrices

2. Internal Description

SMCLS1 first prints the title 'Perfield classification Study..' along with a statistics summary of the considered classes and channels. A call to WRTTRN then prints out the saved training fields. If requested, as indicated by the presence of a PRINT card, the standard deviation and correlation matrix are calculated. The mean, standard deviation and correlation matrix are then printed for each of the defined classes. Finally SAMINV is called to calculate the inverse covariance matrix and the determinant of that matrix for each defined class. Control returns to SAMSUP.

List of subroutines called by SMCLS1:

GETIME
WRTTRN
WRTMTX
SAMINV

Commons in SMCLS1:

GLOCOM
SAMCOM

3. Input Description

Not Applicable

4. Output Description

Error message 160 to the printer and typewriter: Covariance matrix is singular - processing terminated

File name - Title and statistics summaries
DSRN - PRNTR
Device type - Printer
Usage - Output
Description - Two separate summaries are printed. One listing the selected classes and channels with related calibration codes and spectral band information. The other listing the saved training fields.

File name - Optional statistics information
DSRN - PRNTR
Device type - Printer
Usage - Output
Description - Mean, standard deviation and correlation matrix are printed for each selected class.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 286

MODULE IDENTIFICATION

Module Name: SMCLS2 Function Name: SAMPLECLASSIFY

Purpose: Classifications made on a per field basis

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. K. Addessio Date: 11/30/72

MODULE ABSTRACT

SMCLS2 does the real work of classifying on a per field basis.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SMCLS2

CALL SMCLS2(COVMTX,AVEMTX,FLDARY,SUMAB,DET,RDATA,DATBS3)

Input arguments:

COVMTX - REAL*4(VARSZ3,NOPOOL), inverse of reduced covariance matrix where VARSZ3 identifies the size of the reduced covariance matrix and NOPOOL indicates the number of pooled classes.

AVEMTX - REAL*4(NOFET3,NOPOOL), reduced mean matrix where NOFET3 indicates the number of channels requested and NOPOOL indicates the number of pooled classes.

FLDARY - INTEGER*4(10,NOFLDS), array containing test field information where NOFLDS is the number of fields to be classified.

```
FLDARY(1,N)    = run number
FLDARY(2-4,N)  = line information
FLDARY(5-7,N)  = column information
FLDARY(8,N)    = input group
FLDARY(9,N)    = classification result pool
FLDARY(10,N)   = sort pointer for line and
                 run numbers.
```

SUMAB - REAL*4(VARSZ3), work area allotted for the sum of the arrays A B

DET - REAL*4(60), array containing determinants.

RDATA - REAL*4(1), work area allotted as data buffer for GADLIN.

DATBS3 - INTEGER*4, starting point of RDATA in core.

2. Internal Description

SMCLS2 first initializes for classification, sorts group names for printing and forms statistics products via SMMULT. The DO 410 loop is then entered to classify and list one field at a time in the order found on the data tape.

The inner workings of the DO 410 loop include a call to GETRUN to find the desired run and mount and position the Multispectral Image Storage Tape. The calibration set vector is then set up and a check is made for new calibration values. The run information header is printed via a call to HEADER after which the field size is computed and a check is made for core overflow. At this time, an inner DO 280 loop is reached which reads in data from the storage tape and sets up the variables to calculate the field means and covariances. Field statistics are then formed by a sequence of calls to SAMINV and SMMULT. Calculations and comparisons are made for classification and the classified field information is then output to the printer. As a result, at the end of the DO 410 loop we have a list of classified fields as ordered on the data storage tape by increasing line number.

At this point the field classification results are reordered and printed in the order supplied by the user under the title of 'Summary of Results'. Performance information is then tabulated and printed under the heading of 'Performance Summary'.

Processing is completed and control returns to LARSMN via SAMSUP.

List of subroutines called by SMCLS2:

TSTREQ
GETRUN
RUNERR
GETIME
HEADER
GADLIN
LINERR
SAMINT
SUMMULT
ERPRNT

Commons in SMCLS2:

GLOCOM
SAMCOM

3. Input Description

File name - Multispectral Storage Tape
DSRN - DATAPE
Device type - Tape
Usage - Input
Description - Master input tape containing digitized scanner data.

Field description cards are input via RDFLDS.

4. Output Description

File name - Printed output
DSRN - PRNTR
Defice type - Printer
Usage - Output
Description - The following four tables are printed:

1) Classification run identification table
2) Classification results as ordered on the tape
3) Classification results as input by the user
4) Classification Performance summary.

File name - Optional printed output
DSRN - PRNTR
Device type - Printer
Usage - Output
Description - Statistics summary showing the mean and the standard deviation of the response for each channel, and a correlation matrix of the channels for each class used.

File name - Informational and error messages
DSRN - PRNTR and TYPEWR
Device type - Printer and typewriter
Usage - Output
Description - Message numbers are printed below. For text of message see User's Manual

MESSAGES

INFORMATIONAL	ERROR
23	178
212	376
213	101
214	
215	

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

469

LARS Program Abstract 287

MODULE IDENTIFICATION

Module Name: SMMULT Function Name: SAMPLECLASSIFY

Purpose: Compute vector and scalar products

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: B. K. Addessio Date: 11/29/72

MODULE ABSTRACT

This subroutine computes vector and scalar products of a matrix and a vector.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SMMULT

CALL SMMULT(A,B,C,WORK,NOFEAT)

Input Arguments:

A - REAL*4, the unique elements (lower triangular portion) of a symmetric matrix which is NOFEAT by NOFEAT. The elements are stored as follows: $A_{11}\ A_{21}\ A_{22}\ A_{31}\ A_{32}\ A_{33}\cdots$

B - REAL*4, vector with NOFEAT elements

WORK - REAL*4, allotted working area with NOFEAT elements

NOFEAT - INTEGER*4, number of elements in vector B

Output Arguments:

B - REAL*4, product of A and B

C - REAL*4, scalar product B·(AB)

2. Internal Description

Initializes values and forms the vector and scalar products for NOFEAT elements. All data is passed via arguments thus making it available for general use.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 300

MODULE IDENTIFICATION

Module Name: SEPSUP Function Name: SEPARABILITY

Purpose: Supervisor for the Separability function

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 12/18/72

MODULE ABSTRACT

SEPSUP, being the supervisor routine for SEPARABILITY, calls on subroutines which read the control cards, set up dynamic core allocation, perform the divergence calculations, print the results, and allow the user to input new parameters and see the results.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SEPSUP

CALL SEPSUP

No parameters are passed to SEPSUP, since it is the supervisor for the Separability function and is called when *SEPARABILITY has been specified.

2. Internal Description

SEPSUP begins by calling SEPINT to read the function control cards and statistics, and set up the initial array bases. If the checkout option is in effect and no calculations are to be performed (i.e. if CHKOUT is true) a return is executed. The call to DIVRG1 causes information about the channels, classes and training fields to be output to the printer. If SEPSCD equals SEPTRX, the subroutine MOUNT requests that a scratch tape be mounted. After calling ARBASE to compute the remaining array bases, SEPINT calls DIVRG2 to perform the divergence calculations. If STOPFG now equals 2, some statistics were ill-conditioned, and the next combination request is obtained. If STOPFG equals 1, the user halted the calculations, so the best set of channels is stored and a return is executed. DIVPRT is called to get these divergence results and use them to compute separability and print the results. If IBR equals 2, USER is called to allow the user to enter options, and then DIVPT1 is called to compute the separability and print the results. If IBR equals 1 the next combination request is obtained. When all requests have been processed, the best set of channels is stored in TEMPAS to pass on to CLASSIFYPOINTS, and a return is executed. Both GLOCOM and SEPCOM are used in SEPSUP.

3. Input Description

Not Applicable

4. Output Description

SEPSUP begins by printing at the typewriter:

```
I0111 SEPARABILITY FUNCTION REQUESTED        (SEPSUP)
```

and ends by printing:

```
I0011 SEPARABILITY FUNCTION COMPLETED        (SEPSUP)
```

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

474

LARS Program Abstract 301

MODULE IDENTIFICATION

Module Name: ARBASE Function Name: SEPARABILITY

Purpose: Calculates array bases

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 12/15/72

MODULE ABSTRACT

ARBASE calculates the remaining array bases for the current combination request.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

ARBASE

CALL ARBASE is all that is needed to use this module since no parameters are passed to it. The remaining array bases will be calculated.

2. Internal Description

ARBASE calculates the number of channel combinations and corresponding divergences (i.e. the numbers printed on one line of the SEPARABILITY output) which can be stored in the remaining space of ARRAY, and then computes the array bases for the current combination request. If core overflow occurs, ERPRNT is called to print an error message and the core overflow flag IERR is set to 1 before returning to the calling routine. Both GLOCOM and SEPCOM are present in this module.

3. Input Description

The only information needed is that passed in the common SEPCOM.

4. Output Description

A 'MAIN STORAGE OVERFLOW' message is printed if core overflow occurs. The calculated array bases are returned via SEPCOM.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

476

LARS Program Abstract 302

MODULE IDENTIFICATION

Module Name: BSTCHK Function Name: SEPARABILITY

Purpose: Checks Validity of Combination and Show Requests

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 12/15/72

MODULE ABSTRACT

BSTCHK determines whether each combination request is valid and deletes those larger than the number of channels given. The entry point SHWCHK scans each show request and ignores those with invalid channels.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

BSTCHK

CALL BSTCHK is all that is required to use this part of the module. Each number from the COMBINATION request is checked to be sure it is not greater than the number of channels available for use in SEPARABILITY.

SHWCHK

CALL SHWCHK(SHWVEC)

Input argument:

SHWVEC - INTEGER*2, Array (400)

Output argument:

SHWVEC - INTEGER*2, Array (400) SHWCHK examines each show request found in SHWVEC and returns, in SHWVEC, those that are correct.

2. Internal Description

BSTCHK determines the validity of the numbers in the combination request by comparing them to the number of channels being used(NOFET3). When a number in the combination request is larger than the number of channels, it is deleted from further consideration, and a warning message is printed on the typewriter.

ENTRY SHWCHK looks at each show request and determines its validity by checking for ordering of the channels by increasing size and deciding whether each channel is given. An invalid show request causes a message indicating this to be written at the terminal and the request is deleted from further consideration. Both GLOCOM and SEPCOM are found in this module.

3. Input Description

Not Applicable

4. Output Description

When an invalid combination (k) is discovered in a combination request, the message

```
I0017 "BEST" K IS GREATER THAN NO.  OF FEATURES
       IN GIVEN DATA...IGNORED
```

appears on the typewriter.

The entry point SHWCHK prints the message

```
I0031 INVALID "SHOW" REQUEST....N1 N2....NK
```

on the typewriter when a show request N1, N2, ..., NK containing invalid channels is found.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

479

LARS Program Abstract 303

MODULE IDENTIFICATION

Module Name: DIVERG Function Name: SEPARABILITY

Purpose: Computes Separability Using Divergence

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: R.L. Kettig Date: 10/16/74

MODULE ABSTRACT

DIVERG computes the divergence for each pair of classes for each set of channels.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

480

1. Module Usage

DIVERG

CALL DIVERG (COV,AVE,INVCOV,BUF,DIVSUM)

Input Arguments:

AVE - REAL*4, Array (BSTFET*,NOPOOL) class mean matrix for current set of channels.

COV - REAL*4, Array (BSTVAR*,NOPOOL) covariance matrix for current set of channels.

INVCOV - REAL*4, Array (BSTVAR*,NOPOOL) inverse covariance matrix.

Output Arguments:

BUF - REAL*4 Array (DIVSIZ), Divergence for each class combination.

DIVSUM- REAL*4, Sum of divergence for all class combinations.

DIVERG calculates the divergence for each pair of classes for one set of channels.

2. Internal Description

DIVERG computes the divergence for each pair of classes for the current set of channels. Usually DIVERG will be called on to perform these calculations for all sets of channels corresponding to the current combination request. The mean vector, covariance matrix and inverse covariance matrix for the current set of channels for each class are used for these computations. The formula implemented is:

$$D = \tfrac{1}{2}\,\mathrm{tr}\left\{[K_i^{-1} + K_j^{-1}]([K_i+K_j]+[M_i-M_j][M_i-M_j]^T)\right\} -2n$$

where

tr(A) = trace of A=sum of elements along the main diagonal of A

A^T = transpose of A

K_i = covariance matrix for class i

K_i^{-1} = inverse of the covariance matrix for class i

M_i = mean vector for class i

n = number of channels in current set.

Elements of the common block SEPCOM are used in DIVERG.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

482

LARS Program Abstract 304

MODULE IDENTIFICATION

Module Name: DIVPRT Function Name: SEPARABILITY

Purpose: Prints results from Separability Step

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 12/15/72

MODULE ABSTRACT

DIVPRT obtains the separability information for the current combination request from either tape or disk, makes sure the results are ordered as specified by the user (either by D(AVE) or DIJ(MIN)) and prints the results on the printer.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

DIVPRT

CALL DIVPRT (SYMCOM,INTWGH,SPLCOM,IBUF,WEIGHT,DIVSUM,
MINTAB,DIVTAB,DIR,REVTAB,KEPVEC,SAVQUE)

Input arguments:

SYMCOM - INTEGER*2 Array(DIVSIZ), contains all symbol pairs used to represent class combinations, based on the symbols supplied on the SYMBOLS card.

INTWGH - INTEGER*2 Array(NOWTS) contains a pointer to the position of a symbol pair in SYMCOM followed by its associated weight, etc., for all symbol pairs assigned a weight on a WEIGHTS card.

SPLCOM - INTEGER*2 Array(NOSHW), contains all SHOW requests initially entered on the input control cards, in the format of the number of channels in a request followed by the set of channels, etc., for each SHOW request. A zero appears after the last SHOW request.

IBUF - INTEGER*4, Array(DIVSIZ) position of IBUF in dynamically allocated array is passed.

Output arguments:

WEIGHT - INTEGER*2 Array(DIVSIZ), contains the weights of all symbol (or class) pairs, with a default weight of 10 assigned to all pairs not specified on a WEIGHTS card. The number found in WEIGHT(I) is the weight for the symbol pair in SYMCOM (I).

DIVSUM - READ*4, Array (KEEPLV) contains the sum of all separations for each channel combination.

MINTAB - INTEGER*4, Array(KEEPLV) contains the minimum separation of the class pairs for each channel combination.

DIVTAB - INTEGER*2, Array(DIVSIZ,KEEPLV) contains the divergences for each class pair for each channel combination. DIVTAB(I,J) is the divergence of Ith class pair (in SYMCOM) and Jth channel combination (in KEPVEC).

DIR - INTEGER*2, Array(KEEPLV) vector used to order DIVSUM, MINTAB, and DIVTAB according to D(AVE) or DIJ(MIN) as specified by the user. DIR(I) points to the position in DIVSUM, MINTAB and DIVTAB for the information which goes on the Ith line of results on the printer output.

REVTAB - INTEGER*2, Array(SPLMAX) contains the positions of the SHOW requests in the ordering of divergences for all channel combinations.

KEPVEC - INTEGER*2, Array(KEEPLV,BSTFET) contains all the channel combinations which can be stored, with the SHOW requests appearing first.

SAVQUE - INTEGER*2, Array(BSTFET,SPLMAX) contains the SHOW requests which have the same number of channels as the current combination request. SAVQUE(I,J) contains the Ith channel of the Jth SHOW request.

DIVPRT prints the results for SEPARABILITY by setting up page formats, retrieving the data from the scratch device, ordering the information by DIJ(MIN) if MINFLG=1, or D(AVE) otherwise, and printing the channel combinations and their separabilities for each class pair.

<u>DIVPT1</u>

```
CALL DIVPT1(SYMCOM,INTWGH, SPLCOM,IBUF,WEIGHT,DIVSUM,
            MINTAB,DIVTAB,DIR,REVTAB,KEPVEC,SAVQUE)
```

This entry point is called after the user has entered options at the terminal, and avoids some initialization and obtaining SHOW requests again. In all other respects, DIVPT1 AND DIVPRT are the same.

2. Internal Description

The initial page formats are set up according to the number of channels in each combination. Weights from the initial control cards are saved and SHOW requests containing the correct number of channels are obtained by calling GETSHW. Data is retrieved from the scratch device (tape or disk) by calling GETDAT and then ordered according to D(AVE) if MINFLG=0, or DIJ(MIN) otherwise. Using the number of class pairs (DIVSIZ), the remaining page formats are set up and the separability information is printed, as ordered, with a channel combination, followed by the minimum, the average, and the individual divergence values for each class pair for that channel combination. If there are too many class pairs to fit across one page, more pages will be set up containing the channel combination and a continuation of the divergence values for the class pairs. If DIVFLG is equal to 1 then DIVPRT will print the "CLASSES THAT MAY BE COMBINED" list and the "RESULTS OF SEPARABILITY GROUPING" table. The grouping table consists of the group numbers and their corresponding classes, symbols and point totals. The classes are optionally grouped by a call to CGROUP which uses a user supplied valve as a threshold (See supplemental information). When all results are printed, a return is executed to obtain options from the terminal, unless STOPFG = 1, in which case the initial control card values are stored, the best set of channels is saved, and a return is executed. Both SEPCOM and GLOCOM are used in this subroutine.

Subroutines called by DIVPRT

GETSHW	CGROUP
GETDAT	
ERPRNT	

3. Input Description

GETDAT is called to obtain the separability information from the scratch device (tape or disk).

4. Output Description

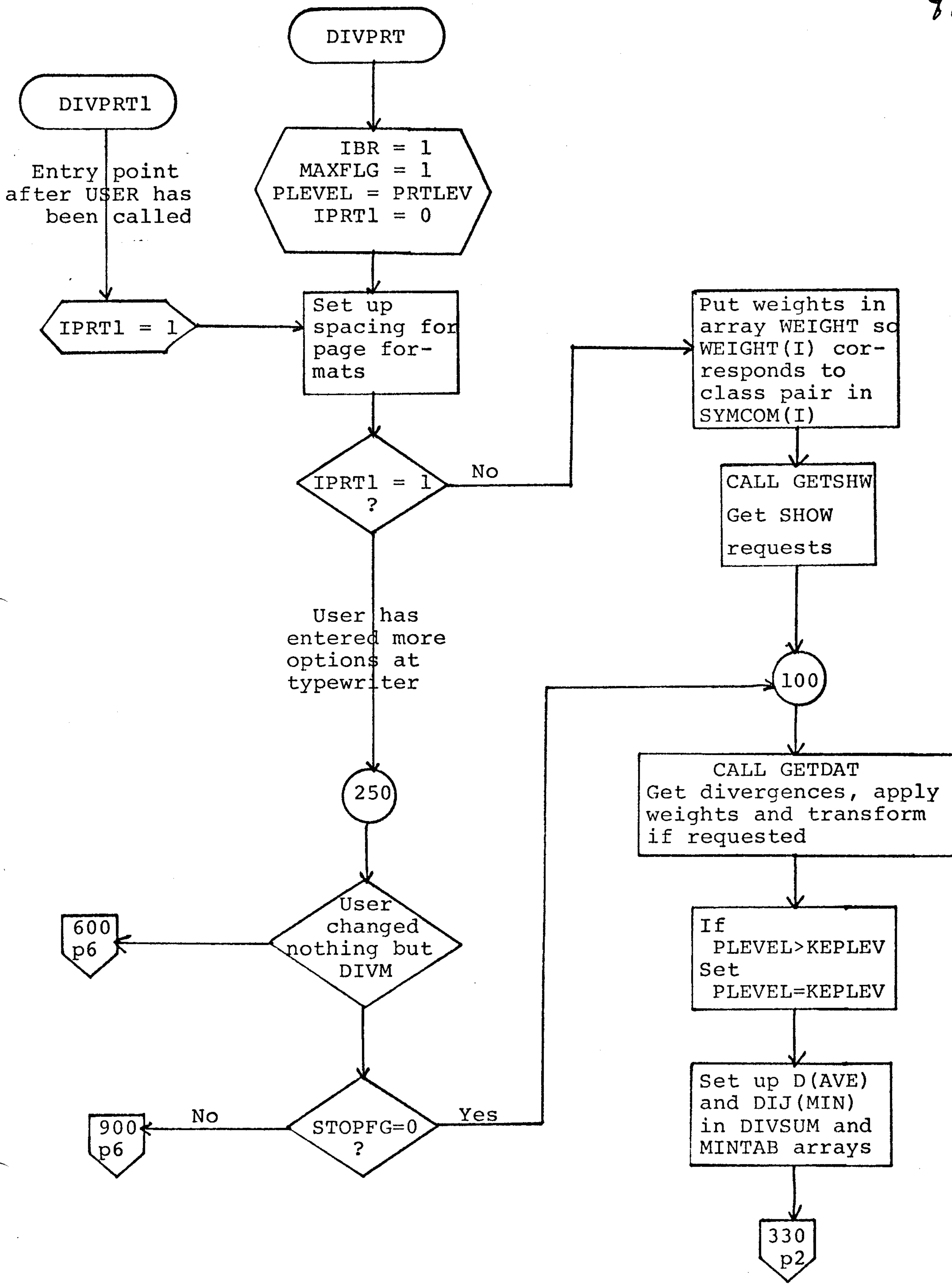
Examples of the printed output from DIVPRT may be found in the User's Manual.

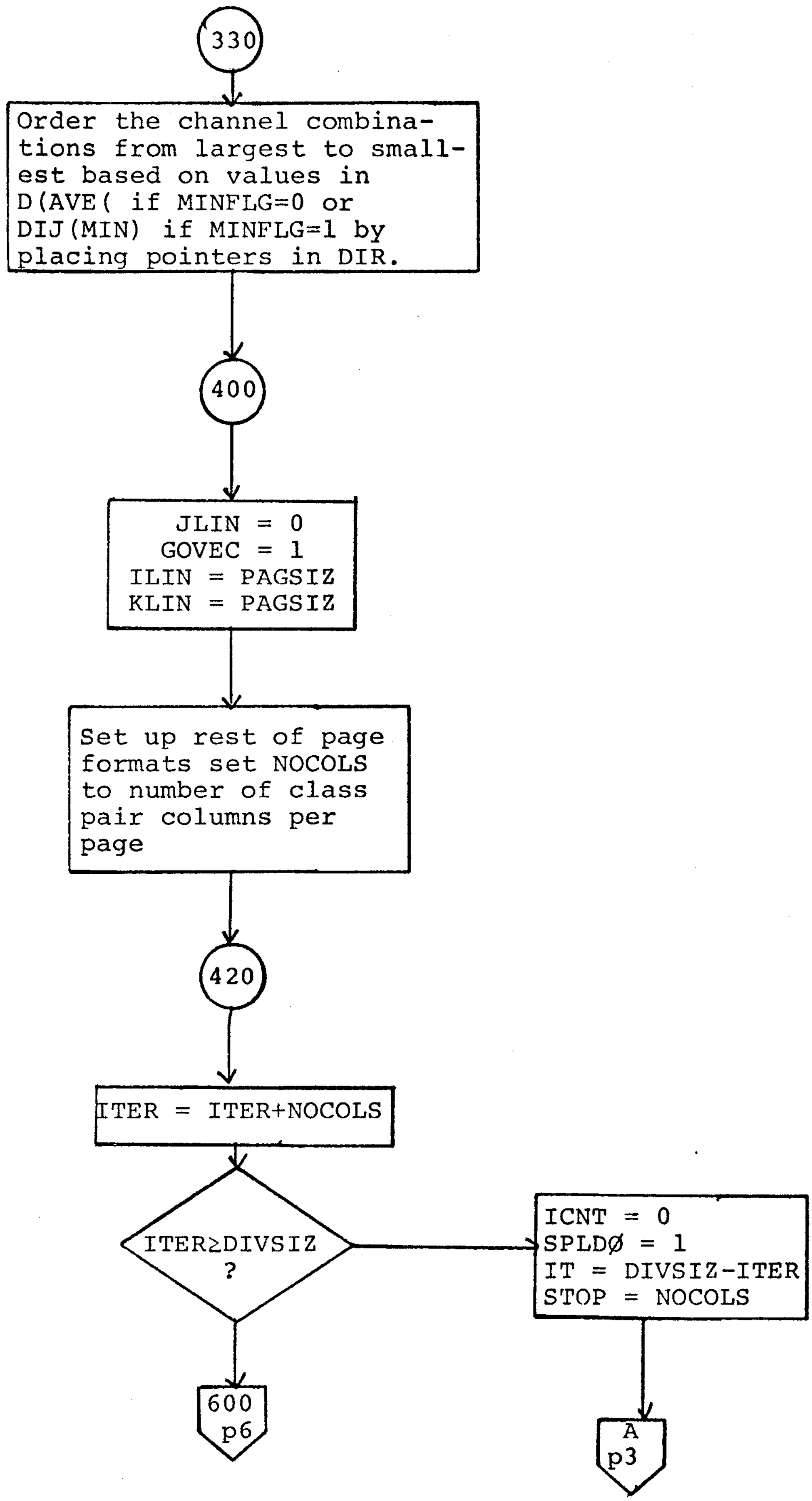
5. Supplemental Information

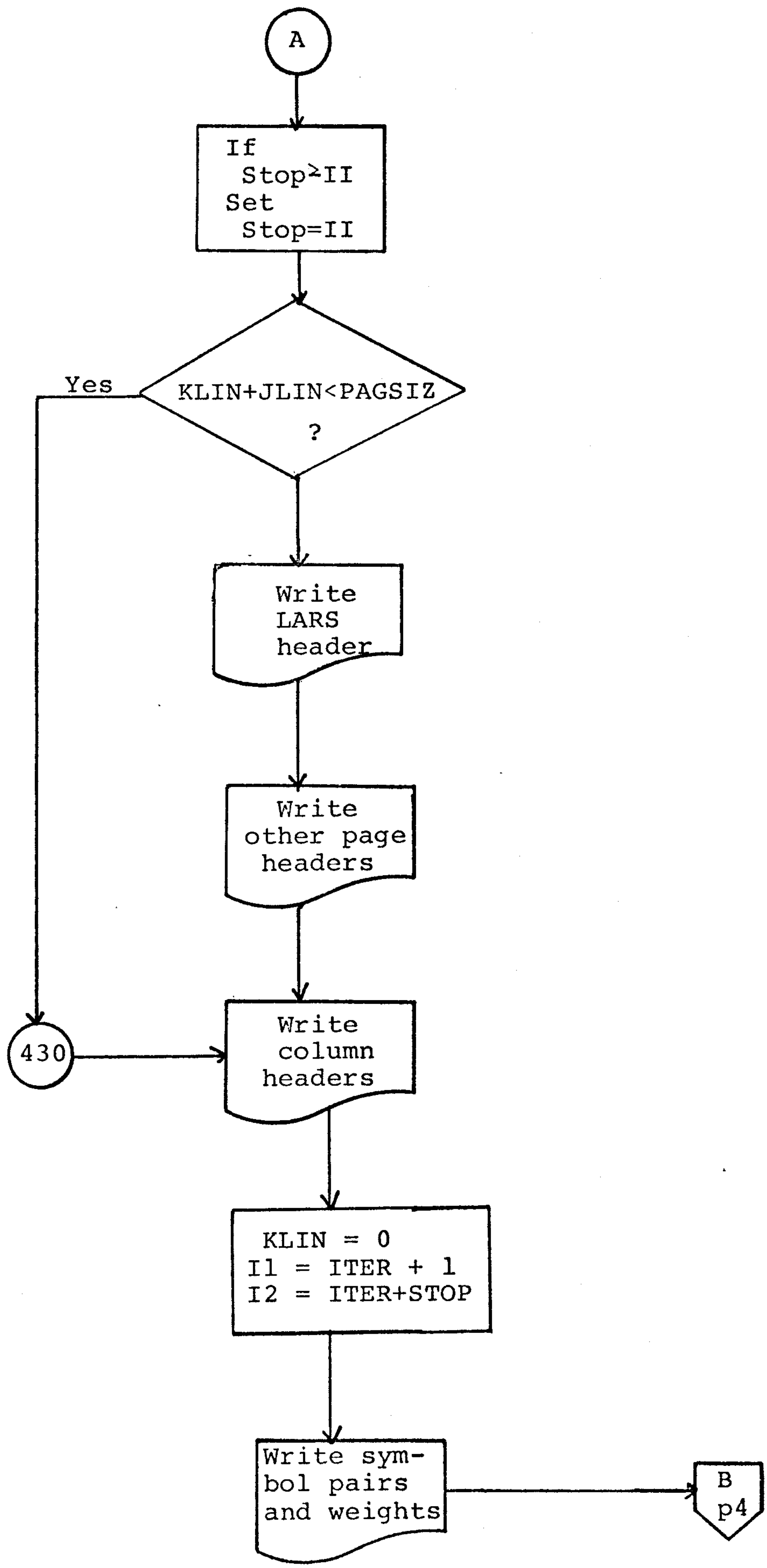
See *SEPARABILITY control cards (DIV(VALUE) OPTION)

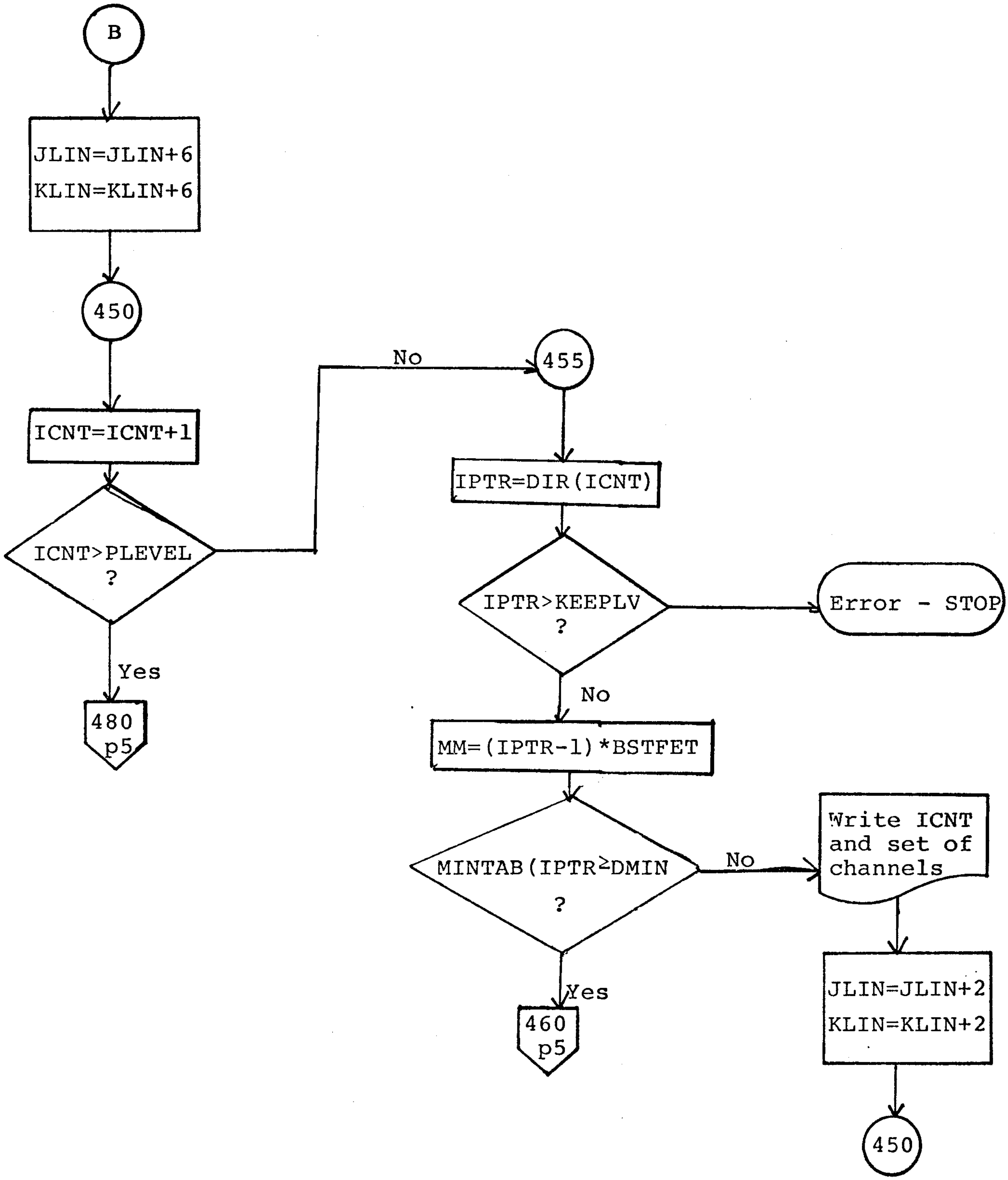
6. Flowchart

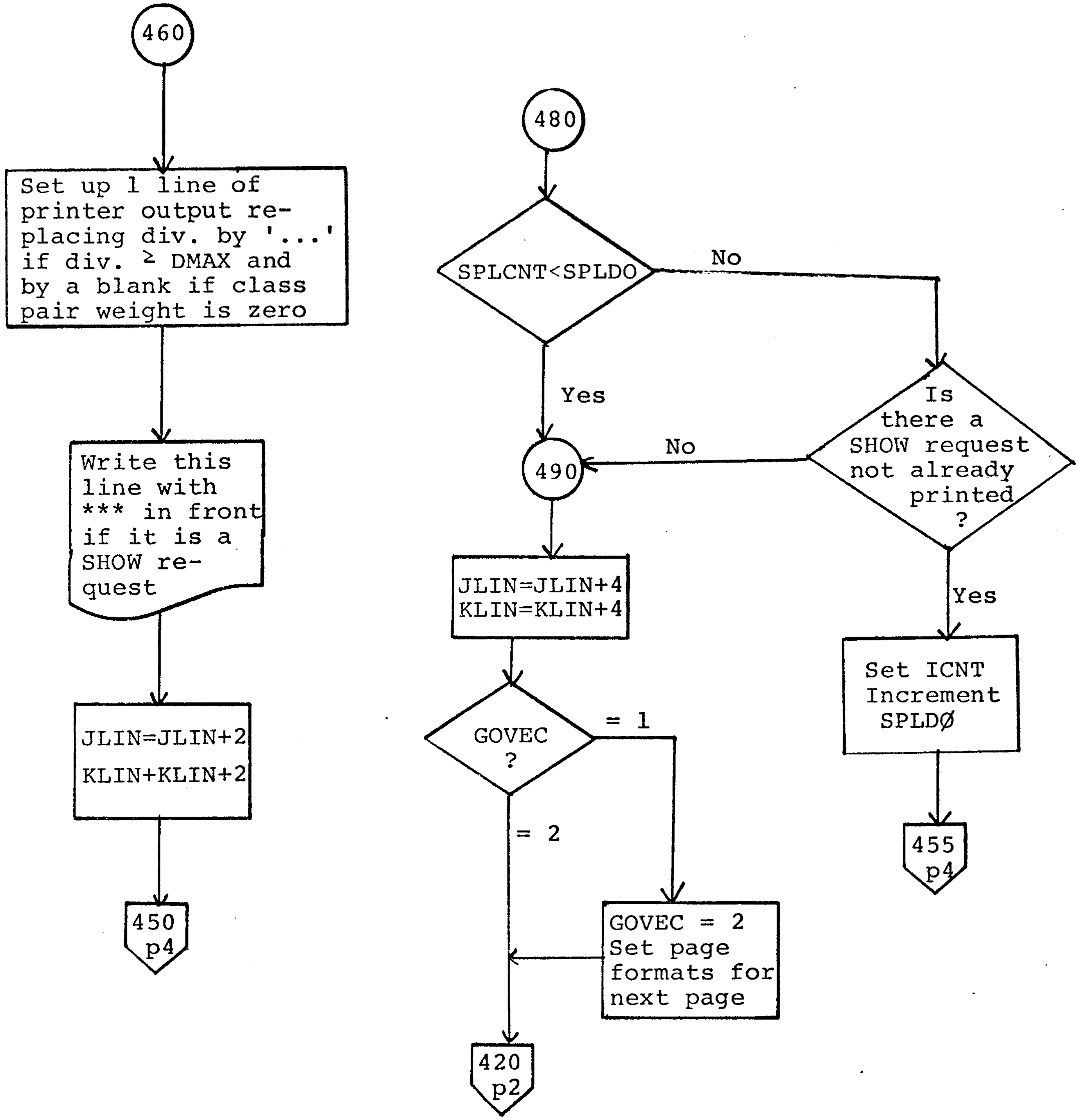
(Page numbers refer to pages of the flowchart.)


DIVPRT
DIVPRT1
Entry point after USER has been called
IBR = 1
MAXFLG = 1
PLEVEL = PRTLEV
IPRT1 = 0
IPRT1 = 1
Set up spacing for page formats
IPRT1 = 1 ?
No
Put weights in array WEIGHT so WEIGHT(I) corresponds to class pair in SYMCOM(I)
CALL GETSHW
Get SHOW requests
User has entered more options at typewriter
100
250
CALL GETDAT
Get divergences, apply weights and transform if requested
600 p6
User changed nothing but DIVM
If PLEVEL>KEPLEV Set PLEVEL=KEPLEV
Set up D(AVE) and DIJ(MIN) in DIVSUM and MINTAB arrays
900 p6
No
STOPFG=0 ?
Yes
330 p2

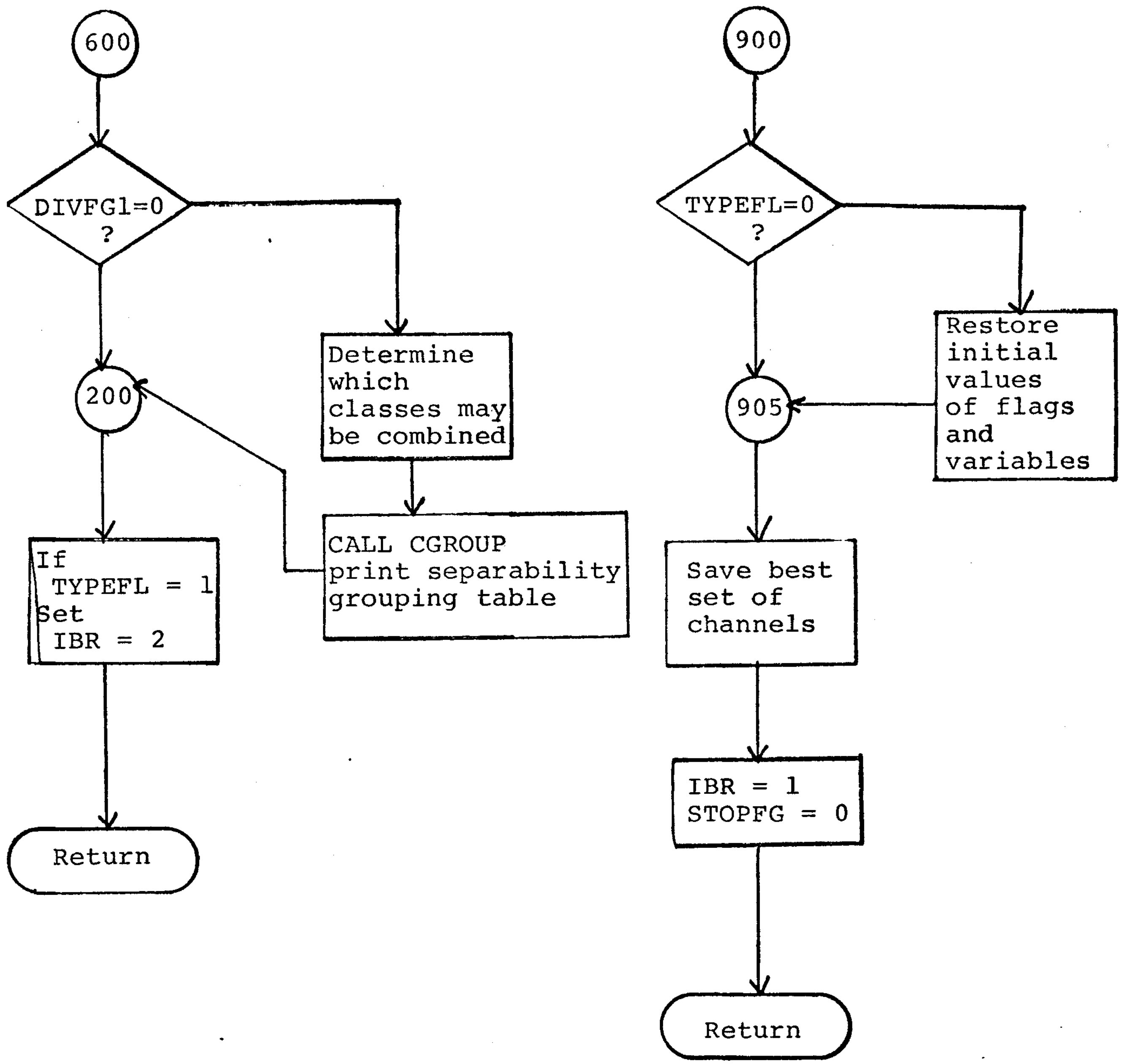

330
Order the channel combinations from largest to smallest based on values in D(AVE(if MINFLG=0 or DIJ(MIN) if MINFLG=1 by placing pointers in DIR.
400
JLIN = 0
GOVEC = 1
ILIN = PAGSIZ
KLIN = PAGSIZ
Set up rest of page formats set NOCOLS to number of class pair columns per page
420
ITER = ITER+NOCOLS
ITER≥DIVSIZ ?
ICNT = 0
SPLDØ = 1
IT = DIVSIZ-ITER
STOP = NOCOLS
600 p6
A p3


A
If
Stop≥II
Set
Stop=II
KLIN+JLIN<PAGSIZ
?
Yes
Write
LARS
header
Write
other page
headers
430
Write
column
headers
KLIN = 0
I1 = ITER + 1
I2 = ITER+STOP
Write sym-
bol pairs
and weights
B
p4


B
JLIN=JLIN+6
KLIN=KLIN+6
450
ICNT=ICNT+1
ICNT>PLEVEL ?
No
Yes
480 p5
455
IPTR=DIR(ICNT)
IPTR>KEEPLV ?
Error - STOP
No
MM=(IPTR-1)*BSTFET
MINTAB(IPTR≥DMIN ?
No
Write ICNT and set of channels
Yes
460 p5
JLIN=JLIN+2
KLIN=KLIN+2
450



(Note: 'div.' is abbreviation for 'divergence')


600
DIVFG1=0 ?
Determine which classes may be combined
200
CALL CGROUP print separability grouping table
If TYPEFL = 1 Set IBR = 2
Return
900
TYPEFL=0 ?
Restore initial values of flags and variables
905
Save best set of channels
IBR = 1 STOPFG = 0
Return

492

LARS Program Abstract 305

MODULE IDENTIFICATION

Module Name: DIVRG1 Function Name: SEPARABILITY

Purpose: Print and Save Divergence Information

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 12/15/72

MODULE ABSTRACT

DIVRG1 prints out the classes being considered and their corresponding symbols, the channels being considered, and the set of training fields being used. If requested, the means, standard deviations and correlation matrix for each class will also be printed.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

DIVRG1

CALL DIVRG1 (COVMTX,AVEMTX,COR)

Input Arguments:

COVMTX - REAL*4 Array (VARSZ3,NOPOOL), Reduced covariance matrix

AVEMTX - REAL*4 Array (NOFET3,NOPOOL), Reduced mean matrix.

Output Arguments:

COR - REAL*4 Array(NOFET3,NOFET3) correlation matrix.

This module outputs the initial information it receives, on the printer. The classes and their corresponding symbols, and the channels and their spectral band and calibration code will be printed. Then the training fields are listed. If statistics are requested, the means, standard deviations and correlation matrix will be printed for each class.

2. Internal Description

After printing the classes and channel information, WRTTRN is called to write out the training field information. Then STATKY is tested. If it is zero, no statistics have been requested, and a return is executed. If STATKY=1, then for each class, the correlation matrix is calculated and the means, standard deviations, and correlation matrix are printed before a return is executed. WRTMTX is used to print the correlation matrix and GETIME is also called each time a new page is to be printed. Both SEPCOM and GLOCOM appear in this module.

3. Input Description

Not Applicable

4. Output Description

The headings

CLASSES CONSIDERED		CHANNELS CONSIDERED		
SYMBOL	CLASS	CHANNEL NO.	SPECTRAL BAND	CAL. CODE

will appear on the printer, followed by the appropriate information. If statistics have been requested, then for each class, the headings

'CLASS...,
CHANNEL
SPECTRAL
BAND
MEAN
STD DEV

will appear followed by the indicated information. Then

CORRELATION MATRIX

appears, followed by the matrix. Examples of these outputs appear in the User's Manual.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

495

LARS Program Abstract 306

MODULE IDENTIFICATION

Module Name: DIVRG2 Function Name: SEPARABILITY

Purpose: Supervises Divergence Calculations

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: R.L. Kettig Date: 10/16/74

MODULE ABSTRACT

DIVRG2 calls subroutines which calculate the inverse of the covariance for each class, and the divergence for each class combination for each combination of channels. This information is saved either on disk or tape.

1. Module Usage

DIVRG2

CALL DIVRG2 (COVMTX,AVEMTX,INVCOV,IBUF,REDCOV,REDAVE)

Input Arguments:

COVMTX - REAL*4, Array (VARSZ3,NOPOOL), covariance matrix for all classes or pools.

AVEMTX - REAL*4, Array (NOFET3,NOPOOL) class means for each channel. AVEMTX (I,J) is the mean for class J and channel I.

IBUF - REAL*4, Array(DIVSIZ) temporary storage of divergences for each class combination.

DIVRG2 uses the COVMTX and AVEMTX matrices to compute REDCOV and REDAVE, the reduced matrices for each channel combination, and to compute INVCOV, the inverse of the reduced covariance matrix. REDAVE, REDCOV, and INVCOV are then used to compute the divergences for each channel combination. These are written on the scratch device.

2. Internal Description

DIVRG2 first calculates the number of channel combinations which can be formed from the number of channels (NOFET3) available. This indicates how many sets of divergence calculations will be performed. The computation loop is then entered, beginning by calling GETINV to find the next channel combination and to compute the reduced mean, covariance matrices and the inverse of the covariance matrix for that channel combination. Then DIVERG is called to compute the divergences and the loop is completed by writing these divergences on the scratch device (disk or tape). If the user has not requested that the calculations stop (CALL TSTREQ), the next set of calculations is performed. Otherwise, STOPFG is set to 1 and a return is executed. A return may also be executed after GETINV is called if 1) no more channel combinations exist, in which case a record starting with -1.0 is written on the scratch device, or 2) statistics for some class are ill-conditioned, in which case STOPFG is set to 2. Both SEPCOM and GLOCOM common blocks are required for this module.

3. Input Description

Not Applicable

4. Output Description

The divergences are output to the scratch device (SEPSCD), which may be disk or tape. Each record contains a REAL*4 sum of divergences (DIVSUM), the current channel combination (each channel in INTEGER*2 format) and finally a list of divergences for each channel combination in REAL*4 format.

Every 100 iterations the message:

I0021_____OF_____CALCULATIONS COMPLETED.

appears at the terminal to indicate how the calculations are progressing. When all computations are completed, the message

I0022 DIVERGENCE CALCULATIONS COMPLETED --READY TO ORDER AND PRINT RESULTS

will appear at the typewriter. If the user has requested that the calculations be stopped, the following message is sent to the terminal:

I0023 PROCESSOR TERMINATING DUE TO STOP COMMAND

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 307

MODULE IDENTIFICATION

Module Name: GETDAT　　Function Name: SEPARABILITY

Purpose: Retrieves Separability Information from Disk or Tape

System/Language: CMS/FORTRAN

Author:　　Date:

Latest Revisor: S. K. Schwingendorf　　Date: 12/15/72

MODULE ABSTRACT

GETDAT reads a record from the disk or tape, processes it according to options specified by the user (such as exclude requests, show requests, and TRANS or UNTRANS) and decides whether it is to be stored in core. This is repeated until the record containing -1.0 as the sum of divergences is found.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL GETDAT (RBUF,VEC,DIVSUM,DIVTAB,SAVQUE,WEIGHT,
DEFWGH,KEPVEC)

Input Arguments:

SAVQUE - INTEGER*2 Array (BSTFET,SPLMAX),SAVQUE(I,J) through SAVQUE(BSTFET,J) contains the channels of the Jth SHOW request having the appropriate number of channels for the current combination request.

WEIGHT - INTEGER*2 Array (DIVSIZ), WEIGHT(I) contains the weight for the symbol pair in SYMCOM(I).

DEFWGH - REAL*4 Default weight for symbol pairs.

IBUF - REAL*4 Array (DIVSIZ), temporary storage of divergences for each class combination.

Output Arguments:

DIVSUM - REAL*4 Array (KEEPLV), DIVSUM(I) contains the sum of divergences for all class pairs for channel combination I.

DIVTAB - INTEGER*2 Array (DIVSIZ, KEEPLV), DIVTAB(I,J) contains the divergence for class pair I and channel combination J with saturating transform and weights applied if requested.

KEPVEC - INTEGER*2 Array (KEEPLV*BSTFET) contains all channel sets for current combinations request.

GETDAT reads the separability information off the scratch device, one set at a time, and applies the saturating transform if requested and the weights (from WEIGHT) to the previously computed divergences. If this set of information was called for by a SHOW request, (as determined by searching through SAVQUE), or if there is room, it is immediately saved. Otherwise, if core is full, it replaces the line with the smallest sum of divergences.

2. Internal Description

GETDAT begins by reading the separability results for one channel combination from the scratch device, and checks this set of channels for an excluded combination (by comparing it with the channels in EXCVEC from the SEPCOM common block). If it contains an excluded channel combination, it is ignored, and the next channel combination is read. If it was requested by the user, the saturating transform is applied, and the line sum is computed. A search through SAVQUE indicates whether this is a SHOW request and must be saved. If core is full (i.e., the number of sets of information already saved equals KEEPVL) then the line with the smallest stored line sum is replaced. The weighted sum is stored in DIVSUM and the weighted interclass divergences are stored in DIVTAB. GETDAT continues to read and process the channel combinations and their divergences stored in DIVTAB. GETDAT continues to read and process the channel combinations and their divergences until a negative sum of divergence is found, indicating all information has been read, and a return is executed. Both GLOCOM and SEPCOM appear in GETDAT.

3. Input Description

The input file for GETDAT will be located on disk (symbolic name SEPARX) whenever there is room, and on tape (SEPTPX) otherwise. This is a scratch file for SEPARABILITY, and stores the initial calculations so they can be used or modified by the user. Each record contains unweighted and untransformed divergences as follows:

1) REAL*4, Sum of divergences.
2) INTEGER*2, Set of channels.
3) REAL*4, Interclass divergences for all class pairs for this set of channels.

4. Output Description

All information is passed back through the output arguments.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 308

MODULE IDENTIFICATION

Module Name: GETINV Function Name: SEPARABILITY

Purpose: Computes Inverse Covariance for Next Set of Channels

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: R.L. Kettig Date: 10/16/74

MODULE ABSTRACT

GETINV computes the next set of channels for the current combination request and then obtains the mean vector, the covariance matrix, and the inverse of the covariance matrix for this set of channels.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GETINV

CALL GETINV (COV, INV, BUF, VEC, NO, AVE, REDCOV, REDAVE)

Input Arguments:

COV - REAL*4, Array (VARSZ3, NOPOOL), COV(1,I) through COV(VARSZ3,I) contains the covariance matrix for pool I.

VEC - INTEGER*2 Array (30) contains the channel numbers for the previous combination.

NO - INTEGER*4, Number of counts to make to find the next set of channels, e.g. NO = 2 would choose every 2nd set of channels. NO always equals 1 in *SEP.

BUF - REAL*4 Array (BSTFET, BSTFET) scratch area.

AVE - REAL*4, Array (NOFET3,NOPOOL), AVE(1,I) through AVE (NOFET3,I) contains the mean vector for pool I.

Output Arguments:

INV - REAL*4 Array (BSTVAR*NOPOOL), INV(BSTVAR*(I-1)) through INV(BSTVAR*I) contains the inverse of the covariance matrix for pool I.

REDAVE - REAL*4, Array (BSTFET*NOPOOL), REDAVE (BSTFET*(I-1)+1) through REDAVE (BSTFET*I) contains the mean vector for the current set of channels for pool I.

REDCOV - REAL*4, Array (BSTVAR*NOPOOL), REDCOV (BSTVAR*(I-1)+1) through REDCOV(BSTVAR*I) contains the covariance matrix for the current set of channels for pool I.

VEC - INTEGER*2, Array (30) contains the current set of channels.

GETINV = 1 if VEC contains next set of channels.
GETINV = 0 if no more channel combinations are left.
GETINV = -1 if any statistics were ill-conditioned.

GETINV finds the next channel combination when given the previous one, and then computes the inverse of the covariance matrix given.

503

2. Internal Description

Using the previous set of channels, the next set is found by adding 1 to the last channel number in the combination. If this exceeds the highest channel number in the data, then the second to last number is increased by 1 and the last channel is set to 1 more than the second to last channel number. This last number would again be checked against the highest channel number available.

The mean vector and covariance matrix for this set of channels, REDAVE and REDCOV, are extracted, one class at a time, from AVE and COV, respectively. The inverse of the covariance matrix is computed by calling MINV.

Both GLOCOM and SEPCOM are used in this routine.

The three exit conditions correspond to the three possible values of GETINV. A return is executed when no more channel combinations exist, when channel statistics for some classes are ill-conditioned, or when the computations are completed.

3. Input Description

Not Applicable

4. Output Description

If the determinant of the inverse covariance matrix is ever zero after the return from MINV, the message:

```
I0020   THE ______ CHANNEL STATISTICS FOR CLASS _____

        ARE ILL-CONDITIONED

"COMBINATIONS _____ " DELETED.                  (GETINV)
```

will appear, where the number of channels in the current combinations request will be found in the first and last space (_____), and the second space is the class number.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 309

MODULE IDENTIFICATION

Module Name: GETSHW Function Name: SEPARABILITY

Purpose: Get SHOW Requests with Correct Number of Channels

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 12/18/72

MODULE ABSTRACT

GETSHW scans the SHOW requests and picks out those with the correct number of channels according to the current combination request.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GETSHW

CALL GETSHW (SAVQUE, BUF, PTR)

Input Arguments

BUF - INTEGER*2, Array may be any size and contain one or many SHOW requests. BUF is formatted as follows:

1) The first integer (n) tells how many numbers are in this SHOW request.
2) The next n integers are the channel numbers in the request.
3) The next integer tells how many numbers are in the next SHOW request, etc.
4) After the last SHOW request a zero will be found.

PTR - INTEGER*4, Indicates how many SHOW requests are already in SAVQUE

Output Arguments

SAVQUE - INTEGER*2, Array (BSTFET, SPLMAX),SAVQUE(1,I) through SAVQUE(BSTFET,I) contains the channels for the I^{th} SHOW request.

PTR - INTEGER*4, Indicates number of SHOW requests currently in SAVQUE.

GETSHW examines the SHOW request(s) in BUF, and any that have 'BSTFET' number of channels are added to SAVQUE, up to a maximum of 'SPLMAX'.

2. Internal Description

The first (or next) item of BUF is obtained. If it is zero, indicating no more SHOW requests, a return is executed. If it does not equal 'BSTFET' then this SHOW request doesn't have the proper number of channels, and is skipped over. This continues until a SHOW request with 'BSTFET' channels is found. Then, if PTR is less than SPLMAX, PTR is incremented by one and the channels of the request are placed in SAVQUE(1,PTR) through SAVQUE(BSTFET,PTR). The search through BUF continues until the zero is found or until SAVQUE is filled. In both cases, a return will be executed. Both GLOCOM and SEPCOM are needed in this routine.

3. Input Description

Not Applicable

4. Output Description

When SAVQUE is full, the following message appears:

I0019 NO MORE SHOW REQUESTS CAN BE ACCEPTED (GETSHW)

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 310

MODULE IDENTIFICATION

Module Name: SEPCOM Function Name: SEPARABILITY

Purpose: Block Data for Separability Common

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 11/30/72

MODULE ABSTRACT

This is the BLOCK DATA routine for the common block SEPCOM used in the SEPARABILITY processor.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 311

MODULE IDENTIFICATION

Module Name: SEPINT   Function Name: SEPARABILITY

Purpose: Reads Control Cards and Sets up Array Bases

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf   Date: 12/18/72

MODULE ABSTRACT

SEPINT calls on SEPRDR to read the control cards, reads and saves the statistics, checks the validity of CHANNELS, CLASSES, and COMBINATION and SHOW requests, calculates array bases and decides whether disk or tape will be used for the calculations.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SEPINT

CALL SEPINT

No parameters are passed to this subroutine. The purpose of SEPINT is to initialize the SEPARABILITY processor by having the control cards read, having the statistics read and saved, checking the validity of requests and calculating array bases.

2. Internal Description

All pertinent information is passed via the GLOCOM and SEPCOM common blocks. SEPINT first calls SEPRDR to read and check the control card input deck. Then the statistics are read, from punched cards if DECK = 1 or from disk if DECK = 0, by calling STAT. Array bases to accomodate these statistics are calculated, and if there is insufficient core, ERPRNT will print the appropriate message and stop the program. SEPINT calls REDSTA to read the statistics into core and CLSCHK to check the validity of the classes. If too few symbols were given, ERPRNT indicates this and requests the SYMBOLS card to be reentered. CTLWRD is called to read this new card and BCDVAL reads the symbols. After placing the symbols in SYMMTX, SEPINT calls FETCHK to check the validity of the channels, and BSTCHK and SHWCHK to check the COMBINATIONS and SHOW requests respectively. The bases of the reduced arrays, caused by any pooling of classes and statistics, are then calculated, and if all results will not fit on disk, SEPSCD is set to SEPTPX to indicate a tape will be needed. This was accomplished by calling TSPACE to find the free space on the T-disk. Finally, the supervisor information is printed, REDSAV is called to save the reduced statistics in core, and SYMSET is called to process the WEIGHT and SHOW requests, and set up the symbol pairs. At this point a return is executed.

3. Input Description

Not Applicable

4. Output Description

SEPINT outputs to the printer some supervisor information such as the number of classes, channels, BEST requests, SHOW requests, and WEIGHT requests. An example of this output can be found in the LARSYS User's Manual. Before executing the return, SEPINT prints, at the typewriter:

I0034 ALL CONTROL AND DATA CARDS HAVE BEEN READ (SEPINT)

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 312

MODULE IDENTIFICATION

Module Name: SEPRDR Function Name: SEPARABILITY

Purpose: Reads Control Cards

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 12/18/72

MODULE ABSTRACT

SEPRDR reads the control cards input by the user and saves all requests made. If an error in the card's format is detected, the card may be reentered at the typewriter.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SEPRDR

CALL SEPRDR (SHWVEC, WGHVEC, SYMCNT)

Output Arguments

SHWVEC - INTEGER*2, Array (400) contains the SHOW requests from the input cards.

WGHVEC - INTEGER*2, Array (1000) contains the WEIGHT requests from the input cards in the format: symbol pair, weight, symbol pair, weight, etc.

SYMCNT - INTEGER*4, Number of symbols entered on SYMBOLS card.

SEPRDR reads the control card input deck and sets the flags and fills the arrays accordingly. If any cards are incorrect or missing, they may be entered at the typewriter when indicated.

2. Internal Description

After initializing all the flags, SEPRDR calls CTLWRD to read a card and set CODE according to which card was read. If a keyword was misspelled, the card may be retyped at the typewriter. Based on the value of CODE, a jump to the appropriate place in SEPRDR is made. The control cards read by SEPRDR, and the actions taken on each are as follows:

COMBINATIONS - IVAL is called to read the numbers. Then the combinations requested are stored in the array BESTVC, and NOBEST is set to the number of combinations in the request.

SYMBOLS - BCDVAL reads the symbols, the symbols are stored in the array SYMMTX, and SYMCNT is set to the number of symbols stored.

CLASSES - The routine POLSCN is called to set up the desired classes.

CHANNEL - The routine CHANEL is called to place the requested channels in the array FETVC3 and set up calibration information.

CARDS - The flag DECK is set to 1 to indicate the statistics are expected from the card deck.

WEIGHT - The symbols which are to be combined into all possible pairs are read, the symbol pairs are formed and all pairs are placed in WGHVEC. Then the weight to be assigned each symbol pair is read and placed in WGHVEC after each symbol pair just entered.

PRINT - This card has four different control parameters which may follow it. They are read by CTLPRM.

BEST - IVAL reads the number of channel combinations to be printed and places this number in the variable PRTLEV.

STATS - The flag STATKY is set to 1 to indicate statistics are to be printed.

SHOW - The number of channels in the SHOW request followed by the channel combination to be printed is saved in the array SHWVEC.

DIV - The flag DIVFG1 is set to 1 and the value to be used is placed in DIVM.

OPTIONS - Six different options may be specified on this card. CTLPRM determines which ones are requested.

MAX - The flag MAXFLG is set to 1 and the value for the upper bound is placed in DMAX.

MIN - The value for the lower bound on the separation measure is placed in DMIN.

EXCLUDE - IVAL reads the number of channels in the request, and the channel subset and these are stored in the array EXCVEC. The flag EXCFLG is set to 1.

TYPE - The flag TYPEFL is set to 1.

UNTRANS - The flag SATFLG is set to 1.

SORT - The flag MINFLG is set to 1.

If no combinations or symbols are entered, this information will be requested, as well as definitions of undefined groups and additional symbols if too few were entered.

Both GLOCOM and SEPCOM variables are needed for SEPRDR. After a 'DATA' or 'END' card has been read, the presence of all information is checked, options are listed, and a return is executed.

3. Input Description

A deck of input cards is read by CTLWRD from this subroutine.

4. Output Description

When an error is encountered on a control card, ERPRNT will print an appropriate message at the typewriter. If -TYPE was entered in the control card deck, the message

I0012 TYPE IN SEPARABILITY CONTROL CARDS

will appear at the typewriter.

If more symbols are entered than can be stored, SEPRDR prints

E169 YOU HAVE ENTERED _____ SYMBOLS. THE MAXIMUM NUMBER THAT CAN BE STORED IS _____.

CORRECT AND TYPE IN ALL SYMBOLS CARDS. (SEPRDR)

After all cards are read, SEPRDR outputs to the printer a summary of the options requested by the user. An example of this may be seen in the User's Manual.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 313

MODULE IDENTIFICATION

Module Name: SYMSET Function Name: SEPARABILITY

Purpose: Relocates preset Weight and SHOW Requests

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 12/18/72

MODULE ABSTRACT

SYMSET sets up the symbol pairs and moves the preset weights and SHOW requests into the dynamically allocated array.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SYMSET

CALL SYMSET(SYMCOM,WEIGHT,SPLCOM,WGHVEC,SHWVEC)

Input arguments:

WGHVEC - INTEGER*2 Array(1000), contains the WEIGHT requests as read off the input control cards: WGHVEC(2I-1) contains Ith input symbol pair WGHVEC(2I) contains the weight of this symbol pair

SHWVEC - INTEGER*2 Array(400), contains SHOW requests as read off the input control cards. Format: number of channels in request, channel numbers, number of channels, channels, etc.

Output arguments;

SYMCOM - INTEGER*2 Array(DIVSIZ), contains all symbols pairs formed from input symbols.

WEIGHT - INTEGER*2 Array(DIVSIZ) contains the weights obtained from the input cards, as found in WGHVEC, except that if WGHVEC(I) contains a symbol pair, then WEIGHT(I) = J where SYMCOM(J) contains the symbol pair

Note that the weights are moved from WGHVEC to WEIGHT because WEIGHT is dynamically allocated and hence won't waste space.

SPLCOM - INTEGER*2 Array(NOSHW), contains the SHOW requests copied directly from SHWVEC. SPLCOM is dynamically allocated so that it is the size required by the SHOW requests and no larger.

The purpose of SYMSET is to set up the symbol combinations and relocate the preset WEIGHT and SHOW requests.

2. Internal Description

All symbol pairs for the symbols in SYMMTX are formed and placed in SYMCOM. The weights are moved from WGHVEC to WEIGHT with the symbol pair replaced by a pointer to the symbol pair in SYMCOM. The SHOW requests are copied directly from SHWVEC into SLPCOM, and a return is executed. Both GLOCOM and SEPCOM appear in SYMSET.

3. Input Description

Not Applicable

4. Output Description

When relocating the weights, if a symbol pair is not found, the message

```
I0018 INVALID CLASS COMBINATION..._____.
      WEIGHT IGNORED     (SYMSET).
```

appears on the printer and typewriter.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 314

MODULE IDENTIFICATION

Module Name: USER Function Name: SEPARABILITY

Purpose: Reads User Options from Typewriter

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: S. K. Schwingendorf Date: 12/18/72

MODULE ABSTRACT

USER reads control cards entered at the typewriter after the initial set of calculations have been made and printed.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

USER

CALL USER(IBUF,DIVSUM,DIVTAB,WEIGHT,SYMCOM,KEPVEC,SAVQUE)

Input arguments:

IBUF - INTEGER*2 Array(DIVSIZ) I/0 buffer for DIVTAB - address of position in dynamically allocated array is passed in call statement.

DIVSUM - REAL*4 Array(KEEPLV) DIVSUM(I) contains the sum of all divergence for the Ith channel combination.

DIVTAB - INTEGER*2 Array(DIVSIZ,KEEPLV) DIVTAB(1,I) DIVTAB(DIVSIZ,I) contains the interclass divergences for the Ith channel combination.

WEIGHT - INTEGER*2 Array(DIVSIZ) WEIGHT(I) contains the weight for the symbol pair in SYMCOM(I)

SYMCOM - INTEGER*2 Array(DIVSIZ) contains all symbol pairs

KEPVEC - INTEGER*2 Array(BSTFET*KEEPLV) contains all sets of channels.

SAVQUE - INTEGER*2 Array(BSTFET,SPLMAX) contains up to 'SPLMAX' SHOW requests with the correct number of channels

Output Arguments:

DIVSUM, DIVTAB, KEPVEC - Same as above, except they will be reordered if SHOW requests were entered

WEIGHT - Same as above, with several weights changed if a WEIGHT request was typed in

SAVQUE - Same as above, with additional SHOW requests inserted if a SHOW request was typed in.

USER allows the user to type in different sets of control cards to change options and get different printed output.

2. Internal Description

Control cards entered by the user are read by CTLWRD, and a jump is executed to the appropriate place. The control cards read, and actions taken on each, are as follows:

PRINT: There are three control parameters which may follow this.

BEST(N): This will print results for the best N combinations by placing in PLEVEL, the variable indicating the number of levels to be printed, the number N. The flag PRTFG1 is set to 1 to indicate the print level has been changed.

SHOW(F1,F2,...): The combination F1,F2,...is read by IVAL and checked to see that it has the correct number of channels for the current combination request, and to be sure the channels are valid. The show request and its divergences are then moved to be with the other show requests.

DIV(VALUE): The variable DIVM is changed to VALUE and the flag DIVFG1 is set to 1 to indicate this option is in effect.

OPTIONS: Ten different control parameters may be used with this keyword.

MAX(VALUE) : VALUE is read by IVAL and assigned to the variable DMAX and the flag MAXFLG is set to 1 to show DMAX has been changed.

MIN(VALUE): VALUE is read by IVAL and is assigned to the variable DMIN.

EXCLUDE(F1,...): This option resets the vector EXCVEC to the number of channels in the request followed by the channels F1,... to be excluded. The flag EXCFLG is set to 1 to indicate an exclude request is in effect.

UNTRANS: The flag SATFLG is set to 1 to indicate that the transformed divergence should not be calculated.

TRANS: The flag SATFLG is set to 0 meaning the transformed divergences are to be used.

SORT: The flag MINFLG is set to 1 so that the printed results will be sorted on DIJ(MIN)

NOSORT: The flag MINFLG is set to 0 to indicate the results will be sorted on D(AVE).

RESET: All parameters and flags are set to their card input or default values.

HELP: A list of possible requests is printed at the typewriter.

TABLE: A table of the symbols and their corresponding class names is printed at the typewriter.

Both GLOCOM and SEPCOM are needed in USER.

3. Input Description

USER expects to read cards input at the typewriter, with the keyword beginning in column 1, and any control parameters beginning at least 1 space after the key word. The control cards are read by CTLWRD.

4. Output Description

Each time a set of calculations have been performed and printed, the message

I0024 ENTER OPTIONS (USER)

will appear at the typewriter to indicate another set of options may be entered. When a keyword or control parameter is misspelled, a message from ERPRNT indicating this, appears at the typewriter and the corrected card may then be entered.

If an improper SHOW request is entered, one of the following self-explanatory messages may appear

I0025 SHOW REQUEST CANNOT BE LOCATED

I0026 SHOW REQUEST PREVIOUSLY SAVED

If the OPTION HELP is typed by the user a list of possible inputs will be typed for the user to see.

When OPTION SORT or OPTION NOSORT is requested and already exists, one of the following messages will appear

I0027 SORT ON DIJ(MIN) ALREADY IN EFFECT

I0028 SORT HAS NOT BEEN IN EFFECT

Similarly, if OPTION TRANS or OPTION UNTRANS is typed and already was in effect one of these messages will appear

I0029 SATURATING TRANSFORM ALREADY IN EFFECT

I0030 SATURATING TRANSFORM HAS NOT BEEN IN EFFECT

If OPTION TABLE is entered, a table of the symbols and corresponding class names is output to the typewriter.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 320

MODULE IDENTIFICATION

Module Name: STASUP Function Name: STATISTICS

Purpose: Supervisor for the statistics function.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Earl Rodd Date: 11/22/72

MODULE ABSTRACT

Supervisor for the statistics function. STASUP calls other subroutines which actually do the work.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

STASUP

Called by LARSMN to supervise the statistics function.

2. Internal Description

Many array bases for dynamically dimensioned arrays are local variables in STASUP. They are arguments to STAINT which computes them and then used as indexes into ARRAY in arguments to LEARN.

3. Input Description

Not Applicable

4. Output Description

Information messages 198 and 199.
Error message 250 for insufficient main storage for dynamically dimensioned arrays.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 321

MODULE IDENTIFICATION

Module Name: CLSHIS Function Name: STATISTICS

Purpose: Prints field and class histograms

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Earl Rodd Date: 11/27/72

MODULE ABSTRACT

CLSHIS prints the histograms for all channels for one class.

Entry FLDHIS prints histograms for all channels for one field.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLSHIS

CALL CLSHIS(TALLY,HISBUF,TTL,BINSIZ,LLMT,PTS)

Input Arguments:

TALLY - I*4, Histogram array. TALLY(I,J) = number of points in the Ith histogram bin for the Jth channel being histogrammed.

HISBUF - I*4, Work area provided by the caller. The work area must have as many words as there are histogram bins.

TTL - I*4, Class name (field name for entry FLDHIS). This is 8 characters of EBCDIC.

BINSIZ - R*4, BINSIZ(I) = the width of one histogram bin for the Ith channel being histogrammed.

LLMT - R*4, LLMT(I) = value of the left end of the abscissa for the Ith channel being histogrammed.

PTS - I*4, Number of points in the class (field for entry FLDHIS) being histogrammed.

Output Arguments:

TALLY - I*4, Returned with all zeroes. The loop which zeroes TALLY assumes that there are 100 histogram bins. (i.e. that TALLY is dimensioned (100,number of channels histogrammed)).

FLDHIS

CALL FLDHIS(TALLY,HISBUF,TTL,BINSIZ,LLMT,PTS)

Input Arguments:

Same as for CLSHIS except where noted.

Output Arguments:

Same as for CLSHIS except where noted.

This subroutine is called to print the histograms for all channels for one class or field. Whether a class or field is histogrammed is determined only by the contents of the input arguments.

The only difference in CLSHIS and entry FLDHIS is in the format and content of the headers for each histogram.

2. Internal Description

The program uses IOR to compute the correct hexadecimal representation for the histogram symbol to print for the excess points beyond an even number of '*' symbols. (The form of the histogram is to have each '*' represent a certain number of points and then above the last '*' print a number (or letter if the number is greater than nine) for the remaining points.)

3. Input Description

Not Applicable

4. Output Description

Field and class histograms including headings.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 322

MODULE IDENTIFICATION

Module Name: CLSSPC Function Name: STATISTICS

Purpose: Prints spectral plots

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Earl Rodd Date: 11/27/72

MODULE ABSTRACT

CLSSPC prints the spectral plot for a class.

Entry FLDSPC prints the spectral plots for a field.

Entry MULSPC prints one coincident spectral plot.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CLSSPC

CALL CLSSPC(MEAN,COVAR,PTRVEC,PLOT,PTS)

Input arguments:

MEAN - R*4 MEAN(I) = the mean of data values for the I'th channel used for this class.

COVAR - R*4 The covariance matrix for this class. The arrangement is the same as the vector FLDVAR in LEARN.

PTRVEC - I*2 In this entry, PTRVEC is a dummy with elements 1 and 61 = 1 and all others = 0.

PLOT - I*2 Work area of 95 halfwords provided by the caller. The work area is used for the print buffer to create print lines.

PTS - I*4 Number of points in the class.

CLSSPC is called to print the spectral plot for one class.

FLDSPC

CALL FLDSPC(DMEAN,DCOVAR,PTRVEC,PLOT,MEAN,COVAR,FLDPTS)

Input arguments:

DMEAN - R*8 MEAN(I) = the mean of data values for the I'th channel used for this field.

DCOVAR - R*8 The covariance matrix for the field. This is the same as the vector FLDVAR in LEARN.

PTRVEC - I*2 In this entry, PTRVEC is a dummy with elements 1 and 61 = 1 and all others = 0.

PLOT - Same as for CLSSPC.

FLDPTS - I*4 Number of points in this field.

Output arguments:

MEAN - R*4 Single precision version of DMEAN.

COVAR - R*4 Single precision version of DCOVAR.

FLDSPC is called to print the spectral plot for a field.

MULSPC

CALL MULSPC(MEAN,COVAR,JDVEC,PTRVEC,PLOT)

Input arguments:

MEAN - R*4 This is the array AVAR from LEARN.

COVAR - R*4 This is the array COVAR from LEARN.

JDVEC - I*4 This is the vector of class names (CLSDES in LEARN).

PTRVEC - I*2 Vector indicating which classes are to be plotted on this coincident spectral plot. PTRVEC(61) = the number classes in this plot. PTRVEC(I) = class number of the I'th class included in the plot.

PLOT - Same as CLSSPC.

MULSPC is called to print one coincident spectral plot.

2. Internal Description

The standard deviation is always computed to plot the mean plus or minus one standard deviation.

3. Input Description

Not Applicable

4. Output Description

Spectral plots for fields and classes are printed and the coincident spectral plots are printed.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 323

MODULE IDENTIFICATION

Module Name: FLDCOV Function Name: STATISTICS

Purpose: Prints statistics

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Earl Rodd Date: 11/27/72

MODULE ABSTRACT

FLDCOV converts sums to means, sums of products to covariances, computes the correlation matrix and prints them.

FLDCOV processes one field.

CLSCOV processes a class.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

FLDCOV

CALL FLDCOV(COR,DEV,MEAN,VAR,PTS,GO)

Input arguments:

COR - R*4 Work array for computing correlation matrix. Has the same arrangement as the covariance matrix except contains correlation values.

MEAN - R*8 This is the array FLDMEN in LEARN. Upon entry it contains sums of data values.

VAR - R*8 This is the array FLDVAR in LEARN. On entry it contains sums of products of data values.

PTS - I*4 The number of data points in the field.

GO - I*2 = 0 if no printout of field statistics was requested, = 1 if the printout was requested.

Output arguments:

DEV - R*4 The standard deviation vector for the field. DEV(I) = standard deviation of data values for the Ith channel used.

MEAN - Now contains means.

VAR - Now contains the covariance matrix.

CLSCOV

CALL CLSCOV(COR,DEV,MEAN,VAR,PTS,GO)

Input arguments:

COR - Same as in FLDCOV.

MEAN - Array CLSVAR from LEARN.

VAR - Array CLSVAR from LEARN.

PTS - Number of points in the class.

GO - = 0 if printout of class statistics was not requested, = 1 if printout was requested.

Output arguments:

DEV - Same as in FLDCOV except the standard deviation vector is for the class.

MEAN - Now contains means.

VAR - Now contains the covariance matrix.

FLDCOV is called for each field. The statistics are computed for the field and printed if requested. CLSCOV is called for each class. The only difference is that for CLSCOV the input arrays contain class statistics rather than field statistics and the header of the printout is different.

2. Internal Description

See arguments

3. Input Description

Not Applicable

4. Output Description

Information messages 207 and 208.

If requested the statistics printout is printed. For field statistics, the main header was printed in LEARN. For class statistics, the number of points in the class is added to the header. FLDCOV (and CLSCOV) print the mean and standard deviation vectors and call WRTMTX to print the correlation matrix.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 324

MODULE IDENTIFICATION

Module Name: LEARN Function Name: STATISTICS

Purpose: Controls statistics computation

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Earl Rodd Date: 11/27/72

MODULE ABSTRACT

LEARN computes statistics and generates histograms and spectral plots for all training fields. Some computation is done by LEARN and some via calls to subroutines.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LEARN

```
CALL LEARN(SPEC,COVAR,AVAR,CLSDES,FLDMEN,FLDVAR,CLSMEN,
           CLSVAR,HFTALY,HCTALY,RDATA,DATBAS,WORK)
```

Input arguments:

SPEC - I*2 Array containing information on coincident spectral plots. See the ARRAY SPCVEC in STAINT.

DATBAS - I*4 The index of the first element of ARRAY available for use for dynamically dimensioned arrays.

The following arrays are computed and used within LEARN and the subroutines it calls. All are dynamically dimensioned and physically located in ARRAY. STAINT computed the bases of all of them.

RDATA - R*4 Space for use as a buffer for reading in the data tape lines. This is ARRAY(DATBAS) in the call from STASUP.

WORK - R*4 This is a work array used by LEARN and passed to other subroutines to use as a work array. It is dimensioned 1000 and was ARRAY(1) in the call from SEPSUP. Thus it is the first 500 doublewords of ARRAY.

COVAR - R*4 Array of covariance matrices for all classes. COVAR(I,J) contains the Ith element of the covariance matrix for class J. The lower triangular portions of the symmetric covariance matrices are stored. The sequence of elements in each covariance matrix is $c_{11},c_{21},c_{22},c_{31},c_{32},c_{33},c_{41}\ldots$

AVAR - R*4 The mean vectors for all classes. AVAR(I,J) = the mean of data points for the Ith channel used for class J.

CLSDES - I*4 Array of class names. CLSDES(1,J) = first four characters (in EBCDIC) of name of class J. CLSDES(2,J) = second four characters.

FLDMEN - R*8 Vector of means for a field. FLDMEN(I) = mean of data points for the Ith channel used. This vector is computed by first summing all data points for a field. Subroutine FLDCOV converts the sum to a mean.

FLDVAR - R*8 This is the covariance array for one field. The lower triangular part of the array is stored in the one-dimensional vector FLDVAR in the arrangement described for COVAR. FLDVAR is initially computed as the sums of products (thus the diagonals of the covariance matrix are initially sums of squares). Subroutine FLDVAR converts FLDVAR to covariances.

CLSMEN - R*8 Mean vector for a class. CLSMEN(I) = mean of data points for the Ith channel used. Like FLDMEN, CLSMEN initially contains the sum of values and is converted to a mean in CLSCOV.

CLSVAR - R*8 Covariance matrix for a class. This has the same arrangement as FLDVAR. It is converted to a covariance from sums of products by CLSCOV.

HFTALY - I*4 Histogram array for field histograms. If field histograms are requested, HFTALY is used only for field histograms. If only class histograms are requested, HFTALY is used to sum the histogram for the entire class rather than zeroing after each field. HFTALY(I,J) = number of points in histogram bin I from the Jth channel being histogrammed. There are 100 histogram bins. The left end of the first histogram bin is equal to 2 times the value of A used in calibrating the data ($d_c = A*d_r+B$, d_c = calibrated data, d_r = raw data). The width of each bin is equal to the value of B. Data points outlining the range of the 100 bins are placed in either bin 1 or bin 100.

HCTALY - I*4,Same as HFTALY except used for classes when field histograms have been requested and HFTALY is cleared before each field is processed.

2. Internal Description

See flowchart

3. Input Description

The file STATS DATA is read to retrieve the training fields. All training fields are read via calls to LAREAD. When 'END' is encountered, signifying the end of the training fields, processing terminates. Data tapes are read via GETRUN and GADLIN.

4. Output Description

Information messages 23, 202, 203.

Error message 195 is written via ERPRNT if there is insufficient space left in ARRAY for the buffers for reading data tapes.

5. Supplemental Information

If the STOP command is issued, processing continues until the field currently being processed is completed.

6. Flowchart

(Page numbers refer to the pages of the flowchart.)

NOTES: The following notations for user options are used in this flowchart:

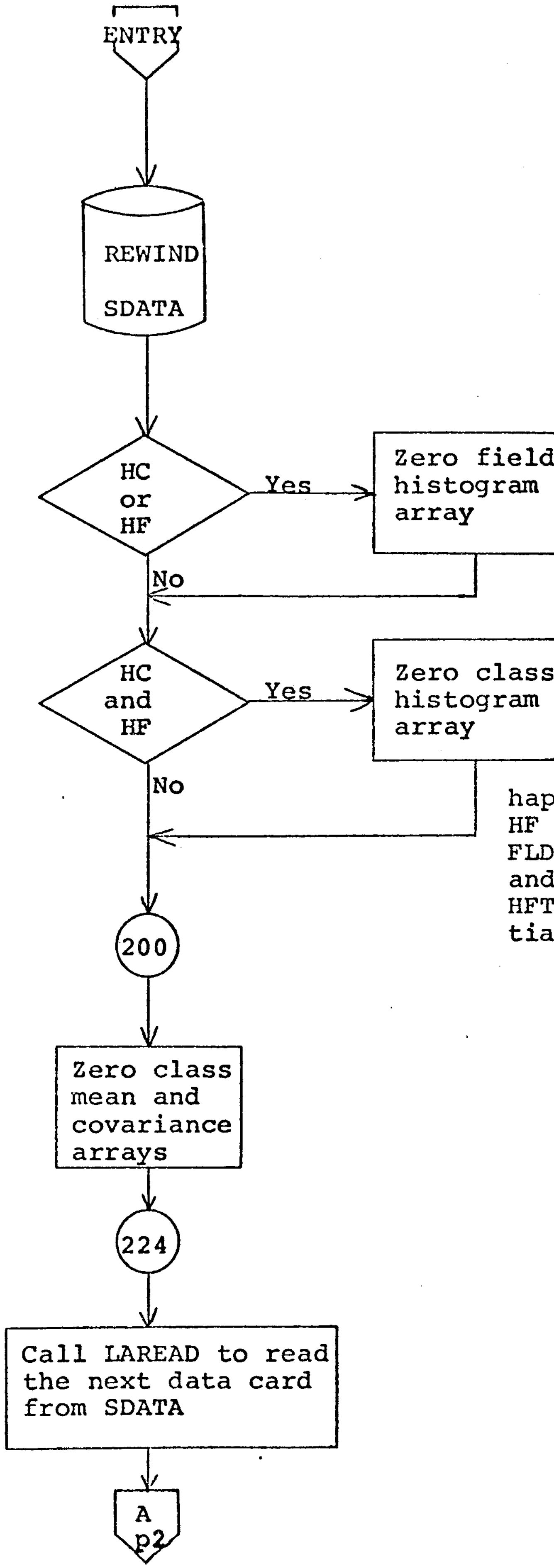
1. HF - print field histograms
   HC - print class histograms

   SF - print spectral plots for fields
   SC - print spectral plots for classes

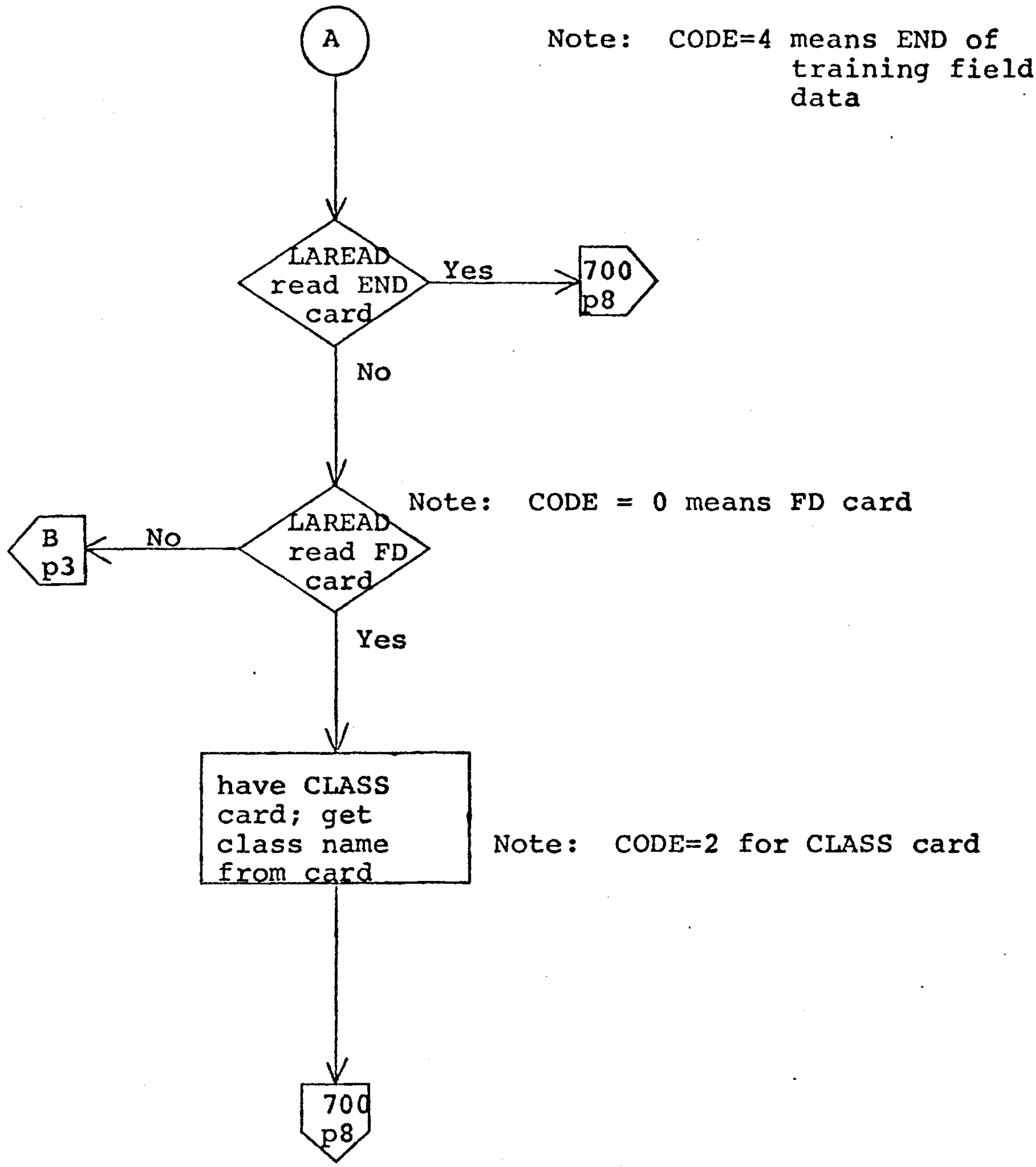
   CF - print means and correlation matrices for fields
   CC - print means and correlation matrices for classes

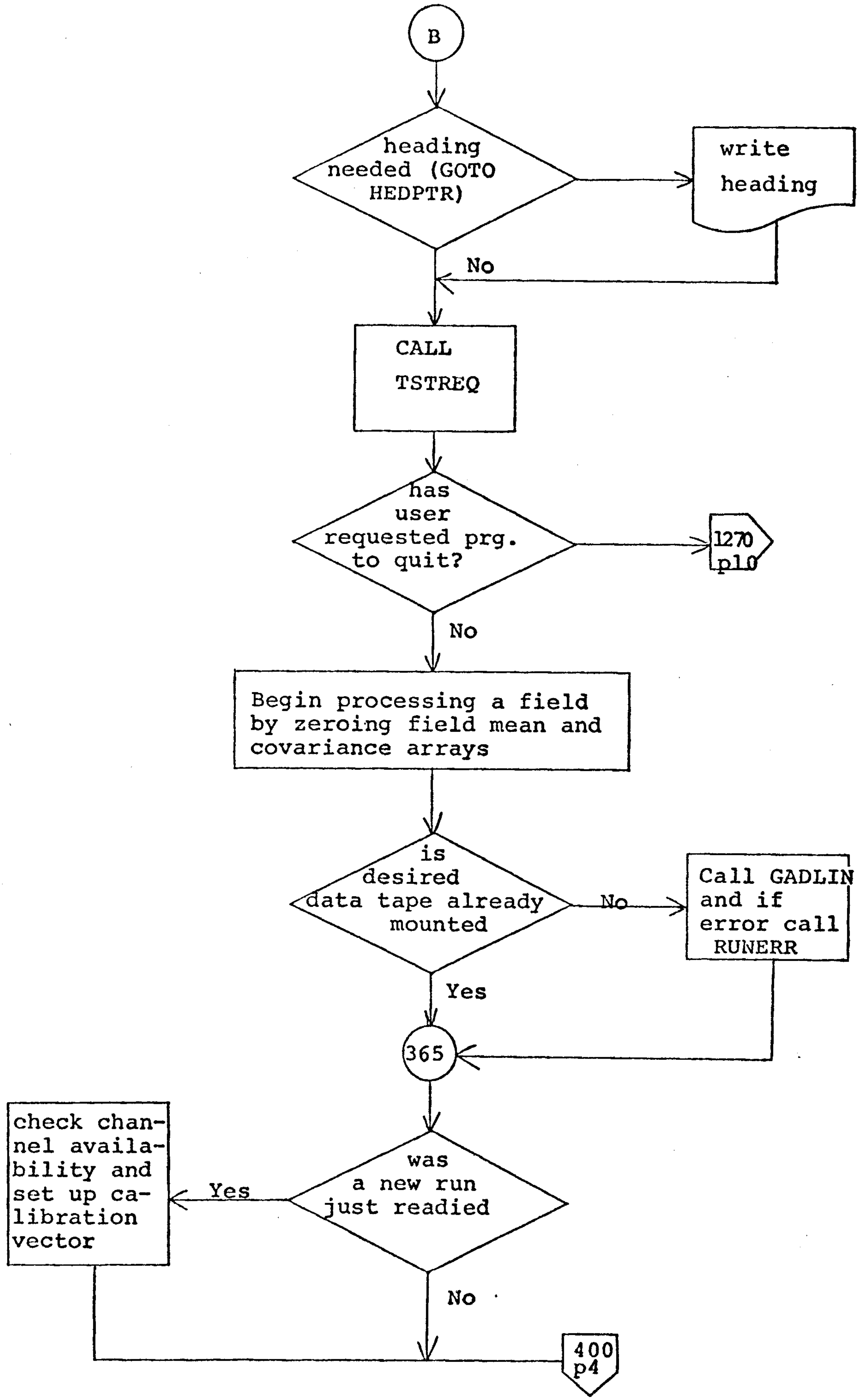
   CG - coincident spectral plots
   PV - punch statistics deck

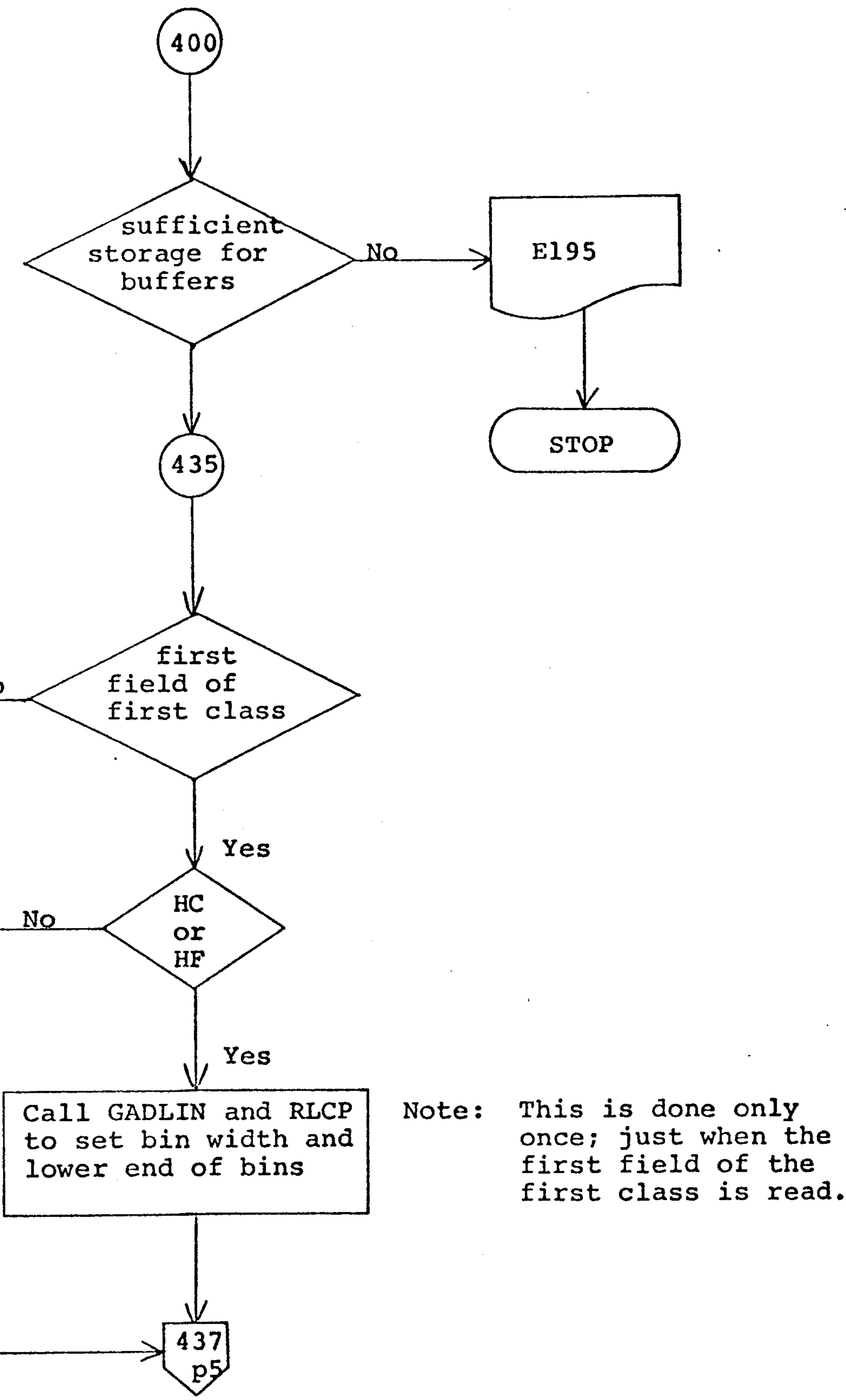
2. CAPITOL letters refer to FORTRAN variables or options above.

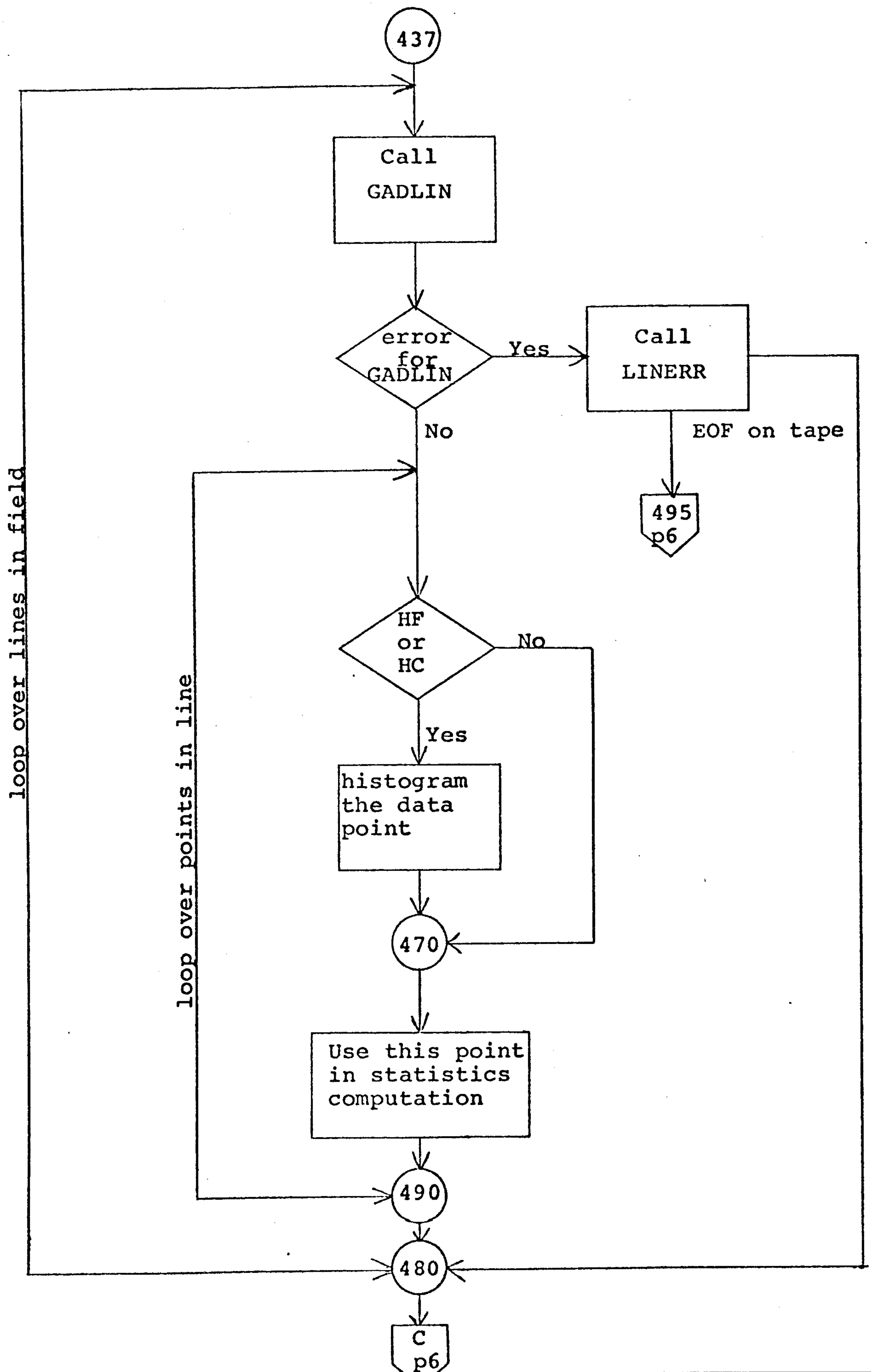


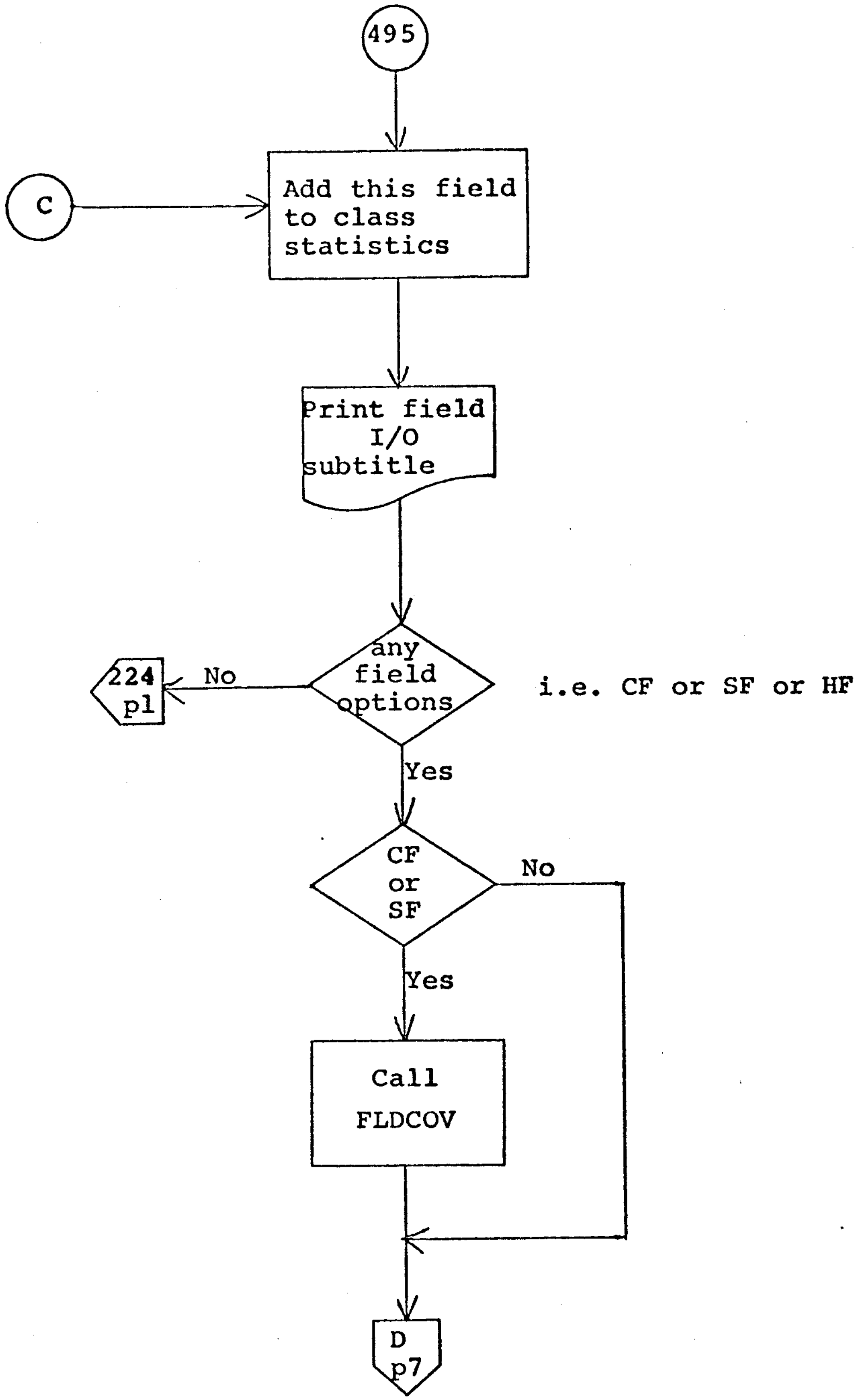
NOTE: IF HC only, the HFTALY will be used for the class histograms. This happens because since HF is not used, FLDHIS is not called and FLDHIS is where HFTALY is reinitialized.

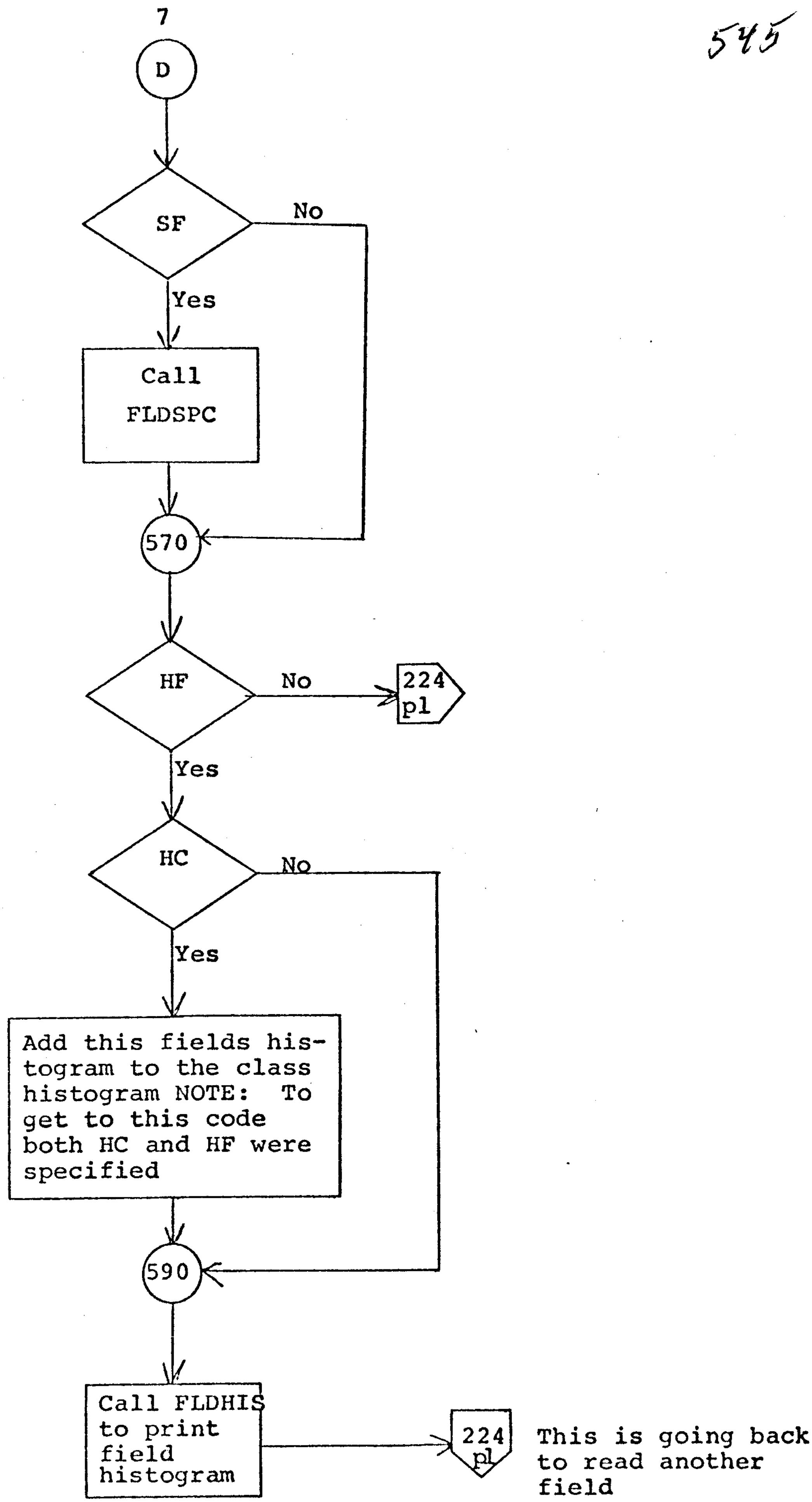

A
Note: CODE=4 means END of training field data
LAREAD read END card
Yes
700 p8
No
Note: CODE = 0 means FD card
B p3
No
LAREAD read FD card
Yes
have CLASS card; get class name from card
Note: CODE=2 for CLASS card
700 p8


B
heading needed (GOTO HEDPTR)
write heading
No
CALL TSTREQ
has user requested prg. to quit?
1270 p10
No
Begin processing a field by zeroing field mean and covariance arrays
is desired data tape already mounted
No
Call GADLIN and if error call RUNERR
Yes
365
was a new run just readied
Yes
check channel availability and set up calibration vector
No
400 p4

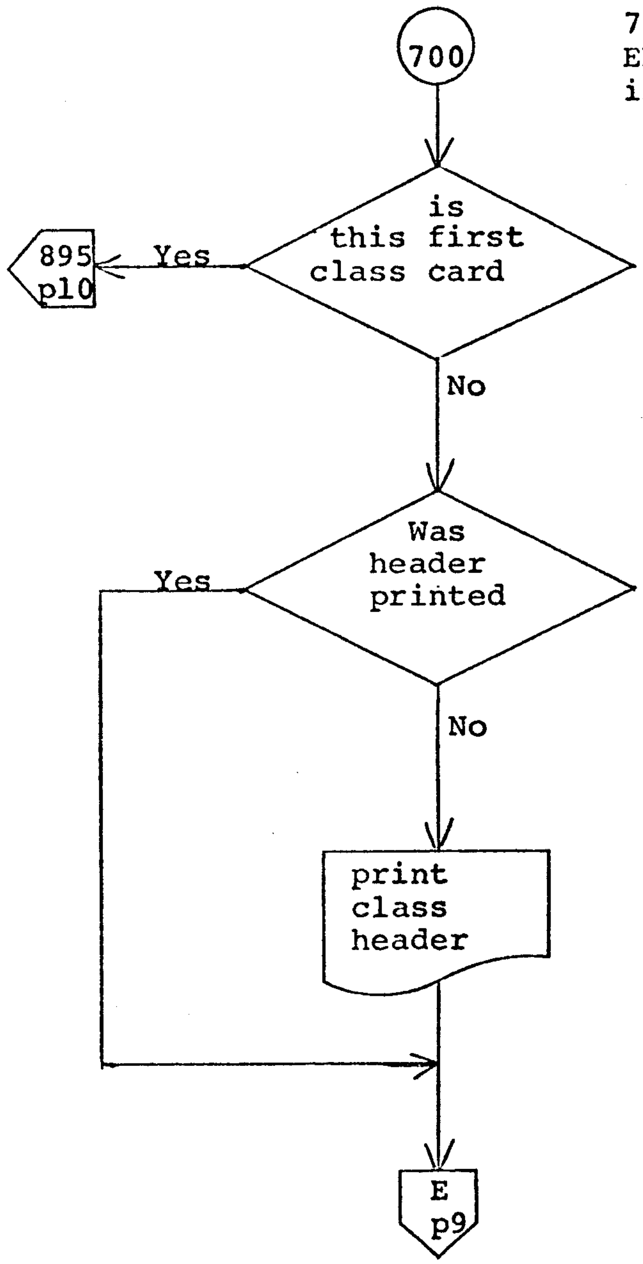

400
sufficient storage for buffers
No
E195
STOP
435
first field of first class
No
Yes
HC or HF
No
Yes
Call GADLIN and RLCP to set bin width and lower end of bins
Note: This is done only once; just when the first field of the first class is read.
437 p5


437
Call
GADLIN
error
for
GADLIN
Yes
Call
LINERR
EOF on tape
495
p6
No
HF
or
HC
No
Yes
histogram
the data
point
470
Use this point
in statistics
computation
490
480
C
p6
loop over lines in field
loop over points in line


495
C
Add this field to class statistics
Print field I/O subtitle
any field options
i.e. CF or SF or HF
No
224 p1
Yes
CF or SF
No
Yes
Call FLDCOV
D p7


7
D
SF
No
Yes
Call
FLDSPC
570
HF
No
224
pl
Yes
HC
No
Yes
Add this fields his-
togram to the class
histogram NOTE: To
get to this code
both HC and HF were
specified
590
Call FLDHIS
to print
field
histogram
224
pl
This is going back
to read another
field

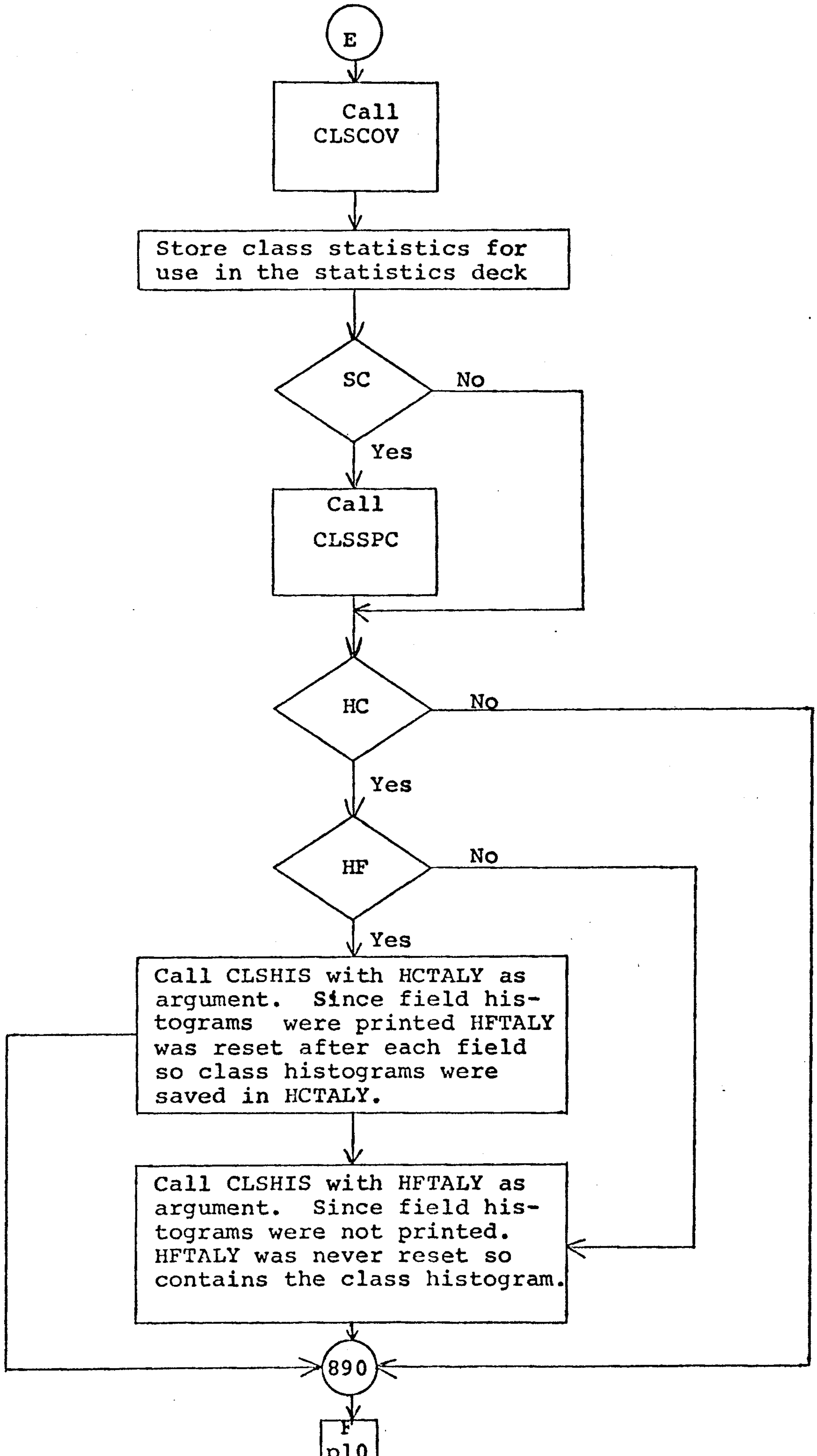
546

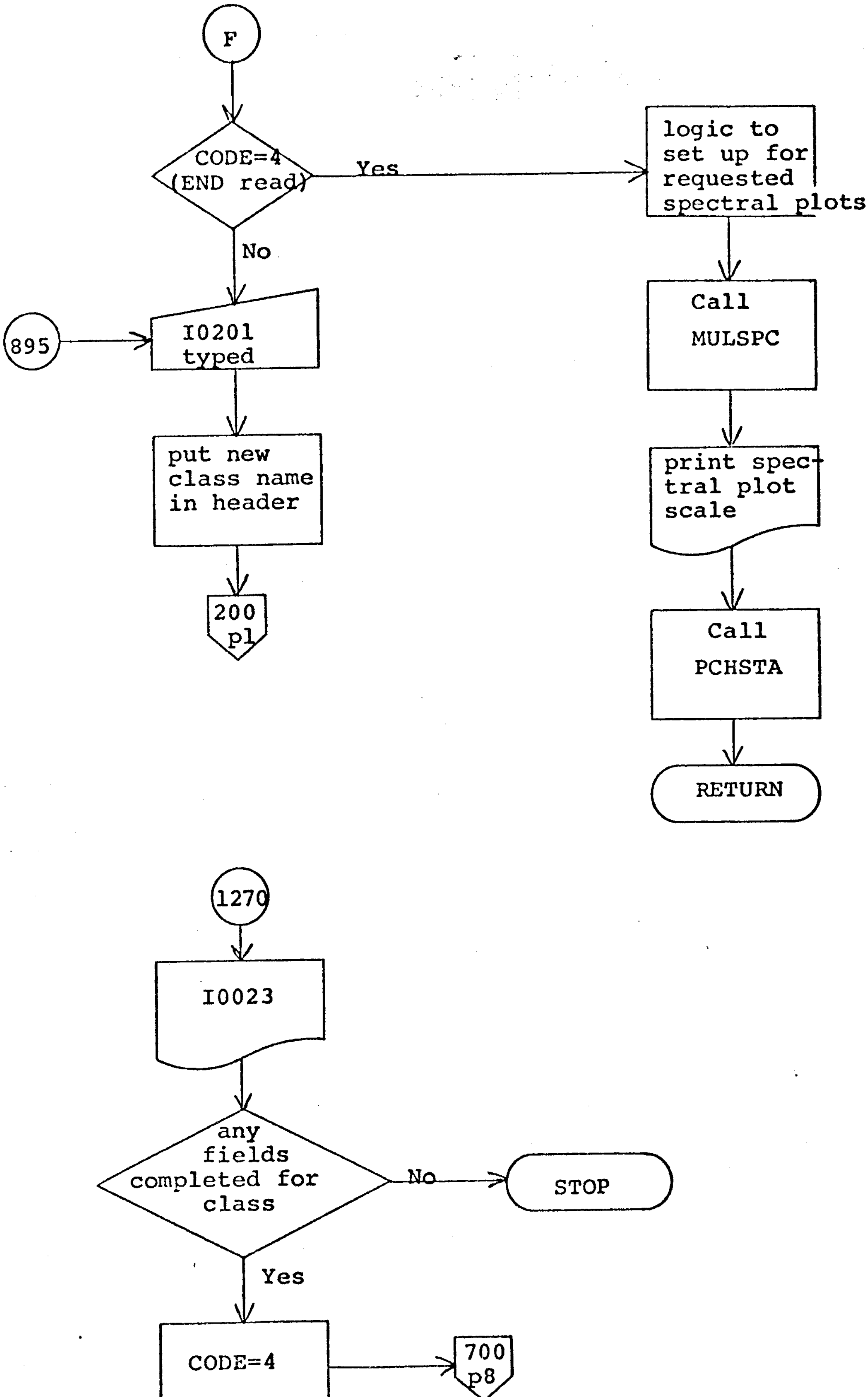


700 is executed when CLAS or END is encountered in training field data.

Note: First class card means very first card of data.

Note: If HF and CF and SF are all no, then no header was printed.


E
Call
CLSCOV
Store class statistics for
use in the statistics deck
SC
No
Yes
Call
CLSSPC
HC
No
Yes
HF
No
Yes
Call CLSHIS with HCTALY as
argument. Since field his-
tograms were printed HFTALY
was reset after each field
so class histograms were
saved in HCTALY.
Call CLSHIS with HFTALY as
argument. Since field his-
tograms were not printed.
HFTALY was never reset so
contains the class histogram.
890
F
p10


F
CODE=4
(END read)
Yes
No
logic to
set up for
requested
spectral plots
Call
MULSPC
print spec-
tral plot
scale
Call
PCHSTA
RETURN
895
I0201
typed
put new
class name
in header
200
p1
1270
I0023
any
fields
completed for
class
No
STOP
Yes
CODE=4
700
p8

LARS Program Abstract 325

MODULE IDENTIFICATION

Module Name: PCHSTA Function Name: STATISTICS

Purpose: Writes statistics to statistics file and punches statistics deck.

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Earl Rodd Date: 11/26/72

MODULE ABSTRACT

PCHSTA writes the count record, channel information, record of class points, statistics and EOS record to the statistics file. If a statistics deck was requested, it is punched.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

PCHSTA

CALL PCHSTA(AVAR,COVAR)

Input arguments:

AVAR - R*4; array of mean vectors for all classes. AVAR(I,J) = the mean value of data points for the Ith channel for class J. The Ith channel means the Ith channel of the channels requested, not channel number I.

COVAR - R*4; array of covariance matrices for all classes. COVAR(I,J) = Ith element of the covariance matrix for class J. See LEARN for the arrangement of the covariance matrices.

PCHSTA is called to write the remainder of the statistics file (after the training fields) to disk. If a punched statistics deck was requested, it is punched.

2. Internal Description

Not Applicable

3. Input Description

The file STATS DATA (statistics file) is read if a punched statistics deck is requested.

4. Output Description

Information message 208.

The statistics file (STATS DATA, DSRN DATA) is completed. The training field portion of the file has been read by LEARN including the record containing 'END' denoting the end of the training fields. PCHSTA first backspaces over the record containing "END". The remainder of the statistics file including the EOS record is written.

The statistics deck is punched if requested. This is accomplished by rewinding SDATA and reading each record and punching it to cards.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 326 552

MODULE IDENTIFICATION

Module Name: STACOM Function Name: STATISTICS

Purpose: Statistics common BLOCK DATA

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Earl Rodd Date: 11/22/72

MODULE ABSTRACT

This is the BLOCK DATA subroutine for the statistics common block, STACOM.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 327

MODULE IDENTIFICATION

Module Name: STAINT Function Name: STATISTICS

Purpose: Initiator for statistics; processes data deck

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Earl Rodd Date: 09/22/72

MODULE ABSTRACT

STAINT is the initiator for the statistics function. It reads the data deck, computes array bases and prints a list of control card information.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

STAINT

```
CALL STAINT (SPCVEC,SPEC1,COVAR1,AVAR1,CLSID1,FLMEN1,
             FLVAR1,CLMEN1,CLVAR1,HFTAL1,HCTAL1,DATBAS)
```

Input Arguments:

SPCVEC - Array of coincident spectral plot requests. SPCVEC(I,J) = class number of Ith class requested in Jth plot.

Output Arguments:

SPCVEC - I*2 If the user requested no specific coincident spectral plots (that is he asks for the default), SPCVEC(1,1) = 0.

SPEC1 - I*4 Base address in ARRAY of SPCVEC array (always = 501)

COVAR1 - I*4 Base address in ARRAY of COVAR array. For the description of this and other arrays whose bases are listed below, see module document of subroutine LEARN.

AVAR1 - I*4 Base address in ARRAY of AVAR

CLSID1 - I*4 Base address in ARRAY of CLSDES

FLMEN1 - I*4 Base address in ARRAY of FLDMEN

FLVAR1 - I*4 Base address in ARRAY of FLDVAR

CLMEN1 - I*4 Base address in ARRAY of CLSMEN

CLVAR1 - I*4 Base address in ARRAY of CLSVAR

HFTAL1 - I*4 Base address in ARRAY of HFTALY

HCTAL1 - I*4 Base address in ARRAY of HCTALY

DATBAS - Next element of ARRAY which is free. This is the element after the last word of HCTALY.

STAINT computes the above array bases. Note that ARRAY is REAL*8 and that the bases are the element number in ARRAY. STAINT also reads the training field data deck and checks it for validity.

2. Internal Description

Not Applicable

3. Input Description

STAINT reads the training field data deck via calls to LAREAD. This deck is checked for errors. A DATA card found is interpreted as an END card.

4. Output Description

The first card of the statistics file is written to unit SDATA (file STATS DATA). Then the entire training field deck is written onto the statistics file. Any CLASS cards which were read without a class name on them are written with the class name 'NONE'. (The user is informed of such via error message). At the end of the training fields a record is written with the characters 'END' so that LEARN can correctly identify the end of the training fields.

If the last class read had no fields (or there were no fields at all), SDATA is rewound and the characters EOS are written.

A list of control information is printed.

Information message 200 is typed.

Error messages 123, 126, 127, 128, 194, 196, 197.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

556

LARS Program Abstract 328

MODULE IDENTIFICATION

Module Name: STARDR Function Name: STATISTICS

Purpose: Card reader for statistics function

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Earl Rodd Date: 11/22/72

MODULE ABSTRACT

STARDR interprets and checks all control cards for the statistics function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

STARDR

CALL STARDR(SPCVEC)

Output Arguments:

SPCVEC - I*2 Array containing information concerning coincident spectral plots requested. SPCVEC(I,J) = class number of the Ith class requested in the Jth spectral plot request.

STARDR uses standard card reader techniques. Communication back to the caller is via variables in common and SPCVEC. Below is a list of actions taken when the control cards are encountered.

PRINT CORRE(F)	- CFDKEY = 1
PRINT CORRE(C)	- CCLKEY = 1
PRINT HIST(F)	- HFDKEY = 1
PRINT HIST(C)	- HCLKEY = 1
PRINT SPECTRL(F)	- SFDKEY = 1
PRINT SPECTRL(C)	- SCLKEY = 1
PRINT COSPEC	- SPCVEC computed
CHANNEL	- CSEL, CSET, NCR, and FETVEC are computed.
SCALE SPCLOW(n)	- SPCBAS = n
SCALE SPCINT(m)	- SPCINT = m
OPTIONS HIST	- NOHIST = number of channels to be histogrammed and HISVEC is computed.
PUNCH	- PCHKEY = 1. If the CHARACTERS option is requested, BINFLG = 0.

STARDR reads all function control cards and the DATA card.

2. Internal Description

Standard control card reader techniques.

3. Input Description

Control cards.

4. Output Description

A list of options selected is printed. Information message 197. Error messages 117 - 122, 124, 125 and 199 and 251.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

MODULE IDENTIFICATION

Module Name: MERSUP Function Name: MERGESTATISTICS

Purpose: Supervisor for MERGESTATISTICS function

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Joan Buis Date: 3/31/80

MODULE ABSTRACT

MERSUP calls MERRDR to decode the control cards. Then it calls MERSTT to combine specified classes from specified stat decks into one stat deck and writes this deck to disk. Finally MERSUP calls POLMER to pool any classes of the newly created statdeck which is on disk if pooling is requested, punches stats if requested, and creates plots if requested.

1. Module Usage

MERSUP

CALL MERSUP

MERSUP is called by LARSMN to execute the MERGESTATISTICS function.

2. Internal Description

MERSUP decodes the control cards through a call to MERRDR, creates a new stat deck through a call to MERSTT, and pools classes if requested through a call to POLMER.

3. Input Description

The card deck described in the MERGESTATISTICS control card listing is required.

4. Output Description

Standard supervisor information messages.

5. Supplemental Information

See system manual for supervisor requirements.

6. Flowchart

Not applicable.

561

MODULE IDENTIFICATION

Module Name: MERCOM Function Name: MERGESTATISTICS

Purpose: Common for MERGESTATISTICS

System/Language: CMS/FORTRAN

Author: William W. Freestone Date: 7/20/77

Latest Revisor: Joan S. Buis Date: 3/31/80

MODULE ABSTRACT

This is the BLOCK DATA subroutine for the MERCOM common area. It is used by the following routines:

MERSUP
MERRDR
MERSTT
STATRD
MERINT
POLMER

561

MODULE IDENTIFICATION

Module Name: MERINT Function Name: MERGESTATISTICS

Purpose: Reads statistics files

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: Joan Buis Date: 4/24/80

MODULE ABSTRACT

MERINT reads statistics files and sets up DCKPTR, DCKSTK, POLPTR and POLSTK.

1. Module Usage

MERINT

CALL MERINT (CLSPTR,CLSSTK, STK1, STK2, STK3,*)

Input Arguments:

CLSPTR - I*2 array dimensioned (3,60) where the I-th column refers to the I-th statistics file. The first row contains the number of classes, the second row contains a pointer to the entry in CLSSTK where the class numbers for file I begin. The third row contains a code indicating the parameter DELETE, ENTIRE or INCLUDE.

CLSSTK - I*2 array dimensioned (120) which contains class numbers.

STK1 - I*2 array dimensioned (2,60) which contains the number of decks in a pool and a pointer into STK2.

STK2 - I*2 array dimensioned (3,120) which contains the deck numbers, number of classes in that deck in a pool, and a pointer into STK3.

STK3 - I*3 array dimensioned (120) which contains the class numbers of the pooled classes.

Non-standard Return

A RETURN 1 is executed if anyone of the following errors occurs:

1. There is an error returned from STATRD.
2. A class requested in a pool was not included on a classes card.
3. Error in deck numbering.

A RETURN 2 is executed if more than 60 classes are requested without pooling, more than 60 pools are requested or more than 120 classes are requested.

2. Internal Description

MERINT first writes to the printer which options have been selected, which classes from which decks will be used, and which classes will be pooled. MERINT then calls STATRD to read the Statistics files. It then uses the INCLUDE, DELETE or ENTIRE parameters to calculate how many classes will be in the new statistics file. It then sets up the arrays with class and pool information (DCKPTR, DCKSTK, POLPTR and POLSTK).

MERINT CALLS:

STATRD

MERINT is called by:

MERRDR

3. Input Description

Not applicable.

4. Output Description

MERINT writes to the printer all of the options which were selected, which classes and pools were selected, which channels will be in the new statistics file and the following information and error messages:

INNNN WARNING -- CHANNEL ' ' WAS NOT IN ANY OF THE INPUT STATISTICS (MERINT)

ENNN POOL ' ' DECK ' ' CLASS ' ' IS NOT INCLUDED ON A CLASSES CARD FUNCTION TERMINATED (MERINT)

ENNN GAP IN DECK NUMBERS - FUNCTION TERMINATED (MERINT) DECK NUMBER IS ' '

ENNN TOO MANY CLASSES REQUESTED - FUNCTION TERMINATED (MERINT)

5. Supplemental Information

Not applicable.

6. Flowchart

Not applicable.

565

MODULE IDENTIFICATION

Module Name: MERRDR Function Name: MERGESTATISTICS

Purpose: Reads function control cards

System/Language: CMS/FORTRAN

Author: William W. Freestone Date: 7/20/77

Latest Revisor: Joan S. Buis Date: 4/3/80

MODULE ABSTRACT

MERRDR reads and interprets all function control cards for MERGESTATISTICS.

1. Module Usage

MERRDR

CALL MERRDR

This section lists the actions taken when the following control cards are read.

PRINT FIELDS	The flag FIEFLG in MERCOM is set to one. This flag is tested in POLMER. If FIEFLG equals one in POLMER, WRTTRN will be called to print out the list of training fields.
PRINT STATS	The flag STAFLG in MERCOM is set to one. This flag is tested in POLMER. If STAFLG equals one in POLMER, POLMER prints the class means, standard deviations, and correlation matrices.
PRINT COSPEC	The flag CSPFLG in MERCOM is set to one. This flag is tested in POLMER. If CSPFLG equals one in POLMER, COSPEC is called to print the coincident spectral plot.
PRINT MEANS	The flag PLTFLG in MERCOM is set to one. FVAL is called to decode the means parameters (which channels should be plotted). POLMER tests PLTFLG; if equal to one, BISPLT is called to print the bispectral plot of the appropriate channels.
PUNCH (NONE)	The flag PUNFLG in MERCOM is set to one and the flag NBNFLG in MERCOM is left set at one. These falgs are tested in POLMER, if PUNFLG and NBNFLG are equal to one, statistics will be punched in binary format.
PUNCH CHARACTER	The flag PUNFLG in MERCOM is set to one and the flag NBNFLG in MERCOM is set to zero. These flags are tested in POLMER, if PUNFLG is equal to one and NBNFLG is equal to zero, statistics will be punched in character format.

PUNCH ONEFIELD	The flag PUNFLG in MERCOM is set to one and the flag FLDFLG in MERCOM is set to one. These flags are tested in MERSTT. If FLDFLG is equal to one, a single dummy field description card is punched for each class of the modified statistics file.
CHANNELS I,J...	The flag CHAFLG in MERCOM is set to one and the routine CHANEL is called to interpret the CHANNELS card.
CLASSES ENTIRE	IVAL is called to interpret the CLASSES ENTIRE card, and the CLSPTR array is modified.
CLASSES DELETE	CLSDEC is called to interpret the deck numbers and class numbers on the CLASSES card. The class numbers are arranged into ascending order and put into a stack, CLSSTK.
CLASSES INCLUDE	CLSDEC is called to interpret the deck numbers and class numbers on the classes card. The class numbers are arranged into ascending order and put into a stack, CLSSTK.
POOL NAME	POLSET is called to decode the POOL card.
DISK READSTATS	The flag DSKFLG in MERCOM is set to one. This flag is tested in STATRD, if DSKFLG is equal to one, unit will be set to input device SDATA. When STAT is called from STATRD, it will read statistics file from disk.
SCALE SPCLOW	FVAL is called to decode the parameters to set the low end of the coincident spectral plot to a value other than zero.
SCALE SPCINT	FVAL is called to decode the parameters to set the coincident spectral plot interval to a value other than one.
SCALE ORIGIN	FVAL is called to decode the parameters and PLTSKI is modified to readjust the origin for the bespectral plot for any channel to a value other than zero.

SCALE UNIT	FVAL is called to decode the parameters PLTSTK is modified to readjust the bispectral plot interval to a value other than one.
DATA	A check is done to make sure NDECK is not equal to zero, if zero, an error indicating no classes card read before data card is printed. MERINT is then called to read the statistics file and set up DCKPTR, DCKSTK, POLPTR and POLSTK.
END	Control is returned to MERSUP, either through normal termination (RETURN) or termination resulting from an error (RETURN 1).

2. Internal Description

MERRDR uses standard card reader logic in using CTLWRD, CTLPRM, IVAL, and FVAL in reading and interpreting the control cards.

MERRDR begins by setting up DATA lists to compare to the control cards. It then initializes arrays and flags. MERRDR then functions as a loop for reading and interpreting control cards -ntil a DATA card is read, indicating the end of the functional control cards. MERRDR calls CLSDEC, POLSET, and CHANEL to facilitate decoding control cards, it calls MERINT to read the statistics decks, and it calls RTMAIN in case of an error from CTLWRD. An error message is printed and the function is terminated if an END card is read before a DATA card when the stats are expected from cards, there is an error inside the parenthesis on a CLASSES card, or there is an error on a POOL card. Control is returned to MERSUP.

3. Input Description

Control cards for the *MERGESTATISTICS processor.

4. Output Description

The control card error messages that are written include:

ENNN END CARD READ BEFORE DATA CARD WHEN STATISTICS EXPECTED FROM CARDS - FUNCTION TERMINATED (MERRDR)

ENNN NO PARAMETER ON ABOVE CARD - CARD IGNORED (MERRDR)

ENNN SYNTAX ERROR ON ABOVE CARD - TYPE CORRECT CARD (MERRDR)

ENNN ERROR INSIDE PARENTHESES ON CLASSES CARD - FUNCTION TERMINATED (MERRDR)

ENNN DECK ' ' PREVIOUSLY MENTIONED ON CLASSES CARD - REFERENCE IGNORED (MERRDR)

ENNN NO CLASSES CARD READ - TYPE IN A CLASSES CARD (MERRDR)

ENNN ERROR ON POOLS CARD. CORRECT ALL POOL CARDS AND START OVER (MERRDR)

5. Supplemental Information

See system manual for card reader requirements.

6. Flowchart

Not applicable.

MODULE IDENTIFICATION

Module Name: COSPEC Function Name: MERGESTATISTICS

Purpose: To print a coincident spectral plot

System/Language: CMS/FORTRAN

Author: B. J. Davis Date:

Latest Revisor: Date:

MODULE ABSTRACT

COSPEC prints a coincident spectral plot of all the classes in a given statdeck. It will also readjust the origin and plot interval for specified channels as requested.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

COSPEC

```
CALL COSPEC (MEAN,COVAR,CLSDES,NOCLS,NOFEAT,FETVEC,VARSIZ,
             SPCBAS,SPCINT)
```

Input Arguments:

MEAN - R*4. Array of class means

COVAR - R*4. Covariance array

CLSDES - I*4. (2,NOCLS) array containing class name for each class in new statistics file.

NOCLS - I*4. Number of classes in new statistics file.

NOFEAT - I*4. Number of features (channels) in new statistics file.

FETVEC - I*2. (30) array of channel numbers used in statistics file.

VARSIZ - I*4. Number of elements in covariance matrix for a class.

SPCBAS - R*4. Low end of spectral plot if user set SPCLOW to a value other than zero.

SPCINT - R*4. Interval of spectral plot if user set SPCINT to a value other than one.

There are no non-standard returns.

2. Internal Description

COSPEC prints a header line and a legend showing which symbols will be used to represent each class in the plot. It labels the spectral bands on the X axis and data values on the Y axis. It calculates the mean plus and minus one standard deviation for each class in each channel. It prints out one line at a time, each line showing the class mean and the spread of plus and minus one standard deviation for each class in each channel. It prints out one line at a time, each line showing the class mean and the spread of plus and minus one standard diviation.

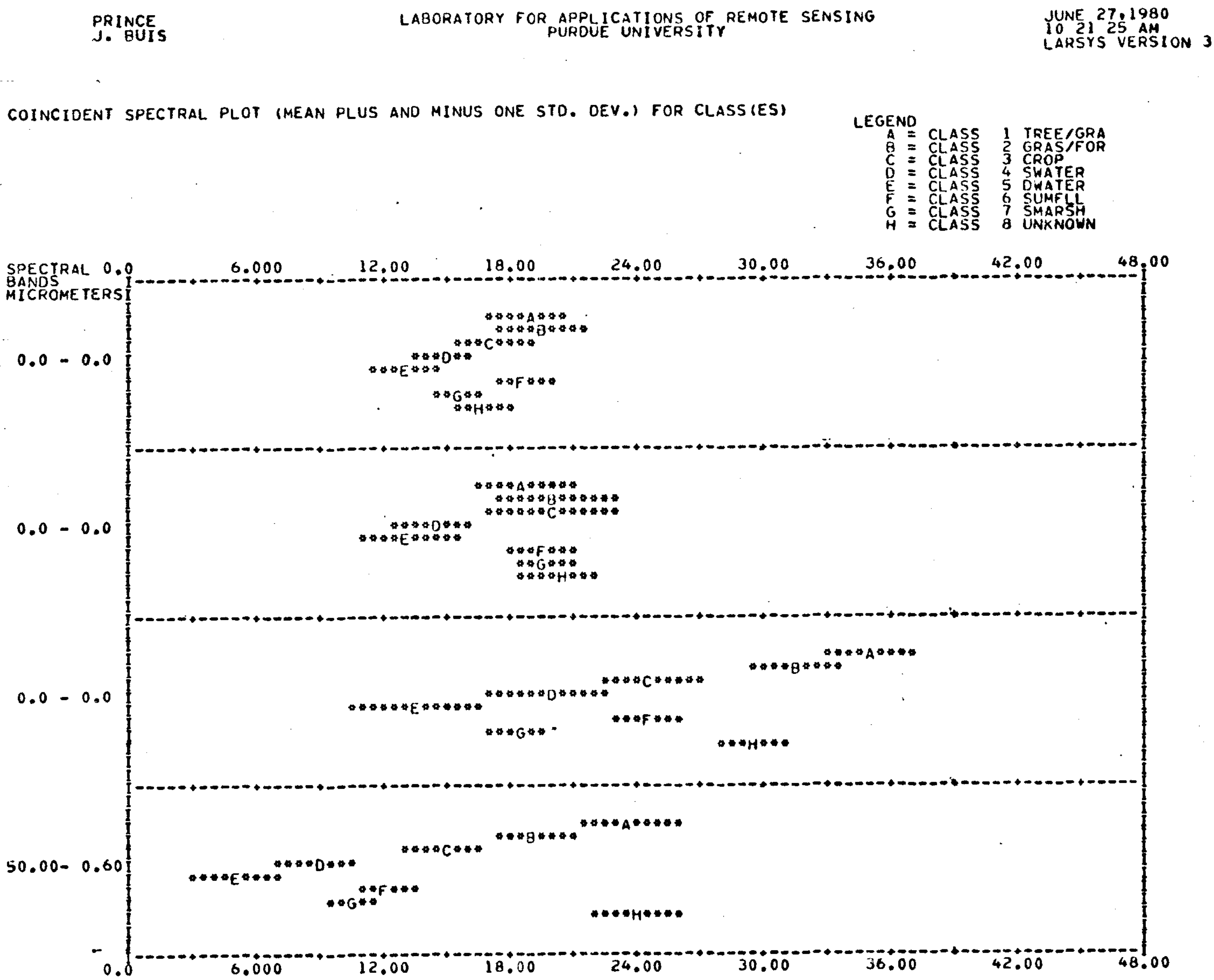
COSPEC calls:	COSPEC is called by:
GETIME	POLMER

3. Input Description

Not Applicable

4. Output Description

An example spectral plot follows.



5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

MODULE IDENTIFICATION

Module Name: BISPLT Function Name: MERGESTATISTICS

Purpose: Prints bi-spectral plot of classes in stat file

System/Language: CMS/FORTRAN

Author: C. A. Pomalaza Date:

Latest Revisor: J. S. Buis Date:

MODULE ABSTRACT

BISPLT prints a bi-spectral plot of all the classes in a given statdeck. It plots a symbol representing class means for any channel plotted against any other channel, or the average of 2 channels against the average of 2 other channels. It will re-adjust the origin and plot interval for specified channels as requested. It prints summary information, including the class number, symbol, value on each axis, class name and the number of points in each class.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

BISPLT

CALL BISPLT (MEAN,CLSDES,NOCLS,NOFEAT,FETVEC,NPLT,IPLSTK, KEPPTS)

Input Arguments:

MEAN - R*4. Array of class means

CLSDES - I*4. (2,NOCLS) array containing class number and class name for each class in the new statistics file.

NOCLS - I*4. Number of classes in new statistics file.

NOFEAT - I*4. Number of features (channels) in new statistics file.

FETVEC - I*2. (30) array of channel numbers used in statistics file.

NPLT - I*4. Number of bi-spectral plots requested.

IPLSTK - I*2. (4,30) entry (I,J) is the channel number of bi-spectral plot J.

PLTSTK - R*4. (2,30) entry (1,I): origin for channel I in the plot, entry (2,I): interval for channel I in the plot.

KEPPTS - I*4. (60) array to store the number of points in each class.

There are no non-standard returns.

2. Internal Description

BISPLT compares the channels requested to be plotted with the channels available in the statdeck and prints an error message if requested channels are not found, and an informational message if they are found. BISPLT then uses the class means to fill the plot array, noting if two or more classes would have been superimposed (CODE = 2) or a symbol would have fallen outside of the area displayed (CODE = 1). BISPLT prints out header information and a legend, and then the actual plot. The plot is followed by summary information.

BISPLT calls:	BISPLT is called by:
GETIME	POLMER

3. Input Description

Not Applicable

4. Output Description

An example bispectral plot follows.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

LARS Program Abstract 336

578

MODULE IDENTIFICATION

Module Name: STATRD Function Name: MERGESTATISTICS

Purpose: reads statistics files

System/Language: CMS/FORTRAN

Author: B. J. Davis Date: 08/24/73

Latest Revisor: J. S. Buis Date: 15/12/79

MODULE ABSTRACT

STATRD calls STAT, REDSTA, MEXPAN and REDFLD to read the statistics file. It modifies the mean and covariance arrays according to channels selected.

Entry Points: None

1. Module usage

STATRD

CALL STATRD

NON-STANDARD RETURN

A return 1 is executed if a core overflow occurs or there is no possible subset of channels and the function is terminated.

2. Internal description

STATRD first calls STAT to read the statistics file from the cardreader or the disk file. It then initializes arrays and calls REDSTA to fill the arrays with the statistics. If channels were specified, STATRD will then modify the mean and covariance arrays according to the channels specified. It checks to make sure core overflow will not occur; if it will, it prints an error message and terminates processing. It calls REDFLD to read the training fields from the statistics file.

3. Input description

Not applicable.

4. Output description

Not applicable.

5. Supplemental information

Not applicable.

6. Flowchart

Not applicable.

580

MODULE IDENTIFICATION

Module Name: REDFLD Function Name: MERGESTATISTICS

Purpose: To interpret training field cards

System/Language: CMS FORTRAN

Author: B. J. Davis Date:

Latest Revisor: Joan Buis Date: 05/80

MODULE ABSTRACT

REDFLD uses LAREAD to interpret training field cards from a statistics file on SDATA. If the card is a valid training field card, it will write the field coordinate information to file TTFLDX. If the card is non-recognizable, it will check to see if is has read all the training field cards for the number of classes in the statistics file, if so, it will do a standard return, if not, it will call ERPRNT with code #302.

Entry Points: None

1. Module Usage

REDFLD

CALL REDFLD (NOCLS, DCKNUM)

Input Arguments:

NOCLS - I*4 Number of classes in statistics deck
DCKNUM - I*4 Number of current statistics file being read

There is no non-standard returns.

2. Internal Description

Described in Module abstract.

REDFLD Calls:

LAREAD
ERPRNT

REDFLD is called by:

POLMER

3. Input Description

Not applicable.

4. Output Description

REDFLD writes field information, deck number and class number to file TTFLDX.

5. Supplemental Information

Not applicable.

6. Flowchart

Not applicable.

MODULE IDENTIFICATION

Module Name: MERSTT Function Name: MERGESTATISTICS

Purpose: Writes merged statdeck to disk.

System/Language: CMS/FORTRAN

Author: B. J. Davis Date:

Latest Revisor: Joan S. Buis Date: 4/28/80

MODULE ABSTRACT

MERSTT creates the merged statistics deck from the various input decks and writes this new deck to the temporary disk (SDATA) in a file called STATS DATA.

Entry Points: None

1. Module Usage

MERSTT

CALL MERSTT

MERSTT is called by MERSUP.

Non-Standard Return

A RETURN 1 is executed if one of the following error occurs:

1. Insufficient core to store reduced statistics.
2. Error in number of field cards.

2. Internal Description

MERSTT writes to the temporary disk the new merged statistics file. It writes record types 1-8 from information contained in DCKPTR and DCKSTK which was set up in MERINT. To aid in writing the means and covariance cards, it calls REDSAV to reduce the mean vectors and covariance matrices.

MERSTT Calls:	MERSTT is called by:
REDSAV	MERSUP

3. Input Description

Not Applicable.

4. Output Description

In addition to the merged statistics file being written to the temporary disk, the following information and error messages are written to the console and/or the printer.

IOXXX	STATISTICS BEING MERGED (MERSTT)
ENNN	CORE OVERFLOW BY ' ' BYTES - FUNCTION TERMINATED (MERSTT)
ENNN	ERROR IN NUMBER OF FIELD CARDS - FUNCTION TERMINATED (MERSTT)

5. Supplemental Information

Not applicable.

6. Flowchart

Not applicable.

MODULE IDENTIFICATION

Module Name: MEXPAN Function Name: MERGESTATISTICS

Purpose: expand the mean and covariance arrays

System/Language: CMS/FORTRAN

Author: C. A. Pomalaza Date: 10/31/76

Latest Revisor: Date:

MODULE ABSTRACT

MEXPAN expands covariance arrays and mean arrays to a size specified by the calling routine, it fills the empty positions with zeros and then moves the mean and covariance values into the expanded arrays.

Entry Points: none

1. Module Usage

MEXPAN

CALL MEXPAN (AVARX, COVARX, AVART, COVART, NCHAN, VARSIZ, UNIFET, UNIVAR, NCLA, CHAN, FETVEC)

Input Arguments:

AVARX	- R*4	(NCHAN, NCLA) mean matrix
COVARX	- R*4	(VARSIZ, NCLA) covariance matrix
NCHAN	- I*4	Number of features (channels) from statistics
VARSIZ	- I*4	Size of covariance matrix
UNIFET	- I*4	Number of features (channels) in expanded statistics
UNIVAR	- I*4	Size of the expanded covariance matrix
NCLA	- I*4	Number of classes in statistics
CHAN	- I*2	(UNIFET) channels for the expanded statistics
FETVEC	- I*2	(NCHAN) channels used for statistics

Output Arguments:

AVART	- R*4	(UNIFET, NCLA) expanded mean matrix
COVART	- R*4	(UNIVAR, NCLA) expanded covariance matrix

There are no non-standard returns

2. Internal Description

Described in module abstract.

MEXPAN calls:

NONE

MEXPAN is called by:

STATRD

3. Input Description

Not applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Flowchart

Not Applicable

MODULE IDENTIFICATION

Module Name: POLMER Function Name: MERGESTATISTICS

Purpose: Pools statistics for appropriate classes

System/Language: CMS/FORTRAN

Author: B. J. Davis Date:

Latest Revisor: Joan S. Buis Date: 4/28/80

MODULE ABSTRACT

POLMER pools appropriate classes and writes new statdeck to disk if pooling is requested, it punches the statistics if requested, and calls other routines to generate plots or write lists depending on option specified in control cards.

Entry Points: None

1. Module Usage

POLMER

CALL POLMER

POLMER is called by MERSUP.

Non Standard Return

A RETURN 1 is executed if one of the following errors occurs.

1. An error is returned from COSPEC.
2. Error in number of field cards.

2. Internal Description

POLMER calls STAT to read the statistics file from the file STATS DATA on the temporary disk. It then calls REDSTA to interpret the statistics file and save it in core. If pooling has been requested, POLMER calls REDFLD to read the training fields from the statistics file and put the training field coordinates in file TTFLDX. POLMER then writes a new statistics file to SDATA (on temporary disk). To write the means and covariance cards, it calls REDSAV to actually pool the statistics of the specified subset of classes. If punching of statistics was requested, SDATA is read from the disk (written in binary format if NBNFLG=1, in character format if NBNFLG=0) and written to the punch. If a coincident spectral plot was requested, BISPLT is called; if the printing of training fields was requested, WRTTRN is called to do this. If the printing of the new statistics was requested, POLMER writes to the printer the class means, standard deviations and correlation matrices (through a call to WRTMTX).

POLMER Calls:	POLMER is called by:
STAT	MERSUP
REDSTA	
REDFLD	
REDSAV	
COSPEC	
BISPLT	
WRTTRN	
WRTMTX	
GETIME	

3. Input Description

Not applicable.

4. Output Description

In addition to writing and possibly punching the new statistics file, calling appropriate routines to write plots or lists, POLMER prints the means, standard deviations and correlation matrices if printing the statistics was requested. An example of the format of the output follows:

MERGED STATISTICS

CLASS.... NS- 2/8

CHANNEL	1	2	3	4
SPECTRAL BAND	0.5 - 0.6	0.6 - 0.7	0.7 - 0.8	0.8 - 1.1
MEAN	19.19	19.35	34.84	23.16
STD. DEV.	1.28	1.97	1.52	1.81

CORRELATION MATRIX

SPECTRAL BAND	0.5 - 0.6	0.6 - 0.7	0.7 - 0.8	0.8 - 1.1
0.5 - 0.6	1.00			
0.6 - 0.7	0.57	1.00		
0.7 - 0.8	0.37	0.13	1.00	
0.8 - 1.1	0.01	0.14	0.53	1.00

POLMER also writes to the printer a list of all the classes in the new deck. The following information and error messages are also written by POLMER.

IOXXX STATISTICS BEING POOLED (POLMER)

IO208 STATISTICS BEING PUNCHED (POLMER)

ENNN ERROR IN NUMBER OF FIELD CARDS-FUNCTION TERMINATED (POLMER)

ENNN ERROR IN CONSTRUCTING POOLED CLASSES-FUNCTION TERMINATED (POLMER)

5. Supplemental Information

Not Applicable.

6. Flowchart

Not applicable.

590

MODULE IDENTIFICATION

Module Name: POLSET Function Name: MERGESTATISTICS

Purpose: Decode the pool card

System/Language: CMS FORTRAN

Author: B. J. Davis Date: 08/27/74

Latest Revisor: Joan Buis Date: 05/80

MODULE ABSTRACT

POLSET calls LOCATE, BCDFIL and POLDEC to decode the POOL cards. It arranges class numbers in ascending order and puts them in a stack.

Entry points: None

1. Module Usage

POLSET

CALL POLSET (POLNAM, NOPOOL, STK1, STK2, STK3, PTR1, PTR2, COL, CARD, *)

Input Arguments:

POLNAM	- I*4 (2,60)	Array to store pool names.
NOPOOL	- I*4	Number of pools in job.
STK1	- I*2 (2,60)	Number of decks in a pool and pointer to STK2.
STK2	- I*2 (3,60)	Deck number, number of classes in that deck in a pool, and pointer into STK3.
STK3	- I*2 (120)	Class numbers of pooled classes.
PTR1	- I*4	Pointer into STK2.
PTR2	- I*4	Pointer into STK3.
COL	- I*4	Column pointer in card.
CARD	- I*4 (18)	18 fullword array containing the control card.

Non-Standard Return

A RETURN 1 is executed if an error is returned from LOCATE or POLDEC.

2. Internal Description

POLSET first calls LOCATE and BCDFIL to update the number of pools in the job, read and store the pool name, and update STK1. It calls POLDEC to decide the portion of the pool card enclosed in parenthesis and updates PTR1 and STK2. It then arranges the class numbers in ascending order and updates PTR2 and STK3. It updates STK1, and, if finished with the card, it will return to MERRDR if not repeat.

POLSET CALLS	POLSET is called by:
LOCATE	MERRDR
BCDFIL	
POLDEC	

3. Input Description

Not applicable.

4. Output Description

Not applicable

5. Supplemental Information

Not applicable.

6. Flowchart

Not applicable.

LARS Program Abstract 350

MODULE IDENTIFICATION

Module Name: SMOSUP Function Name: SMOOTHRESULTS

Purpose: Supervisor for SMOOTHRESULTS

System/Language: CMS/FORTRAN

Author: Joah Cain Date: 4/18/80

Latest Revisor: Date:

MODULE ABSTRACT

SMOSUP supervises the SMOOTHRESULTS function by calling two subroutines: one to read the control cards, and one to modify the input results file.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL SMOSUP

No arguments are used in the call to SMOSUP. This SMOOTHRESULTS supervisor routine is called from LARSMN. Upon completion of the function, control is returned to LARSMN.

2. Internal Description

SMOSUP contains two common blocks - GLOCOM and SMOCOM. This supervisory routine calls SMORDR to read in the function control cards. After the cards have been interpreted, SMOSUP calls SMOINT which initiates the modification of an area. When SMOINT returns control to SMOSUP, the supervisor indicates that the function is completed and returns control to LARSMN. Subroutines called by SMOSUP:

SMORDR
SMOINT

3. Input Description

Not applicable.

4. Output Description

Standard supervisor information messages.

5. Supplemental Information

Not Applicable.

6. Flowchart

Not applicable.

LARS Program Abstract 351

MODULE IDENTIFICATION

Module Name: SMOCOM Function Name: SMOOTHRESULTS

Purpose: Block common for SMOOTHRESULTS module

System/Language: CMS/FORTRAN

Author: John Cain Date: 4/18/80

Latest Revisor: Date:

MODULE ABSTRACT

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

596

LARS Program Abstract 352

MODULE IDENTIFICATION

Module Name: SMOINT Function Name: SMOOTHRESULTS

Purpose: Modify the input Classification Results File

System/Language: CMS/FORTRAN

Author: John Cain Date: 4/18/80

Latest Revisor: Date:

MODULE ABSTRACT

SMOINT reads the input Classification Results File "n" lines at a time (where n is the number of lines in a 'cell'), reassigns class numbers to the dominant class (depending on user-input parameters) and writes out a modified Classification Results File.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SMOINT

CALL SMOINT

There are no arguments to SMOINT; which is called by SMOSUP and returns control to SMOSUP when the function terminates.

2. Internal Description

SMOINT reads from the input classification Results File and modifies each record type in the following way:

RECORD TYPE 1 - Record type 1 is read and checked to see whether it is valid (i.e., not a startup file). The new tape and file numbers are written onto the output Classification Results File.

RECORD TYPE 2 - The number of classification pools is changed to the number of group names and the pool pointer and stack arrays are changed accordingly. If no grouping was done, the pools are then considered classes in themselves. If there are any weights, then they are checked and normalized before written onto the output results file.

RECORD TYPE 3 - The first card of the statistics deck is marked indicating that the deck is invalid due to execution of the *SMOOTHRESULTS function. This deck is then copied onto the output results file.

RECORD TYPE 4 - unchanged.

RECORD TYPE 5 - SMOINT checks the area requested by the user to be sure it exists. If only part of the area requested exists, then the user is given the option to continue or terminate the function. If the user continues, the parameters are changed so that they are now valid. These are written to the output results RECORD TYPE 5. If no BLOCK card was used this record remains unchanged.

RECORD TYPE 6 - This record is read into buffers "n" lines at a time, (where "n" is the number of lines in a cell) and shifted until only the columns requested are considered. SMOINT then calls SMOOTH to modify the buffer data. Upon return, SMOINT writes the modified lines to the output Classification Results file.

RECORD TYPES 7,8 - These record types are created according to the required Classification Results File format. A final record TYPE 1 is written, all tapes are detached and control returns to SMOSUP. Subroutines called by SMOINT:

SMOOTH

3. Input Description

SMOINT reads the first 6 record types off of a LARSYS Classification Results File. It also checks the weights card and can read in a corrected version if required.

4. Output Description

Several information messages may be issued. They are as follows:

a) I*** THIS IS A RESTART FILE -- RUN CLASSIFYPOINTS FIRST.

b) I*** FILE LENGTH IS ONLY ONE RECORD -- TRY RUNNING LISTRESULTS.

c) I*** AREA REQUESTED ONLY PARTIALLY WITHIN THIS CLASSIFICATION AREA. DO YOU WISH TO CONTINUE?

d) I*** EXECUTION TERMINATED BY USER. DO NOT CONSIDER PARTIAL AREA.

All eight record types are written to either tape or disk in the LARSYS Classification Results File format.

5. Supplemental Information

Not applicable.

6. Flowchart

Not applicable.

LARS Program Abstract 353

MODULE IDENTIFICATION

Module Name: SMORDR Function Name: SMOOTHRESULTS

Purpose: Read and interpret control cards for SMOOTHRESULTS

System/Language: CMS/FORTRAN

Author: John Cain Date: 4/18/80

Latest Revisor: Date:

MODULE ABSTRACT

SMORDR reads and interprets all control cards for SMOOTHRESULTS and prints out a summary of the user's requests.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SMORDR

CALL SMORDR

There are no arguments to SMORDR. SMORDR interprets the following function control cards and takes the indicated actions:

CONTROL CARD	ACTION
INRESULTS	INRES is set to the correct DSRN for tape or disk. If it is a tape that is requested, MMTAPE is called to mount and position it.
CELLSIZE	CELROW is set equal to the number of lines per cell (length) and CELCOL is set to the number of columns (width).
OUTRESULTS	OUTRES is set to the correct DSRN for tape or disk. If it is a tape, MMTAPE is called to mount and position it.
PRIORITY	The vector PCLASS is filled with user defined priority classes.
GROUP	GRPNAM and GRPSTK are computed by a call to GRPSCN.
WEIGHTS	These weights are read into PROB, a REAL*4 vector of dimension GO.
BLOCK	The first 6 words of array BLOCK are used to contain this field boundary definition and the run number is put in RUNNUM.
MIXCLASS	The array MIXDAT is filled with the low and high range percentage values to be later compared to each group in each cell. The array is filled by a call to READMX.

2. Internal Description

SMORDR initializes necessary variables and then uses CTLWRD to interpret the keyword on each card. An unexpected end of file for the control card input causes a call to ERPRNT and termination of the function. Once the keyword has been interpreted, SMORDR uses CTLPRM, IVAL, and FVAL to further interpret the card. If a disk is requested the function TSPACE is used to calculate the amount of available space on the disk. Execution is terminated if this amount is not more than 20% greater than the amount required. READMX is called to read the MIXCLASS card and passes back the array of percentages.

After the END card is read, checks are made to determine whether user requests are valid and complete.

Subroutines called by SMORDR:	MMTAPE	READMX	FVAL
	CTLWRD	CTLPRM	TSPACE
	ERPRNT	IVAL	

3. Input Description

Control cards are read by calls to CTLWRD. If a disk is used, it is first checked to make certain that the file exists.

4. Output Description

"Options chosen" messages are typed in addition to requests for more information.

5. Supplemental Information

Not applicable.

6. Flowchart

Not applicable.

LARS Program Abstract 354

MODULE IDENTIFICATION

Module Name: READMX   Function Name: SMOOTHRESULTS

Purpose: Interprets the MIXCLASS control card

System/Language: CMS/FORTRAN

Author: John Cain   Date: 4/18/80

Latest Revisor: Date:

MODULE ABSTRACT

READMX interprets the MIXCLASS control card(s) for the SMOOTHRESULTS function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

READMX

CALL READMX (LCARD, COL)

Input Arguments:

LCARD - L*1, Card image of the MIXCLASS card being interpreted.

COL - I*4, the column number preceeding the first name on the MIXCLASS card (i.e., the first nonblank after the keyword.)

Output Arguments:

NCLNAM - I*4, NCLNAM is a two dimensional vector that is filled with the new class names from the MIXCLASS card.

NEWCLS - I*4, the number of new classes. NEWCLS is incremented as each new name is read.

MIXDAT - I*4, MIXDAT is filled with the percentage ranges specified on the MIXCLASS card. MIXDAT (I,1) is the lower boundary of the percentage range and MIXDAT (I,2) is the higher boundary.

2. Internal Description

READMX can be reduced to several functions - stripping off the name, scanning and modifying the user supplied ranges, and the reading and loading of the ranges into MIXDAT.

The first function is performed by using LOCATE to find the left parenthesis of the class being interpreted and then by a call to BCDFIL. The data for this particular name is then scanned and the positions of all hyphens are noted by setting a corresponding flag in a flag vector. After a hyphen is read it is changed to a comma so that once the data is finished being scanned it can be read by IVAL. Once IVAL has been called the flag vector is used to transfer the data read into the array MIXDAT. This is done by placing the first data value into MIXDAT (I,1) (i.e. the lower boundary of the user defined percentage range) and then checking the corresponding flag in the flag vector. If the flag was not set, then this lower boundary is simply copied into the upper boundary, MIXDAT (I,2). If the flag was set then the next data value is treated as the upper boundary. This is continued until all values have been loaded.

Subroutines called by READMX: LOCATE

BCDFIL

IVAL

3. Input Description

READMX checks for various syntax errors on the MIXCLASS card. It is therefore capable of requesting a new MIXCLASS card by using CTLWRD.

4. Output Description

READMX can write a syntax error message to the user. This gives the user the approximate location of the error and asks for the card to be retyped.

5. Supplemental Information

Not applicable.

6. Flowchart

Not applicable.

605

LARS Program Abstract 355

MODULE IDENTIFICATION

Module Name: SMOOTH Function Name: SMOOTHRESULTS

Purpose: Compute the dominant class in each cell and reclassify

System/Language: CMS/FORTRAN

Author: John Cain Date: 4/18/80

Latest Revisor: Date:

MODULE ABSTRACT

SMOOTH takes "n" lines of a Classification Results File (where n is the number of lines in a cell), determines which group or class makes up the largest weighted percentage of each cell, and reloads the cell with that class. User-defined priority classes remain unchanged.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL SMOOTH (NOCOLS)

Input Arguments:

NOCOLS - I*4, NOCOLS is the number of columns (i.e. the buffer length) that is to be segmented into individual cell widths.

The following input variables are located in the SMOCOM common:

CELCOL - I*4, the number of columns in the user defined cell.

CELROW - I*4, the number of rows in the user defined cell.

2. Internal Description

Upon entry, SMOOTH uses NOCOLS and the user defined parameters CELCOL and CELROW to determine the total number of cells to be processed. Using this information the main program loop is entered. The main loop begins by tallying the number of samples of each group in the cell. If the "weight" option is in effect then each group's tally is weighted. The total weight is then found by summing all the weighted values. Each group weight is then divided by the total weight to obtain a cell percentage for its group. These percentages are compared to user supplied group percentages from the MIXCLASS card. If the values fall within the MIXCLASS ranges, then all the samples of the cell are changed to this MIXCLASS group and rewritten into the main buffer. Any user defined priority samples (from the PRIORITY card) will not be altered. If none of the MIXCLASS ranges correspond to the calculated percentages, then the nonpriority group with the greatest percentage must be determined. For efficiency the groups not found in the cell are removed from the percentage array (compressed). The array is then sorted so that the group with the greatest percentage is first and the one with the smallest is last. The number of the first group found that is not a priority class is then loaded into a pointer array. Groups defined as priority are loaded (overlayed) into their corresponding position in this array. The cell is then reloaded by using this pointer array for indexing. This loop is repeated for each CELCOL group of columns across the input Classification Results lines. When all cells have been modified, control is returned to SMOINT.

3. Input Description

Not applicable.

4. Output Description

Not applicable.

5. Supplemental Information

The buffer of samples modified by SMOOTH is passed through GLOCOM in ARRAY. It's dimensions are CELROW by NOCOLS.

6. Flowchart

Not applicable.

LARS Program Abstract 356

MODULE IDENTIFICATION

Module Name: SMOTAB Function Name: SMOOTHRESULTS

Purpose: Tabulate results

System/Language: CMS/FORTRAN

Author: J.S. Cain Date:

Latest Revisor: Date:

MODULE ABSTRACT

Tabulate results for the SMOOTHRESULTS function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 360

MODULE IDENTIFICATION

Module Name: SECSUP Function Name: SECSUP

Purpose: Supervised ECHO supervisor routine.

System/Language: CMS/Fortran

Author: P.D. Alenduff Date: 3/30/77

Latest Revisor: Date:

MODULE ABSTRACT

SECSUP is the supervisor routine for the supervised ECHO classifier.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SECSUP

Calling sequence:

CALL SECSUP

SECSUP is the supervisor routine for the supervised ECHO classifier. It is called with no arguments.

2. Internal Description

The program SECSUP basically controls the program sequencing for the many subprograms involved in the supervised ECHO classifier. Two subroutines are called by SECSUP: SECRDR and SECINT.

SECRDR is the control card reader for reading the user input data cards and is called first. SECINT is the initialization routine for the SECHO function and is called to handle utilization of all input/output operations as well as classification processing. For more detailed descriptions of these modules, see their program abstracts.

Two common blocks are used in SECSUP. GLOCOM, the main LARSYS common block and SECCOM, the common block for the supervised ECHO function are included.

3. Input Description

Not applicable

4. Output Description

Two messages are produced and written to unit TYPEWR, the console.

SUPERVISED ECHO FUNCTION REQUESTED

Signifies beginning of the function.

SUPERVISED ECHO FUNCTION COMPLETED

Signifies end of the function.

5. Supplemental Information

Two COMMON blocks are included in SECSUP: GLOCOM
SECCOM

LARS Program Abstract 361

MODULE IDENTIFICATION

Module Name: SECCOM Function Name: SECSUP

Purpose: Common block for supervised ECHO function

System/Language: CMS/FORTRAN

Author: P. D. Alenduff Date: 03/30/77

Latest Revisor: Date:

MODULE ABSTRACT

SECCOM is the common block for the supervised ECHO classifier.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

SECCOM

Variable Description

BUFROZ	INTEGER*4 Number of rows of cells in buffer.
CSELEC	REAL*4 Cell selection parameter.
CSET (3,30)	REAL*4 Channel calibration code vector from statistics.
CSET3 (3,30)	REAL*4 Channel calibration code vector from channels card.
THRES	REAL*4 Annexation threshold.
CELSIZ	INTEGER*4 Number of data points in a cell.
CELWTH	INTEGER*4 Number of rows in cell.
INFO (16)	INTEGER*4 Array for storing field description information and areas to be classified.
INPUT	INTEGER*4 Unit number of input tape.
INTTAP	INTEGER*4 Tape number of intermediate tape.
INTFIL	INTEGER*4 File number of intermediate tape.
JPTS	INTEGER*4 Number of cells horizontally in data area.
LINES	INTEGER*4 Number of rows of cells in area to be processed.
NOCLS	INTEGER*4 Number of classes in statistics deck.
NOFET	INTEGER*4 Number of channels in statistics deck.
NOFET3	INTEGER*4 Number of channels to be used in classification.
NOPOOL	INTEGER*4 Number of pools to be used in classification.
NWORD	INTEGER*4 Integer array word counter.
NWORD2	INTEGER*4 Integer word counter.

NVR	INTEGER*4 Number of channels on data tape.
OUTPUT	INTEGER*4 Unit number of output file.
POLNAM (2,60)	INTEGER*4 Names of statistics classes.
PREFIX(2)	INTEGER*4 Prefix information used in writing results.
RQTAPE	INTEGER*4 Tape number of the results tape.
RQFILE	INTEGER*4 File number of the results file.
SYMCNT	INTEGER*4 Number of user supplied symbols.
VARSZ3	INTEGER*4 Number of words to store half of a triangular covariance matrix for classes (channels * (channels + 1))/2.
CSEL(30)	INTEGER*2 Channel select vector from statistics.
CSEL3(30)	INTEGER*2 Channel select vector from channels card.
FETVEC(30)	INTEGER*2 Array of channel numbers from statistics deck.
FETVC3(30)	INTEGER*2 Array of channel numbers from channels card.
POLPTR(2,60)	INTEGER*2 Array of classification pools and pooling information.
POLSTK(60)	INTEGER*2 Array of stacked class numbers organized by pool request.
CDFLAG	LOGICAL*1 Logical variable signalling input of statistics on cards in control card deck.
CLSMAP	LOGICAL*1 Logical variable requesting production of classification map.
CSET1(3,30)	LOGICAL*1 Logical array indicating storage of calibration values by user.
OBJMAP	LOGICAL*1 Logical variable requesting production of singular cell map in phase 2 (annexation).

PHASE1	LOGICAL*1 Logical variable requesting initial cell processing is to be carried out.
PHASE2	LOGICAL*1 Logical variable indicating annexation of cells is to be carried out.
POLNM1(60)	LOGICAL*1 Logical array signalling that the name for POOL (I) has been stored.
PRSTAT	LOGICAL*1 Logical variable indicating request to print statistics information.
SYM(60)	LOGICAL*1 Array of user supplied symbols.
SYMMTX(60)	LOGICAL*1 Array of default symbols.

LARS Program Abstract 362

MODULE IDENTIFICATION

Module Name: SECRDR Function Name: SECSUP

Purpose: Control card reader for Supervised ECHO Function

System/Language: CMS/FORTRAN

Author: P.D. Alenduff Date: 3/30/77

Latest Revisor: Date:

MODULE ABSTRACT

SECRDR is the control card reader for the Supervised ECHO function. After user requests are read, many different checks are made to detect any control card errors. The input or output classification or intermediate tapes are then mounted on the appropriate devices, and an options list is printed to document user selections.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SECRDR

Calling sequence:

CALL SECRDR

SECRDR is called without parameters to read user option selections and detect any errors on specified combinations of options. Upon error detection, either corrected data is requested or the function is terminated following an information message detailing the error. Output and/or intermediate tapes and the multispectral image storage tape are mounted and a list of selected options is produced.

2. Internal Description

SECRDR first performs initialization on all variables stored in SECCOM common block to set up any default options and clear any previously selected options. Control card decoding is heavily dependent upon LARSYS control card processing routines CTLWRD and BCDVAL. CTLWRD is called with a list of acceptable keywords to read input control cards. The number of the matching keyword is returned to SECRDR and a branch is made to the code that will appropriately decode the remainder of the control card. Other special purpose LARSYS routines used for control card decoding include CHANEL for decoding the CLASSES card.

After encountering a DATA or END card, a check is made for a request of at least one channel on the CHANNELS card. The cell width parameter is then checked to be greater than or equal to 2. The cell selection option must also be non-negative, and annexation threshold THRES must be zero or positive. If both an object map and classification map are requested, only an object map is produced. At this point, tape and disk parameters are checked. If OPTIONS INTERMEDIATE (start processing with the intermediate tape) is selected, a tape number must be supplied. If an intermediate tape number is supplied, a file number must also be specified.

Upon specification of OPTIONS INTERMEDIATE, processing begins with the intermediate tape and a results file is produced. If, however, no OPTIONS INTERMEDIATE request is made and both an intermediate tape number and a results location is specified, a conflict is recognized and EXECUTION is terminated. If a results location is needed and none is specified, more data is requested. A file number must be specified when a tape location is selected. Specifications of both tape and disk options for classification results information causes termination.

After the error checking sequence is finished, tapes are mounted on the appropriate devices by calls to MTAPE. Initialization, if requested, is performed on the results or intermediate tape. A list of the selected options is then produced. Both GLOCOM and SECCOM common blocks are used by SECRDR.

3. Input Description

Input data is read from unit READIN in GLOCOM which is either CRDRDR, the card reader, or KEYBD, the console keyboard, when cards are typed in by the user. This selection is accomplished by the LARSYS system.

4. Output Description

Output is written to both PRNTR, the printer, and TYPEWR, the console device. A variety of error messages is produced by SECRDR, and brief list follows:

ERROR IN TAPE OR FILE SPECIFICATION-TYPE CORRECT CARD

> A non-numeric character was entered as either a tape or file number.

ERROR IN PARAMETER VALUE SPECIFICATION-TYPE CORRECT CARD

> A non-numeric character was specified as the cell size, cell selection, or annexation value.

YOU HAVE ENTERED X SYMBOLS. THE MAXIMUM ALLOWED IS 60. EXCESS SYMBOLS WILL NOT BE USED.

> The maximum number of symbols that can be stored is 60. Only the first 60 can be used.

CELL SIZE MUST BE GREATER THAN OR EQUAL TO TWO-DEFAULT OF 2 ASSUMED-TYPE CORRECT CARD

> The CELL SIZE (X) parameter cannot be less than 2. A corrected card must be supplied.

CELL SELECTION PARAMETER MUST BE GREATER THAN OR EQUAL TO ZERO-TYPE CORRECT CARD

> The CELL SELECT (X) entry cannot be negative, a new card is requested.

ANNEXATION THRESHOLD MUST BE NON-NEGATIVE. TYPE A CORRECTED ANNEXATION THRESHOLD CARD.

> The THRES specified is negative. The user is requested to correct the error.

BOTH SINGULAR CELL AND CLASSIFICATION MAPS REQUESTED. ONLY OBJECT MAP WILL BE PRODUCED.

Only one map can be produced during annexation. If both are requested, only the singular cell map will be printed.

NO INTERMEDIATE TAPE SUPPLIED FOR OPTIONS INTERMEDIATE. TYPE IN INTERMEDIATE CARD

When requesting that processing start from an intermediate tape, an intermediate tape number must be specified.

INTERMEDIATE FILE OR INITIALIZE MUST BE SPECIFIED. TYPE ADDITIONAL INTERMEDIATE CARD

When producing an intermediate tape, some indication of file number must be given.

BOTH INTERMEDIATE TAPE AND RESULTS LOCATION SPECIFIED WITHOUT OPTIONS INTERMEDIATE-JOB TERMINATED

It is impossible to determine which part of the classification should be performed when this set of options is supplied. Execution is terminated.

EITHER INTERMEDIATE TAPE OR RESULTS LOCATION MUST BE SPECIFIED WITHOUT OPTIONS INTERMEDIATE. TYPE IN ADDITIONAL CARD

Some output file is needed for processing. Additional information is requested.

EITHER RESULTS FILE OR INITIALIZE MUST BE REQUESTED. TYPE IN ADDITIONAL RESULTS CARD.

When producing a results tape, some indication of file number must be given.

NO RESULTS DESTINATION SPECIFIED-TYPE IN RESULTS CARD

Either tape and file or disk must be specified when results will be produced. Additional information is requested.

BOTH RESULTS TAPE PARAMETERS AND DISK SPECIFIED-FUNCTION TERMINATED

Either tape or disk can be selected for results but not both.

BOTH FILE AND INITIALIZE REQUESTED FOR INTERMEDIATE TAPE-FILE REQUEST IGNORED

The intermediate output tape is initialized. Only file 1 can be initialized.

BOTH FILE AND INITIALIZE REQUESTED FOR RESULTS TAPE-FILE REQUEST IGNORED

The results tape is initialized. Only file 1 can be initialized.

In addition to these messages, the list of options selected is printed on the unit PRNTR.

5. Supplemental Information

Common blocks GLOCOM and SECCOM are used by SECRDR.

LARS Program Abstract 363

MODULE IDENTIFICATION

Module Name: SECINT Function Name: SECSUP

Purpose: Initiator routine for Supervised ECHO classifier.

System/Language: CMS/FORTRAN

Author: P.D. Alenduff Date: 4/27/77

Latest Revisor: Date:

MODULE ABSTRACT

SECINT carries out required initialization of the rest of variables used by SECSUP and finishes reading any data cards as well as the statistics to be used with the processor. In addition to the array allocation performed, some option information is printed and a loop is entered that carries out the desired stages of classification for each area to be classified.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SECINT

CALL SECINT

The program SECINT is called with no arguments. Any variables to be used or changed are contained in common blocks GLOCOM and SECCOM, both required by SECINT. All information required to perform the classification is gathered by SECINT from the appropriate source (i.e., statistics and areas to be classified) and array allocation is carried out using variables supplied to common block SECCOM by the control card reader SECRDR. Statistics are reduced by calling REDSAV to save statistics for only classes and channels requested and statistics information is printed by calling SECPRT. Initial records are then written to the output device. A loop is entered and, for each area to be classified, a sequence of tests is performed. If multispectral data is required, the correct run is requested and existance of requested channels is confirmed. Further allocation is performed and SECHO1 is called to perform the needed processing for the area.

2. Internal Description

The first information handled by SECINT is the statistics file. If processing is starting from an intermediate tape, records 1 and 2 are read from the intermediate tape to re-establish information used in initial processing such as channels selected and calibration used, and classes selected and pooling requested. The statistics file is then read from the tape and written onto the unit SDATA, the statistics file on disk. The next card is checked for an "end" card and if any area description cards are required, the function is terminated.

Otherwise, CLASSX, the storage disk file for all classification areas for this file, is rewound and processing loops until all areas are stored on disk. Some initial allocation is then performed as array storage is set up for the covariance and mean matrices for the entire statistics file on disk. REDSTA transfers the file into memory. CLSCHK and FETCHK then check for fatal errors in selection of pooling and channels. The addresses of reduced arrays for storing only needed information are then computed and REDSAV reduces the information to that required.

Initial records are then produced and written to OUTPUT. The entire statistics file from SDATA is then transfered to the output file, and SECPRT is called to print the statistics information, if requested, and to produce the record type 4 with covariance and means matrices of the classes. A loop is then started and executed once for each area to be classified. If processing begins with the data type, the correct input run is requested and calibration information is established. If processing begins with an intermediate type, all areas on the tape in this file are used. Field size and the allocation of all other arrays used by SECH01 are calculated. Then a record type 5, area identification record, is produced and SECH01 is called once for each area to be processed. When finished, the final record is written to the output device.

3. Input Description

Input cell information for SECINT is read from CPYOUT if an intermediate tape is requested as the starting point of processing. The format is identical to that of a classification results file. SDATA, the statistics file on disk is also read to be copied into the output file. CLASSX, the areas to be classified as stored on disk, is also used to store the list of areas to be processed.

4. Output Description

A copy of the statistics file is written to SDATA from the intermediate tape is used as input. CLASSX is written as the list of areas to be classified. The PRNTR is used for messages and output information. The file OUTPUT is the result of processing with the same format as a classification results file.

5. Supplemental Information

This program uses common blocks GLOCOM and SECCOM. See LARSYS Systems Manual for classification results file format.

LARS Program Abstract 364

MODULE IDENTIFICATION

Module Name: SECRD Function Name: SECSUP

Purpose: Read data arrays from intermediate tape.

System/Language: CMS/Fortran

Author: P.D. Alenduff Date: 4/12/77

Latest Revisor: Date:

MODULE ABSTRACT

SECRD reads data arrays from the intermediate tape for annexation processing by SECHO.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SECRD

CALL SECRD (COVMTX, AVEMTX)

Input Arguments:

COVMTX (VARSZ3, NOPOOL)	R*4 An array dimensioned (channels * (channels + 1))/2 fullwords for storing the covariance matrices of classes used in classification.
AVEMTX (NOPET3, NOPOOL)	R*4 An array dimensioned channels by classes words for storing the class means of classes used in the classification.

Output Arguments:

Same as above.

SECRD is called to read one record from the intermediate tape. COVMTX and AVEMTX are read in an unformatted fashion from the unit INPUT. This program uses common blocks GLOCOM and SECCOM.

2. Internal Description

See Above.

3. Input Description

SECRD reads from unit INPUT a record type 4 from the classification format file.

4. Output Description

Not applicable.

5. Supplemental Information

See LARSYS Systems Manual for format of classifcation results file.

625

LARS Program Abstract 365

MODULE IDENTIFICATION

Module Name: SECPRT Function Name: SECSUP

Purpose: Statistics printing and output routine for SECHO

System/Language: CMS/FORTRAN

Author: P. D. Alenduff Date: 4/12/77

Latest Revisor: Date:

MODULE ABSTRACT

SECPRT produces tabular statistics for user information as well as carrying out minor data array processing.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SECPRT

CALL SECPRT (COVMTX, AVEMTX, SCAPRD, DETCOV, CONST, WORK)

Input Arguments:

COVMTX (VARSZ3, NOPOOL)	R*4 Covariance matrix for each class.
AVEMTX NOFET3, NOPOOL)	R*4 Mean matrix for each class.

Output Arguments:

SCAPRD (NOPOOL)	R*4 Scalar product for each class.
DETCOV (NOPOOL)	R*4 Determinant of the covariance matrix for each class.
CONST (NOPOOL)	R*4 Constant formed for computation in classification for each class.
WORK (NOFET3 * (NOFET3 + 2))	R*4 Buffer array for inverting covariance matrix at least channels by channels + 2 words.

SECPRT is called to print the statistics information to be used for classification, if requested, and produce the record type 4 of a classification results format file for SECHO. Also the determinant, scalar product, and constant arrays, DETCOV, SCAPRD, and CONST, are produced for later usage.

2. Internal Description

Header information is printed on the Unit PRNTR followed by the writing of record type 4 on the unit OUTPUT. This classification format record contains COVMTX and AVEMTX, the covariance and mean matrices used in the classification for each channel. This the information concerning classes used is printed and WRTMTX is called to print the correlation matrix. The determinant of the covariance matrices is formed by a call to SAMINV, and a check is made to determine if any determinants were zero or negative. Then SMMULT is called to produce the array SCAPRD, the scalar product, and a calculation is performed to produce the constant array. Execution is terminated if any of the determinants were zero or negative. This routine uses common blocks GLOCOM and SECCOM.

3. Input Description

Not Applicable.

4. Output Description

A record type 4 of the classification format output file is written to unit OUTPUT.

Also, statistics information is produced on unit PRNTR.

5. Supplemental Information

Not applicable.

LARS Program Abstract 366

MODULE IDENTIFICATION

Module Name: FILBUF Function Name: SECSUP

Purpose: Retrieve cell statistics

System/Language: CMS/FORTRAN

Author: P.D. Alenduff Date: 4/18/77

Latest Revisor: Date:

MODULE ABSTRACT

FILBUF returns a set of buffers with cell information from either direct processing from the data tape or reading from the intermediate tape.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

FILBUF

CALL FILBUF (ICLSCV, VECPRD, SCAPRD, SETCOV, CONST, BDATA, RDATA, PIXCOR, PIXVAL, CELCOR, CELSUM, CELIKE, CELCLS, PIXCLS, CURLIN, *)

Input Arguments:

ICLSCV (VARSZ3, NOPOOL)	R*4 Class covariance matrices from statistics dimensioned classes by (channels * (channels + 1))/2.
VECPRD NOFET3, NOPOOL)	R*4 Vector product matrix transposed times the covariance matrix for the class, dimensioned channels by classes.
SCAPRD (NOPOOL)	R*4 Scalar product of the classes for all statistics classes.
DETCOV (NOPOOL)	R*4 Determinant of the covariance matrix for each class.
CONST (NOPOOL)	R*4 Constant array computed from ln (2*PI*class covariance matrix) for each class.
BDATA (NVR, ID(6))	L*1 Buffer array used by GADLIN for reading data tape dimensioned channels on tape by samples per line on tape.
RDATA (NVR, CELWTH)	R*4 Array used to store calibrated data values from tape for one row of cells (multiple data lines).
PIXCOR (VARSZ3, CELWTH^2)	R*4 Array used to store cross product data values in one cell for a correlation type indication.
PIXVAL (NOFET3, CELWTH^2)	R*4 Array used to store pixel data values for a cell.
CELCOR (VARSZ3)	R*4 Sum of the PIXCOR values for a cell being processed.

Output Arguments:

CELSUM (NOFET3)	R*4 The sum of all data points in a cell for each channel.
CELIKE (NWORD, JPTS)	R*4 Likelihood values for each class for each cell in a row of cells.
CELCLS (JPTS)	I*2 Class numbers for each cell in a row of cells.
PIXCLS (NWORDS, JPTS)	I*2 Class numbers of pixels for all ce-ls in a row of cells.
CURLIN	I*4 Current line number requested.

FILBUF is called to fill up CELCLS and CELIKE for annexation or writing to the intermediate tape. By either reading from the intermediate tape for annexation processing, or reading from the data tape and carrying out processing for the area, cell information can be produced.

2. Internal Description

FILBUF produces arrays CELCLS and CELIKE from either reading the intermediate tape that is in a classification format or from reading the input data tape via GADLIN. The raw information is then processed by producing cell statistics with GATHER, establishing cell classes with SAMCLS, testing for cell homogeneity, and outputting a row of cell information. This routine uses common blocks GLOCOM and SECCOM.

3. Input Description

The only input is possible input from the intermediate tape which stores the same format used by classification tapes and stores cell statistics for later annexation processing.

4. Output Description

Not applicable.

5. Supplemental Information

Not applicable.

631

LARS Program Abstract 367

MODULE IDENTIFICATION

Module Name: SAMCLS Function Name: SECSUP

Purpose: Classify cell using maximum likelihood

System/Language: CMS/Fortran

Author: P.D. Alenduff Date: 4/14/77

Latest Revisor: Date:

MODULE ABSTRACT

SAMCLS performs a maximum likelihood classification by calling LOGLIK and finding the maximum likelihood class for a cell.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SAMCLS

```
CALL SAMCLS (CLASS, CHISQR, LIKELY, COVMTX, VECPRD, SCAPRD,
             CONST, SAMCOR, SAMSUM, SAMPTS, NOFET3, MTXSIZ,
             NOPOOL, RETLIK)
```

Input Arguments:

COVMTX (VARSZ3,NOPOOL)	R*4 An array dimensioned (channels * (channels + 1)/2) * number of classes, containing the upper half of the triangular covariance matrices for each of the classes used for classification.
VECPRD (NOFET3,NOPOOL)	R*4 An array dimensioned channels * classes containing the vector product of the classes for each channel from the inverse covariance matrix and the determinant of the covariance matrix.
SCAPRD (NOPOOL)	R*4 An array dimensioned classes for the scalar product of the mean and covariance matrices for each class.
CONST (NOPOOL)	R*4 An array dimensioned "classes" words containing the constant term used consisting of an expression.
SAMSUM (NOFET3)	R*4 Sum of data values for all points in cell for given channel.
SAMPTS	I*4 Number of points per cell.
NOFET3	I*4 Number of channels.
MTXSIZ	I*4 Number of words needed to store half of triangular covariance matrix.
NOPOOL	L*4 Number of classification pools.
RETLIK	L*4 Indication of whether or not likelihoods are to be returned in LIKELY.

Output Arguments:

CLASS	I*4 Class number of cell.
CHISQR	R*4 Resultant Chi-square value from LOGLIK and classification.
LIKELY (NOPOOL)	R*4 Log likelihood of cell belonging to pools.

2. Internal Description

SAMCLS calls LOGLIK once for each class and is returned a log likelihood for the cell for each class. If the likelihood is greater than the previously stored, the new likelihood is saved with the class number and, when done, this class and value are returned.

3. Input Description

Not Applicable.

4. Output Description

Not Applicable.

5. Supplemental Information

Not Applicable.

LARS Program Abstract 368

MODULE IDENTIFICATION

Module Name: LOGLIK Function Name: SECSUP

Purpose: Produce Log Likelihood of a cell

System/Language: CMS. Fortran

Author: P.D. Alenduff Date: 4/13/77

Latest Revisor: Date:

MODULE ABSTRACT

LOGLIK produces a log likelihood for a class in the input statistics.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LOGLIK

```
CALL LOGLIK (LIKELY, CHISQR, ICLSCV, VECPRD, SCAPRD, CONST,
             DIMEN, SAMCOR, SAMSUM, SAMPTS)
```

Input Arguments:

ICLSCV (MTXSIZ)	R*4 Class covariance matrix from statistics, dimensioned channels * (channels + 1)/2.
VECPRD (DIMEN)	R*4 Vector product of class covariance matrix determinant and means dimensioned "channels" words.
SCAPRD	R*4 The scalar product term for the class being considered.
CONST	R*4 The constant term derived in SECPRT.
DIMEN	I*4 Number of channels of data in cell information.
SAMCOR (MTXSIZ)	R*4 The cross product terms from the cells data values dimensioned as ICLSCV.
SAMSUM (DIMEN)	R*4 The sum of the data values for each channel for the cell.
SAMPTS	I*4 Number of data points in a cell, actually (cell width)^2.

Output Arguments:

LIKELY	R*4 Log likelihood value for the cell.
CHISQR	R*4 Chi-square value returned for the cell.

LOGLIK is called to return a log likelihood and a Chi-square value for the cell being considered.

2. Internal Description

The Chi-square value is calculated by the equation:

$$CHISQR = tr(C_j^{-1} \sum^{m} Y_i Y_i') - 2M_j' C_j^{-1} \sum^{m} Y_i + mM_j' C_j^{-1} M_j$$

and the log likelihood is calculated by:

$$LIKELY = -.5 \ (m \cdot \ln |2\pi \ C_j| + CHISQR)$$

Where:

C_j is the class covariance matrix for class j.

M_j is the mean vector for class j.

Y_i is the channel response vector for pixel i of the cell.

m is the number of pixels in the cell.

3. Input Description

Not applicable.

4. Output Description

Not applicable.

5. Supplemental Information

Not applicable.

637

LARS Program Abstract 369

MODULE IDENTIFICATION

Module Name: STATS2 Function Name: SECSUP

Purpose: Carry out annexation tests and processing

System/Language: CMS/FORTRAN

Author: P.D. Alenduff Date: 4/20/77

Latest Revisor: Date:

MODULE ABSTRACT

STATS2 contains entry points LIKRAT, ADD2, and BEGFLD. LIKRAT produces a likelihood test by subtracting log - likelihoods and returns a resultant likelihood value. ADD2 adds a cell to the statistics of a field. BEGFLD initializes field statistics from a fields statistics.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LIKRAT

CALL LIKRAT (STAT, CELIKE, FLDLIK, AUXLIK, CELCLS, FLDCLS, AUXCLS)

Input Arguments:

CELIKE (NOPOOL)	R*4 Log likelihood values for this cell for each class in the statistics.
FLDLIK (NOPOOL)	R*4 Log likelihood values of all cells annexed to the field for each class in the statistics.
CELCLS	I*2 Class number of cell being considered.
FLDCLS	I*2 Class number of field being tested.

Output Arguments:

AUXLIK (NOPOOL)	R*4 Resultant log likelihood of annexation of cell to given field for each class.
AUXCLS	I*2 Resultant class of annexation test for this cell.
STAT	R*4 Resulting statistic concerning annexation performance for this cell to the field being considered.

LIKRAT produces a resulting likelihood, class and statistic is the given cell is annexed to the supplied field.

ADD2

CALL ADD2 (FLDLIK, AUXLIK, FLDCLS, AUXCLS, FLDSIZ)

Input Arguments:

AUXLIK (NOPOOL)	R*4 Likelihood values formed with temporary annexation of cell to field.
AUXCLS	I*2 Class number of cell when annexation of cell to field was successful.
FLDSIZ	I*4 Number of pixels included in the given field.

Output Arguments:

FLDLIK (NOPOOL)	R*4 Likelihood values of new field formed.
FLDCLS	I*2 Class of field formed.
FLDSIZ	I*4 (see above).

ADD2 adds the cell supplied to the field information supplied. Temporary likelihood values are assigned as new likelihoods for field.

BEGFLD

```
CALL BEGFLD (CELCLS, FLDCLS, CELIKE, FLDLIK, FLDSIZ)
```

Input Arguments:

CELCLS	I*2 See above.
CELIKE (NOPOOL)	R*4 See above.

Output Arguments:

FLDCLS	I*2 See above.
FLDLIK (NOPOOL)	R*4 See above.
FLDSIZ	I*4 See above.

BEGFLD initialize FLDCLS, FLDLIK, and FLDSIZ to be equal to CELCLS, CELIKE, and CELSIZ.

2. Internal Description

See above.

3. Input Description

Not applicable.

4. Output Description

Not applicable.

5. Supplemental Information

Not applicable.

640

LARS Program Abstract 370

MODULE IDENTIFICATION

Module Name: SECH01 Function Name: SECSUP

Purpose: Performs Supervised ECHO Processing on area.

System/Language:

Author: P.D. Alenduff Date: 4/11/77

Latest Revisor: Date:

MODULE ABSTRACT

This routine is called once for each area to be classified and performs all classification on each line of data and produces the output file.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

SECHO1

CALL SECHO1 (ICLSCV, VECPRD, SCAPRD, DETCOV, CONST, BDATA, RDATA, PIXCOR, PIXVAL, CELCOR, CELSUM, CELIKE, CELCLS, PIXCLS, STACK, OPEN, CLOSED, FLDSIZ, FLDLIK, PRTBUF, BUFFER)

Input Arguments:

ICLSCV (VARSZ3, NOPOOL)	R*4 An array dimensioned (channels * (channels + 1)/2) * number of classes, containing the upper half of the triangular covariance matrices for each of the classes used for classification.
VECPRD (NOFET3, NOPOOL)	R*4 An array dimensioned channels * classes containing the vector product of the classes for each channel from the inverse covariance matrix and the determinant of the covariance matrix.
SCAPRD (NOPOOL)	R*4 An array dimensioned "classes" for the scalar product of the mean and covariance matrices for each class.
DETCOV (NOPOOL)	R*4 An array dimensioned "classes" words for the determinant of the covariance matrix for each class.
CONST (NOPOOL)	R*4 An array dimensioned "classes" words containing the constant term used consisting of an expression with determinant and scalar product for each class.
DDATA ((NOFET3*ID(6))/2)	I*2 The array used for reading the byte information of data values from data tape.
RDATA (NVR, CELWTH)	R*4 The array dimensioned "(ID(6)-6) * cell width" words for reading actual data values from tape and calibrating them.

PIXCOR (VARSZ3 * CELSIZ)	R*4 The array dimensioned (channels * (channels + 1)/2) by (cell width)^2 words for storing the cross product terms of the data values of a cell (correlation term).
PIXVAL (NOFET3 *CELSIZ)	R*4 An array dimensioned "channels * (cell width)^2" for storing actual pixel data values for each channel for 1 cell.
CELCOR (VARSZ3)	R*4 An Array dimensioned (channels * (channels + 1))/2 words for saving the cross product of the pixel values.
CELSUM (NOFET3)	R*4 An array dimensioned "channels" words used to store the sum of data values for each pixel in the cell for a given channel number.
CELIKE (NWORD, JPTS)	R*4 An array dimensioned "NWORD by columns of cells" for storing the cell likelihoods or pixel values for each cell in the line.
CELCLS (JPTS)	I*2 An array dimensioned "column" of cells halfwords used to contain the most likely class number of all cells in a given row.
PIXCLS (NWORD, JPTS)	I*2 The array dimensioned NWORD2 by "column" halfwords for storing the class number of each pixel in a given cell for singular cells (equivalent to CELIKE).
STACK (JPTS)	I*2 A stack of field numbers for the cell in the given position for a row of cells.
OPEN (JPTS)	L*1 An array of "column" bytes used as logical flags to store whether or not a particular column of cells has been opened as a field.
CLOSED (JPTS)	L*1 An array of logical values for each column of cells to indicate whether the field for each column has been opened for annexation.

FLDSIZ (JPTS)	I*4 The array of "column" halfwords to store the length of the current field.
FLDCLS (JPTS)	I*2 The array of "column" halfwords to store the length of the current fields as the number of previously annexed rows in this column.
FLDLIK (NOPOOL, JPTS)	R*4 The array of likelihoods dimensioned columns of cells by classes words used to store the likelihood of each field for each class.
PRTBUF (JPTS)	I*2 The output print buffer dimensioned "columns of cells" halfwords.
BUFFER (BUFROZ * JPTS)	I82 The buffer array used for annexation of neighboring rows of cells dimensioned BUFROZ * JPTS.

SECH01 performs processing of an area to be classified. If only an intermediate tape is to be produced, cell information is gathered and written to tape. Otherwise all information is used in annexation and classification, and the classification results are written to the output device. Any maps are also produced.

2. Internal Description

Initially, column headers are printed if any map is to be produced. All buffer pointers and cell arrays are initialized so that the arrays OPEN and CLOSE indicate that no annexation and classification has taken place. Then a loop is entered and executed once for each row of cells (two or more data lines). The routine FILBUF is called to gather statistics on one row of cells, either directly from processing the data tape, or from reading the intermediate tape. If no annexation is to be performed, the information is written to the output device and this loop is executed until this entire area is completed and a return is executed. Otherwise, a series of calls to LIKRAT for a comparison of the likelihood of the cell being annexed to the cell above, left, and right is performed. The cell is annexed to the most likely cell if the comparison with the annexation threshold is successful. This is continued for all cells in the row. When finished, any fields formed from adjoining cells that are closed and no longer continued are classified and then any new fields are opened in that statistics from the initial cell are copied into the array storing field statistics. Then LIKRAT is called to classify closed areas. A print buffer is then filled with symbols for either the cell map or classification map and printed out. Statistics are handled for remaining open fields and the row is written to the output tape. This is continued until the area has been processed.

3. Input Description

Not Applicable.

4. Output Description

The unit PRNTR receives the map and column header produced if a map is to be printed.

The unit OUTPUT is used as the output unit for either intermediate tape - cell information or classified cell lines. These lines follow the format of regular classification records.

5. Supplemental Information

This program uses common blocks GLOCOM and SECCOM.

LARS Program Abstract 371

MODULE IDENTIFICATION

Module Name: GATHER Function Name: ECHO FUNCTIONAL SUPPORT

Purpose: Gather cell statistics information for processing.

System/Language: CMS/FORTRAN

Author: P.D. Alenduff Date: 4/12/77

Latest Revisor: Date:

MODULE ABSTRACT

GATHER computes information about cell statistics for actual classification by SECHO

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GATHER

```
CALL GATHER (RDATA,PIXVAL,PIXCOR,CELSUM,CELCOR,CELWTH,
             VECSIZ,MTXSIZ,NSR,NVR)
```

Input Arguments:

RDATA (NSR,NVR) - R*4 Calibrated data values from data tape with NSR samples and NVR channels.

CELWTH - I*4 Number of data points in width of cell.

VECSIZ - I*4 Number of channels to be used in classification.

MTXSIZ - I*4 Number of entries in the triangular covariance matrix. Equal to (channels * (channels + 1))/2.

NSR - I*4 Number of data samples per line in RDATA.

NVR - I*4 Number of channels of data in RDATA array.

Output Arguments:

PIXVAL (NOFET3, CELWTH*CELWTH) - R*4 An array of actual data values in the cell dimensioned channels by (cell width)^2.

PIXCOR (MTXSIZ, CELWTH*CELWTH) - R*4 Cross product terms for all data points in a cell, dimensioned (cell width)^2 by MTXSIZ.

CELSUM (VECSIZ) - R*4 Sum of data values in the cell for each channel of data (dimensioned channels).

CELCOR (MTXSIZ) - R*4 Correlation matrix for data values in cell consisting of a sum of products of data values in cell.

GATHER is called to return information concerning the data characteristics of one cell of data.

2. Internal Description

GATHER loops for each channel of data and each pixel within the cell to produce the data array returned.

3. Input Description

Not applicable.

4. Output Description

Not applicable.

5. Supplemental Information

Not applicable.

LARS Program Abstract 380

MODULE IDENTIFICATION

Module Name: COMSUP Function Name: COMPARERESULTS

Purpose: Supervisor for the function.

System/Language: CMS/FORTRAN

Author: John Cain Date: 6/1/79

Latest Revisor: Date:

MODULE ABSTRACT

Supervisor for the COMPARERESULTS function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL COMSUP

There are no arguments to COMSUP. It is called from LARSMN when the COMPARERESULTS function is requested. Control returns to LARSMN upon completion of the function.

2. Internal Description

COMSUP receives control from LARSMN to perform the COMPARERESULTS processing. COMSUP calls COMRDR to read and interpret the control cards. Upon return from COMRDR, COMSUP calls CHANGE to finish the processing. Subroutines called by COMSUP: COMRDR
CHANGE

3. Input Description

Not Applicable

4. Output Description

Message numbers are listed below, see User's Manual for text of message.

MESSAGES

INFORMATIONAL

I 26
I 264

5. Supplemental Information

Not Applicable.

6. Flowchart

Not Applicable.

650

LARS Program Abstract 381

MODULE IDENTIFICATION

Module Name: COMCOM Function Name: COMPARERESULTS

Purpose: Block data

System/Language: CMS/FORTRAN

Author: John Cain Date: 6/1/79

Latest Revisor: Date:

MODULE ABSTRACT

This is the BLOCK DATA subroutine for the COMPARERESULTS common block COMCOM.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 382

MODULE IDENTIFICATION

Module Name: COMRDR Function Name: COMPARERESULTS

Purpose: Reads and interprets function control cards.

System/Language: CMS/FORTRAN

Author: John Cain Date: 6/1/79

Latest Revisor: Date:

MODULE ABSTRACT

COMRDR interprets all function control cards for COMPARERESULTS. Checks are made for complete and valid specifications and the proper input-output devices are attached.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL COMRDR (Z,NAME)

Output Arguments:

Z-LOGICAL*1 each element initialized to .FALSE.

Z(i,1,j)=.TRUE. - if class i from the first classification is part of user-defined class j.

Z(m,2,n)=.TRUE. - if class m from the 2nd classification is part of user-defined class n.

NAME-I*4 - contains the names of the user-defined classes.

Listed below are the actions taken when the following control cards are read.

FIRSTRESULTS (SECONDRESULTS)	TAPE - the variable TAPE1 (TAPE2) is set to the given tape number. FILE - the variable FILE1 (FILE2) is set to the given file number. DISK - DISKFG is checked to be sure that the DISK option is not already in effect, the tape and file numbers are checked to be sure that both the DISK option and TAPE option are not being used simultaneously. If they are, then an error message will be printed and the DISK will be used. RESLT1 (RESLT2) is set equal to CLASSR.
NEWRESULTS	TAPE - the variable TAPE3 is set equal to the given tape number. FILE - the variable FILE3 is set equal to the given file number. INIT - the variable INITFG is set equal to 1. DISK - the same checks are made as above in addition to a check to see whether the INIT and DISK option were used simultaneously. DISKFG is set equal to one and RESLT3 is set equal to CLASSR.
BLOCK	RUN - the variable RUNNUM is set equal to the given run number. LINE - STALIN is set equal to the first entry (the starting line of the area to be investigated). LASLIN (last line) is set equal to the second entry and finally LININT (line interval) is set equal to the last entry. COL - same as above where the variables are: STACOL - first entry, LASCOL - second entry and COLINT - final entry.

DATA	A check is made for the presence and validity of all information.
CLASS name	The name given is stored in the array NAME.
FIRST N1, N2,... SECOND M1, M2,...	Using the given class numbers the appropriate locations in the Z(64,2,64) array are set .TRUE. (i.e. if these are the FIRST and SECOND cards for the jth user-defined CLASS, then the following assignments are made for array Z:

```
     Z(N1,1,j) = .TRUE.
     Z(N2,1,j) = .TRUE.
     .
and  .
     Z(M1,2,j) = .TRUE.
     Z(M2,2,j) = .TRUE.
     .
     .
```

2. Internal Description

COMRDR uses the standard card reader logic in using CTLWRD, CTLPRM and IVAL to read and interpret control cards.

COMRDR begins by initializing all flags and arrays that are used to convey control card information. It then goes into a loop of reading and interpreting the input specifications and the BLOCK card. When the DATA card is read COMRDR checks for the presence of all information and its validity. Another loop is entered and the CLASS cards and their corresponding FIRST and SECOND cards are read. The class numbers from the FIRST and SECOND cards are used to set appropriate values in the Z array to a logical .TRUE.

Z(i,1,j)=.TRUE. if the Ith class from the FIRST results file is part of user-defined class j.

Z(k,2,m)=.TRUE. if class k from the SECOND results file is part of user-defined class m.

This loop is exited when an END card is read. Once this card is read, COMRDR calls CHTAPE to mount the specified tapes. If a disk was specified as an input device, COMRDR first checks to be certain both a tape and disk were not specified for a single input. It then reads from the results file to be sure it exists on the disk. If a disk was specified as an output device, checks are made to be sure there is sufficient space for the output results. TSPACE makes a search for a larger disk if necessary. COMRDR finally returns control back to COMSUP. Subroutines called by COMRDR:

CTLWRD	CTLPRM	TSPACE
BCDFIL	CHTAPE	RTMAIN
IVAL	ERPRNT	

3. Input Description

Function control cards for COMPARERESULTS are read via CTLWRD.

4. Output Description

Message numbers are listed below, see User's Manual (vol. 3) for text and explanation of message.

MESSAGES

INFORMATIONAL	ERROR
I0261 I0262	E620-E633

5. Supplemental Information

Not applicable.

6. Flowchart

Not applicable.

LARS Program Abstract 383

MODULE IDENTIFICATION

Module Name: COMPAR   Function Name: COMPARERESULTS

Purpose: Compare 2 lines of classification results

System/Language: CMS/FORTRAN

Author: Susan Schwingendorf   Date: 3/28/79

Latest Revisor: ______   Date: ______

MODULE ABSTRACT

COMPAR compares two lines of classification results (presumably from two different classifications which are registered to each other) against user defined change classes in a logical array, and writes the output class number in the output vector.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL COMPAR (NCOLS,NCLASS,Z,BUFF1,BUFF2,BUFF3)

Input Arguments:

NCOLS - INTEGER*4, the number of columns of classified data.

NCLASS - INTEGER*4, the number of classes defined by the user in array Z.

Z - LOGICAL*1 (64, 2, 64) array containing user defined change classes. (Initialized to .FALSE.)

Z(I,1,K) = .TRUE.
Z(J,2,K) = .TRUE. means a point in class

I from classification 1 (BUFF1) and in class J on classification 2 (BUFF2) should be assinged to class K in BUFF3.

BUFF1 - LOGICAL*1 (2*NCOLS + 4) vector containing classified data from first classification. First full word is line number. Then the second byte of each halfword contains the next class number.

BUFF2 - LOGICAL*1 (2*NCOLS + 4) vector containing classified data from the second classification. First full word (4 bytes) is the line number. Then the second byte of each halfword contains the next class number.

Output Arguments:

BUFF3 - LOGICAL*1 (2*NCOLS + 4) vector of change classes for this line. The first full word contains the line number. Then the second byte of each halfword contains the assigned change class number.

2. Internal Description

The line number is written in the first word of BUFF3. The next class number is then extracted from BUFF1 and BUFF2 and assigned to integer variables CLASS1 and CLASS2. A loop through the logical array Z determines which output class to assign this point to. If Z (CLASS1,1,J) and Z (CLASS2,2,J) are true, then the point is assigned to class J. If it belongs to none of the defined output classes, then it is assigned a class number NCLASS+1. The output class numbers are written in BUFF3.

3. Input Description

Not Applicable.

4. Output Description

Not Applicable.

5. Supplemental Information

Not Applicable.

6. Flowchart

Not Applicable.

LARS Program Abstract 384

MODULE IDENTIFICATION

Module Name: CHANGE Function Name: COMPARERESULTS

Purpose: Compares two classification results files and outputs the compared results.

System/Language: CMS/Fortran

Author: John Cain Date: 6/1/79

Latest Revisor: Date:

MODULE ABSTRACT

CHANGE is the main subroutine for COMPARERESULTS. It reads from two input tapes (or one disk and one tape), calls COMPAR, then outputs the data in standard LARSYS classification results file format to tape or disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CHANGE

CALL CHANGE(Z,NAME)

Input Arguments:

Z-Logical*1 - Z(i,1,j)=.TRUE. if class i from the first classification is part of user-defined class j.

Z(m,2,n)=.TRUE. if class m from the 2nd classification is part of user-defined class n.

NAME - I*4. - Contains the names of the user-defined classes.

Output Arguments:

Not Applicable.

2. Internal Description

CHANGE first reads the file numbers from the input tapes, and the tape numbers passed through the common block, and creates a code that takes the place of the CLASSIFICATION STUDY number. The code format is the first tpae and file numbers followed by the second tape and file numbers. CHANGE then reads the area identification record (record type 5) from both input sources and checks to see whether they are valid for the given BLOCK CARD; if not, appropriate error messages are printed and the function is terminated. Record types 1-5 are written to the output tape (DISK). The inputs are positioned to the correct line number and shifted to the correct column number. CHANGE then calls COMPAR to determine which class each point belongs to and this information is used to create file type 6. Finally record types 7 and 8 are written and control is returned to COMSUP. If the output device is a tape, then a final record type 1 and END OF FILE Mark are written before returning to the supervisor. Subroutines called by CHANGE:

COMPAR
RTMAIN
TAPOP

3. Input Description

Record types 1, 5, 6 of the LARSYS classification results files are read from the two input devices, RESLT1 and RESLT2. One of these may be a disk (DSRN CLASSR). Tape drives 181 (CPYOUT) and 182 (SCNDTP defined in COMCOM) are used as inputs.

4. Output Description

The output device RESLT3 initially has a DSRN of MAPTAP. If a disk is used (only if one is not used for input), the DSRN is changed to CLASSR. Tape drive 180 is used for output to facilitate the run of PRINTRESULTS on the output data immediately after the COMPARE-RESULTS run. The output is a classification results file in standard LARSYS format.

Message numbers are listed below, see User's Manual for text and explanation of message.

MESSAGES

INFORMATIONAL	ERROR
I0263	E634-E638

5. Supplemental Information

Not Applicable.

6. Flowchart

Not Applicable.

LARS Program Abstract 385

MODULE IDENTIFICATION

Module Name: CHTAPE Function Name: COMPARERESULTS

Purpose: Mounts and positions results tapes

System/Language: CMS/FORTRAN

Author: E.M. Rodd Date: 9/5/72

Latest Revisor: J. Cain Date: 6/1/79

MODULE ABSTRACT

CHTAPE mounts and positions the results tape (or a tape to be used as output for copying results files.)

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CHTAPE

CALL CHTAPE (RQTAPE,RQFILE,MODE,UNIT)

Input Arguements

RQTAPE - I*4. Tape number of requested tape. A tape number of 0 is a request for a scratch tape.

RQFILE - I*4. File number of requested file. If RQFILE is = 0, then the tape will be initialized by writing a record type 1 on the results tape with filetype= 0.

MODE - I*4. Flag indicating usage of CHTAPE. MODE = -1 indicates CHTAPE has been called to mount and position a tape to be used for copying results files onto. MODE = 0 indicates that a results tape is being mounted for reading a results file. In this case, the tape is mounted ring out. Also, if MODE = 0, RQFILE = 0 is invalid and will cause an error when an attempt is made to write on the tape. MODE = 1 indicates a tape is being mounted for writing a new results file (or continuing a suspended classificaiton). Since the unit value is passed in the call, Mode(1) = Mode(-1).

UNIT - I*4. DSRN of tape being mounted.

Output Arguments

RQTAPE - I*4. When MODE = 0, set to -1 if requested tape file was full and user decided to use disk for results. Otherwise, remains unchanged.

RQFILE - I*4. When MODE = 1, set to -1 if requested tape file was full and user decided to use disk for results. Otherwise, sends back current file position of tape.

CHTAPE checks the validity of the tape by reading the record type 1 from the tape and verifying the tape and file number as well as checking for the correct type of file. Any attempt to overwrite an existing file causes CHTAPE to ask the user (via the typewriter) if he wishes to overwrite the file, respecify a new results card, or terminate the function. Note, however, that if a request has been made to initialize a tape, no checking is performed on previous contents.

2. Internal Description

See Output Description. Subroutine called by CHTAPE:

TAPOP	RINGIN	IVAL
MOUNT	CTLWRD	ERPRNT
CPFUNC	CTLPRM	RTMAIN

3. Input Description

The record type 1 of the results tape is read for each file up to and including the file needed. That is, if file 4 is requested the record type 1 is read from files 1-4.

4. Output Description

The following information messages are issued under the circumstances listed. The term filetype means the filetype code from record type 1 of results file (the program uses variable CHECK for this number).

I0042 is typed when a tape has been mounted and before CHTAPE positions it. This message is not typed when the tape is being initialized or when the correct tape number was already mounted.

I0043 is typed when MODE = ± 1 and filetype of the requested file = 0.

I0044 is typed when MODE = ± 1 and filetype of the requested file = 1 and the restart flag from GLOCOM (RESTRT) is not = 1.

I0045 is typed when the tape is correctly positioned. This is not typed when initializing a tape.

After I0043 and I0044, the user is asked whether he wishes to overwrite the file, respecify a new results card with a new tape and/or file or disk option, or terminate the function.

I0100 is typed to allow entry of the new results card. This occurs when the user requests to respecify the results card.

I0101 is typed to confirm usage of disk for results and occurs whenever disk is specified on the results card.

The following error messages are typed under the conditions listed.

E361 is written when the tape is being filed forward and a file is encountered with filetype other than zero before the requested file is reached and MODE = 0.

E362 is written when the circumstance for E361 occurs and MODE = 1. It is also written when MODE = 1 and the filetype of the file requested is = -1.

E363 is written if the RESTRT flag is = 1 and the filetype of the requested file is not = 1.

E364 is written when MODE = 1 and the filetype of the file requested = 1.

E365 is written when an EOF is read on the results file. This should never occur with valid results files.

For message texts refer to the User's Manual.

5. Supplemental Information

This section deals with the handling of tapes by CHTAPE.

Input:

If a tape is mounted on the device and it is the incorrect tape number (as noted from the appropriate status words in GLOCOM), TOPRU is called to unload the tape before the correct tape is mounted. If the correct tape is mounted, CHTAPE will check for the ring in if MODE = ±1. If the ring is not in, the tape in unloaded and MOUNT is called to mount the tape with the ring in. If the correct tape is mounted, CHTAPE assumes that the file number (as recorded in GLOCOM) is correct and moves the tape backwards or forwards to find the requested file. CHTAPE is a modified version of MMTAPE.

Output:

The tape is mounted with ring in for MODE = ±1 and with ring out for MODE = 0.

The tape is left positioned at the beginning of the requested file. When the tape is initialized a TOPRW is used to do this.

6. Flowchart

Not applicable.

LARS Program Abstract 0390

MODULE IDENTIFICATION

Module Name: RATSUP   Function Name: RATIO

Purpose: Supervisor for the RATSUP load module.

System/Language: CMS/FORTRAN

Author: Randy Alan Culp   Date: 5/7/76

Latest Revisor: Date:

MODULE ABSTRACT

Supervisor for the RATIO processor.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

RATSUP

CALL RATSUP

This routine is called within LARSYS by the main module, LARSMN, when a *RATIO card is detected within the input control cards.

2. Internal Description

RATSUP:

1) Informs the user that the ratio function has been requested.

2) Calls RATRDR to read and interpret the function control cards.

3) If it is not a checkout run and the statistics deck for the ratio calculations is to be read from tape:
   a) Reads the first two records,
   b) Calculates the amount of core needed for moving the statistics deck from tape to disk,
   c) Calls RATTAP to do the moving.

4) Calls STAT(see LARS Abstract 135) to get the size of the statistics deck and to move it from cards to disk if necessary.

5) Return if the user has requested the noratios option, desiring only the moving of the statistics deck to disk.

6) Check that the statistics deck is large enough for default ratio calculation if the default is used. If not, writes an error message and returns.

7) Calculates the amount of core needed by REDSTA (see LARS Abstract 134) to read the statistics deck and calls it to read the statistics into memory.

8) Informs the user that ratios calculation is beginning.

9) Calls RATIO if the user requested the default calculations of the first two channels over the next two channels. (Usually visible channels over infrared).

10) Calls RATIOS if the user specified their own combination of visible and infrared channels.

11) Writes a function complete message and returns to the caller.

Subroutines called by RATSUP:

RATRDR
ERPRNT
TOPRF
RATTAP
STAT
REDSTA
RATIO
RATIOS

Commons used in RATSUP:

RATCOM
GLOCOM

3. Input Description

Not Applicable

4. Output Description

The information and error messages described under the Internal Description section are written to the terminal. The error message is also written to the printer.

5. Supplemental Information

See the LARSYS System Manual for a description of supervisor module requirement.

6. Flowchart

Not Supplied

LARS Program Abstract 0391

MODULE IDENTIFICATION

Module Name: RATCOM Function Name: RATIO

Purpose: Block data for RATSUP load module

System/Language: CMS/FORTRAN

Author: Randy Alan Culp Date: 5/7/76

Latest Revisor: Date:

MODULE ABSTRACT

This is the Block data subroutine for the RATIO processor. The RATCOM common block is contained in RATSUP FORTRAN, RATRDR FORTRAN, RATTAP FORTRAN, and RATIO FORTRAN.

669

LARS Program Abstract 0392

MODULE IDENTIFICATION

Module Name: RATRDR Function Name: RATIO

Purpose: Reads function control cards for RATSUP load module

System/Language: CMS/FORTRAN

Author: Randy Alan Culp Date: 5/7/76

Latest Revisor: Date:

MODULE ABSTRACT

RATRDR interprets all function control cards for the RATIO processor. Checks are made for complete and valid specifications.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

RATRDR

CALL RATRDR

This routine interprets the function control cards and puts the results into variables located in RATCOM. All results are validated and any required tapes are mounted.

2. Internal Description

After initialization of variables, CTLWRD is called to read and interpret the key words. An unexpected end of file for the control card input results in ERPRNT being called to terminate execution. After CTLWRD has determined the key word, a branch is made to sections of code to further interpret each of the possible cards. CTLPRM and IVAL are used to assist with this interpretation. After the END card is detected the user's requests are checked for completeness and validity.

Once all inputs are complete MMTAPE is called to mount any required classification results tapes. The user's requests are written on the line printer and control is returned to the caller.

Subroutines called by RESRDR:

TSTREQ
CTLWRD
ERPRNT
CTLPRM
IVAL
MMTAPE
RTMAIN

Commons used in RATRDR:

GLOCOM
RATCOM

3. Input Description

RATRDR does not actually perform any reading operations. It does invoke CTLWRD which performs reads to the control card input stream (card reader or typewriter). In addition, MMTAPE performs the mounting of the classification results tape.

4. Output Description

File name - Error Messages
DSRN - PRNTR and TYPEWR
Device type - Printer and temminal
Description - Message numbers are listed below, for text see LARSYS User's Manual.

ERROR MESSAGES

431
581
582
588
589
590

Also one that tells the user that he has specified only one option and asks for a card for the other.

File name - Request Selection Summary
DSRN - PRNTR
Device type - Printer
Description - List of all control cards input and options requested.

5. Supplemental Information

See LARSYS System Manual for a description of control card reading routines.

6. Flowchart

Not Applicable

LARS Program Abstract 0393

MODULE IDENTIFICATION

Module Name: RATTAP Function Name: RATIO

Purpose: To move a statistics deck from tape to disk.

System/Language: CMS/FORTRAN

Author: Randy Alan Culp Date: 5/7/76

Latest Revisor: Date:

MODULE ABSTRACT

RATTAP moves a statistics deck from a classification results tape file to disk and optionally prints the tape classified file information.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

RATTAP

```
CALL RATTAP(FRQUP, FRQLW, POLNAM, POLPTR, POLSTK, COVMTX,
            AVEMTX, BUF)
```

Input Arguments:

FRQUP - REAL*4, Array of upper wavelength band values for each channel.

FRQLW - REAL*4, Array of lower wavelength band values for each channel.

POLNAM - REAL*8, Array of names assigned to each pool used in classification.

POLPTR - INTEGER*2, A 2 by i matrix where i = the number of pools. POLPTR(1,i) = the number of classes in pool i, and POLPTR(2,i) = the location of the first class for the pool in POLSTK.

POLSTK - INTEGER*2, Array of class numbers of all classes in the statistics deck grouped by classification pool.

COVMTX - REAL*4, Lower half covariance matrices for each pool.

AVEMTX - REAL*4, Mean vector for each pool.

BUF - INTEGER*2, Buffer array to read in each line classified.

RATTAP is called by RATSUP to move the statistics file from a classification results tape if requested by a user control card.

2. Internal Description

RATTAP reads the first two records from the classification results file and prints header information on the line printer concerning the tape and file used. If the user requested LIST, then channel, calibration and class information is printed. The statistics deck is then moved from tape to disk. If the user did not request LIST, the tape is repositioned to the beginning of the tape file and return is made to the caller. If LIST was requested, the classification information is read from tape and written to the printer before the tape is repositioned and return made.

Subroutines called by RATTAP:

TOPRF
ERPRNT

Commons used in RATTAP

GLOCOM
RATCOM

3. Input Description

File name - Classification Results File
DSRN - MAPTAP
Devise type - Tapes
Description - See LARSYS System Manual

4. Output Description

File name - Error Messages
DSRN - PRNTR and TYPEWR
Device type - Printer and Terminal
Description - Error message numbers are 592, 593, and 594, for text use LARSYS User's Manual

File name - Information Message
DSRN - TYPEWR
Device type - Terminal
Description - Moving of Statistics Complete

File name - Results File Printer Listing
DSRN - PRNTR
Device type - Printer
Description - Classification results file information. For examples see LARSYS User's Manual

File name - Statistics deck
DSRN - SDATA
Device type - Disk
Description - See LARSYS System Manual

5. Supplemental Information

Not Applicable

6. Flowchart

Not Supplied

LARS Program Abstract 0394

MODULE IDENTIFICATION

Module Name: RATIO Function Name: RATIO

Purpose: Calculates and prints ratios using a LARSYS statistic deck.

System/Language: CMS/FORTRAN

Author: Randy Alan Culp Date: 5/7/76

Latest Revisor: Date:

MODULE ABSTRACT

RATIO receives from RATSUP FORTRAN the means and variances from a LARSYS statistics deck from which it calculates and prints the ratios of two visual channels and two infrared channels.

Entry RATIOS allows a variable number of channels for both the visual and infrared.

1. Module Usage

RATIO

CALL RATIO(OUTPUT, CLSNAM, AVAR, COVAR)

Input Argument

OUTPUT - INTEGER*4, DSRN of unit for printer length output.

CLSNAM - INTEGER*4, Array of names of the classes used in calculating statistics.

AVAR - REAL*4, Array containing means organized according to channel and pool.

COVAR - REAL*4, Array containing the lower half covarience matrix for each pool.

RATIOS

CALL RATIOS(OUTPUT, CLSNAM, AVAR, COVAR)

All arguments the same as RATIO.

RATIO and RATIOS are used to calculate and print the ratio of visible over infrared channels in a LARSYS statistics deck. RATIO is called by RATSUP if the user requests default calculations of the first two channels over the second two, which represent the two visible and two infrared channels for most statistics decks generated from satellite data. RATIOS is called if the user has specified some combination of channels for both the visible and infrared. Up to 60 channels can be used for each, thus allowing for weighting since channels can be specified more than once and in both categories, although they appear only once in the statistics deck. RATIOS only writes the ratios organized according to class while RATIO, in addition, sorts and writes them according to the ratio magnitude and the mean total magnitude, along with the variances organized according to class.

2. Internal Description

RATIO

For each class (or pool) a sum is made of the visible and infrared channel means, along with their totals, ratios, and percent of total. Their values are written along with the class name and means, according to class. An array of printer subscripts is then computed so the output can be repeated according to the total magnitude and the ratio magnitude.

The variances from the covariance matrix are written according to class along with the number of pools in each, after which return is made to the caller.

RATIOS

Calculations are made for each class as in RATIO except the users requested channels are used instead of the first four. The output contains only the totals, ratios, percentages, and class names.

RATIO contains the RATCOM common.

3. Input Description

Not Applicable

4. Output Description

The output described under the Module Usage and Internal Description sections are written to unit OUTPUT along with appropriate header information.

5. Supplemental Information

Not Applicable

6. Flowchart

Not Supplied

LARS Program Abstract 400

MODULE IDENTIFICATION

Module Name: BIPSUP Function Name: BIPLOT

Purpose: Supervisor for *BIPLOT Routines

System/Language: CMS/FORTRAN

Author: Mike Kaser Date: 05/10/76

Latest Revisor: B. J. Davis Date: 10/12/77

MODULE ABSTRACT

This program is the supervisor for the *BIPLOT (Bispectral Plotting) Function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

679

LARS Program Abstract 401

MODULE IDENTIFICATION

Module Name: BIPRDR Function Name: BIPLOT

Purpose: Read Control cards for the *BIPLOT Function

System/Language: CMS/FORTRAN

Author: Mike Kaser Date: 05/10/76

Latest Revisor: B. J. Davis Date: 10/12/77

MODULE ABSTRACT

This program reads the control cards from the input device for the *BIPLOT function.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

BIPLTR 680

LARS Program Abstract 402

MODULE IDENTIFICATION

Module Name: BIPLTR Function Name: BIPLOT

Purpose: Plot control

System/Language: CMS/FORTRAN

Author: Mike Kaser Date: 05/10/76

Latest Revisor: B.J. Davis Date: 10/12/77

MODULE ABSTRACT

This program prints feature, and class information and calls the appropriate plotting routine.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 403

MODULE IDENTIFICATION

Module Name: BIPELL Function Name: BIPLOT

Purpose: Plots Ellipsoids

System/Language: CMS/FORTRAN

Author: Mike Kaser Date: 05/10/76

Latest Revisor: B.J. Davis Date: 10/12/77

MODULE ABSTRACT

This program calculates the boundaries of an ellipsoid and fills it in.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 404

MODULE IDENTIFICATION

Module Name: BIPCLA Function Name: BIPLOT

Purpose: Classifies each pixel in classes

System/Language: CMS/FORTRAN

Author: Mike Kaser Date: 05/10/76

Latest Revisor: B.J. Davis Date: 10/12/77

MODULE ABSTRACT

This program is the classifier for *BIPLOT.

Entry Point (Function) LIM finds limits for routine.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 405

MODULE IDENTIFICATION

Module Name: BIPMEN Function Name: BIPLOT

Purpose: Plots Means

System/Language: CMS/FORTRAN

Author: Mike Kaser Date: 05/21/76

Latest Revisor: B.J. Davis Date: 10/12/77

MODULE ABSTRACT

This subroutine plots class means on a grid. This program includes separability code.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 406

MODULE IDENTIFICATION

Module Name: BIPDIV Function Name: BIPLOT

Purpose: Compute Separability

System/Language: CMS/FORTRAN

Author: R.L. Kehig Date: 10/16/74

Latest Revisor: M.D. Kaser Date: 06/01/76

MODULE ABSTRACT

This program computes separability using divergence.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

LARS Program Abstract 407

MODULE IDENTIFICATION

Module Name: BIPCOM Function Name: BIPLOT

Purpose: Common block

System/Language: CMS/FORTRAN

Author: Mike Kaser Date: 05/21/76

Latest Revisor: B.J. Davis Date: 10/12/77

MODULE ABSTRACT

Block data common block for *BIPLOT.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906