Space Science and Engineering Center
University of Wisconsin-Madison

DOCUMENTATION OF THE VAS DATA PROCESSING SOFTWARE

A REPORT from the

Cooperative
Institute for
Meteorological
Satellite
Studies

DOCUMENTATION OF THE

VAS DATA PROCESSING SOFTWARE

by

James P. Nelson, III

Cooperative Institute for Meteorological Satellite Studies
Space Science and Engineering Center
University of Wisconsin
1225 West Dayton Street
Madison, Wisconsin 53706

November 1984

Preface

This manual summarizes the main and auxiliary software used to process VAS (VISSR Atmospheric Sounder) retrievals at the Space Science and Engineering Center at the University of Wisconsin-Madison. Since the software is being developed in a research laboratory, it is constantly undergoing change to upgrade and improve it. The programs described herein, with the exception of VTPZ, are versions from May 1, 1984. The version of VTPZ contained within the text is valid for May 3, 1984.

If there are any questions or suggestions, please mail them to:

Mr. James P. Nelson, III
Space Science and Engineering Center
University of Wisconsin
1225 West Dayton Street
Madison, Wisconsin 53706

TABLE OF CONTENTS

Introduction

A large volume of software has been developed and written at the University of Wisconsin-Madison for the purpose of preparing and processing VAS radiance data and then retrieving profiles of atmospheric temperature and moisture. A person new to the Man-computer Interactive Data Access System (McIDAS) and/or VAS retrieval processing would have a difficult time assimilating and becoming familiar with the software without some type of guidance. This document is intended to alleviate this problem by explaining how the VAS software is operated in terms of McIDAS commands, program flowcharts and program descriptions. The manual does not explain all of the VAS software; rather, it covers only those programs of major importance to the retrieval process.

The text consists of five chapters and six appendices. The first chapter gives a broad overview of the retrieval process. The second chapter presents each program in terms of modular flowcharts. The final three chapters deal with individual programs within the retrieval process itself. These chapters involve three levels of importance, ranging from the most vital software to programs which are rarely used or have been deactivated. Chapter 3 details the software used on an every day basis, while Chapters 4 and 5 contain descriptions of several lesser used retrieval support and other discontinued programs.

The final section of the reference contains six appendices. Appendix I contains descriptions of different McIDAS terms used in the manual, such as sounder area, MD file, etc. Appendix II describes constructs which exist in one type of flow chart contained within the manual. Appendices III and IV contain lists of all program subroutines and functions, including a description of what each subroutine and function does. The fifth appendix describes program code length, number of accesses to storage (I-O), etc., for each program. Finally, Appendix VI is comprised of VAS retrieval software reproductions.

CHAPTER 1

General Retrieval Outline

In this chapter, a general summary of the VAS retrieval process is presented. Included are both a description of the steps undertaken when processing an area of retrievals in real time and a flowchart outlining these steps, as well as a corresponding series of sample processing commands. The flowchart follows directly on the heels of the retrieval process description, and provides a compact summary of the retrieval scheme.

Optional programs within the flowchart are denoted as such. Otherwise, if there is no attendant "optional" note, the user should assume that a given program is required for the retrieval process. In addition, the text description will differentiate between general McIDAS (system) programs and VAS retrieval processing programs by labelling the system programs as such. Furthermore, text that relates to a given sample processing command will be denoted by the number of the corresponding command (in parenthesis). Since there are a total of 30 commands, there will be 30 corresponding numbers located at the proper locations within the text.

It should be noted that although this discussion deals with the processing of an area of retrievals, only minor modifications in the keyin of retrieval program VTPZ are required to process only a single retrieval. In addition, this discussion assumes the user has a McIDAS terminal complete with a television (video) monitor. Again, only minor changes in program commands are needed to generate retrievals without a video monitor (known as the "auto" processing mode). In order to process retrievals in this mode, however, the user must know where VAS data exist in terms of the sounder area's NW and SE corner latitude/longitude coordinates. Any differences between the general discussion given here and the AUTO mode will be duly noted in the appropriate locations within the sample processing commands. Finally, to get help with any program when examining different options to the ones given here, simply key in "HELP program name," or examine the program within the McIDAS User's Manual.

Initially, the radiance data for the retrieval(s) the user wishes to process must be listed using system program LA and then (optional, but highly advisable) loaded into one or more of the user's sounder area(s) using system program AA (1). Following this, system program DF is used to display the band 8 (window) digital radiance data for the given area in a specified image frame on the television (video) monitor (2). In addition, it is also possible to plot a satellite-projection map over the image using system program IC (3). After the band 8 image has been displayed, the sounder area corresponding to the radiance data just displayed must be pointed at using VPVA (4). This has the effect of telling the system what radiance data it is to use for the retrieval processing. Then, as a quick check of the data quality, one can use VDVA, which can be keyed in to list brightness temperatures at the cursor location for all twelve bands used in the retrieval process (5). (Note: band 11 is usually not used, due to excessive noise, so its brightness temperature will be displayed as 999999).

Up to this point, we have dealt only with the raw VAS radiance data. Retrieval processing of VAS radiance information also requires both surface data (if the surface data option is being used) and upper air guess data. This data is stored in both Meteorology Data (MD) files and gridfiles. In addition, the user must have a retrieval MD file to store the results of the retrieval processing, as well as a "scratch" gridfile for temporary storage of grids of retrieval parameters produced by program BNVA. One should check to see if these files exist using system programs MDU and IGU (6). If the files don't exist, they can be created using the same system programs (7). After the files have been made, the upper air guess data is loaded into the upper air guess gridfile using system program NMCU (8). At this point, the following files should exist: sounder area (loaded by AA), surface data MD file (schema: RSVC; for 1000 mb heights, sea level temperatures and surface dewpoint depressions; no data yet), upper air guess MD file (schema: VGSS; no data yet), retrieval MD file (schema: VRET; no data yet), upper air guess gridfile (loaded by NMCU), surface gridfile (no data yet), and the scratch gridfile (no data yet).

The next required step is IDVA, whose function is to initialize the VASTEXT file and the retrieval MD file row header (9). The VASTEXT file can be envisioned as a bookkeeping file whose purpose is to keep track of many different retrieval parameters, such as retrieval MD file number, sounder area, retrieval type, etc. The VASTEXT file contents can be displayed on the CRT at any time by using program LOVA (10).

The next steps involve IGU again and system program IGG. IGU is used to set the grid file pointer to the upper air guess gridfile (11), and IGG is used subsequently to list the grids contained within this gridfile (12). This step is necessary to pick out the grids containing the most suitable guess for the sounder area being processed. Usually, the most suitable guess will consist of the grids closest in time to the sounder area you wish to process. Finally, the gridded guess data is reformatted from the guess gridfile into the upper air guess MD file via GSVA (13). In other words, values from numerous grids (up to 22, consisting of 15 temperature grids, 6 dewpoint grids, and one 1000 mb height grid) for a given gridpoint are stored at a certain row and column coordinate in the guess MD file. This completes the upper air guess data processing.

The next stage of the retrieval processing scheme involves the surface data. This data is prepared using programs CSVA and SRVA. CSVA calculates 1000 mb heights, sea level temperatures and surface dewpoint depressions for each station and places the results in the designated surface MD file (14). Then, SRVA takes this data, generates a grid of one of the parameters, and places the results in a grid location within a designated surface gridfile (15).

The results of the surface data grid analysis can be examined by first setting the gridfile pointer to the number specified in the SRVA command using system program IGU (16), and then using system program IGTV to contour the desired parameter on the video screen (17). If bad analyses of any or all three surface parameters result, the user should first clear the graphics screen and then re-plot the satellite projection map over the image (using system programs EG and IC), and then go back and plot the surface data with PLVA (18). Then, unacceptable reports can be deleted with XRVA (19). Note the MDU SET before the execution of XRVA. Following this, the process from SRVA forward must be done again to generate corrected grids of any or all of the three surface parameters (20).

With the completion of the surface analyses, it is now possible to do the actual VAS retrievals. However, before the retrievals are made, set the "NO. RETRIEVALS=" value in the VASTEXT file to 0 by using program SPVA (21). This will ensure that the correct number of retrievals about to be performed is stored in the VASTEXT file. Then, the retrievals can be made by executing program VTPZ (22). This program generates profiles of temperature and moisture, and also creates an image of total precipitable water vapor. Note the example of the VASTEXT file as it stands after the area of retrievals has been processed.

Up to this point, it has been possible to carry out the retrieval processing via either a video or non-video terminal. However, since the emphasis from now on lies mainly with visual retrieval editing, only a video terminal should be used.

After the retrievals have been generated, various retrieval parameters (such as height, temperature, etc.) can be viewed on the video screen using program PLVA (23). Then, before grids of retrieval parameters are generated by using program BNVA, IGU should be executed again to set the gridfile pointer to wherever the user desires the results of BNVA to be placed (24). Usually, the retrieval parameter grids are stored in a different gridfile (the scratch gridfile) than either the first guess or surface gridfiles. At this point, BNVA is executed, with the results being stored in the scratch gridfile (25).

Since one of the statements outputted by BNVA indicates into which grid of the gridfile the analysis was stored, IGTV can be used again in the same fashion as with the surface data to contour various retrieval data grids (26). Poor analyses can then be corrected in the same manner as the surface data analyses (EG, IC, PLVA, MDU and XRVA) (27). If some deletions are done, the user must repeat the steps from BNVA (step 25) onward to assess how the deletions affect the gridded contours (28).

Finally, VAS gradient winds can be produced and plotted using GWVA (together with height grid(s) generated previously by BNVA) (29). Note that the grid number(s) in which the gridded height results of BNVA were placed WILL HAVE CHANGED if any deletions were done (see step 28 above). This is so because BNVA will place updated retrieval grids (after deletions have been done) in positions after the original unacceptable grids. (In the sample processing commands associated with this discussion, it was assumed that no deleting was necessary and that therefore the 500 mb height grid was grid number 1.)

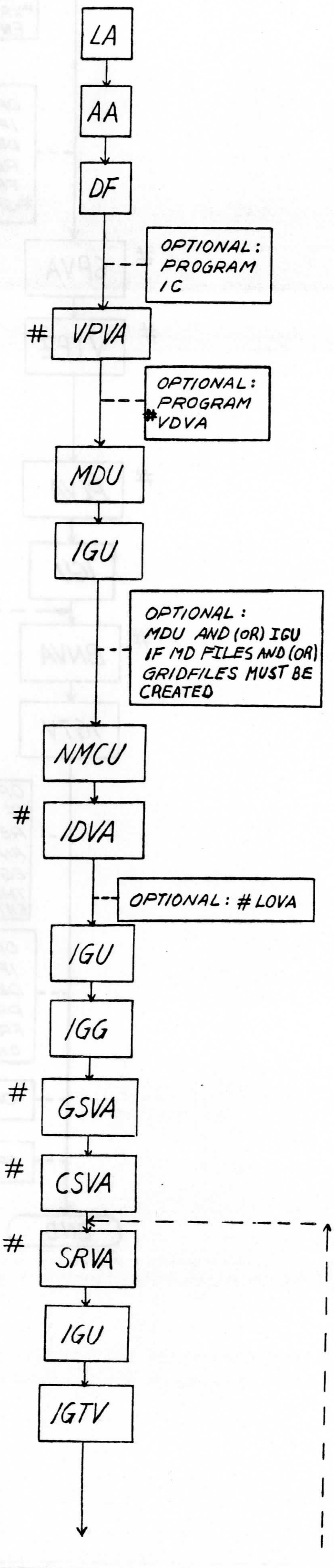
Lastly, if retrievals from a different time period must also be processed, a more up to date (closer in time) first guess can be generated using the just-completed retrievals via program UGVA

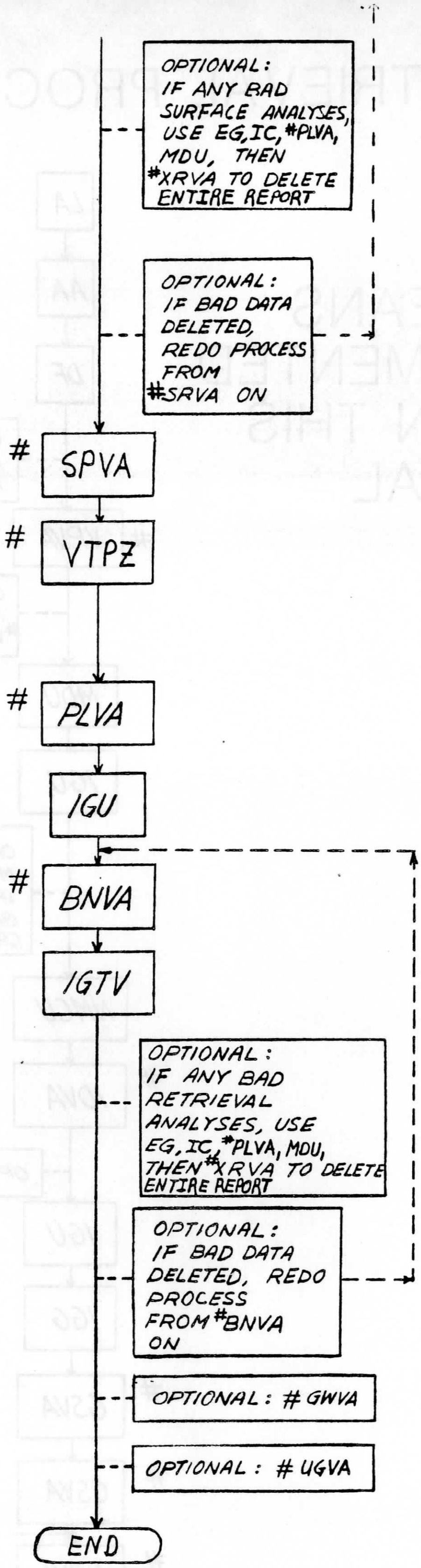
(30), and the entire procedure begins anew.

On the pages following the flowchart, a sample of McIDAS keyins for the processing of a typical retrieval area is presented. These commands correspond directly to the above text.

VAS RETRIEVAL PROCESS FLOWCHART

"#" MEANS DOCUMENTED WITHIN THIS MANUAL




OPTIONAL: IF ANY BAD SURFACE ANALYSES, USE EG,IC,#PLVA, MDU, THEN #XRVA TO DELETE ENTIRE REPORT
OPTIONAL: IF BAD DATA DELETED, REDO PROCESS FROM #SRVA ON
SPVA
VTPZ
PLVA
IGU
BNVA
IGTV
OPTIONAL: IF ANY BAD RETRIEVAL ANALYSES, USE EG,IC,#PLVA, MDU, THEN #XRVA TO DELETE ENTIRE REPORT
OPTIONAL: IF BAD DATA DELETED, REDO PROCESS FROM #BNVA ON
OPTIONAL: #GWVA
OPTIONAL: #UGVA
END

Summary of Files Used

Note: assume the user has been assigned sounder areas 4176-4191, MD files 4176-4191, and gridfiles 4176-4191.

A.) sounder area (data at 1248 GMT, 21 MAY 84): 4190

B.) digital area for storage of total precipitable water vapor image (created by retrieval program VTPZ): 4191

C.) surface data MD file (schema: RSVC; containing Z100, TSL and DD): 4183

D.) upper air guess MD file (schema: VGSS): 4184

E.) retrieval MD file (schema: VRET): 4185

F.) upper air guess gridfile: 4190

G.) surface grid file (containing grids of Z100, TSL and DD): 4189

H.) scratch gridfile for temporary storage of retrieval grids: 4191

Command Sequence

1.) Locate, and move radiance data:

LA 1300 1310 (lists GOES-EAST realtime VAS sounder areas)

then:

AA 1305 4190 ASIZE=ALL (assume area 1305 contains VAS radiance data from 1248 GMT, 21 MAY 84)

2.) Load digital sounder area into given image frame on television monitor: (*** in AUTO mode, numbers 2 and 3 are not done ***)

DF 4190 1 EC 35 90 2

3.) Draw map over image (satellite projection):

IC

4.) Point to correct sounder area:

VPVA 4190

5.) Display brightness temperatures at cursor location:

VDVA B

6.) Check to see if necessary MD and gridfiles exist:

MDU LIST 4176 4191

IGU LIST 4176 4191

7.) Create MD and grid files (if the files do not exist):

MDU MAKE 4183 RSVC 1 X X 84142 "SFC DATA 12GMT 21 MAY 84

MDU MAKE 4184 VGSS 1 X X 84142 "U/AIR GUESS 00GMT 21 MAY 84

MDU MAKE 4185 VRET 1 X X 84142 "VAS RET. 1248 GMT 21 MAY 84

IGU MAKE 4189 "EDITED SFC DATA 12GMT 21 MAY 84

IGU MAKE 4190 "LFM FROM 84 05 21 00

IGU MAKE 4191 "SCRATCH GRIDFILE FOR RET. GRIDS

Note: Text labels after quotation marks are optional (but recommended) and only for the user's benefit.

8.) Load current upper air guess into guess gridfile:

NMCU 0 4190 FILE=LFM00Z "LFM GSS FROM 00GMT 21 MAY 84

9.) Initialize VASTEXT file and row header of retrieval MD file for displayed image:

IDVA 4185 1

(*** in AUTO mode, IDVA 4185 1 45108 35080 AUTO=1, as an example, having specified the NW and SE corners to be covered as 35 to 45 degrees N latitude and 80 to 108 degrees W longitude ***)

10.) Check VASTEXT file contents:

LOVA

11.) Set gridfile number to upper air guess gridfile:

IGU SET 4190

12.) List grids in gridfile 4190:

IGG LIST Note: grid 33 usually starts sequence of 12-hour forecast grids.

13.) Reformat guess from guess gridfile 4190 into guess MD file 4184, for the 12-hour forecast sequence, starting at grid 33:

GSVA 4190 33 4184

14.) Calculate Z100, TSL and DD for each station in real time surface hourly observation MD file (one of MD files 1-10,schema = SVCA), and load into surface MD file (schema = RSVC) 4183:

CSVA 4183 Note: hour of surface data will be 12 GMT (truncation of satellite observation time) and, thus, row 12 in MD file 4183 will be used to store the data.

15.) Create one grid each of Z100, TSL and DD (data taken from surface MD file 4183) and store in grid positions 1-3 in surface gridfile 4189:

SRVA Z100 4189

SRVA TSL 4189

SRVA DD 4189

16.) Examine results; first set gridfile to 4189:

IGU SET 4189

17.) Contour data in gridfile 4189:

IGTV 1 20 SAT (Z100--grid number 1)

IGTV 2 3 SAT (TSL--grid number 2)

IGTV 3 3 SAT (DD--grid number 3)

(*** in AUTO mode, numbers 16-20 are not done ***)

18.) If bad analyses result, examine surface MD file. First, clear the graphics screen:

EG, followed by

IC (draw satellite projection map over image)

then:

PLVA Z100 MDNR=4183 MDRR=12 for Z100 (with MDNR, MDRR indicating MD file and row numbers for surface data, respectively)

then:

PLVA TSL MDNR=4183 MDRR=12 for TSL, and

PLVA DD MDNR=4183 MDRR=12 for DD

19.) Delete bad reports and update surface MD file:

MDU SET 4183

then: (*** in AUTO mode, numbers 16-20 are not done ***)

XRVA (delete ENTIRE report)

20.) If any deletions, repeat (for that parameter) starting at step 15.

21.) Set "NO. RETRIEVALS=" in VASTEXT file to 0:

SPVA NRET=0

22.) Process retrievals for the sounder area:

VTPZ END=3513 6997 SPC=16 11 ARA=4191

(*** in AUTO mode: VTPZ SPC=16 11 ARA=4191 AUTO=1, as an example ***)

In the non-auto mode, the desired ending location (in image coordinates) is determined by setting the cursor near the SE corner of the image and then executing system program I (simply key in the letter "I" by itself).

The starting location will be determined by default from the VASTEXT file. The retrievals generated will be for 11X11 FOV (field of view) boxes (default), separated by 16 FOV's in the y-direction, and 11 FOV's in the x-direction. In addition, a total precipitable water vapor image will be created and stored in digital area 4191. The following example shows what the VASTEXT file should look like at this point. Remember, the VASTEXT file is displayed on the CRT by using program LOVA.

YYDDD	BEGIN	X-RES	Y-RES	LLNW	LLSE	STAT	NSAT	SNDAREA
84142	124800	8	8	47110	33078	1	29	4190

MDNS	MDRS	MDNG	MDRG	MDNR	MDRR
4183	12	4184	0	4185	1

NGFG	NGFS	ZGRID	TGRID	DGRID	PGRID
4190	4189	1	2	3	84142

NO.RETRIEVALS=183
CURRENT RETRIEVAL OPTIONS..

TYP	GSS	SPC	SIZ	SFC	ENDL	ENDE	BEGL	BEGE	TER	PLT
0	0	20	10	0	3421	7149	2417	4673	68	0

Note: The above information pertains to the processing of a sounder area sensed in the small detector mode of VAS. For the large detector mode, variables X-RES and Y-RES would be equal to 16.

NOTE: Beyond this point, only a video terminal should be used. This is so because the editing which follows is most efficiently performed on this type of terminal, where XRVA is used with cursor selection to delete unacceptable retrievals.

23.) Plot retrieval data:

PLVA Z 500 MDNR=4185 MDRR=1

24.) Set gridfile before BNVA:

IGU SET 4191

25.) Generate grids of retrieval parameters and store in gridfile 4191:

BNVA Z 500

26.) Select the appropriate grid (#1, because grid just generated was dumped here by BNVA), and display the contours on the video screen (over the earlier PLVA plot of 500 mb heights):

IGTV 1 30 SAT (contour interval of 30 meters)

27.) If there are poor analyses, go through a similar procedure as was used for the surface data:

EG

IC

PLVA Z 500 MDNR=4185 MDRR=1

MDU SET 4185

XRVA (delete entire poor retrieval)

28.) If there are any deletions from the retrieval MD file, repeat from step 25.

29.) Generate VAS gradient winds (optional):

GWVA 1 500

30.) Update upper air first guess (i.e., grids 33-54 in gridfile 4190) with the just-completed retrievals, if more retrievals are to be made at a different time, and then begin again at step 1 to process the next retrieval area:

UGVA 33 Note: the updated grids will be added onto the end of gridfile 4190.

CHAPTER 2

VAS MODULAR FLOWCHARTS

This chapter contains modular flowcharts for each VAS program discussed in chapters 3-5. The flowcharts are modular in that they consist solely of subroutines which are called successively in a given program. Each subroutine is therefore a module of the program as a whole. The programs are subdivided to systems level or assembly language subroutines in most cases, which is sufficiently detailed for the purposes of this manual.

The first tier of subroutines in each flowchart, which will be referred to as Level I Subroutines, are those routines called explicitly by the given program. Subsequent lower-level subroutines branch out from these main subroutines. When a given subroutine terminates a branch of the program "tree," it means simply that I could no longer trace that particular branch, or that it was not important to go any further.

Before a list of the order of flowchart presentation, I shall discuss briefly the page number system adopted for this chapter. When a given flowchart refers to a certain page number, the user should turn to the page whose SECOND (right) number at the bottom corresponds to the correct page. The user should note that there are 27 such double-numbered pages in Chapter 2.

ORDER OF PRESENTATION

1. VPVA
2. VDVA
3. IDVA
4. GSVA

5. CSVA

6. SRVA

7. XRVA

8. SPVA

9. LOVA

10. GPVA

11. SRAD

12. SRET

13. UGVA

14. PLVA

15. EXVA

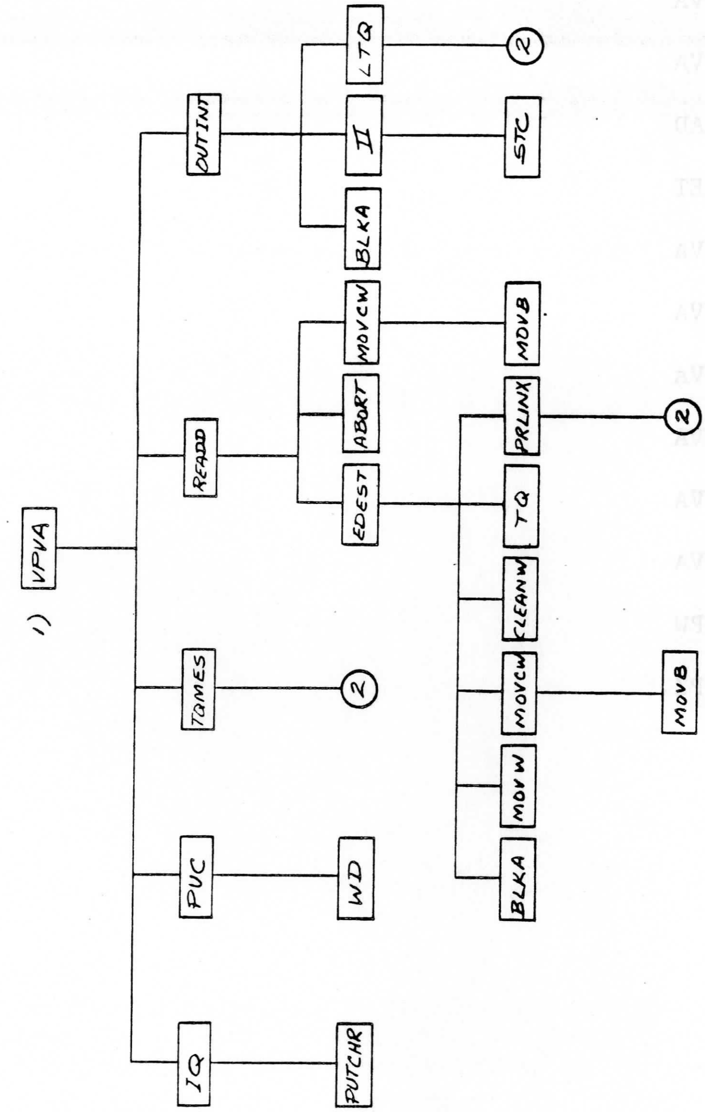
16. BNVA

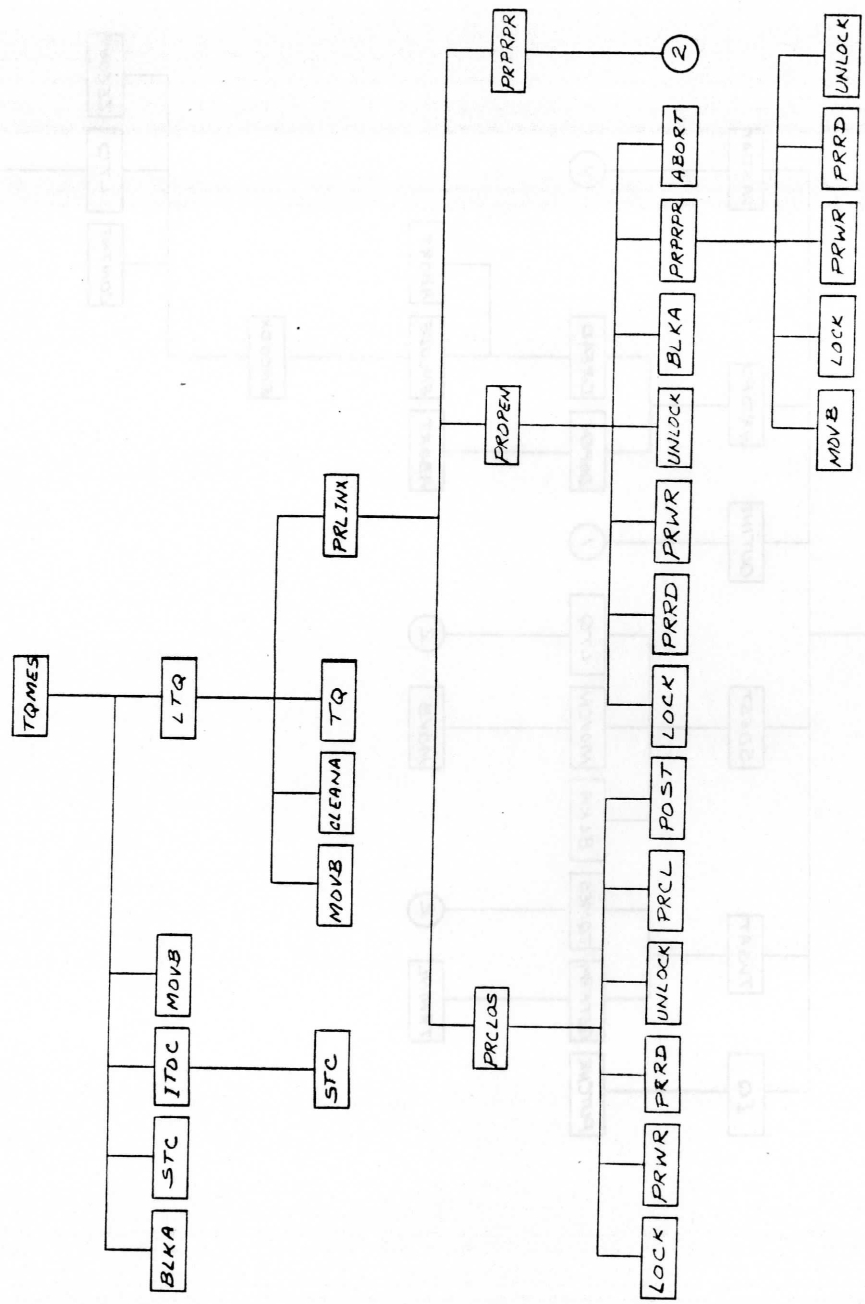
17. GWVA

18. ESVA

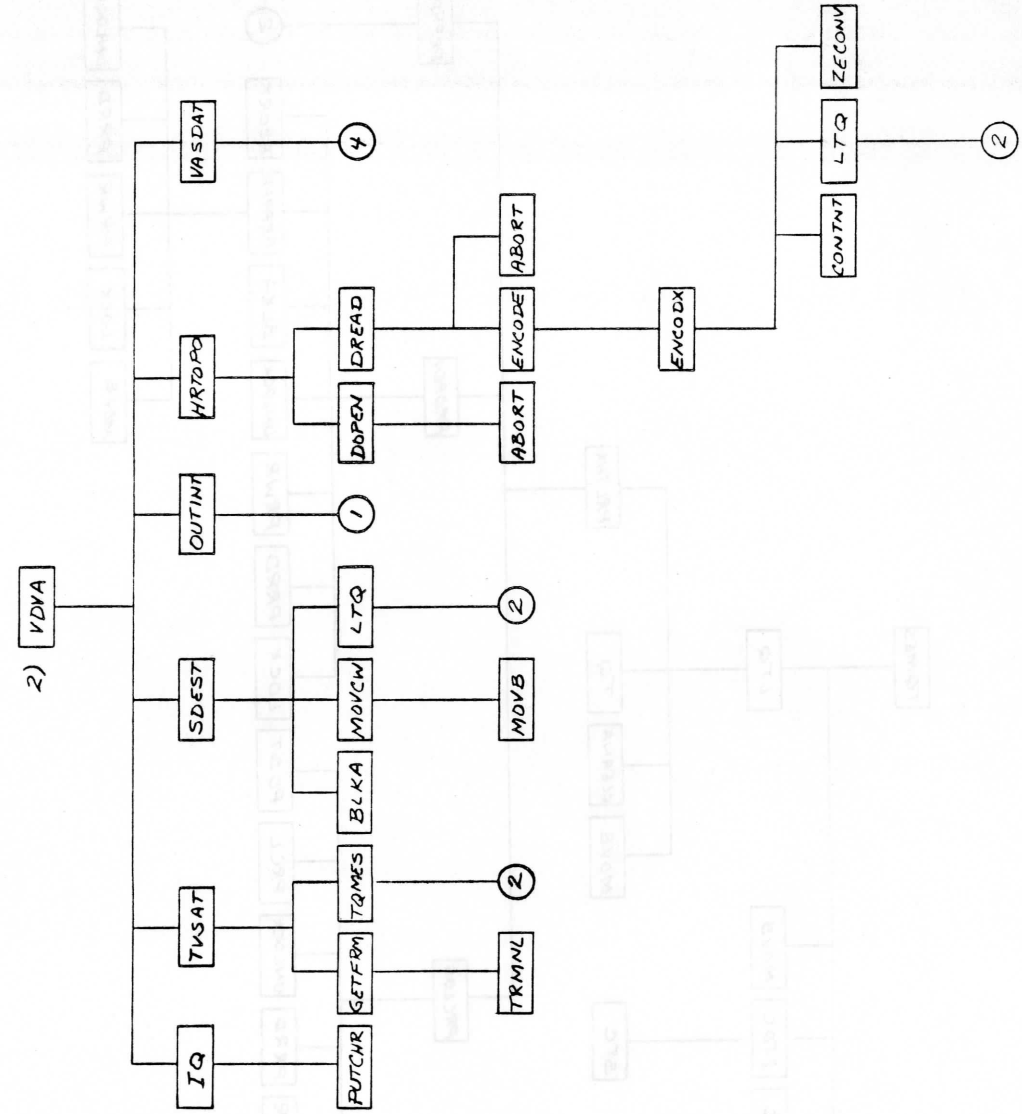
19. VTPW

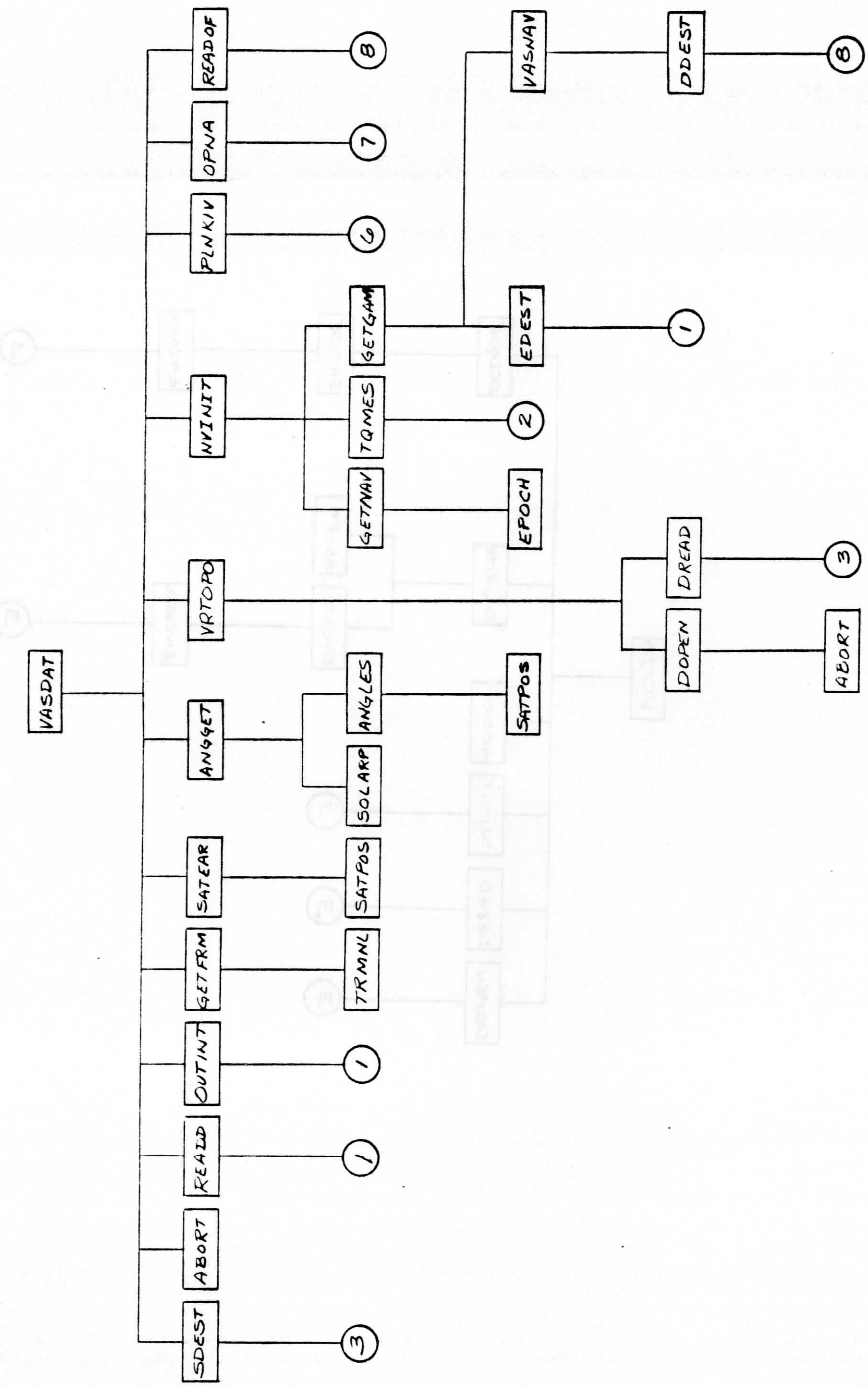
20. VTPZ

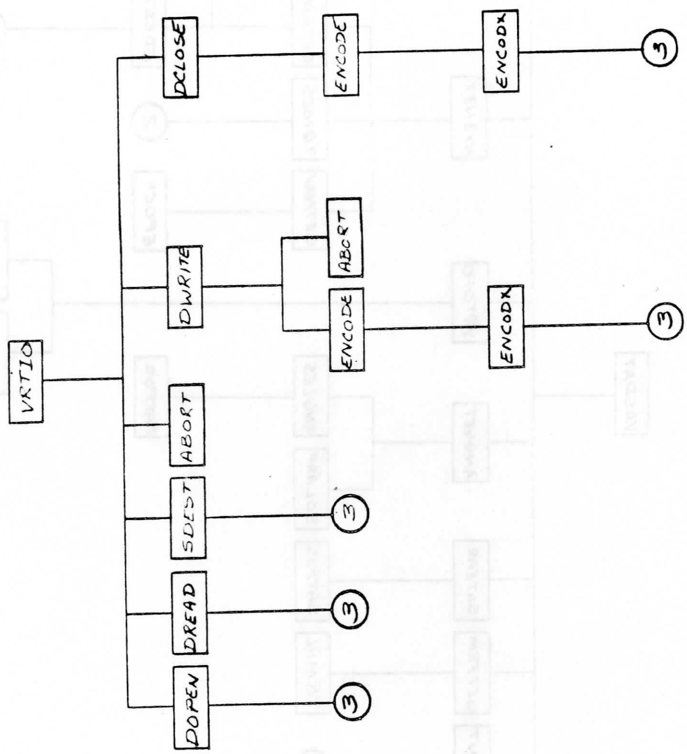

1)
VPVA
IQ
PUTCHR
PUC
WD
TQMES
2
READD
EDEST
BLKA
MOVW
MOVCW
MOVB
CLEANW
TQ
PRLINX
2
ABORT
MOVCW
MOVB
OUTINT
BLKA
II
STC
LTQ
2

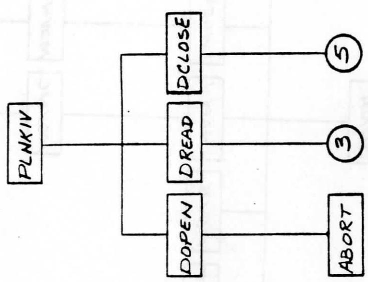

TQMES
LTQ
TQ
CLEANA
MOVB
PRLINX
PRPRPR
2
PROPEN
LOCK
PRRD
PRWR
UNLOCK
BLKA
PRPRPR
ABORT
MOVB
LOCK
PRWR
PRRD
UNLOCK
PRCLOS
LOCK
PRWR
PRRD
UNLOCK
PRCL
POST
BLKA
STC
ITOC
MOVB
STC

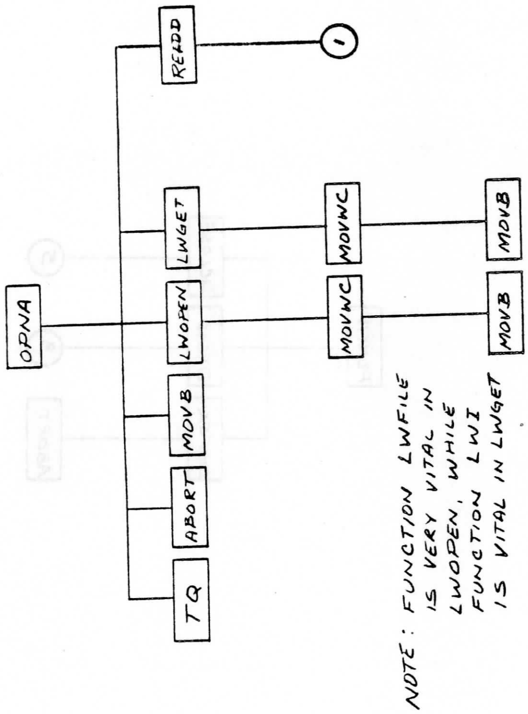
2)

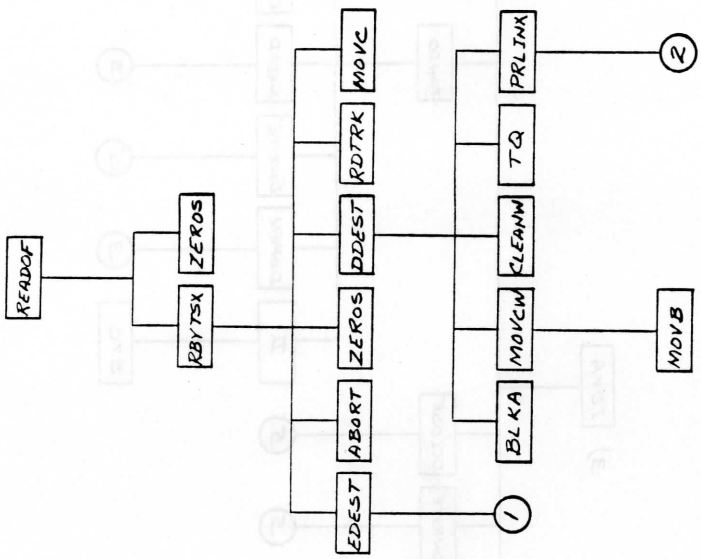

VDYA
IQ
TVSAT
SDEST
OUTINT
HRTOPO
VASDAT
PUTCHR
GETFRM
TQMES
BLKA
MOVCW
LTQ
OUTINT
DOPEN
DREAD
TRMNL
2
MOVB
2
1
ABORT
ENCODE
ABORT
4
ENCODX
CONTNT
LTQ
ZECONV
2

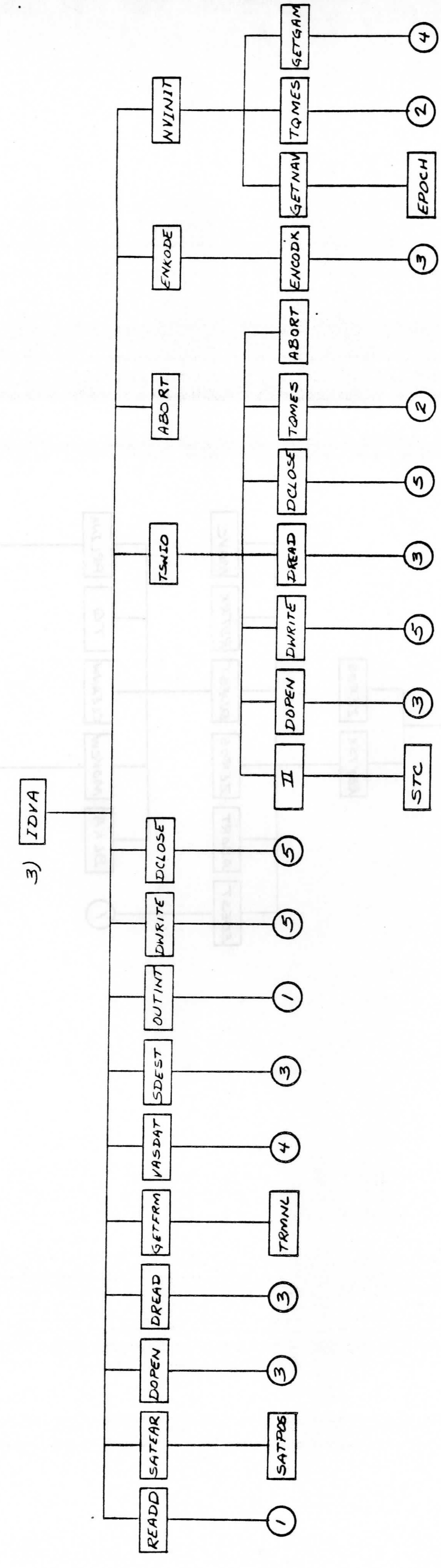

VASDAT
SDEST
3
ABORT
REALD
1
OUTINT
1
GETFRM
TRMNL
SATEAR
SATPOS
ANGGET
SOLARP
ANGLES
SATPOS
VRTOPO
DOPEN
ABORT
DREAD
3
NVINIT
GETNAV
EPOCH
TQMES
2
GETGAM
EDEST
1
VASNAV
DDEST
8
PLNKIV
6
OPNA
7
READOF
8

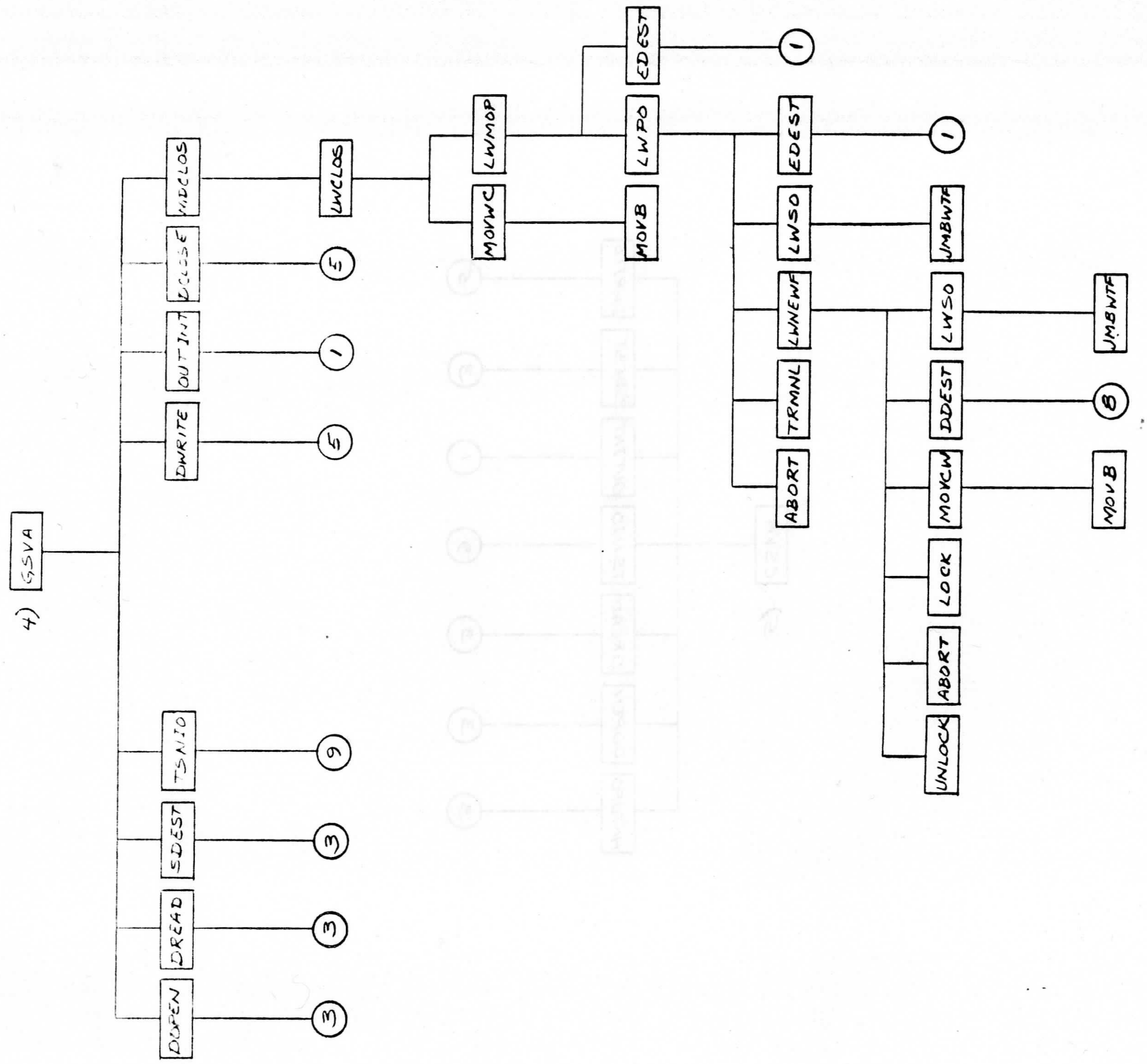

VRTIO
DOPEN
DREAD
SDIST
ABORT
DWRITE
DCLOSE
ABORT
ENCODE
ENCODX
ENCODE
ENCODX
3
3
3
3
3


PLNKIV
DOPEN
DREAD
DCLOSE
ABORT
3
5

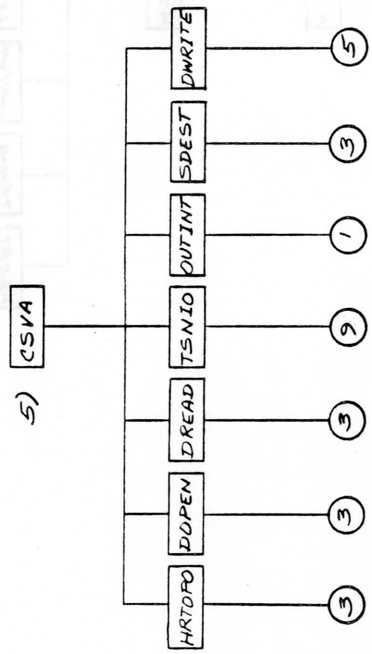

OPNA
TQ
ABORT
MOVB
LWOPEN
LWGET
READD
1
MOVNC
MOVNC
MOVB
MOVB
NOTE: FUNCTION LWFILE
IS VERY VITAL IN
LWOPEN, WHILE
FUNCTION LWI
IS VITAL IN LWGET

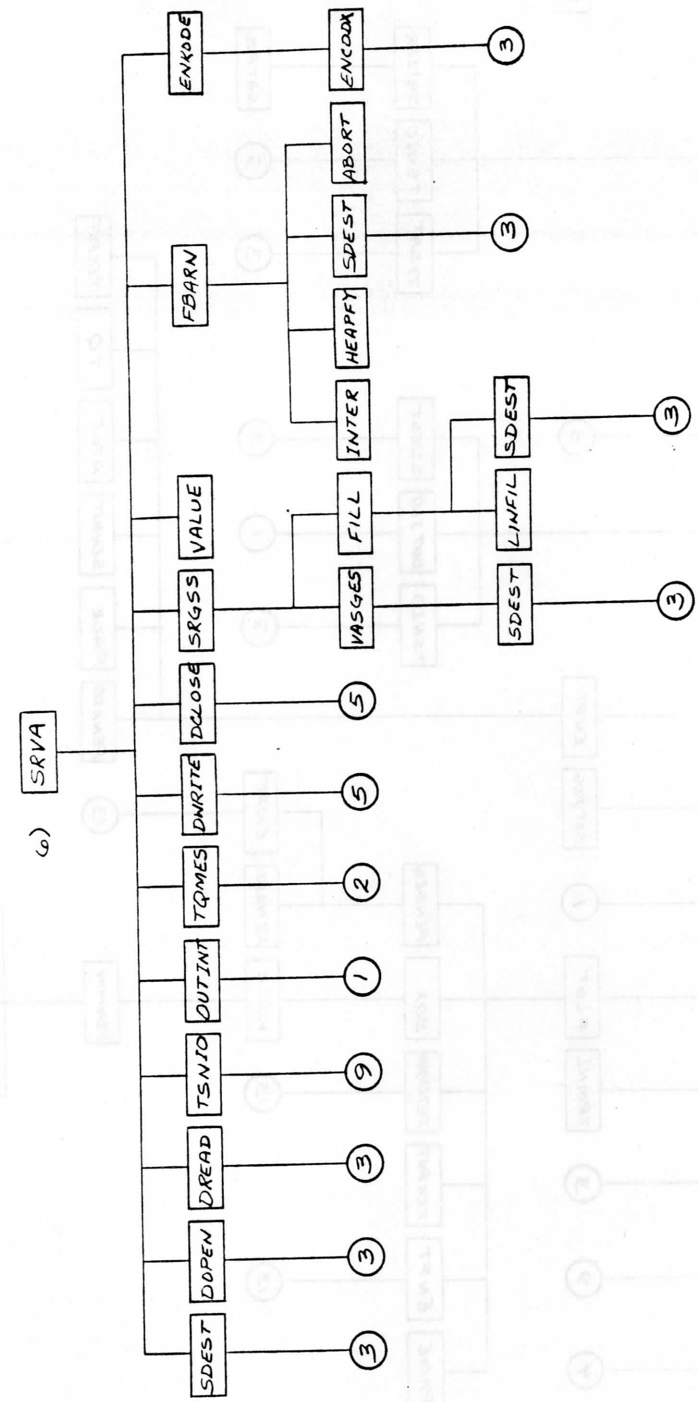

READOF
RBYTSX
ZEROS
EDEST
ABORT
ZEROS
DDEST
RDTRK
MOVC
BLKA
MOVCW
CLEANW
TQ
PRLINX
MOVB
1
2


3) IDVA
READD
SATEAR
SATPOS
DOPEN
DREAD
GETFRM
TRMNL
VASDAT
SDEST
OUTINT
DWRITE
DCLOSE
TSKIO
II
STC
DOPEN
DWRITE
DREAD
DCLOSE
TQMES
ABORT
ABORT
ENKODE
ENCODX
NVINIT
GETNAV
EPOCH
TQMES
GETGAM

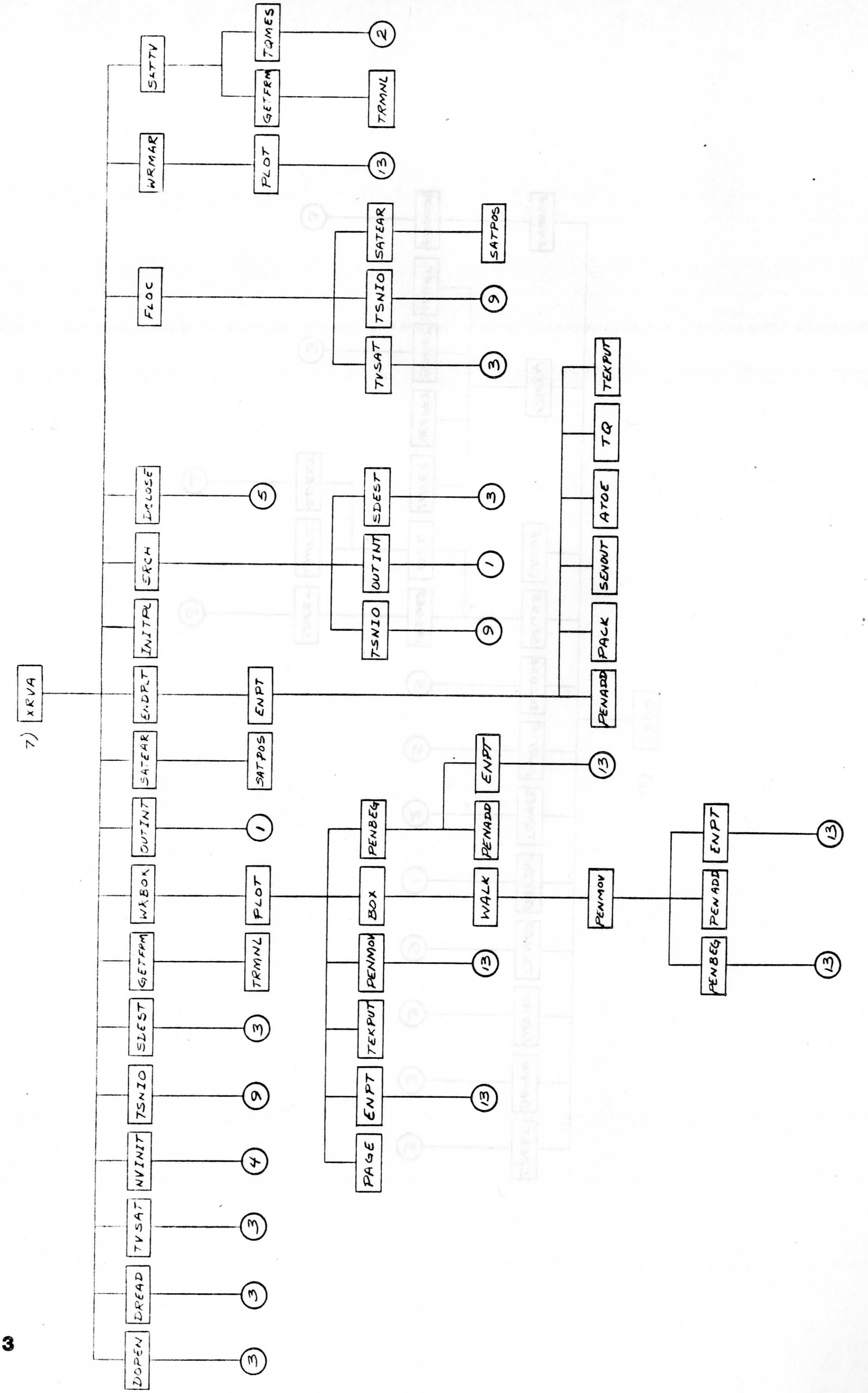

4)
GSVA
DOPEN
DREAD
SDEST
TSNIO
DWRITE
OUTINT
DCLOSE
MDCLOS
LWCLOS
MOVWC
LWMOP
MOVB
LWPO
EDEST
ABORT
TRMNL
LWNEWF
LWSO
EDEST
JMBWTF
UNLOCK
ABORT
LOCK
MOVCW
DDEST
LWSO
MOVB
JMBWTF
3
3
3
9
5
1
5
1
1
8

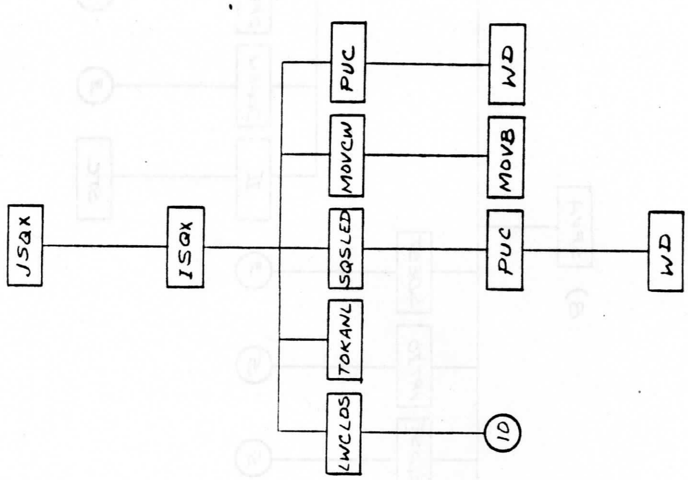
5)

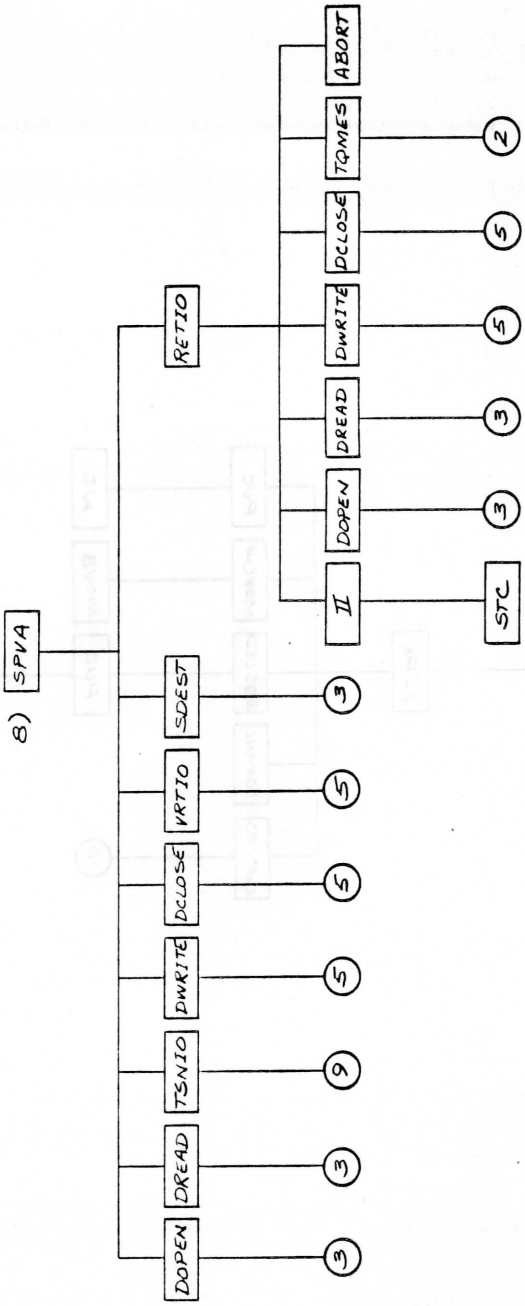



(6)
SRVA
SDEST
DOPEN
DREAD
TSNIO
OUTINT
TQMES
DWRITE
DCLOSE
SRGSS
VALUE
FBARN
ENKODE
VASGES
FILL
SDEST
LINFIL
SDEST
INTER
HEAPFY
SDEST
ABORT
ENCODX
3
3
3
9
1
2
5
5
3
3
3
3

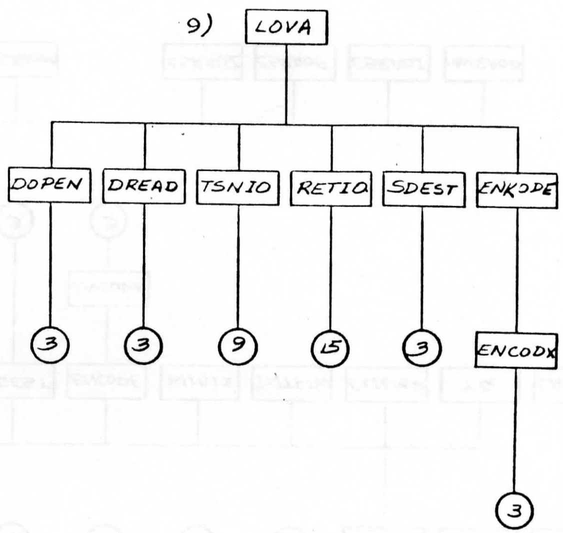
7)

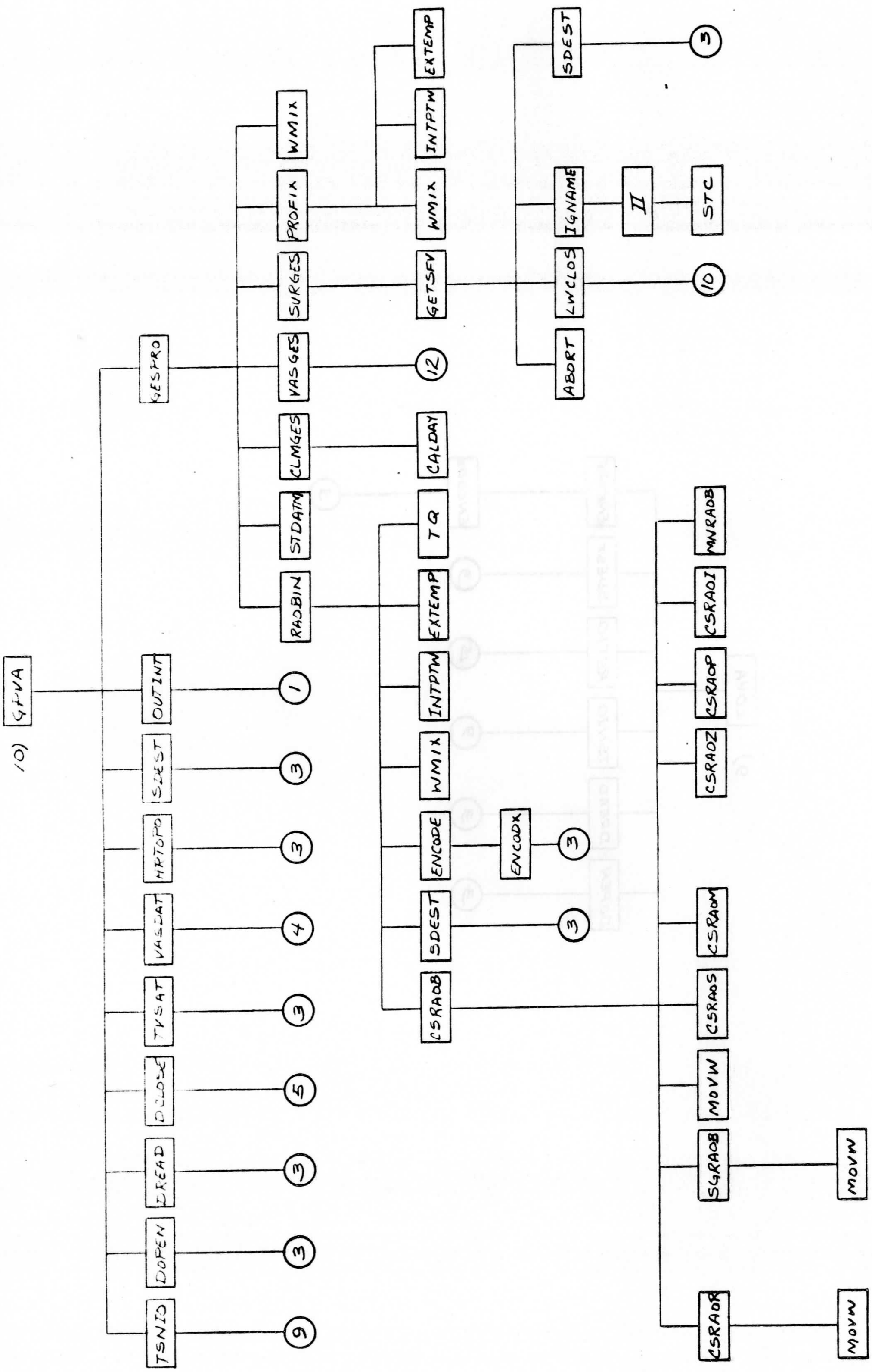

XRVA
DOPEN
DREAD
TVSAT
NVINIT
TSNIO
SDEST
GETFPM
WRBOX
OUTINT
SATEAR
ENDPLT
INITPL
ERCH
DCLOSE
FLOC
WRMAR
SETTV
TRMNL
PLOT
SATPOS
ENPT
PAGE
TEKPUT
PENMOV
BOX
PENBEG
PENADD
WALK
PACK
SENOUT
ATOE
TQ
GETFRM
TQMES
3
4
9
13
1
5
2


JSQX
ISQX
LWCLOS
TOKANL
SQSLED
MOVCW
PUC
PUC
MOVB
WD
WD
1D

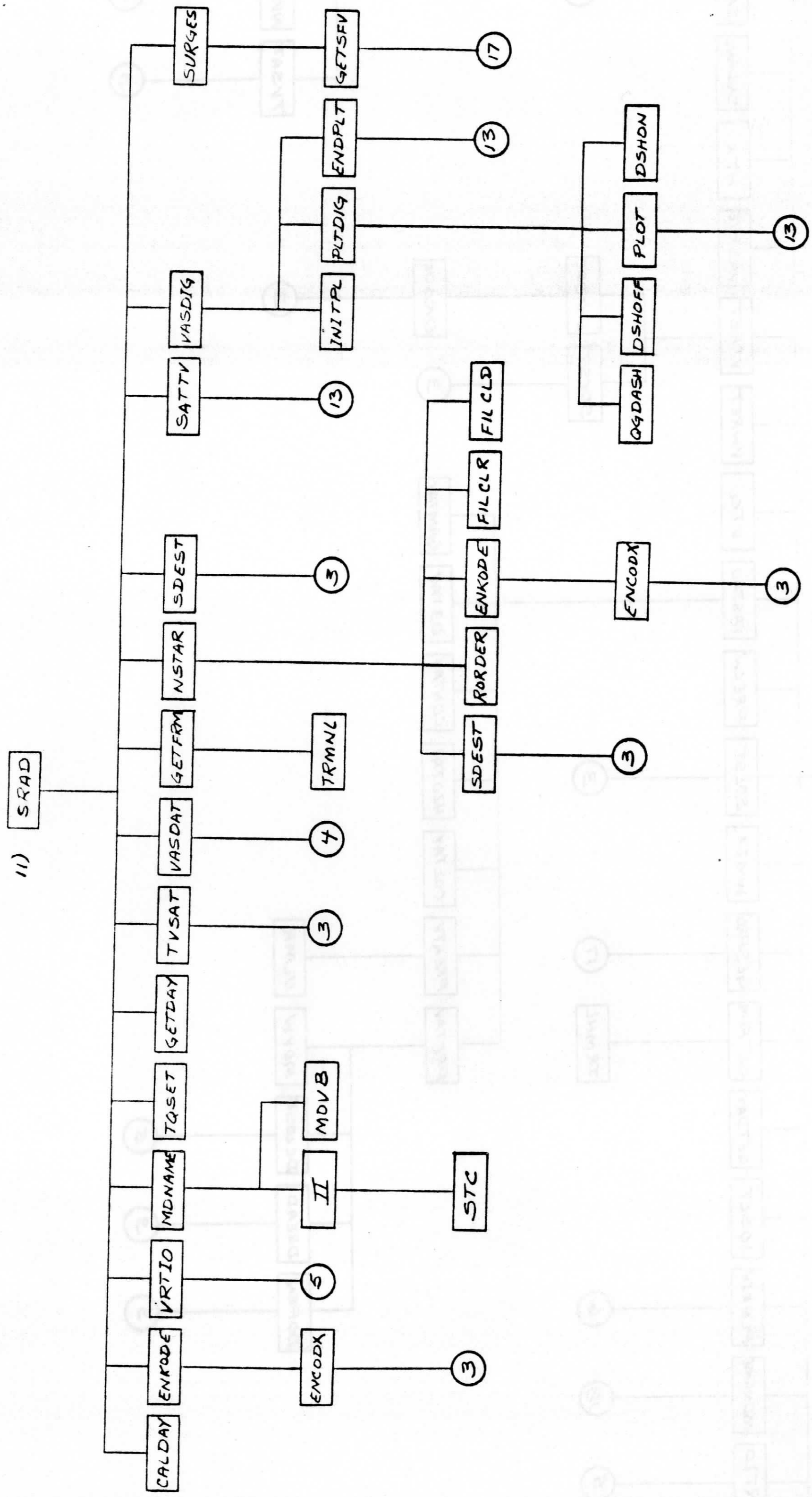

8)
SPVA
DOPEN
DREAD
TSNIO
DWRITE
DCLOSE
WRTIO
SDEST
RETIO
II
DOPEN
DREAD
DWRITE
DCLOSE
TQMES
ABORT
STC
3
3
9
5
5
5
3
3
3
5
5
2

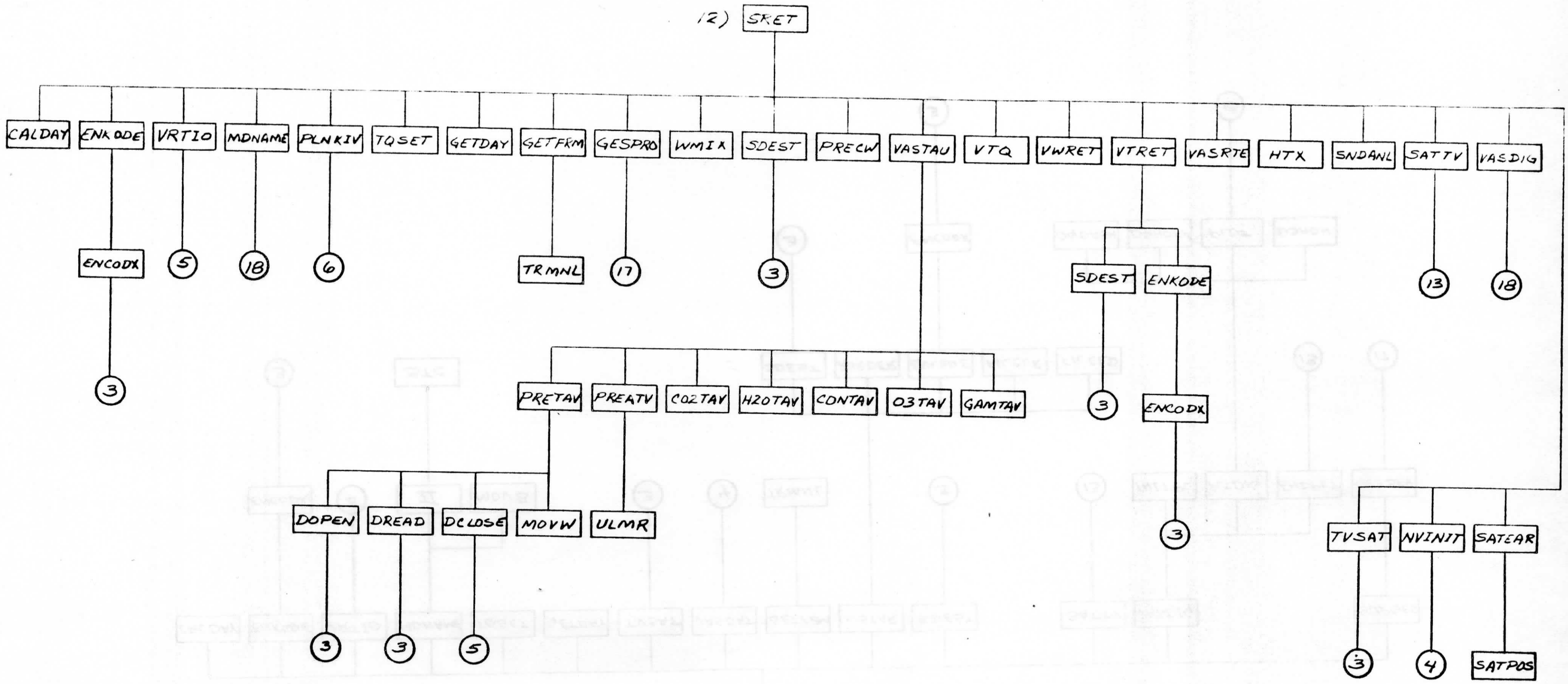
9)



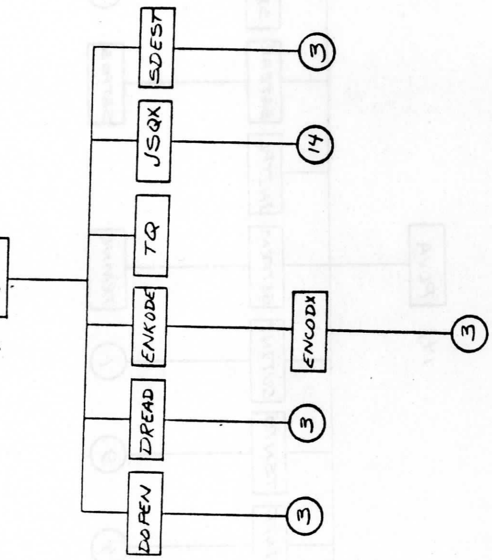

10)
GPVA
TSNID
DOPEN
DREAD
DCLOSE
TVSAT
VASDAT
MRTOPO
SDEST
OUTINT
GESPRO
RADBIN
STDATM
CLMGES
VASGES
SURGES
PROFIX
WMIX
CSRADB
SDEST
ENCODE
ENCODX
WMIX
INTPTW
EXTEMP
TQ
CALDAY
GETSFV
WMIX
INTPTW
EXTEMP
ABORT
LWCLOS
IGNAME
SDEST
II
STC
CSRADR
SGRADB
MOVW
CSRADS
CSRADM
CSRADZ
CSRADP
CSRADI
MNRADB
MOVW
MOVW

11)

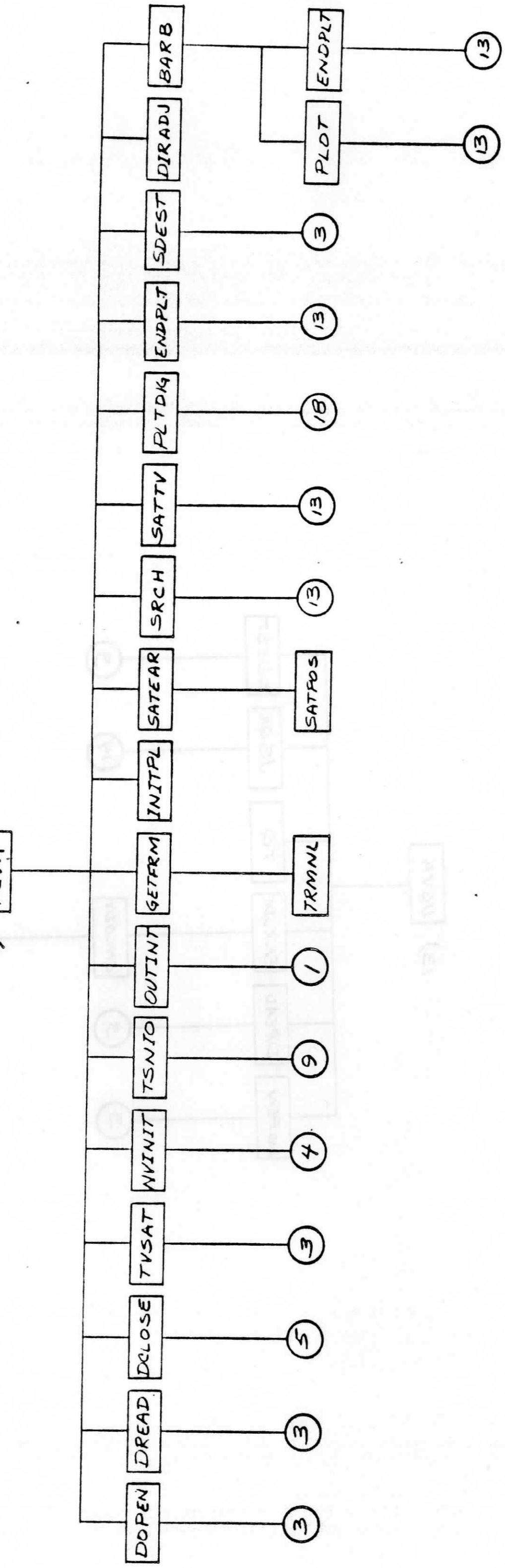

SRAD
CALDAY
ENKODE
ENCODX
3
VRTIO
5
MDNAME
II
MDVB
STC
TQSET
GETDAY
TVSAT
3
VASDAT
4
GETFRM
TRMNL
NSTAR
SDEST
3
RORDER
ENKODE
ENCODX
3
FILCLR
FIL CLD
SDEST
3
SATTV
13
VASDIG
INITPL
PLTDIG
QGDASH
DSHOFF
PLOT
DSHON
13
ENDPLT
13
SURGES
GETSFV
17


(2) SKET
CALDAY
ENKODE
VRTIO
MDNAME
PLNKIV
TQSET
GETDAY
GETFRM
GESPRO
WMIX
SDEST
PRECW
VASTAU
VTQ
VWRET
VTRET
VASRTE
HTX
SNDANL
SATTV
VASDIG
ENCODX
5
18
6
TRMNL
17
3
SDEST
ENKODE
13
18
PRETAV
PREATV
CO2TAV
H2OTAV
CONTAV
O3TAV
GAMTAV
3
ENCODX
3
DOPEN
DREAD
DCLOSE
MOVW
ULMR
3
TVSAT
NVINIT
SATEAR
3
3
5
3
4
SATPOS

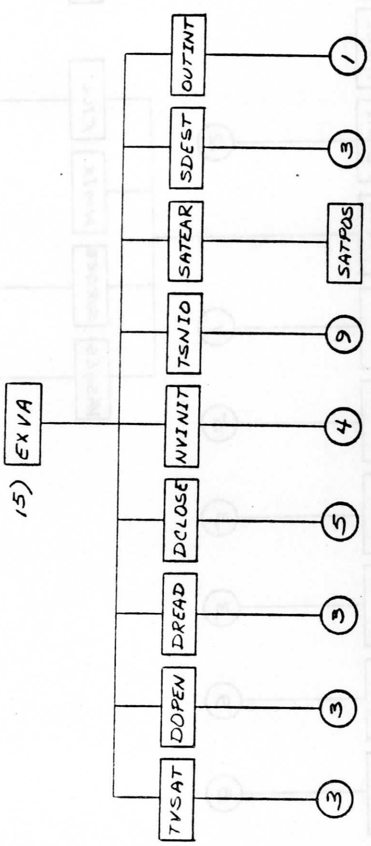
13)



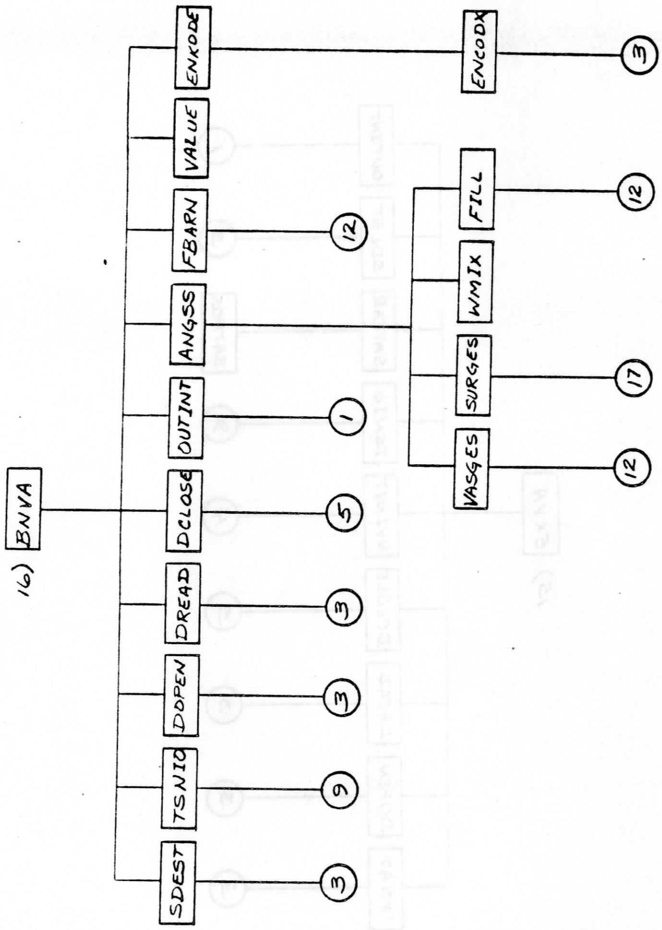
14)

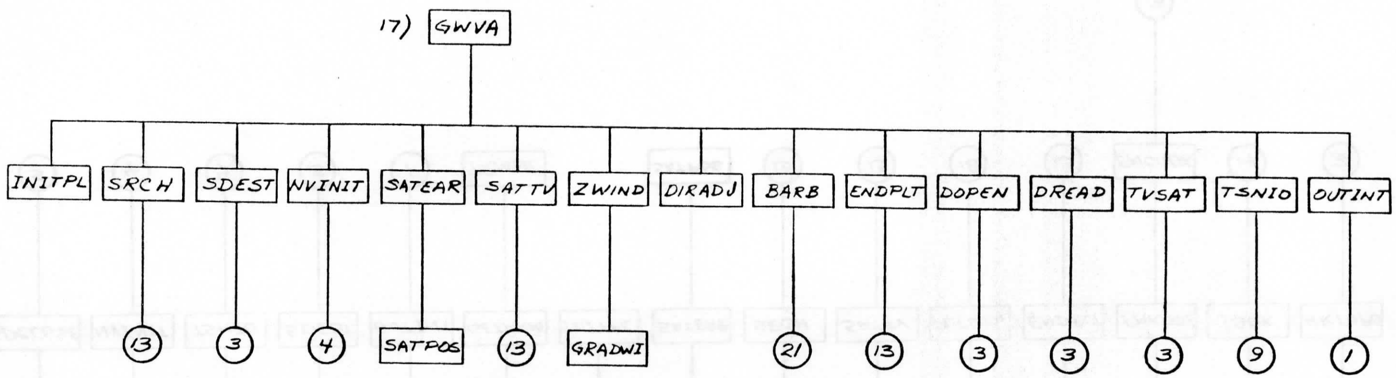


PLVA
DOPEN
DREAD
DCLOSE
TVSAT
NVINIT
TSNIO
OUTINT
GETFRM
TRMNL
INITPL
SATEAR
SATPOS
SRCH
SATTV
PLTDIG
ENDPLT
SDEST
DIRADJ
BARB
PLOT
ENDPLT
3
3
5
3
4
9
1
13
13
18
13
3
13
13

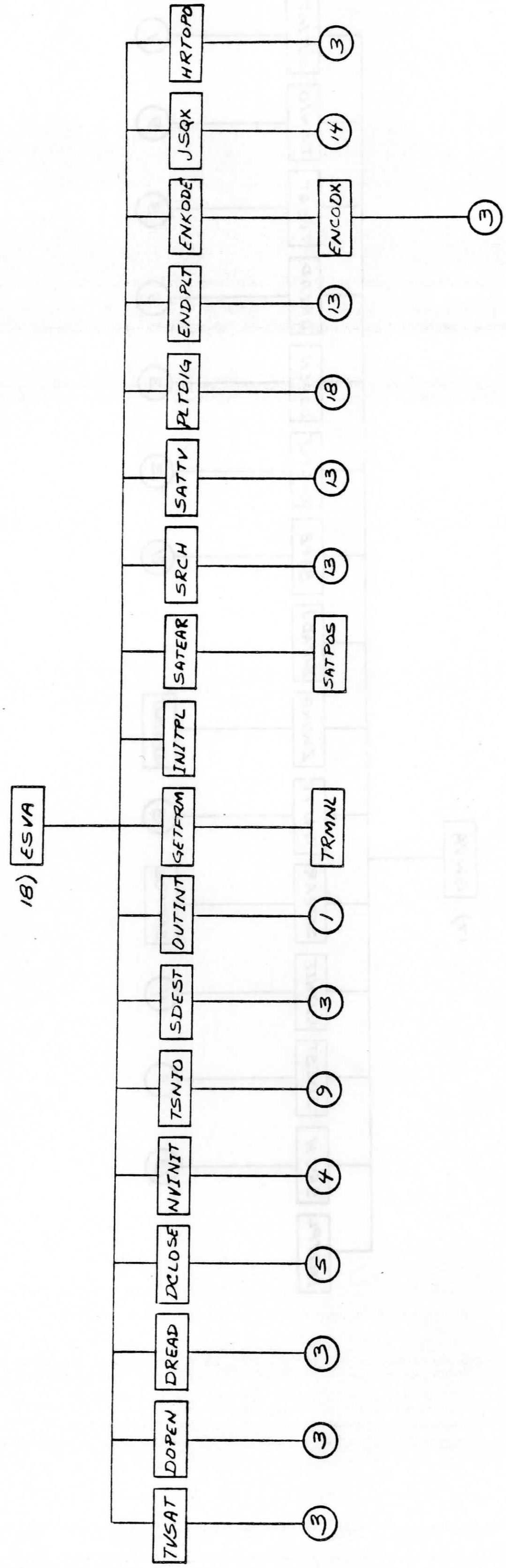


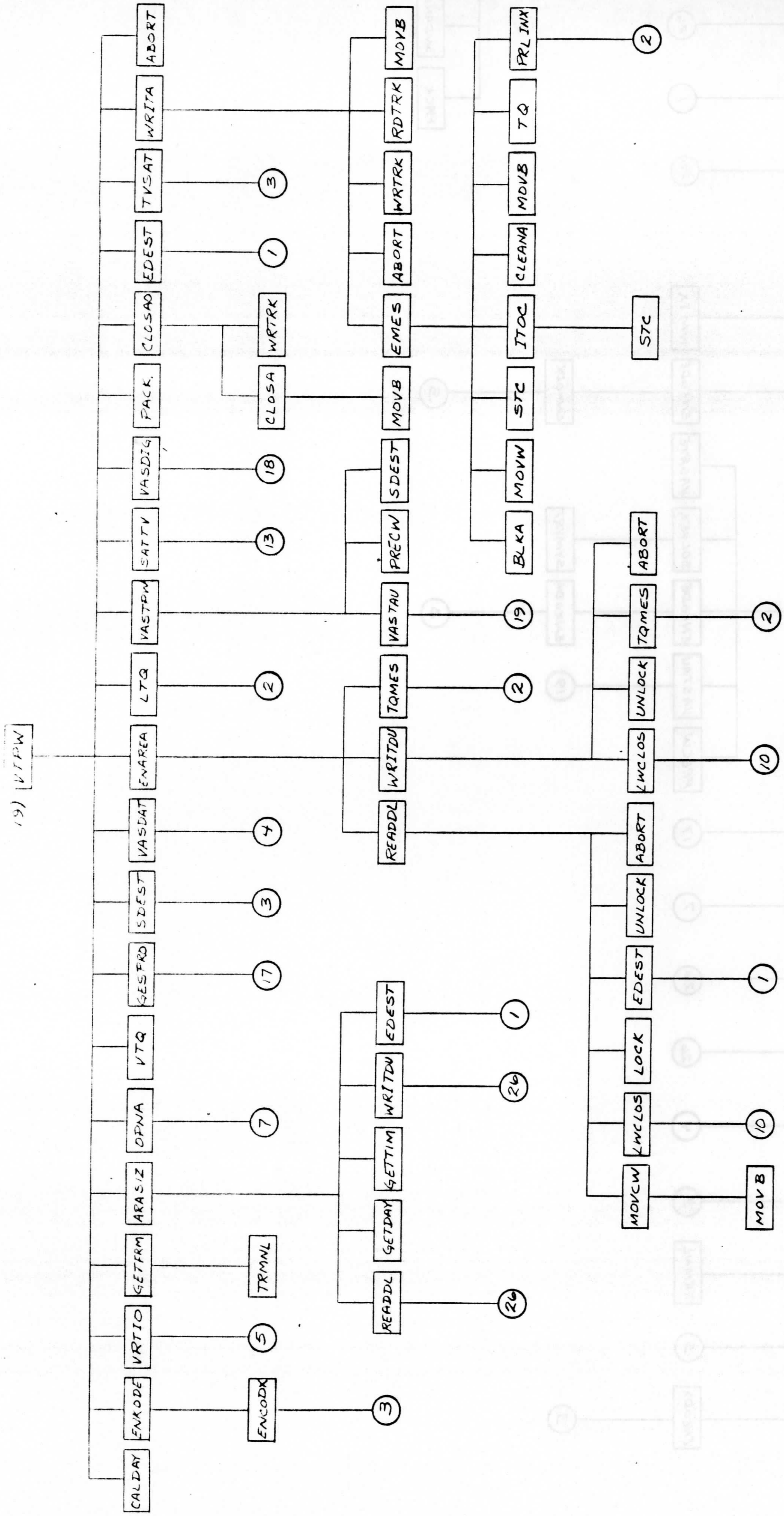
EXVA
(5)
TVSAT
DOPEN
DREAD
DCLOSE
NVINIT
TSNIO
SATEAR
SDEST
OUTINT
SATPOS
3
3
3
5
4
9
3
1

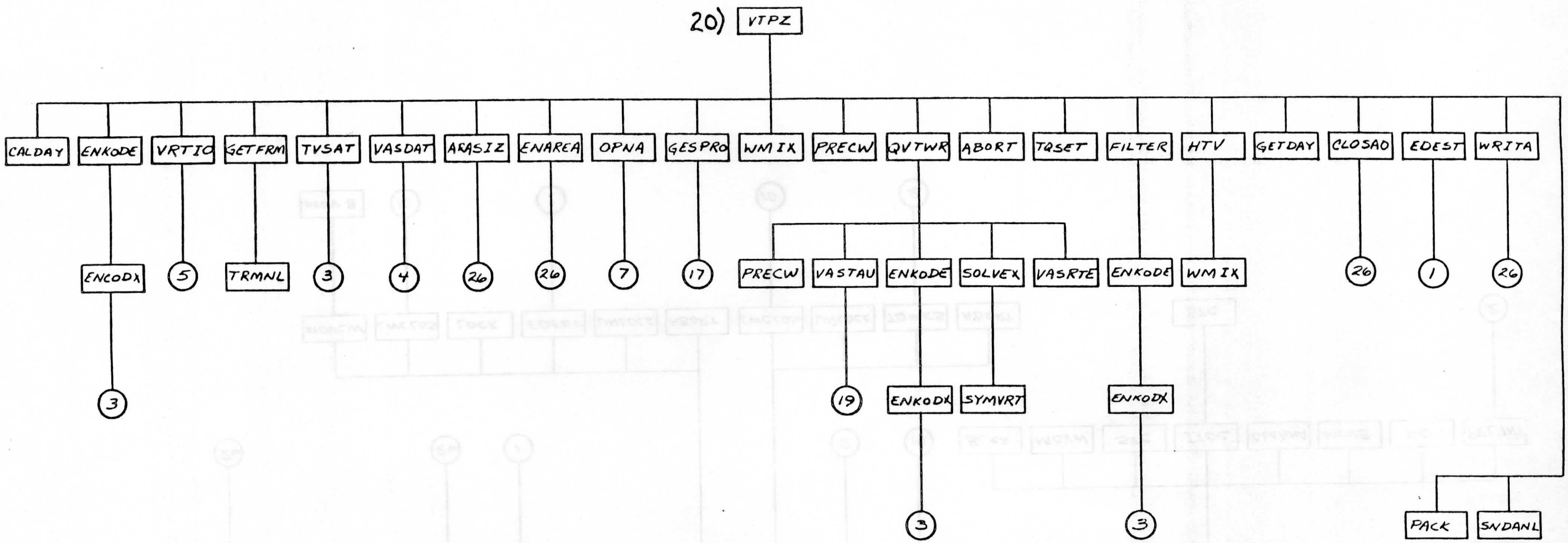

16) BNVA
SDEST
TSNIO
DOPEN
DREAD
DCLOSE
OUTINT
ANGSS
FBARN
VALUE
ENKODE
VASGES
SURGES
WMIX
FILL
ENCODX
3
9
3
3
5
1
12
12
17
12
3


17)
GWVA
INITPL
SRCH
SDEST
NVINIT
SATEAR
SATTV
ZWIND
DIRADJ
BARB
ENDPLT
DOPEN
DREAD
TVSAT
TSNIO
OUTINT
13
3
4
SATPOS
13
GRADWI
21
13
3
3
3
9
1

18)


ESVA
TVSAT
3
DOPEN
3
DREAD
3
DCLOSE
5
NVINIT
4
TSNIO
9
SDEST
3
OUTINT
1
GETFRM
TRMNL
INITPL
SATEAR
SATPOS
SRCH
13
SATTV
13
PLTDIG
18
ENDPLT
13
ENKODE
ENCODX
3
JSQX
14
HRTOPO
3


(9) VTPW
CALDAY
ENKODE
VRTIO
GETFRM
ARASIZ
OPNA
VTQ
GESFRD
SDEST
VASDAT
ENAREA
LTQ
VASTPN
SATTV
VASDIG
PACK.
CLOSAD
EDEST
TVSAT
WRITA
ABORT
ENCODK
TRMNL
CLOSA
WRTRK
READDL
GETDAY
GETTIM
WRITDU
EDEST
READDL
WRITDU
TQMES
VASTAU
PRECW
SDEST
MOVB
EMES
ABORT
WRTRK
RDTRK
MOVB
MOVCW
LWCLOS
LOCK
EDEST
UNLOCK
ABORT
LWCLOS
UNLOCK
TQMES
ABORT
MOV B
BLKA
MOVW
STC
ITOC
CLEANA
MOVB
TQ
PRLINX
STC



20)
VTPZ
CALDAY
ENKODE
VRTIO
GETFRM
TVSAT
VASDAT
ARASIZ
ENAREA
OPNA
GESPRO
WMIX
PRECW
QVTWR
ABORT
TQSET
FILTER
HTV
GETDAY
CLOSAO
EDEST
WRITA
ENCODX
5
TRMNL
3
4
26
26
7
17
PRECW
VASTAU
ENKODE
SOLVEX
VASRTE
ENKODE
WMIX
26
1
26
3
19
ENKODX
SYMVRT
ENKODX
3
3
PACK
SNDANL

CHAPTER 3

Main VAS Retrieval Software

This chapter contains those programs most vital to the VAS retrieval process. These programs, in addition to the other supporting McIDAS software mentioned in chapter 1, are necessary to process VAS data sets completely. Much of the software discussed in this chapter and the two which follow can apply to either VAS or TOVS. However, I will deal exclusively with the VAS aspects of the programs.

The presentation of each program in this chapter and the two which follow is similar. First, a general description of the program and how it operates is given. This should help someone unfamiliar with the software to get at least a basic understanding of its function. Next, a different type of flowchart than the Modular Flowchart, called a Level I Flowchart, is presented. In addition to Level I Subroutines, these flowcharts include comments, DO LOOPs and any IF statements that affect the calling of a Level I Subroutine. Finally, an attempt is made to define each of the variables within the IF statements themselves with at least one arithmetic statement, and more than one if the value of the variable changes within the program proper. The constructs used in the flowcharts are described in Appendix II. To gain the fullest understanding of each program, the Level I Flowchart should be used concurrently with both the program description and its code reproduction in Appendix VI.

The final section of each program discussion consists of a list of the subroutines called within that program. Level I

Subroutines are indicated at the head of each list with a capital "I." This list should correspond exactly to the subroutines included in the program's Modular Flowchart in Chapter 2.

VPVA

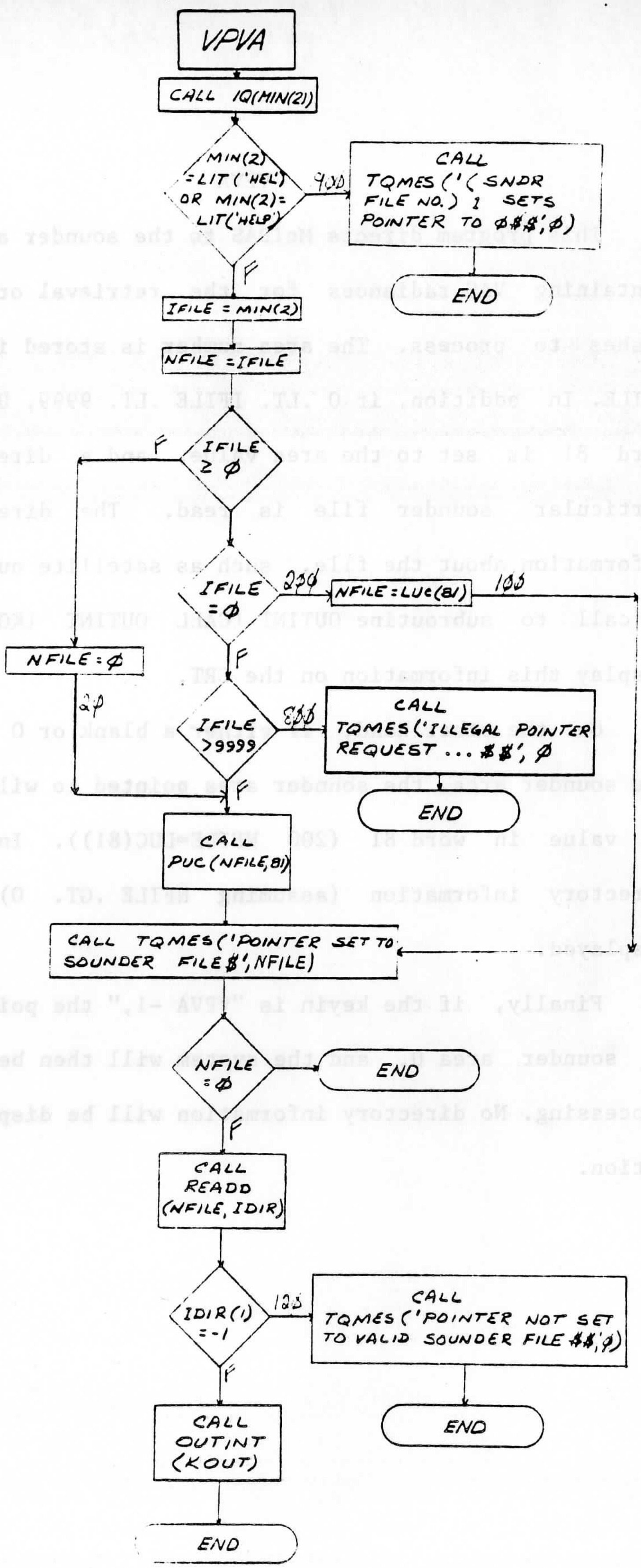
This program directs McIDAS to the sounder area/sounder file containing VAS radiances for the retrieval or retrievals one wishes to process. The area number is stored in the variable IFILE. In addition, if 0 .LT. IFILE .LT. 9999, User Common (UC) word 81 is set to the area value, and a directory for that particular sounder file is read. The directory contains information about the file, such as satellite number, date, etc. A call to subroutine OUTINT (CALL OUTINT (KOUT)) will then display this information on the CRT.

On the other hand, if either a blank or 0 is keyed in for the sounder area, the sounder area pointed to will be the current UC value in word 81 (200 NFILE=LUC(81)). In addition, the directory information (assuming NFILE .GT. 0) will again be displayed.

Finally, if the keyin is "VPVA -1," the pointer will be set to sounder area 0, and the system will then be set for TOVS processing. No directory information will be displayed using this option.

READ KEYIN

SET USER COMMON WORD 81 TO KEYED-IN SOUNDER AREA NUMBER (ASSUMING, OF COURSE, THAT SOUNDER AREA NUMBER HAS BEEN KEYED IN)



Subroutines used by VPVA:

1) IQ-I
2) PUC-I
3) TQMES-I
4) READD-I
5) OUTINT-I
6) PUTCHR
7) WD
8) BLKA
9) STC
10) ITOC
11) MOVB
12) LTQ
13) CLEANA
14) TQ
15) PRLINX
16) PRCLOS
17) PROPEN
18) PRPRPR
19) LOCK
20) PRWR
21) PRRD
22) UNLOCK
23) PRCL
24) POST
25) ABORT
26) II
27) EDEST
28) MOVW
29) MOVCW
30) CLEANW

IDVA

IDVA is used to initialize the VASTEXT documentation file and a VRET-schema retrieval MD file row header. The discussion which follows will deal with two modes of operation. In the first mode, I will assume no latitude/longitude boundaries for processing have been keyed in. In the second mode, known as the "auto" mode, I will assume keyword parameter AUTO has been keyed in to some non-zero value, and I will also assume that both the NW and SE image corners have been keyed in via positional parameters LALONW and LALOSE. The first mode is used when image information is available to the user; the second mode is used when no image information is available, but the NW and SE corners of the area to be processed are known.

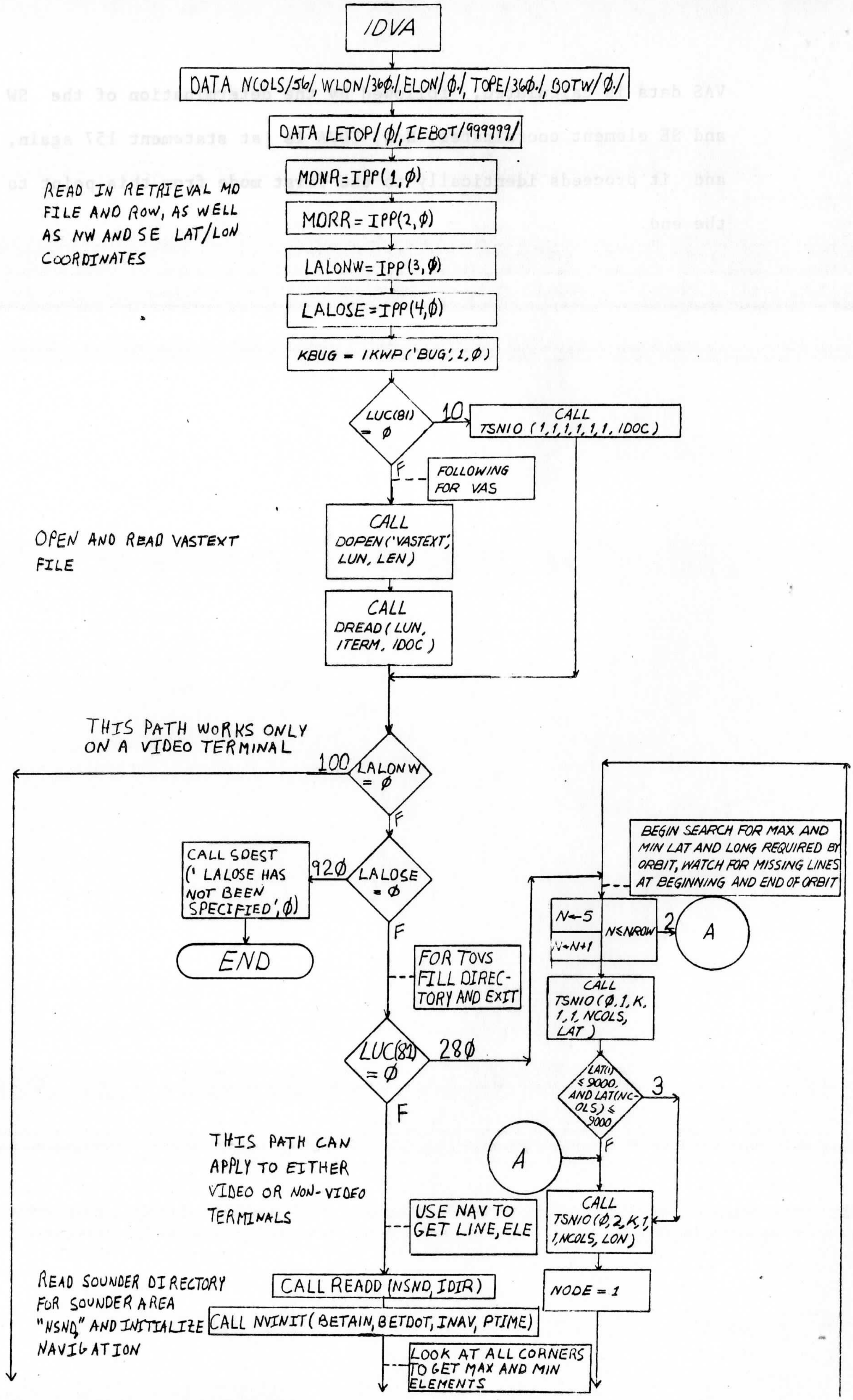
In the first mode case, variables MDNR, MDRR and KBUG (retrieval MD file number, retrieval MD file row number, and debug option, respectively) are read in via positional and keyword parameters (MDNR=IPP(1,0), MDRR=IPP(2,0), KBUG=IKWP('BUG',1,0)). Then, after the VASTEXT file is opened and read, control passes to statement 20 and then immediately to statement 100 (note that LALONW=0 for this mode, since it was not keyed in). At this point, image information is accessed by subroutine GETFRM, including such things as the upper left corner line and element of the TV frame (IL,IE). Subsequent steps result in the picking up of the sounder area (NSND) and the calculation of the lower right corner of the TV frame itself in terms of line and element coordinates (LL,LE). Then, subroutine VASDAT is used to locate the upper bounds of the VAS data by moving successively down the TV screen. At least 4 consecutive lines of data must be found before control will transfer to statement 130. This constraint guarantees that the maximum number of small-detector fields of view will be used when generating retrievals later on in VTPZ. After the upper bounds of the data is located (which gives the line coordinate of the NW corner of the area (image) to be processed), the leftmost boundary of the data (element coordinate of the NW corner) is located (see code between statements 133 and 135). At this point, control passes to statement 138, where a message saying "...UPPER LEFT DONE" is displayed on the CRT. Now, the same process is done for the lower right (SE) corner of the satellite image (find lower bounds of data, (giving the lower right line coordinate of the image), then locate lower right element coordinate of the image). This is done approximately between statements 140 and 157. Keep in mind that LALONW=0 during this entire process. After the SE corner of the VAS data has been successfully located, variables IE, LE, LOW and LOE are determined (IE=MIN(IETOP,IEBOT), etc.), and a message saying "...LOWER RIGHT DONE" is displayed on the CRT. Then, LALONW and LALOSE are calculated and stored in IDOC(25) and (26), respectively. Further steps initialize the retrieval MD file row header (assuming that MDNR and MDRR have been keyed in .NE. 0) and the VASTEXT file. The row header is initialized by putting the array IRET into the header via function MDPUT (IOK=MDPUT(MDNR,MDRR,0,IRET), while the VASTEXT file is updated by placing several pertinent variables such as the retrieval MD file and row numbers, status flag, sounder area number, spacing defaults, etc. into the array IDOC, which is then written, via subroutine DWRITE, into the VASTEXT file itself. This concludes the first mode of operation of IDVA.

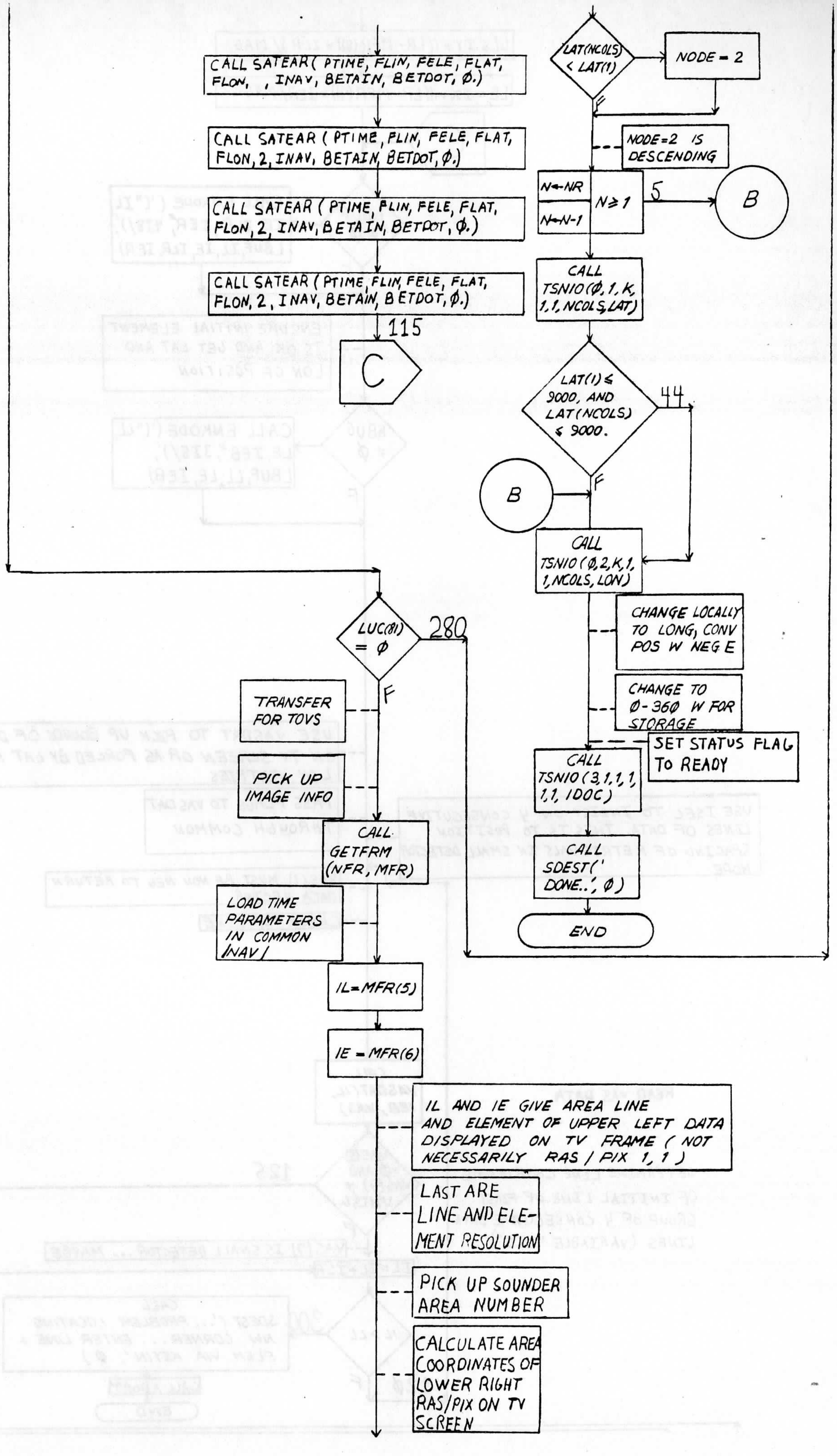
For the second "auto" mode case, IDVA proceeds identically to the first mode up to statement 20. Then, since both LALONW and LALOSE have been keyed in, IDOC(25) and (26) are filled, variables LAN and LOW are determined, and information pertaining to the sounder area is accessed via subroutine READD (CALL READD(NSND,IDIR)). Following this, after the navigation has been initialized (NVINIT), four calls to subroutine SATEAR, in conjunction with other NVINIT calculations, determine the line and element coordinates of the area to be processed (initial line (IL), initial element (IE), last line (LL) and last element (LE)). Note that no mention has been made of the TV image within the discussion of this mode, and that the boundaries of the area to be processed with this mode may not coincide with the actual sounder area boundaries, as they did for the non-"auto" mode.

After control transfers to statement 115, IDVA functions like the previously-discussed non-"auto" mode up to statement 135; that is, the upper bounds of the VAS data is located, giving the NW line coordinate, and then the western boundary is determined, giving the NW element coordinate.

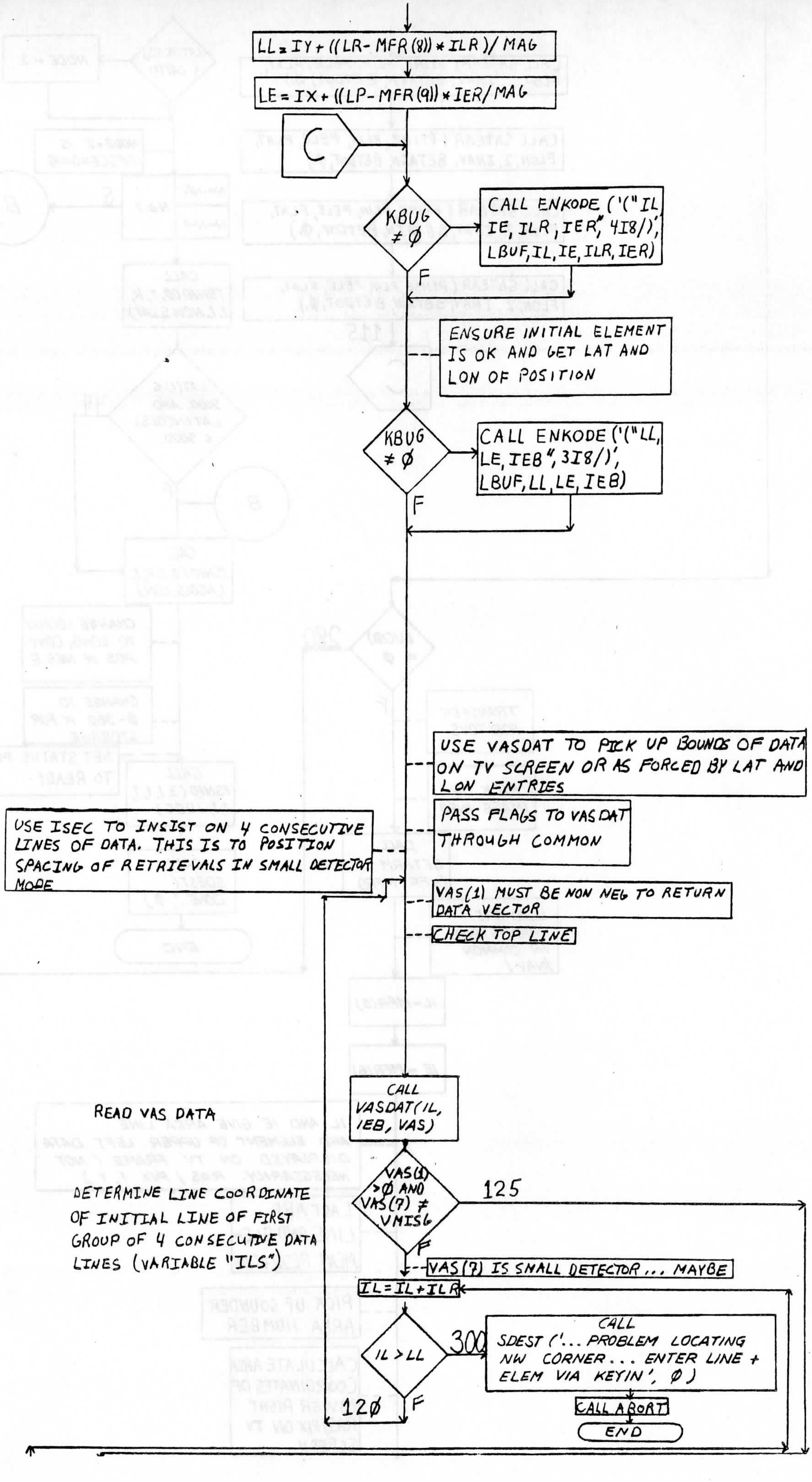
At this point, however, the program functions differently than the first case, in that the element coordinate of the NE corner of the VAS data (LETOP) is calculated, something which is not done in the initial mode of IDVA. Then, after the "...UPPER LEFT DONE" message is printed on the CRT, the lower bounds of the

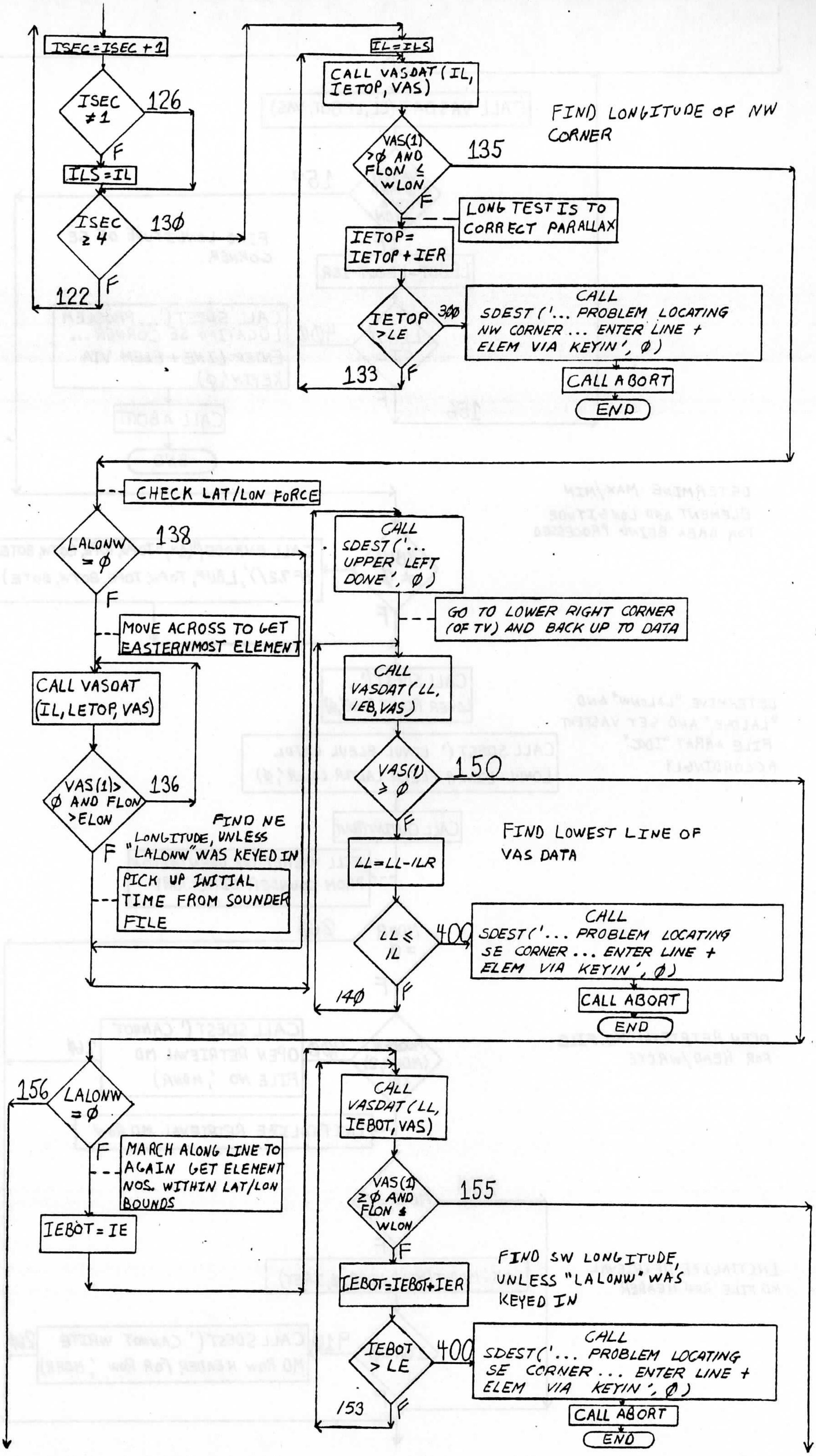
VAS data is calculated, followed by the determination of the SW and SE element coordinates. Now, IDVA is at statement 157 again, and it proceeds identically to the first mode from this point to the end.


IDVA
DATA NCOLS/56/, WLON/36Ø./, ELON/Ø./, TOPE/36Ø./, BOTW/Ø./
DATA LETOP/Ø/, IEBOT/999999/
MDNR=IPP(1,Ø)
MDRR=IPP(2,Ø)
LALONW=IPP(3,Ø)
LALOSE=IPP(4,Ø)
KBUG = IKWP('BUG',1,Ø)
READ IN RETRIEVAL MD FILE AND ROW, AS WELL AS NW AND SE LAT/LON COORDINATES
LUC(81) = Ø
10
CALL TSNIO (1,1,1,1,1,1, IDOC)
F
FOLLOWING FOR VAS
OPEN AND READ VASTEXT FILE
CALL DOPEN('VASTEXT', LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
THIS PATH WORKS ONLY ON A VIDEO TERMINAL
100
LALONW = Ø
F
CALL SDEST (' LALOSE HAS NOT BEEN SPECIFIED', Ø)
92Ø
LALOSE = Ø
END
F
FOR TOVS FILL DIRECTORY AND EXIT
LUC(81) = Ø
28Ø
F
BEGIN SEARCH FOR MAX AND MIN LAT AND LONG REQUIRED BY ORBIT, WATCH FOR MISSING LINES AT BEGINNING AND END OF ORBIT
N←5
N←N+1
N≤NROW
2
A
CALL TSNIO(Ø,1,K,1,1,NCOLS,LAT)
LAT(1) ≤ 9000. AND LAT(NCOLS) ≤ 9000
3
F
A
CALL TSNIO(Ø,2,K,1,1,NCOLS,LON)
NODE = 1
THIS PATH CAN APPLY TO EITHER VIDEO OR NON-VIDEO TERMINALS
USE NAV TO GET LINE, ELE
READ SOUNDER DIRECTORY FOR SOUNDER AREA "NSND" AND INITIALIZE NAVIGATION
CALL READD(NSND, IDIR)
CALL NVINIT(BETAIN, BETDOT, INAV, PTIME)
LOOK AT ALL CORNERS TO GET MAX AND MIN ELEMENTS

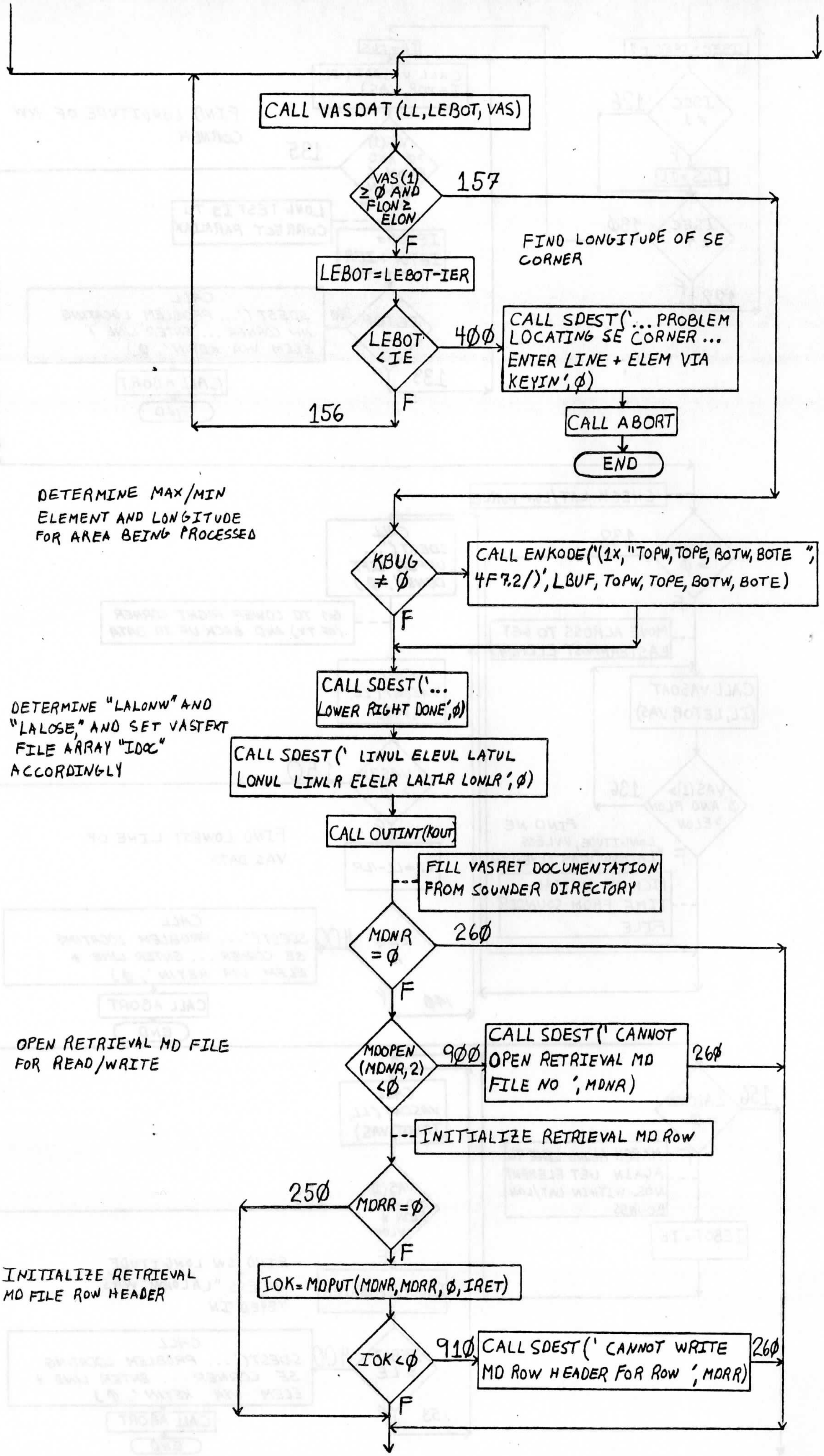


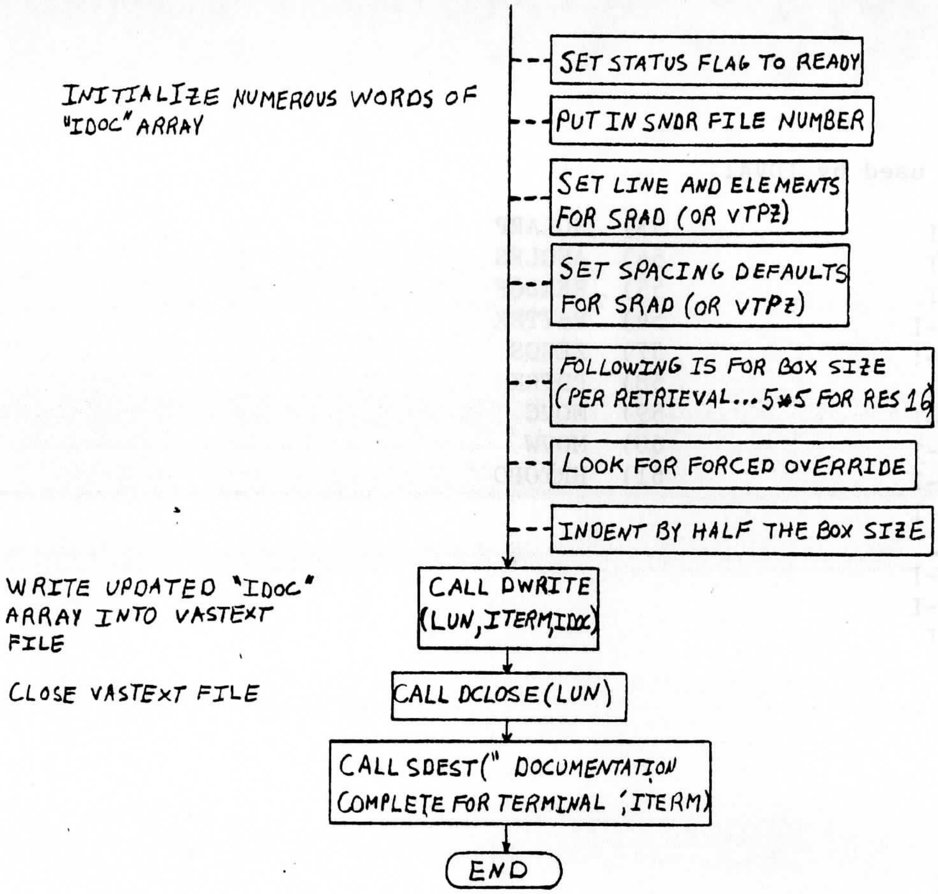
CALL SATEAR(PTIME, FLIN, FELE, FLAT, FLON, , INAV, BETAIN, BETDOT, Ø.)
CALL SATEAR (PTIME, FLIN, FELE, FLAT, FLON, 2, INAV, BETAIN, BETDOT, Ø.)
CALL SATEAR (PTIME, FLIN, FELE, FLAT, FLON, 2, INAV, BETAIN, BETDOT, Ø.)
CALL SATEAR (PTIME, FLIN, FELE, FLAT, FLON, 2, INAV, BETAIN, BETDOT, Ø.)
115
C
LUC(81) = Ø
280
F
TRANSFER FOR TOVS
PICK UP IMAGE INFO
CALL GETFRM (NFR, MFR)
LOAD TIME PARAMETERS IN COMMON /NAV/
IL=MFR(5)
IE = MFR(6)
IL AND IE GIVE AREA LINE AND ELEMENT OF UPPER LEFT DATA DISPLAYED ON TV FRAME (NOT NECESSARILY RAS / PIX 1, 1)
LAST ARE LINE AND ELEMENT RESOLUTION
PICK UP SOUNDER AREA NUMBER
CALCULATE AREA COORDINATES OF LOWER RIGHT RAS/PIX ON TV SCREEN
LAT(NCOLS) < LAT(1)
F
NODE = 2
NODE=2 IS DESCENDING
N←NR
N≥1
N←N-1
5
B
CALL TSNIO(Ø, 1, K, 1, 1, NCOLS, LAT)
LAT(1) ≤ 9000, AND LAT(NCOLS) ≤ 9000.
44
F
B
CALL TSNIO (Ø, 2, K, 1, 1, NCOLS, LON)
CHANGE LOCALLY TO LONG, CONV POS W NEG E
CHANGE TO Ø-36Ø W FOR STORAGE
SET STATUS FLAG TO READY
CALL TSNIO (3, 1, 1, 1, 1, 1, IDOC)
CALL SDEST(' DONE..', Ø)
END


LL = IY + ((LR - MFR(8)) * ILR)/MAG
LE = IX + ((LP - MFR(9)) * IER/MAG
C
KBUG ≠ Ø
CALL ENKODE ('("IL IE, ILR, IER", 4I8/)', LBUF, IL, IE, ILR, IER)
F
ENSURE INITIAL ELEMENT IS OK AND GET LAT AND LON OF POSITION
KBUG ≠ Ø
CALL ENKODE ('("LL, LE, IEB", 3I8/)', LBUF, LL, LE, IEB)
F
USE VASDAT TO PICK UP BOUNDS OF DATA ON TV SCREEN OR AS FORCED BY LAT AND LON ENTRIES
PASS FLAGS TO VASDAT THROUGH COMMON
USE ISEC TO INSIST ON 4 CONSECUTIVE LINES OF DATA. THIS IS TO POSITION SPACING OF RETRIEVALS IN SMALL DETECTOR MODE
VAS(1) MUST BE NON NEG TO RETURN DATA VECTOR
CHECK TOP LINE
READ VAS DATA
CALL VASDAT(IL, IEB, VAS)
VAS(1) >Ø AND VAS(7) ≠ VMISG
125
DETERMINE LINE COORDINATE OF INITIAL LINE OF FIRST GROUP OF 4 CONSECUTIVE DATA LINES (VARIABLE "ILS")
F
VAS(7) IS SMALL DETECTOR ... MAYBE
IL=IL+ILR
IL > LL
300
CALL SDEST ('... PROBLEM LOCATING NW CORNER ... ENTER LINE + ELEM VIA KEYIN', Ø)
CALL ABORT
END
120
F



ISEC=ISEC+1
ISEC ≠1
126
F
ILS=IL
ISEC ≥4
130
F
122
IL=ILS
CALL VASDAT(IL, IETOP, VAS)
FIND LONGITUDE OF NW CORNER
VAS(1) >Ø AND FLON ≤ WLON
135
F
LONG TEST IS TO CORRECT PARALLAX
IETOP= IETOP+IER
IETOP >LE
300
CALL SDEST('... PROBLEM LOCATING NW CORNER ... ENTER LINE + ELEM VIA KEYIN', Ø)
CALL ABORT
END
133
F
CHECK LAT/LON FORCE
LALONW =Ø
138
F
MOVE ACROSS TO GET EASTERNMOST ELEMENT
CALL VASDAT (IL, LETOP, VAS)
VAS(1)> Ø AND FLON >ELON
136
F
FIND NE LONGITUDE, UNLESS "LALONW" WAS KEYED IN
PICK UP INITIAL TIME FROM SOUNDER FILE
CALL SDEST('... UPPER LEFT DONE', Ø)
GO TO LOWER RIGHT CORNER (OF TV) AND BACK UP TO DATA
CALL VASDAT(LL, IEB, VAS)
VAS(1) ≥ Ø
150
F
FIND LOWEST LINE OF VAS DATA
LL=LL-ILR
LL ≤ IL
400
CALL SDEST('... PROBLEM LOCATING SE CORNER ... ENTER LINE + ELEM VIA KEYIN', Ø)
CALL ABORT
END
140
F
156
LALONW =Ø
F
MARCH ALONG LINE TO AGAIN GET ELEMENT NOS. WITHIN LAT/LON BOUNDS
IEBOT=IE
CALL VASDAT(LL, IEBOT, VAS)
VAS(1) ≥Ø AND FLON ≤ WLON
155
F
IEBOT=IEBOT+IER
FIND SW LONGITUDE, UNLESS "LALONW" WAS KEYED IN
IEBOT > LE
400
CALL SDEST('... PROBLEM LOCATING SE CORNER ... ENTER LINE + ELEM VIA KEYIN', Ø)
CALL ABORT
END
153
F


CALL VASDAT(LL,LEBOT,VAS)
VAS(1) ≥ Ø AND FLON ≥ ELON
157
F
FIND LONGITUDE OF SE CORNER
LEBOT=LEBOT-IER
LEBOT <IE
400
CALL SDEST('... PROBLEM LOCATING SE CORNER ... ENTER LINE + ELEM VIA KEYIN',Ø)
CALL ABORT
END
F
156
DETERMINE MAX/MIN ELEMENT AND LONGITUDE FOR AREA BEING PROCESSED
KBUG ≠ Ø
CALL ENKODE('(1X,"TOPW,TOPE,BOTW,BOTE ",4F7.2/)',LBUF,TOPW,TOPE,BOTW,BOTE)
F
CALL SDEST('... LOWER RIGHT DONE',Ø)
DETERMINE "LALONW" AND "LALOSE," AND SET VASTEXT FILE ARRAY "IDOC" ACCORDINGLY
CALL SDEST(' LINUL ELEUL LATUL LONUL LINLR ELELR LALTLR LONLR',Ø)
CALL OUTINT(KOUT)
FILL VASRET DOCUMENTATION FROM SOUNDER DIRECTORY
MDNR = Ø
260
F
OPEN RETRIEVAL MD FILE FOR READ/WRITE
MDOPEN (MDNR,2) <Ø
900
CALL SDEST(' CANNOT OPEN RETRIEVAL MD FILE NO ',MDNR)
26Ø
INITIALIZE RETRIEVAL MD ROW
25Ø
MDRR = Ø
F
INITIALIZE RETRIEVAL MD FILE ROW HEADER
IOK=MDPUT(MDNR,MDRR,Ø,IRET)
IOK<Ø
910
CALL SDEST(' CANNOT WRITE MD ROW HEADER FOR ROW ',MDRR)
26Ø
F


INITIALIZE NUMEROUS WORDS OF "IDOC" ARRAY
SET STATUS FLAG TO READY
PUT IN SNDR FILE NUMBER
SET LINE AND ELEMENTS FOR SRAD (OR VTPZ)
SET SPACING DEFAULTS FOR SRAD (OR VTPZ)
FOLLOWING IS FOR BOX SIZE (PER RETRIEVAL...5*5 FOR RES 16)
LOOK FOR FORCED OVERRIDE
INDENT BY HALF THE BOX SIZE
WRITE UPDATED "IDOC" ARRAY INTO VASTEXT FILE
CALL DWRITE (LUN,ITERM,IDOC)
CLOSE VASTEXT FILE
CALL DCLOSE(LUN)
CALL SDEST(" DOCUMENTATION COMPLETE FOR TERMINAL ',ITERM)
END

Subroutines used by IDVA:

1) DOPEN-I
2) DREAD-I
3) TSNIO-I
4) GETFRM-I
5) VASDAT-I
6) SDEST-I
7) OUTINT-I
8) DWRITE-I
9) DCLOSE-I
10) ENKODE-I
11) READD-I
12) NVINIT-I
13) SATEAR-I
14) ABORT-I
15) ENCODE
16) ENCODX
17) CONTNT
18) LTQ
19) MOVB
20) CLEANA
21) TQ
22) PRLINX
23) PRCLOS
24) PROPEN
25) PRPRPR
26) LOCK
27) PRWR
28) PRRD
29) UNLOCK
30) PRCL
31) POST
32) BLKA
33) ZECONV
34) II
35) STC
36) MOVWC
37) TQMES
38) ITOC
39) MOVCW
40) GETNAV
41) EPOCH
42) GETGAM
43) VASNAV
44) DDEST
45) CLEANW
46) EDEST
47) PLNKIV
48) OPNA
49) LWOPEN
50) LWGET
51) SATPOS
52) ANGGET
53) SOLARP
54) ANGLES
55) READOF
56) RBYTSX
57) ZEROS
58) RDTRK
59) MOVC
60) MOVW
61) HRTOPO

LOVA

The purpose of LOVA is to display the contents of the VASTEXT file for the user's local terminal. As always for VAS applications, the sounder area must be set previously by VPVA.

First, the VASTEXT file is opened and read via subroutines DOPEN and DREAD, causing such retrieval processing parameters as sounder area number (NSND), retrieval MD file number, number of retrievals processed (NRET), etc. to be passed to LOVA through the array IDOC (call DREAD(LUN, ITERM,IDOC). A further call to subroutine SDEST outputs data variable TITLE to the CRT, whereupon subroutine ENKODE outputs values corresponding to the list of variables in TITLE. Further calls to SDEST and ENKODE output still more of the VASTEXT file contents.

The list at the end of this brief discussion deals with all of the variables listed by LOVA. The program(s) in the parentheses to the immediate right of each variable indicate(s) which program(s) affect that particular variable. Furthermore, the element of the array IDOC which each variable occupies is also listed. Note that as one goes continually further in the retrieval process, executing more of the pertinent programs, more of these variables will be assigned values. Note also that program SPVA can be used at any time to change different VASTEXT file variables.

1.) YYDDD (IDVA-derived from IDOC(1), CSVA)- Julian date of VAS data

2.) BEGIN (IDVA-IDOC(2))- beginning time of VAS radiance measurements

3.) Y-RES (IDVA-IDOC(9))- "Y" coordinate resolution of infrared measurements (in kilometers)

4.) X-RES (IDVA-IDOC(10))- "X" coordinate resolution of infrared measurements (in kilometers)

5.) LLNW (IDVA-IDOC(25))- latitude and longitude of NW corner of retrieval area to be processed

6.) LLSE (IDVA-IDOC(26))- latitude and longitude of SE corner of retrieval area to be processed

7.) STAT (IDVA-IDOC(28))- status flag (is set to 1 in IDVA, and must be such for CSVA to work)

8.) NSAT (IDVA-IDOC(5))-satellite number (29 for GOES-EAST, 31 for GOES-WEST)

9.) SNDAREA (VPVA/IDVA-IDOC(35))- sounder area currently pointed to

10.) MDNS (CSVA-IDOC(36))- surface data MD file (containing Z100, TSL and DD data)

11.) MDRS (CSVA-IDOC((37))- row of surface data MD file that pertinent surface data is contained in (time dependent)

12.) MDNG (GSVA-IDOC(38))- guess MD file (upper air)

13.) MDRG (GSVA-IDOC(39))- row number of guess MD file (as many rows as needed are filled, so this variable is usually not important)

14.) MDNR (IDVA-IDOC(40))- retrieval MD file number

15.) MDRR (IDVA-IDOC(41))- retrieval MD file row number

16.) NGFG (GSVA-IDOC(42))- guess grid file (upper air) number

17.) NGFS (SRVA-IDOC(29))- grid file containing 3 surface grids of Z100, TSL and DD

18.) ZGRID (SRVA-IDOC(30))- grid of Z100 (1000 mb heights)

19.) TGRID (SRVA-IDOC(31))- grid of TSL (sea level temperatures)

20.) DGRID (SRVA-IDOC(32))- grid of DD (surface dewpoint depressions)

21.) NO. RETRIEVALS (SRAD/VTPZ-IDOC(100))- number of retrievals processed

22.) TYP (IDVA and SRAD-IDOC(50))- type of retrieval to be done

23.) GSS (IDVA, SRAD and SRET-IDOC(51))- type of guess to be used for the retrieval

24.) SPC (IDVA and SRAD-IDOC(52))- spacing of attempted retrievals (in fields of view)

25.) SIZ (IDVA and SRAD-IDOC(53))- length of one side of a retrieval box in fields of view (retrieval boxes are usually 11*11 fields of view for small detector data, and 5*5 fields of view for large detector data)

26.) SFC (IDVA, SRAD and SRET-IDOC(54))- surface option

27.) ENDL (IDVA-IDOC(55))- ending line of retrieval area

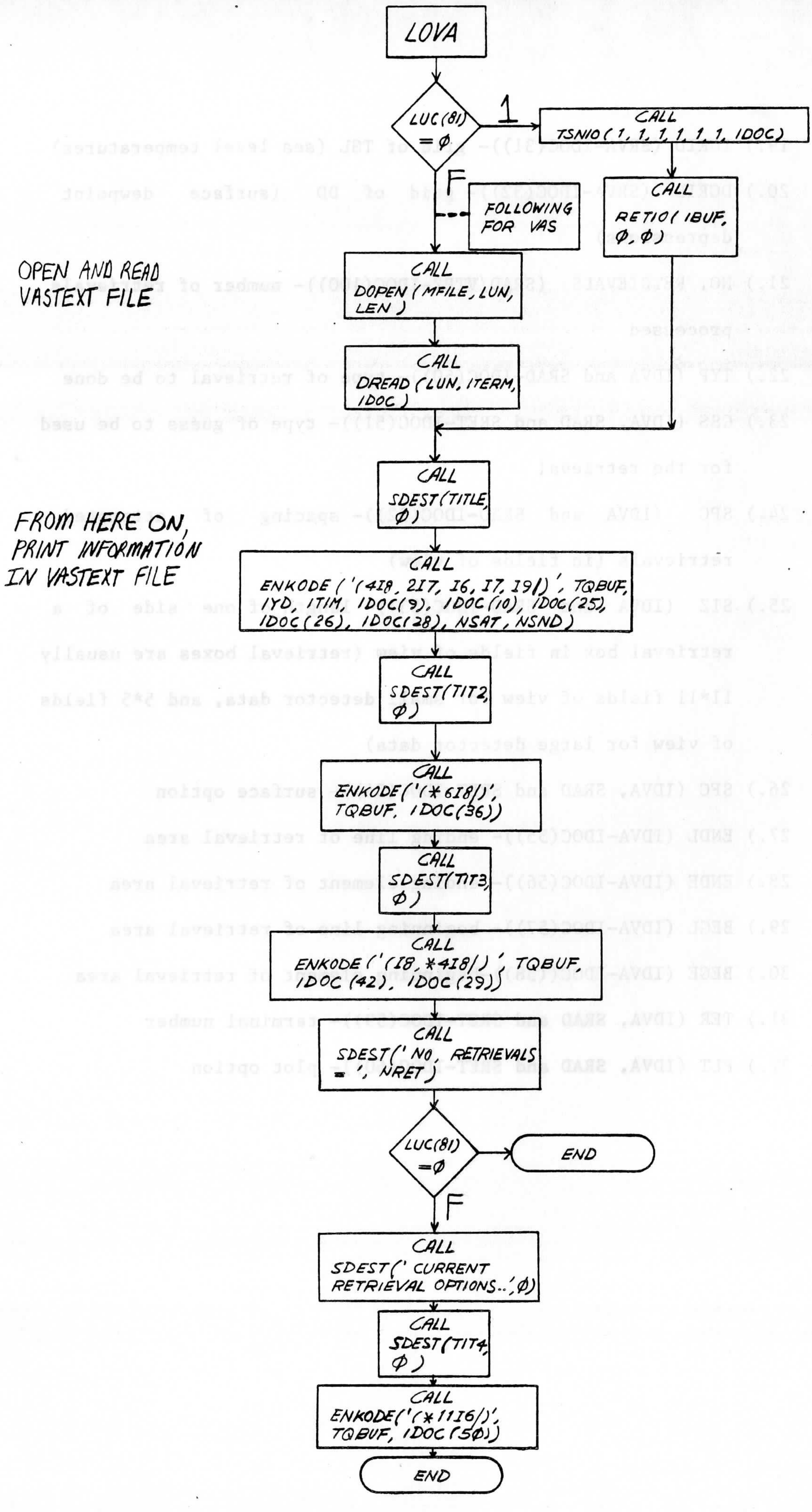
28.) ENDE (IDVA-IDOC(56))- ending element of retrieval area

29.) BEGL (IDVA-IDOC(57))- beginning line of retrieval area

30.) BEGE (IDVA-IDOC((58))-beginning element of retrieval area

31.) TER (IDVA, SRAD and SRET-IDOC(59))- terminal number

32.) PLT (IDVA, SRAD and SRET-IDOC(60))- plot option


LOVA
LUC(81) = Ø
1
CALL TSNIO(1, 1, 1, 1, 1, 1, IDOC)
CALL RETIO(IBUF, Ø, Ø)
F
FOLLOWING FOR VAS
OPEN AND READ VASTEXT FILE
CALL DOPEN(MFILE, LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
CALL SDEST(TITLE, Ø)
FROM HERE ON, PRINT INFORMATION IN VASTEXT FILE
CALL ENKODE('(4I8, 2I7, I6, I7, I9/)', TQBUF, IYD, ITIM, IDOC(9), IDOC(10), IDOC(25), IDOC(26), IDOC(28), NSAT, NSND)
CALL SDEST(TIT2, Ø)
CALL ENKODE('(*6I8/)', TQBUF, IDOC(36))
CALL SDEST(TIT3, Ø)
CALL ENKODE('(I8, *4I8/)', TQBUF, IDOC(42), IDOC(29))
CALL SDEST(' NO, RETRIEVALS =', NRET)
LUC(81) = Ø
END
F
CALL SDEST(' CURRENT RETRIEVAL OPTIONS..', Ø)
CALL SDEST(TIT4, Ø)
CALL ENKODE('(*11I6/)', TQBUF, IDOC(5Ø))
END

Subroutines used by LOVA:

1) DOPEN-I
2) DREAD-I
3) TSNIO-I
4) RETIO-I
5) SDEST-I
6) ENKODE-I
7) ENCODE
8) ENCODX
9) ABORT
10) CONTNT
11) LTQ
13) ZECONV
14) MOVB
15) CLEANA
16) TQ
17) PRLINX
18) PRCLOS
19) PROPEN
20) PRPRPR
21) LOCK
22) PRWR
23) PRRD
24) UNLOCK
25) PRCL
26) POST
27) BLKA
28) II
29) STC
30) TQMES
31) ITOC
32) MOVCW

SPVA

SPVA enters keyed in values into either the VASTEXT file or the TOVS documentation file. Specifically for VAS, it can be used to set (or change) MD and gridfile pointers, number of retrievals, grid numbers within the surface gridfile, etc. for VAS retrieval processing. A list of the quantities that can be manipulated is given at the end of this discussion.

First, the VASTEXT file is opened and read via subroutines DOPEN and DREAD. Then, the NW and SE latitude/longitude coordinates of the image area being processed are set or changed in the VASTEXT file (IF (LLNW.NE.0) IDOC(25)=LLNW,etc.). Next, the MD pointers (files or rows) within the VASTEXT file are changed, depending on which keywords and their respective values have been entered in the original keyin (IDOC(35+M)=MESS). Following this, the VASTEXT file words corresponding to the surface data gridfile and its three associated grids (NGFS, ZGRID, TGRID and DGRID) are changed.

In the next stage of the program, the pointer for the first guess gridfile (NGFG) is checked and changed if necessary, and the results up to this point are written into the VASTEXT file via subroutine DWRITE.

In the final step, the keyword parameter for the number of retrievals is checked. If nothing has been keyed in for this parameter (NRET= -1), SPVA is done. Otherwise, the retrieval MD file row header is accessed, which contains the old value of NRET. Then, the respective words of the arrays IRET and IDOC are updated with the new keyed-in value of NRET, after which a call to subroutine VRTIO (CALL VRTIO(IRET,0,1)) writes the updated row header back into row MDRR of the retrieval MD file, and the updated IDOC array back into the VASTEXT file. At this point, SPVA is finished.

List of quantities that can be set by SPVA:

1.) NRET- number of retrievals

2.) MDNS- MD file for surface data

3.) MDRS- row of surface data MD file

4.) MDNG- MD file for the first guess

5.) MDRG- row of first guess MD file

6.) MDNR- MD file for the VAS retrievals

7.) MDRR- row of VAS retrieval MD file

8.) NGFG- gridfile for the first guess

9.) NGFS- gridfile for surface data

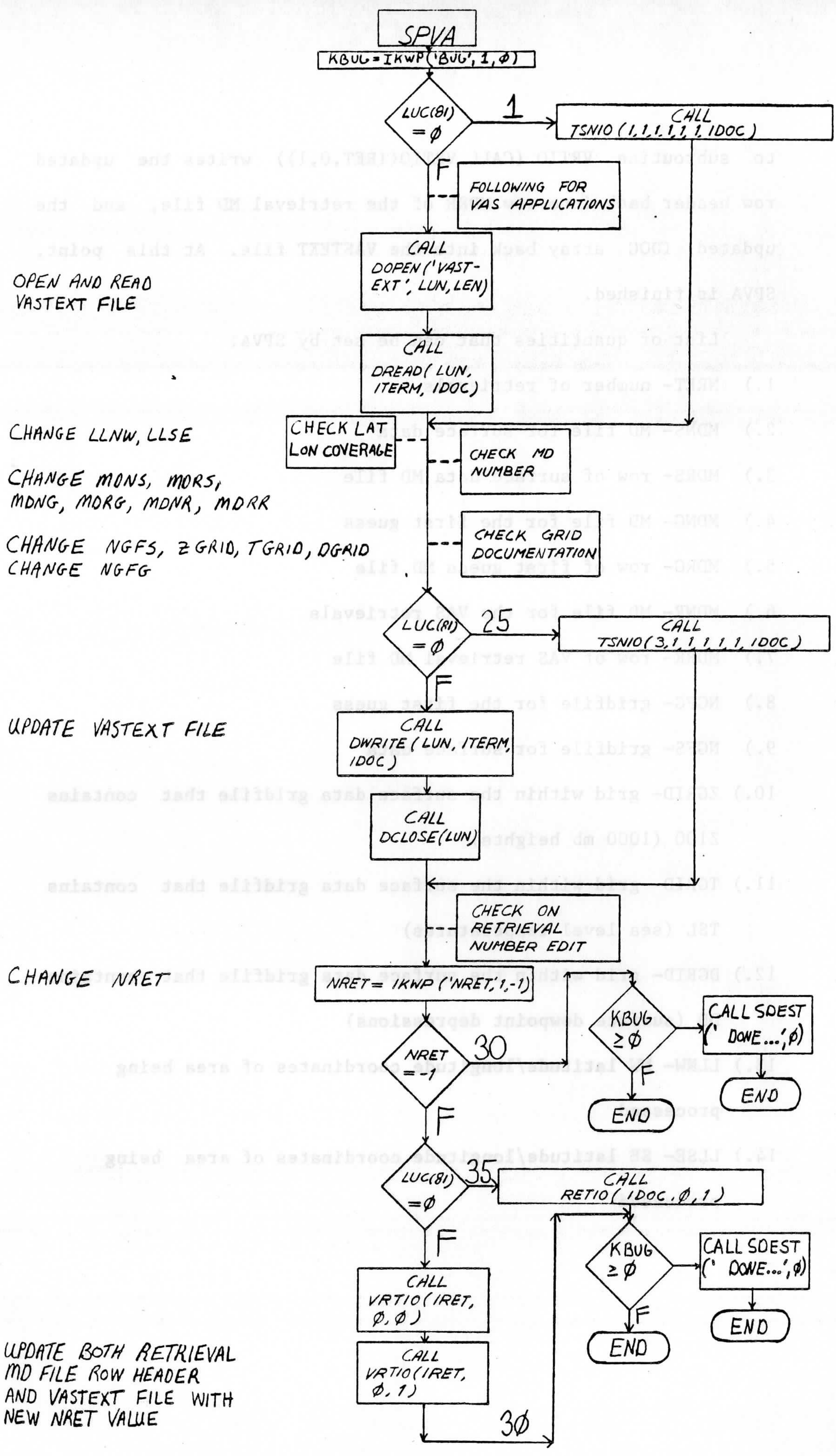
10.) ZGRID- grid within the surface data gridfile that contains Z100 (1000 mb heights)

11.) TGRID- grid within the surface data gridfile that contains TSL (sea level temperatures)

12.) DGRID- grid within the surface data gridfile that contains DD (surface dewpoint depressions)

13.) LLNW- NW latitude/longitude coordinates of area being processed

14.) LLSE- SE latitude/longitude coordinates of area being processed


SPVA
KBUG=IKWP('BUG', 1, Ø)
LUC(81) = Ø
1
CALL TSNIO (1,1,1,1,1,1, IDOC)
F
FOLLOWING FOR VAS APPLICATIONS
OPEN AND READ VASTEXT FILE
CALL DOPEN('VAST-EXT', LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
CHANGE LLNW, LLSE
CHECK LAT LON COVERAGE
CHECK MD NUMBER
CHANGE MONS, MORS, MDNG, MORG, MDNR, MDRR
CHANGE NGFS, ZGRID, TGRID, DGRID
CHANGE NGFG
CHECK GRID DOCUMENTATION
LUC(81) = Ø
25
CALL TSNIO(3,1,1,1,1,1, IDOC)
F
UPDATE VASTEXT FILE
CALL DWRITE (LUN, ITERM, IDOC)
CALL DCLOSE(LUN)
CHECK ON RETRIEVAL NUMBER EDIT
CHANGE NRET
NRET= IKWP ('NRET', 1,-1)
KBUG ≥ Ø
CALL SDEST (' DONE...', Ø)
NRET = -1
30
F
END
END
F
LUC(81) = Ø
35
CALL RETIO (IDOC, Ø, 1)
F
KBUG ≥ Ø
CALL SDEST (' DONE...', Ø)
CALL VRTIO(IRET, Ø, Ø)
CALL VRTIO(IRET, Ø, 1)
F
END
END
UPDATE BOTH RETRIEVAL MD FILE ROW HEADER AND VASTEXT FILE WITH NEW NRET VALUE
3Ø

Subroutines used by SPVA:

1) DOPEN-I
2) DREAD-I
3) TSNIO-I
4) DWRITE-I
5) DCLOSE-I
6) VRTIO-I
7) RETIO-I
8) SDEST-I
9) ENCODE
10) ABORT
11) ENCODX
12) CONTNT
13) LTQ
14) ZECONV
15) MOVB
16) CLEANA
17) TQ
18) PRLINX
19) PRCLOS
20) PROPEN
21) PRPRPR
22) LOCK
23) PRWR
24) PRRD
25) UNLOCK
26) PRCL
27) POST
28) BLKA
29) II
30) STC
33) ITOC
34) MOVCW
35) TQMES

GSVA

This program prepares the upper air guess MD file by reformatting the first guess grids in a particular grid file into guess vectors within the guess MD file. Each row and column coordinate within the guess MD file refers to a guess vector at particular latitude and longitude coordinates. Upon completion of the reformatting, the VASTEXT file is updated to show both the original guess gridfile and the resulting guess MD file numbers.

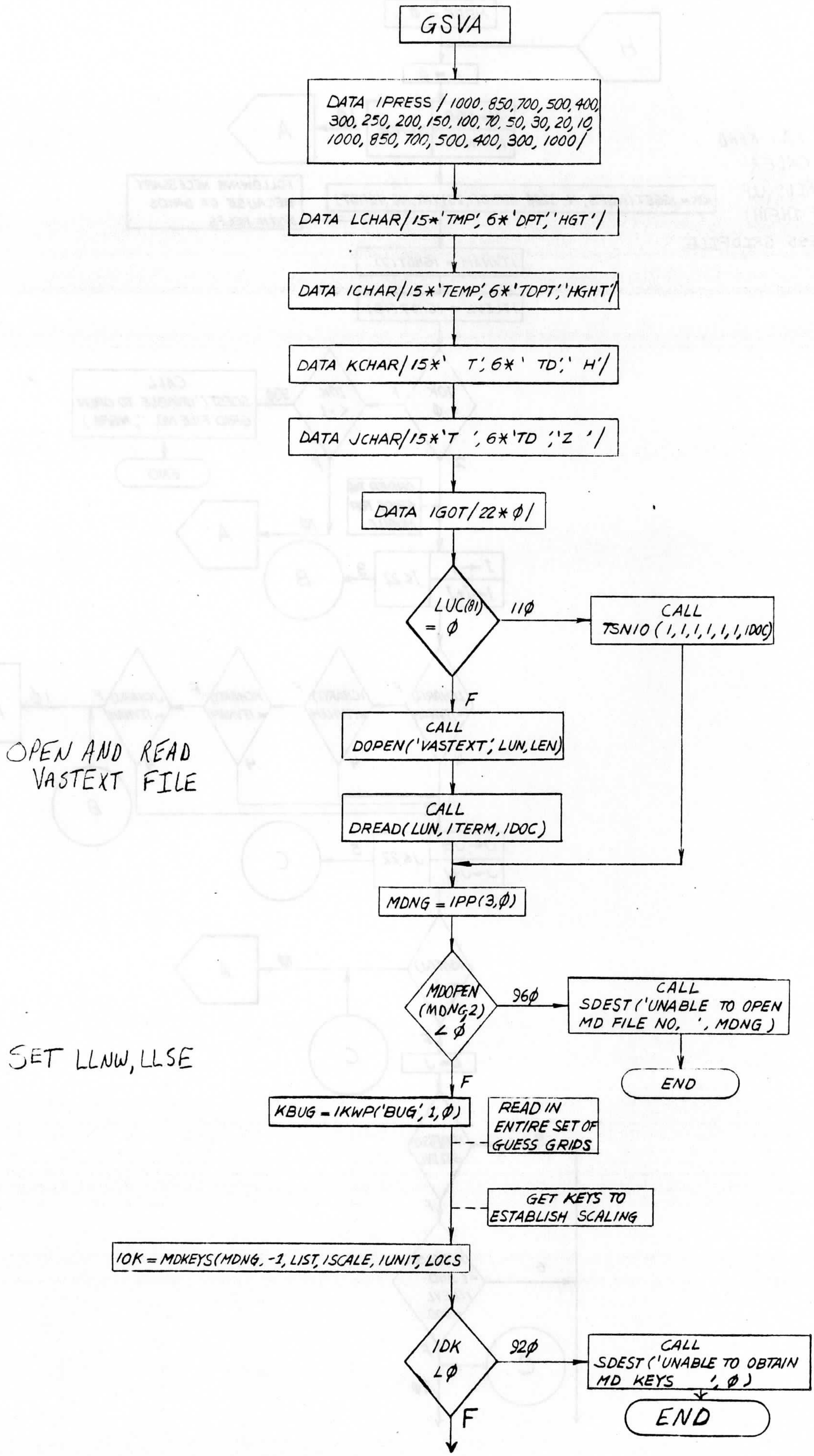
The first major step is to read the first guess gridfile (NGFG) and then open and read the VASTEXT file. Then, the first grid to be reformatted into the guess MD file, as well as the guess MD file itself, are placed into variables NGB and MDNG. Next, after such quantities as the NW and SE guess MD file boundaries (VASTEXT defaults), lat/lon increment and debug option are set, but before the guess grids themselves are read in DO LOOP 10, function MDKEYS returns the keys for the guess MD file. This step is important because it allows GSVA to establish a relationship between grid and MD file scaling (variable SCL), which is calculated later in the program. Scaling relates to the way variables are stored in a given file. For instance, if temperature is stored as degrees Kelvin * 100 in the guess gridfile, but it is to be stored as degrees Kelvin * 1000 in the guess MD file, then the relationship established via SCL will assure that the correct data is being stored, at least in terms of order of magnitude, in the guess MD file.

Following the reading of the keys, the guess grids are read from guess gridfile NGFG in DO LOOP 10 via function IGGET, one at a time, and ordered for the guess MD file. Variable IOK will equal 0 for each grid read correctly. As many as 22 grids can be accessed from the guess gridfile. Among these 22 grids are 15 grids of temperature from 1000-300 mb, 6 grids of dewpoint temperature from 1000-300 mb, and a grid of 1000 mb heights. The ordered grids are placed in the array IGRID (IGRID(J,L)=IGRID(J,23)), with each grid occupying one column of this array. Upon completion of this task, if the debug option is on (KBUG .NE. 0), information concerning the day, hour (time of model initialization), grid level, etc. is displayed on the CRT via subroutine OUTINT. Note that GSVA will exit DO LOOP 10 if variable NGOT is set to 22 at any point during the execution of the loop (IF (NGOT.EQ.22) GO TO 12).

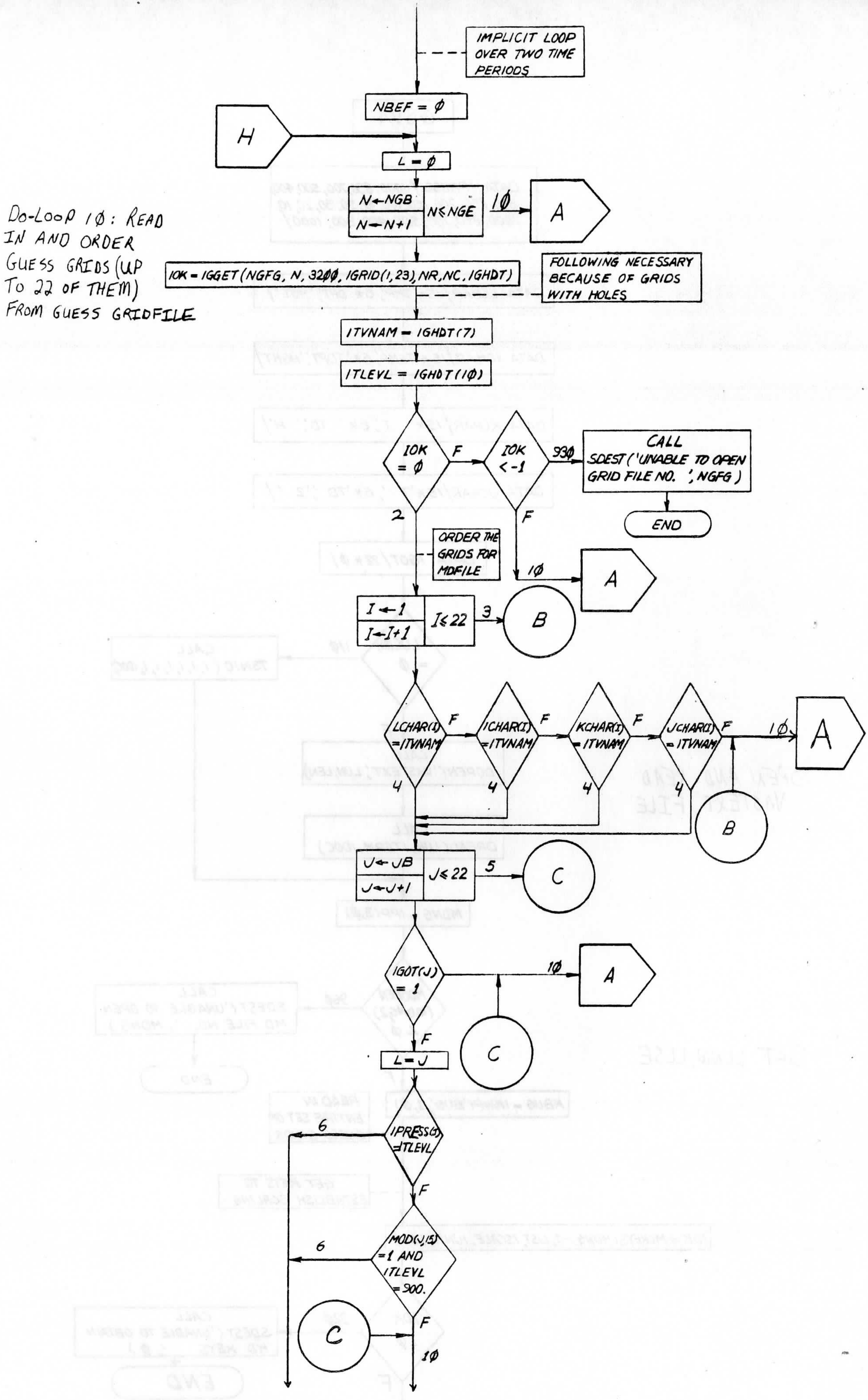
After all the guess grids have been accessed from the guess gridfile, three messages are displayed via the three calls to subroutine SDEST immediately after DO LOOP 10. Then, the lat/lon bounds accessed previously from the VASTEXT file (or explicitly keyed in via positional parameters 4 and 5) are used to generate latitude/longitude extents (variables LAN, LAS, LOW and LOE) for the guess MD file. These extents are limited (obviously) to the extents of the guess grids (see DO LOOPs 50 and 40). DO LOOPs 50 and 40 make up the vector of 26 pieces of guess information (day, hour (of model initialization), latitude, longitude, 15 temperatures (T), 6 dewpoints (TD) and the 1000 mb height (Z100)) for the guess MD file at a particular row and column (latitude and longitude) coordinate. The actual loading of the data into the guess MD file is accomplished by function MDPUT

(IOK=MDPUT(MDNG,NN,MM,IREC)).

The data is reformatted row by row. In other words, starting in the upper left corner of the guess area, data is loaded in increments toward the right, and when that row is finished, data is loaded from the far left in the second row. In addition, if one needs a first guess which lies between two available guess grid times, a time interpolation option updates the earlier 22 grid values of T,TD and Z100 to the correct time by using both time sets of grid values and an interpolation factor (DT). (FINC=DT*FLOAT(JREC(K)-IREC(K)); IREC(K)=IREC(K)+FINC). Finally, the guess MD file is closed, and the VASTEXT file is updated with the day, a quantity combining hour and valid time, guess MD file and row numbers, and the guess gridfile number, via subroutine DWRITE. A subsequent execution of program LOVA will show that both the guess gridfile and guess MD file numbers have been filled in the VASTEXT file (MDRG never set, so takes on a value of 0 by default).

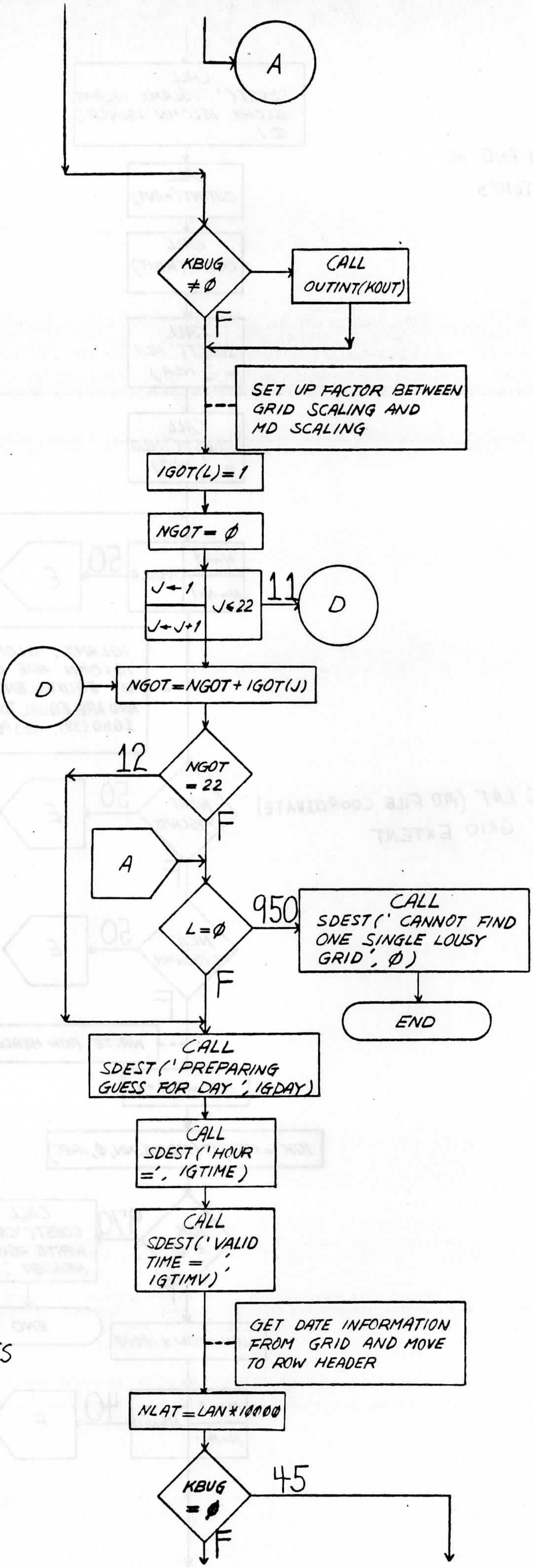


GSVA
DATA IPRESS / 1000, 850, 700, 500, 400, 300, 250, 200, 150, 100, 70, 50, 30, 20, 10, 1000, 850, 700, 500, 400, 300, 1000/
DATA LCHAR/15*'TMP', 6*'DPT', 'HGT'/
DATA ICHAR/15*'TEMP', 6*'TDPT', 'HGHT'/
DATA KCHAR/15*'   T', 6*' TD', '   H'/
DATA JCHAR/15*'T   ', 6*'TD ', 'Z   '/
DATA IGOT/22*Ø/
LUC(81) = Ø
11Ø
CALL TSNIO (1, 1, 1, 1, 1, 1, IDOC)
F
CALL DOPEN('VASTEXT', LUN, LEN)
OPEN AND READ VASTEXT FILE
CALL DREAD(LUN, ITERM, IDOC)
MDNG = IPP(3, Ø)
MDOPEN (MDNG, 2) < Ø
96Ø
CALL SDEST ('UNABLE TO OPEN MD FILE NO. ', MDNG)
END
SET LLNW, LLSE
F
KBUG = IKWP('BUG', 1, Ø)
READ IN ENTIRE SET OF GUESS GRIDS
GET KEYS TO ESTABLISH SCALING
IOK = MDKEYS(MDNG, -1, LIST, ISCALE, IUNIT, LOCS
IOK < Ø
92Ø
CALL SDEST ('UNABLE TO OBTAIN MD KEYS ', Ø)
END
F



IMPLICIT LOOP OVER TWO TIME PERIODS
NBEF = Ø
H
L = Ø
N←NGB
N←N+1
N≤NGE
1Ø
A
Do-Loop 1Ø: READ IN AND ORDER GUESS GRIDS (UP TO 22 OF THEM) FROM GUESS GRIDFILE
IOK = IGGET(NGFG, N, 32ØØ, IGRID(1,23), NR, NC, IGHDT)
FOLLOWING NECESSARY BECAUSE OF GRIDS WITH HOLES
ITVNAM = IGHDT(7)
ITLEVL = IGHDT(1Ø)
IOK = Ø
F
IOK < -1
93Ø
CALL SDEST('UNABLE TO OPEN GRID FILE NO. ', NGFG)
END
2
F
ORDER THE GRIDS FOR MDFILE
1Ø
A
I←1
I←I+1
I≤22
3
B
LCHAR(I) =ITVNAM
F
ICHAR(I) =ITVNAM
F
KCHAR(I) =ITVNAM
F
JCHAR(I) =ITVNAM
F
1Ø
A
4
4
4
4
B
J←JB
J←J+1
J≤22
5
C
IGOT(J) = 1
1Ø
A
F
C
L=J
6
IPRESS(J) =ITLEVL
F
6
MOD(J,15) =1 AND ITLEVL =900.
F
C
1Ø

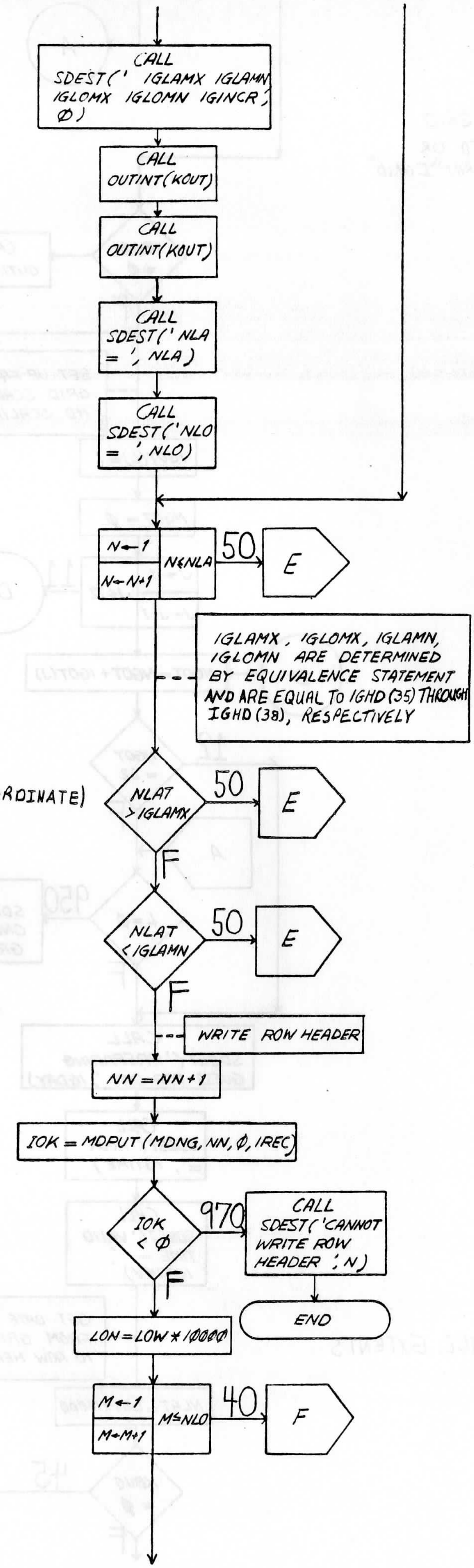
LOAD GUESS GRID
VALUES OF T, TD OR
Z100 INTO ARRAY "IGRID"



SET MD FILE EXTENTS

PRINT GRID AND MD FILE EXTENTS

MAKE SURE LAT (MD FILE COORDINATE) IS WITHIN GRID EXTENT

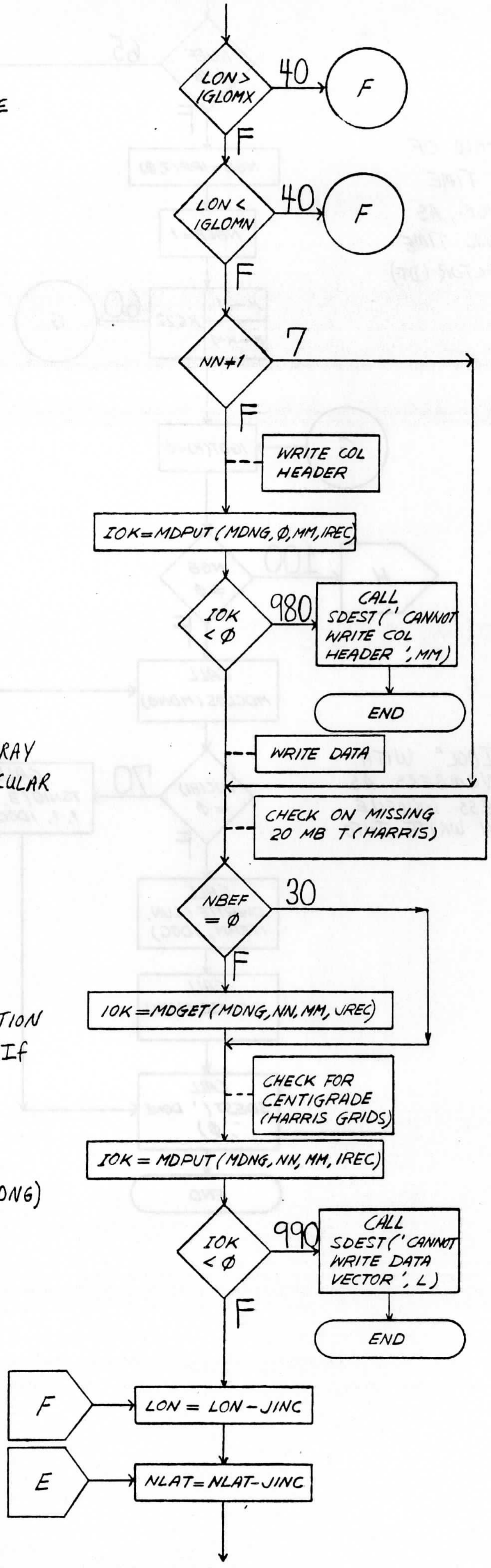


MAKE SURE LON (MD FILE COORDINATE) IS WITHIN GRID EXTENT

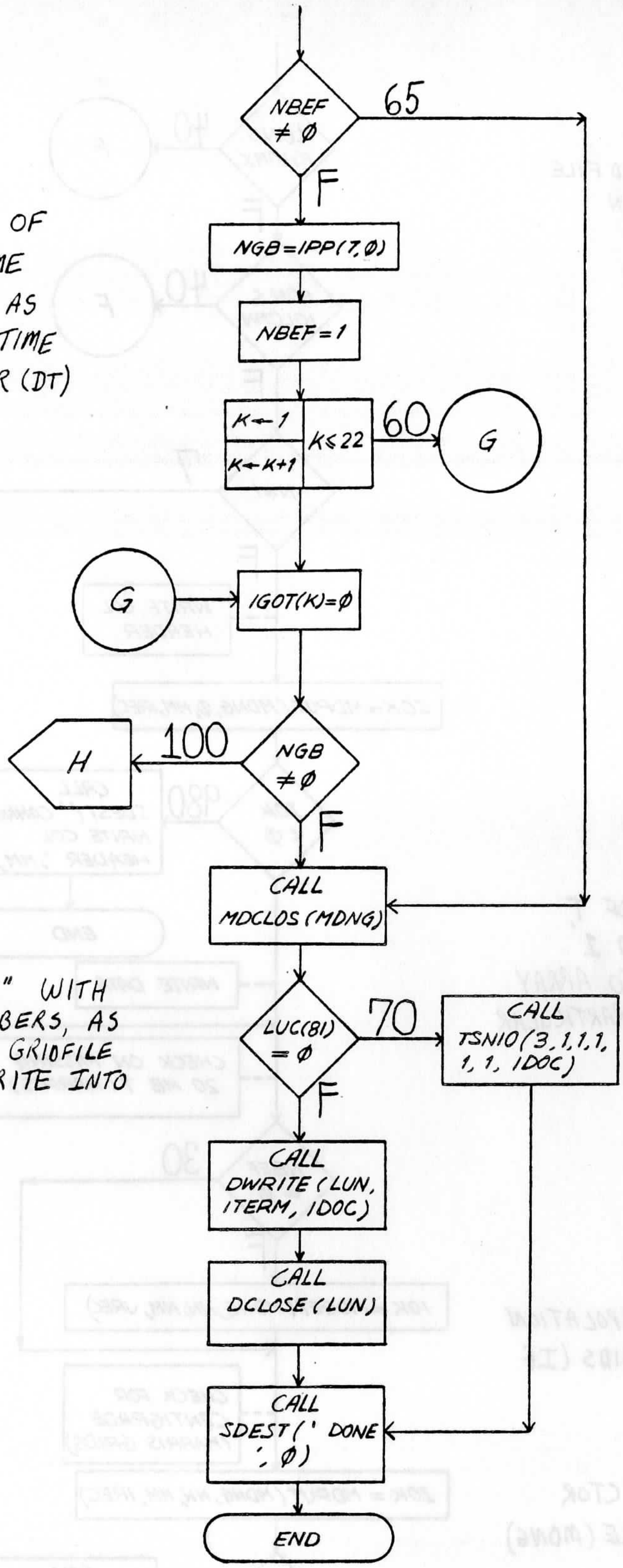
WRITE 15 VALUES OF T, 6 VALUES OF TD AND 1 VALUE OF Z100 INTO ARRAY "IREC" FOR ONE PARTICULAR MD FILE LOCATION

DO TIME INTERPOLATION BETWEEN TWO GRIDS (IF DESIRED)

STORE GUESS VECTOR IN GUESS MD FILE (MDNG)



READ IN FIRST GRID OF EARLIER SET FOR TIME INTERPOLATION (NGB), AS WELL AS FRACTIONAL TIME INTERPOLATION FACTOR (DT)



UPDATE ARRAY "IDOC" WITH GUESS MD FILE NUMBERS, AS WELL AS THE GUESS GRIDFILE NUMBER, AND THEN WRITE INTO VASTEXT FILE

Subroutines used by GSVA:

1) DOPEN-I
2) DREAD-I
3) TSNIO-I
4) OUTINT-I
5) SDEST-I
6) MDCLOS-I
7) DWRITE-I
8) DCLOSE-I
9) ENCODE
10) ENCODX
11) ABORT
12) CONTNT
13) ZECONV
14) LTQ
15) MOVB
16) CLEANA
17) TQ
18) PRLINX
19) PRCLOS
20) PROPEN
21) PRPRPR
22) LOCK
23) PRWR
24) PRRD
25) UNLOCK
26) PRCL
27) POST
28) BLKA
29) II
30) STC
31) LWCLOS
32) MOVWC
33) LWMOP
34) MOVCW
35) ITOC
36) LWPO
37) EDEST
39) MOVW
39) CLEANW
40) LWNEWF
41) LWSO
42) JMBWTF

CSVA

CSVA takes surface data from a given MD file, calculates surface dewpoint depression, sea level temperature and 1000 mb height for each station location, and places the resulting data into a surface MD file (MDNS).

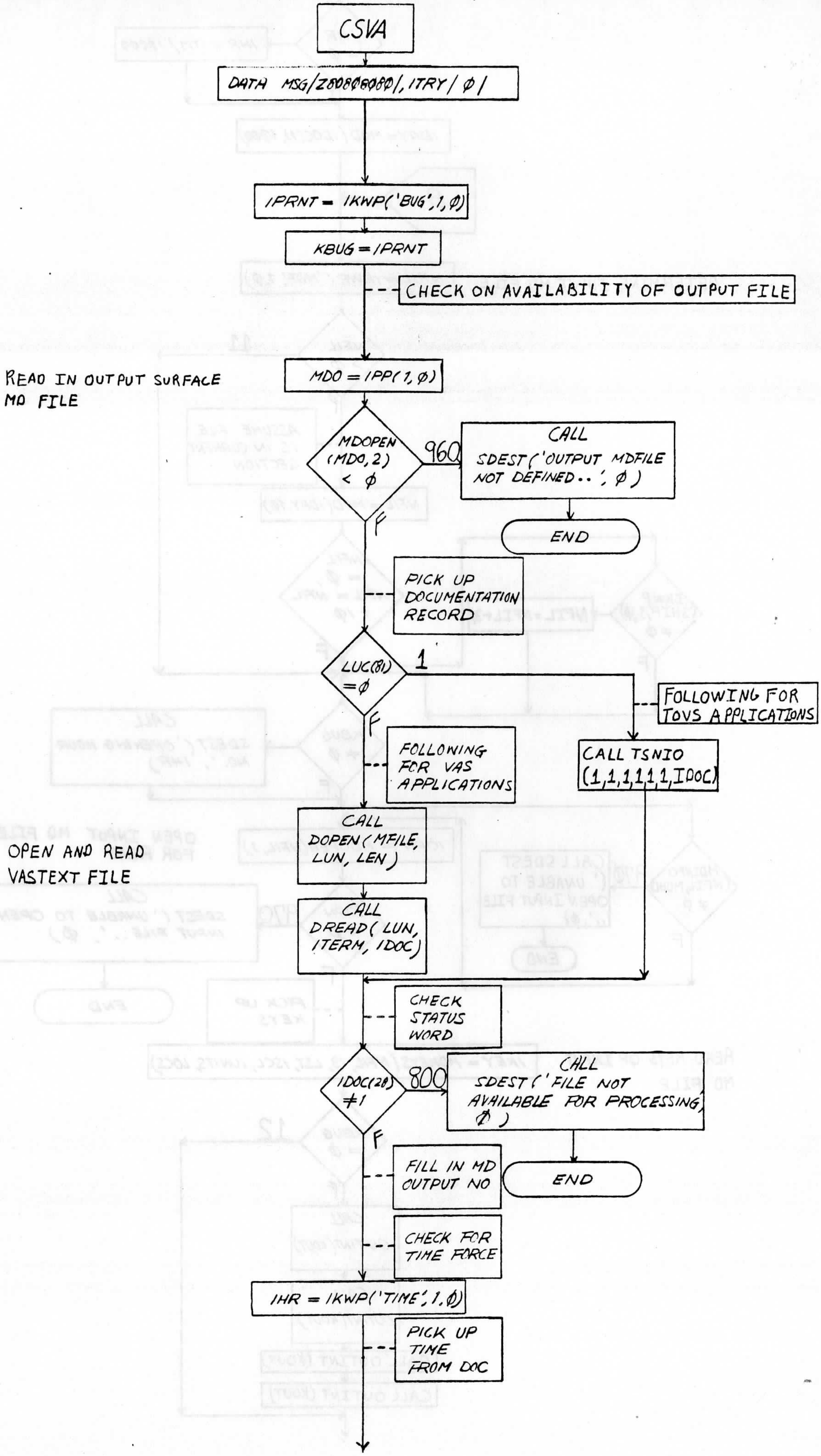
After the output MD file (MDO) is opened for read/write and the existing documentation (VASTEXT file) accessed, the status word (IDOC(28)) is checked (was previously set to 1 by IDVA). Then, after the array IDOC is updated with the output MD file number, the MD file from which the conventional surface data is to be taken for the calculations is determined (NFIL=IKWP('MDF',1,0)). If the input MD file is not specified, it will be determined from variable IDAY, and can be either an MD file very close in time to the VAS retrievals being processed, or a file from an entirely different day. After the input MD file has been determined, it is opened for reading, the keys for the file are read via function MDKEYS, and default latitude and longitude limits are set up via IDOC(25) and (26) (LLNW=IDOC(25); LLSE=IDOC(26)). These limits are used to define the extent from which the raw input MD file observations can be taken. Note that the default limits can be overridden by keyed-in values via keyword parameters LAT and LON. Next, the correct row of the input data MD file for the date and time desired is accessed (note that if keyword parameter AUTO is used, the first row of the correct day will be picked), and a row header for row ITIME of output MD file MDO is written. Then, the program begins gathering station data in implicit DO LOOP 40 from row ISV of input MD file NFIL using function MDGET (IOK=MDGET(NFIL,ISV,NREP,IBUF)), making sure the station is within the latitude-longitude bounds set earlier.

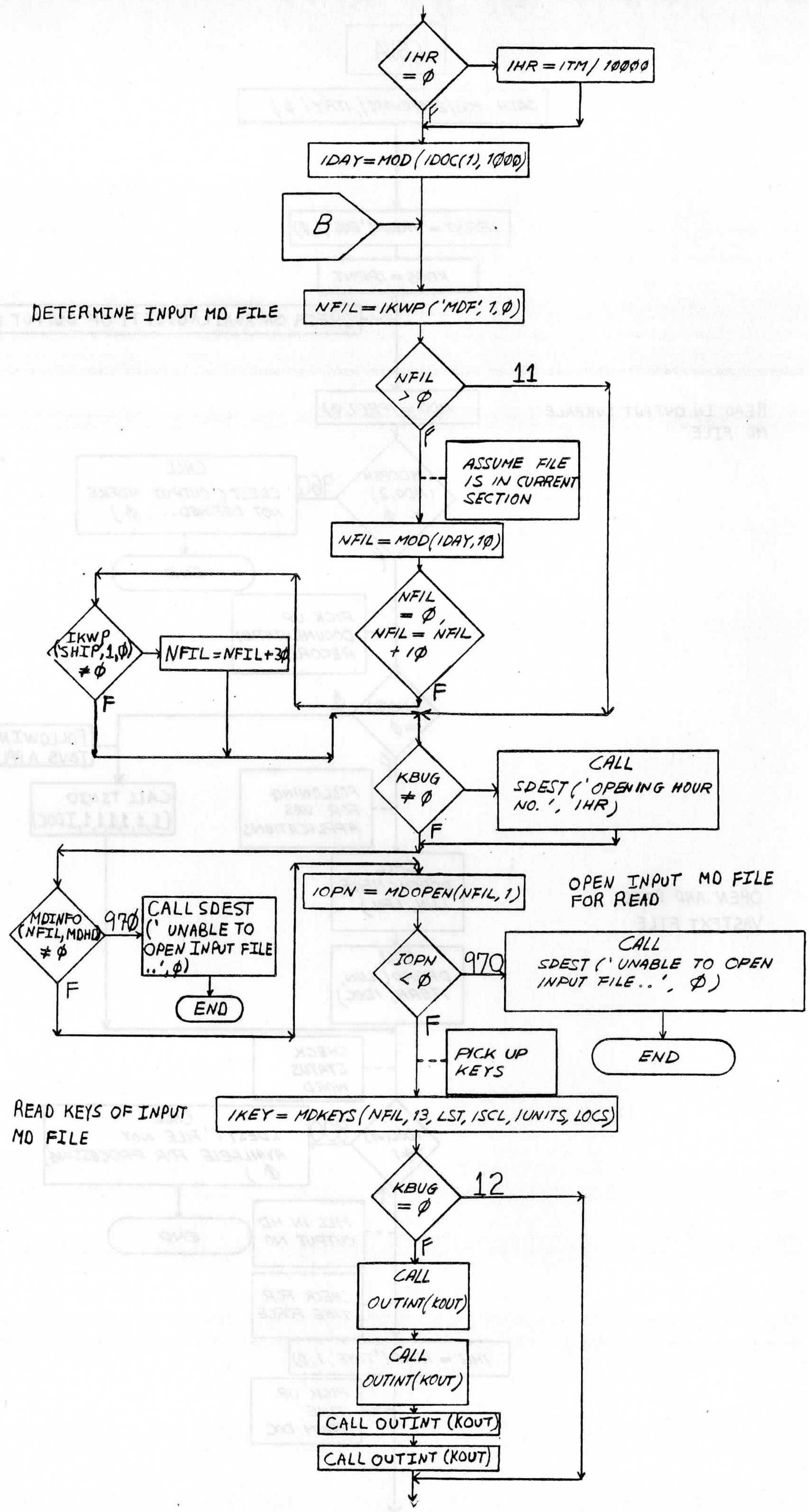
For each station whose data is read, the station dewpoint depression, sea level temperature, and 1000 mb height are calculated. Note that a missing station temperature value will result in none of these quantities being calculated for a given report, and that there must be station temperature and dewpoint values, as well as a sea level pressure value, for all three of the quantities to be determined.

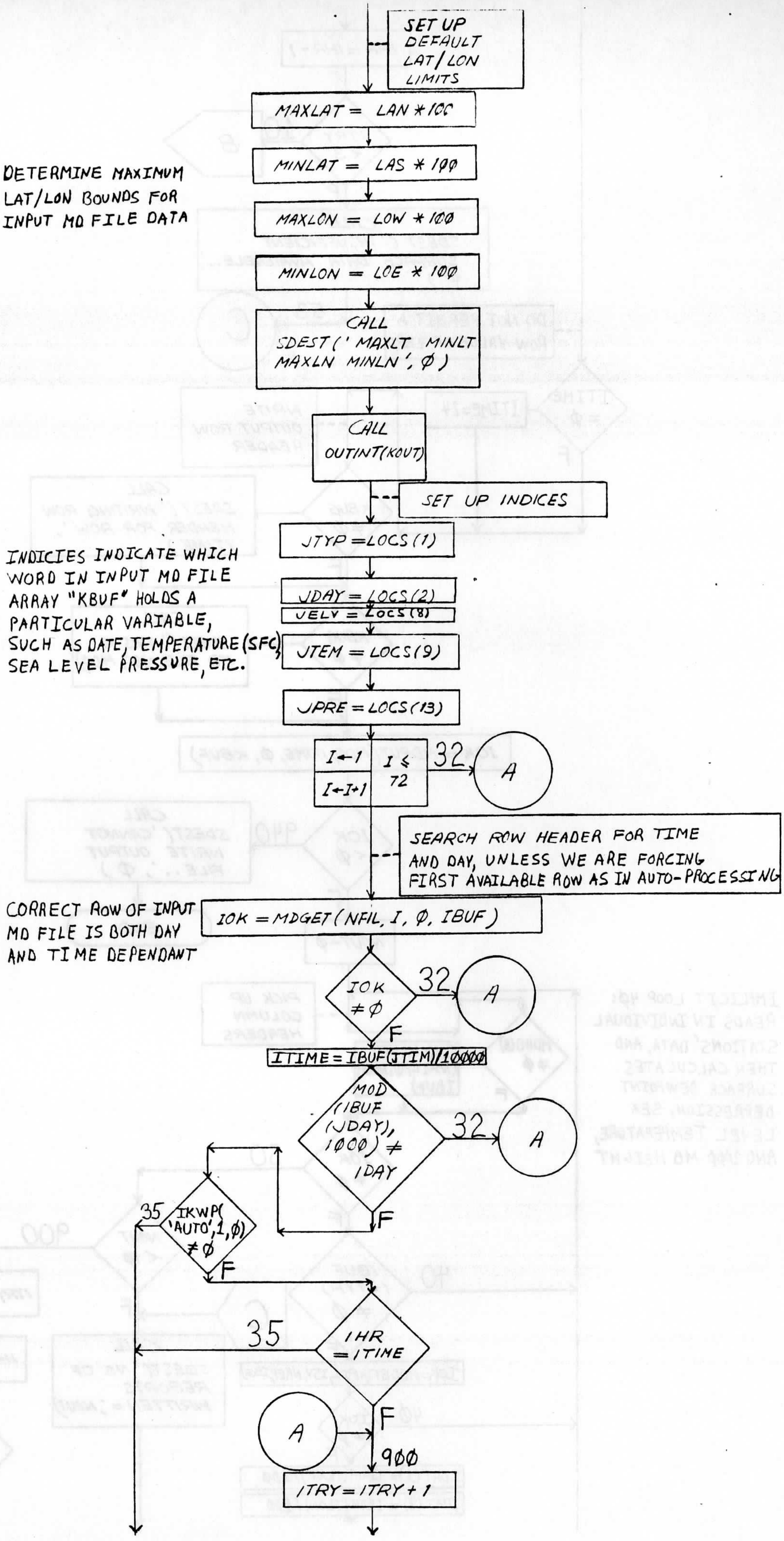
The results for each station (the array KBUF) are placed in row ITIME of output MD file MDO via function MDPUT (IOK=MDPUT(MDO,ITIME,NOUT,KBUF)), and the program goes back to statement 40 to get another station's data. Note that provisions are made in case there are not enough station reports accessed (variable NOUT after code line "50 CONTINUE"). CSVA will try to get surface data from 1, or possibly 2 hours earlier than ITIME, if necessary. Then, if it does not find at least 10 surface reports for either earlier time, a message saying " INSUFFICIENT SURFACE DATA AVAILABLE.." will be printed on the CRT, and the program will transfer to statement 53.

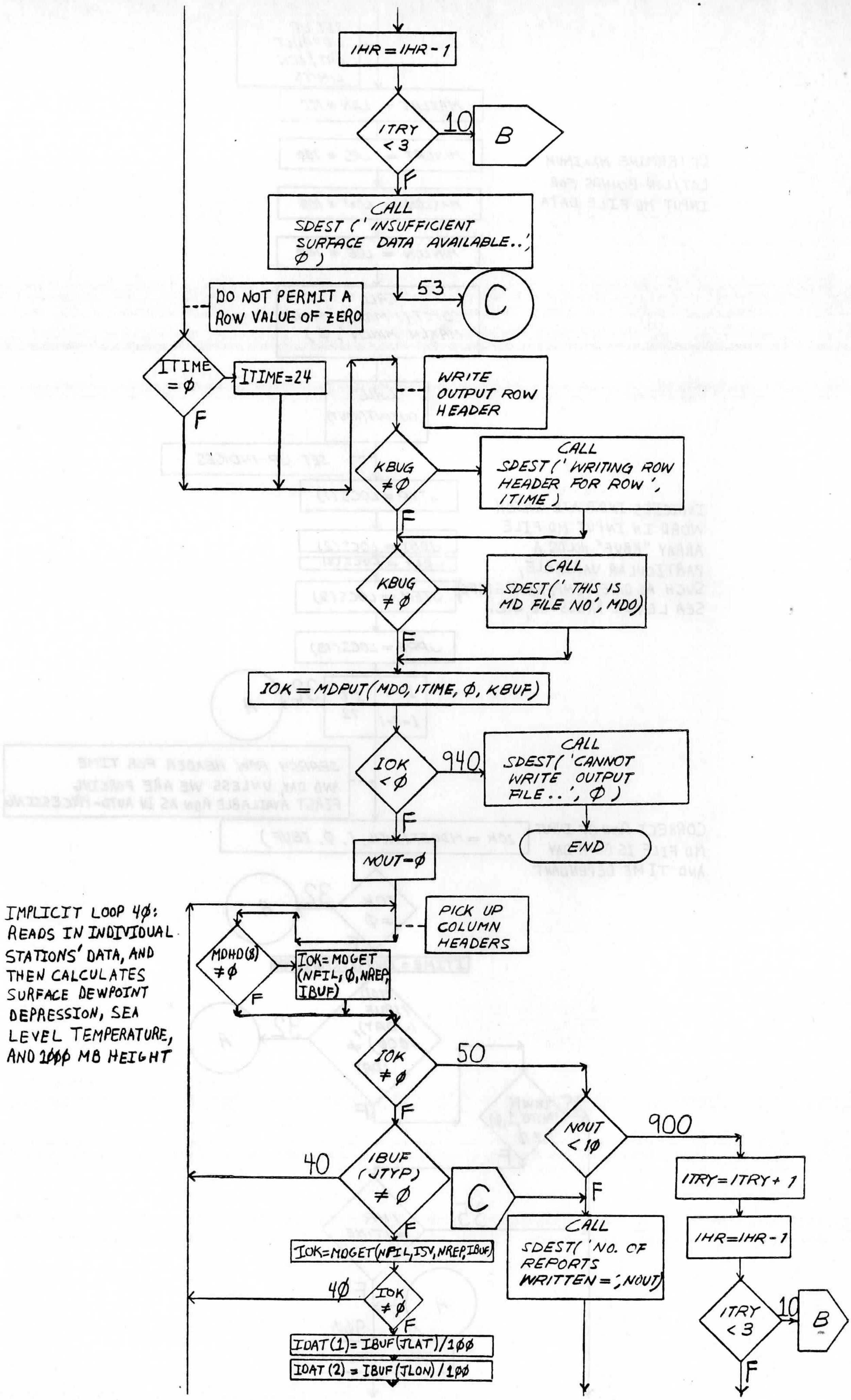
Finally, assuming all available data is accessed with no problems, the number of reports written (statement 53) is outputted to the CRT, and the array IDOC is updated and written into the VASTEXT file (CALL DWRITE(LUN,ITERM,IDOC)). In addition, the row header of the output MD file is re-written to show the number of surface data reports written into that row of the file.

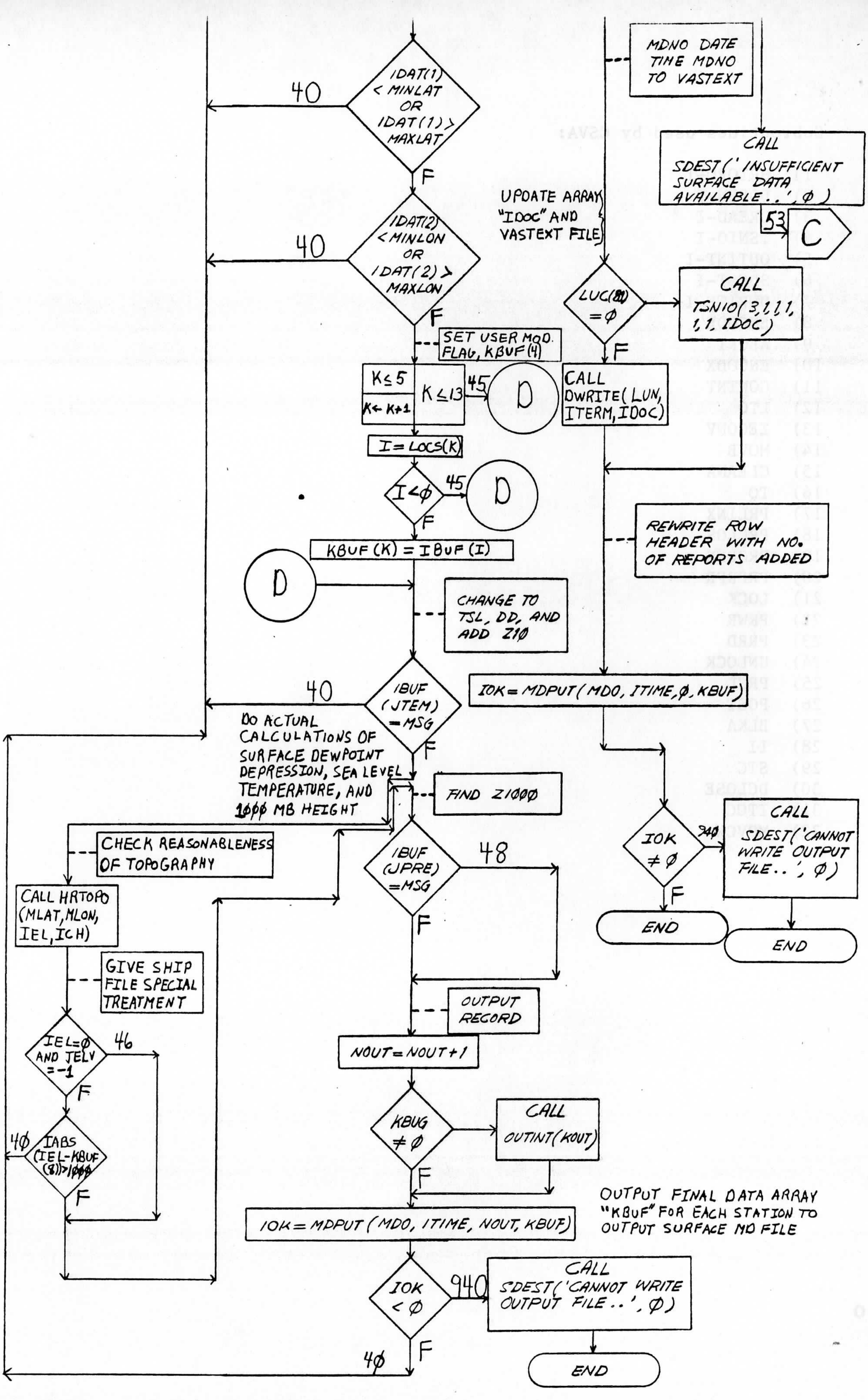
A subsequent execution of program LOVA will now show the retrieval surface MD file number, plus the row of that file which contains the station dewpoint depressions, sea level temperatures and 1000 mb heights for all the stations accessed.


CSVA
DATA MSG/Z80808080/, ITRY/ Ø /
IPRNT = IKWP('BUG', 1, Ø)
KBUG = IPRNT
CHECK ON AVAILABILITY OF OUTPUT FILE
READ IN OUTPUT SURFACE MD FILE
MDO = IPP(1, Ø)
MDOPEN (MDO, 2) < Ø
960
CALL SDEST('OUTPUT MDFILE NOT DEFINED..', Ø)
END
F
PICK UP DOCUMENTATION RECORD
LUC(81) = Ø
1
FOLLOWING FOR TOVS APPLICATIONS
CALL TSNIO (1,1,1,1,1,1,IDOC)
F
FOLLOWING FOR VAS APPLICATIONS
OPEN AND READ VASTEXT FILE
CALL DOPEN(MFILE, LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
CHECK STATUS WORD
IDOC(28) ≠1
800
CALL SDEST('FILE NOT AVAILABLE FOR PROCESSING, Ø)
END
F
FILL IN MD OUTPUT NO
CHECK FOR TIME FORCE
IHR = IKWP('TIME', 1, Ø)
PICK UP TIME FROM DOC


IHR = Ø
IHR = ITM / 10000
F
IDAY = MOD(IDOC(1), 1000)
B
DETERMINE INPUT MD FILE
NFIL = IKWP('MDF', 1, Ø)
NFIL > Ø
11
F
ASSUME FILE IS IN CURRENT SECTION
NFIL = MOD(IDAY, 1Ø)
NFIL = Ø, NFIL = NFIL + 1Ø
F
IKWP (SHIP, 1, Ø) ≠ Ø
NFIL = NFIL + 3Ø
F
KBUG ≠ Ø
CALL SDEST ('OPENING HOUR NO.', IHR)
F
MDINFO (NFIL, MDHD) ≠ Ø
97Ø
CALL SDEST (' UNABLE TO OPEN INPUT FILE ..', Ø)
END
F
IOPN = MDOPEN(NFIL, 1)
OPEN INPUT MD FILE FOR READ
IOPN < Ø
970
CALL SDEST (' UNABLE TO OPEN INPUT FILE ..', Ø)
END
F
PICK UP KEYS
READ KEYS OF INPUT MD FILE
IKEY = MDKEYS(NFIL, 13, LST, ISCL, IUNITS, LOCS)
KBUG = Ø
12
F
CALL OUTINT(KOUT)
CALL OUTINT(KOUT)
CALL OUTINT (KOUT)
CALL OUTINT (KOUT)


SET UP DEFAULT LAT/LON LIMITS
MAXLAT = LAN * 100
MINLAT = LAS * 100
DETERMINE MAXIMUM LAT/LON BOUNDS FOR INPUT MD FILE DATA
MAXLON = LOW * 100
MINLON = LOE * 100
CALL SDEST(' MAXLT MINLT MAXLN MINLN', 0)
CALL OUTINT(KOUT)
SET UP INDICES
JTYP = LOCS(1)
INDICIES INDICATE WHICH WORD IN INPUT MD FILE ARRAY "KBUF" HOLDS A PARTICULAR VARIABLE, SUCH AS DATE, TEMPERATURE (SFC), SEA LEVEL PRESSURE, ETC.
JDAY = LOCS(2)
JELV = LOCS(8)
JTEM = LOCS(9)
JPRE = LOCS(13)
I←1
I←I+1
I ≤ 72
32
A
SEARCH ROW HEADER FOR TIME AND DAY, UNLESS WE ARE FORCING FIRST AVAILABLE ROW AS IN AUTO-PROCESSING
CORRECT ROW OF INPUT MD FILE IS BOTH DAY AND TIME DEPENDANT
IOK = MDGET(NFIL, I, 0, IBUF)
IOK ≠ 0
32
A
F
ITIME = IBUF(JTIM)/10000
MOD (IBUF (JDAY), 1000) ≠ IDAY
32
A
F
35
IKWP('AUTO', 1, 0) ≠ 0
F
35
IHR = ITIME
F
A
900
ITRY = ITRY + 1


IHR=IHR-1
ITRY < 3
10
B
F
CALL SDEST ('INSUFFICIENT SURFACE DATA AVAILABLE..', Ø)
53
C
DO NOT PERMIT A ROW VALUE OF ZERO
ITIME = Ø
ITIME=24
F
WRITE OUTPUT ROW HEADER
KBUG ≠ Ø
CALL SDEST('WRITING ROW HEADER FOR ROW', ITIME)
F
KBUG ≠ Ø
CALL SDEST(' THIS IS MD FILE NO', MDO)
F
IOK=MDPUT(MDO, ITIME, Ø, KBUF)
IOK < Ø
940
CALL SDEST('CANNOT WRITE OUTPUT FILE..', Ø)
END
F
NOUT=Ø
IMPLICIT LOOP 4Ø: READS IN INDIVIDUAL STATIONS' DATA, AND THEN CALCULATES SURFACE DEWPOINT DEPRESSION, SEA LEVEL TEMPERATURE, AND 1ØØØ MB HEIGHT
PICK UP COLUMN HEADERS
MDHD(8) ≠ Ø
IOK=MDGET (NFIL, Ø, NREP, IBUF)
F
IOK ≠ Ø
50
F
NOUT < 1Ø
900
ITRY=ITRY + 1
IHR=IHR-1
ITRY < 3
10
B
F
IBUF (JTYP) ≠ Ø
40
C
F
CALL SDEST('NO. OF REPORTS WRITTEN=', NOUT)
IOK=MDGET(NFIL, ISV, NREP, IBUF)
4Ø
IOK ≠ Ø
F
IDAT(1)= IBUF(JLAT)/1ØØ
IDAT (2) = IBUF (JLON) / 1ØØ


IDAT(1) < MINLAT OR IDAT(1) > MAXLAT
40
F
IDAT(2) < MINLON OR IDAT(2) > MAXLON
40
F
SET USER MOD. FLAG, KBUF(4)
K≤5
K← K+1
K≤13
45
D
I=LOCS(K)
I<Ø
45
D
F
KBUF(K) = IBUF(I)
D
CHANGE TO TSL, DD, AND ADD Z1Ø
IBUF (JTEM) =MSG
40
F
DO ACTUAL CALCULATIONS OF SURFACE DEWPOINT DEPRESSION, SEA LEVEL TEMPERATURE, AND 1ØØØ MB HEIGHT
FIND Z1ØØØ
CHECK REASONABLENESS OF TOPOGRAPHY
CALL HRTOPO (MLAT, MLON, IEL, ICH)
GIVE SHIP FILE SPECIAL TREATMENT
IEL=Ø AND JELV =-1
46
F
IABS (IEL-KBUF (8))>1ØØØ
4Ø
F
IBUF (JPRE) =MSG
48
F
OUTPUT RECORD
NOUT=NOUT+1
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
IOK=MDPUT(MDO, ITIME, NOUT, KBUF)
OUTPUT FINAL DATA ARRAY "KBUF" FOR EACH STATION TO OUTPUT SURFACE MD FILE
IOK < Ø
94Ø
CALL SDEST('CANNOT WRITE OUTPUT FILE..', Ø)
END
F
4Ø
MDNO DATE TIME MDNO TO VASTEXT
CALL SDEST('INSUFFICIENT SURFACE DATA AVAILABLE..', Ø)
53
C
UPDATE ARRAY "IDOC" AND VASTEXT FILE
LUC(8Ø) =Ø
CALL TSNIO(3,1,1,1, 1,1, IDOC)
F
CALL DWRITE(LUN, ITERM, IDOC)
REWRITE ROW HEADER WITH NO. OF REPORTS ADDED
IOK=MDPUT(MDO, ITIME, Ø, KBUF)
IOK ≠ Ø
94Ø
CALL SDEST('CANNOT WRITE OUTPUT FILE..', Ø)
F
END
END

Subroutines used by CSVA:

1) HRTOPO-I
2) DOPEN-I
3) DREAD-I
4) TSNIO-I
5) OUTINT-I
6) SDEST-I
7) DWRITE-I
8) ENCODE
9) ABORT
10) ENCODX
11) CONTNT
12) LTQ
13) ZECONV
14) MOVB
15) CLEANA
16) TQ
17) PRLINX
18) PRCLOS
19) PROPEN
20) PRPRPR
21) LOCK
22) PRWR
23) PRRD
24) UNLOCK
25) PRCL
26) POST
27) BLKA
28) II
29) STC
30) DCLOSE
31) ITOC
32) MOVCW

SRVA

The purpose of SRVA is to generate grids of surface variables previously calculated by program CSVA for use in either VAS or TOVS retrievals. As usual, only VAS will be dealt with here. The output of SRVA is stored in a grid which, in turn, is contained within a gridfile for surface data. Normally, the user will run SRVA three times to generate one grid each of sea level temperature, surface dewpoint depression and 1000 mb height.

First, after the VASTEXT file has been opened and read via subroutines DOPEN and DREAD, variables such as retrieval surface MD file and row (MDNS and MDRS, respectively), guess MD file (MDNG), etc. are set. Then, the surface MD file prepared by CSVA (MDNS) is opened for read/write, and the number of reports to be processed (NRPT) is determined, either by reading the row header for row MDRS or an explicit keyin of keyword parameter RPT. These reports will contain the data that is to be used in constructing the surface grid.

After the operator-forced boundaries (set up in IDVA or keyed in) that will determine the area of the grid are determined, the character whose data is to be gridded is set by function CPP (CHAR=CPP(1,' ')), and the analysis field arrays IDL and ITG are initialized to 0. Note that several different types of variables can be gridded (TSL (sea level temperature), DD (surface dewpoint depression), TS (surface temperature) and Z100 (1000 mb height)). In addition, note the element of the gross error check array (SDL) that is set up after the character to be gridded has been determined. This element is important later in both a gross error check involving the guess MD file (MDNG) guess field (compares raw data value to guess value) and within an edit option quality control check (compares raw data value to final grid value). After the IDL and ITG arrays have been initialized to zero, the grid increment (in tenths of degrees) is established. Keep in mind that this increment can be either keyed in or determined within the program. If the grid increment is NOT keyed in, it will be calculated such that the total number of gridpoints is less than or equal to 2400 (the value of variable NSIZE).

At this point, it is possible to acquire the information to be used in the surface analyses in two different ways. First, if a guess is to be used when generating the grids, such as will be the case if there are large data-void areas within the analysis region, a call to subroutine SRGSS will access the guess (the "VASGSS" route), after which the actual surface data will be gathered via function MDGET (beginning of DO LOOP 148). It should be mentioned here that the guess grid option is never used when executing SRVA and can therefore be ignored within the context of this discussion. Note the "NODAT" option approximately 15-20 lines after the call to subroutine SRGSS. If it is in effect (IKWP('NODAT',1,0) .NE. 0), SRVA will skip to statement 150, and guess data only will be stored in output surface gridfile NGRFS. In addition, note the preparation of variable FCK just before SRVA enters DO LOOP 148. This variable is used later when comparing the raw surface data to a value from the guess field to determine if a given data value should be used in the

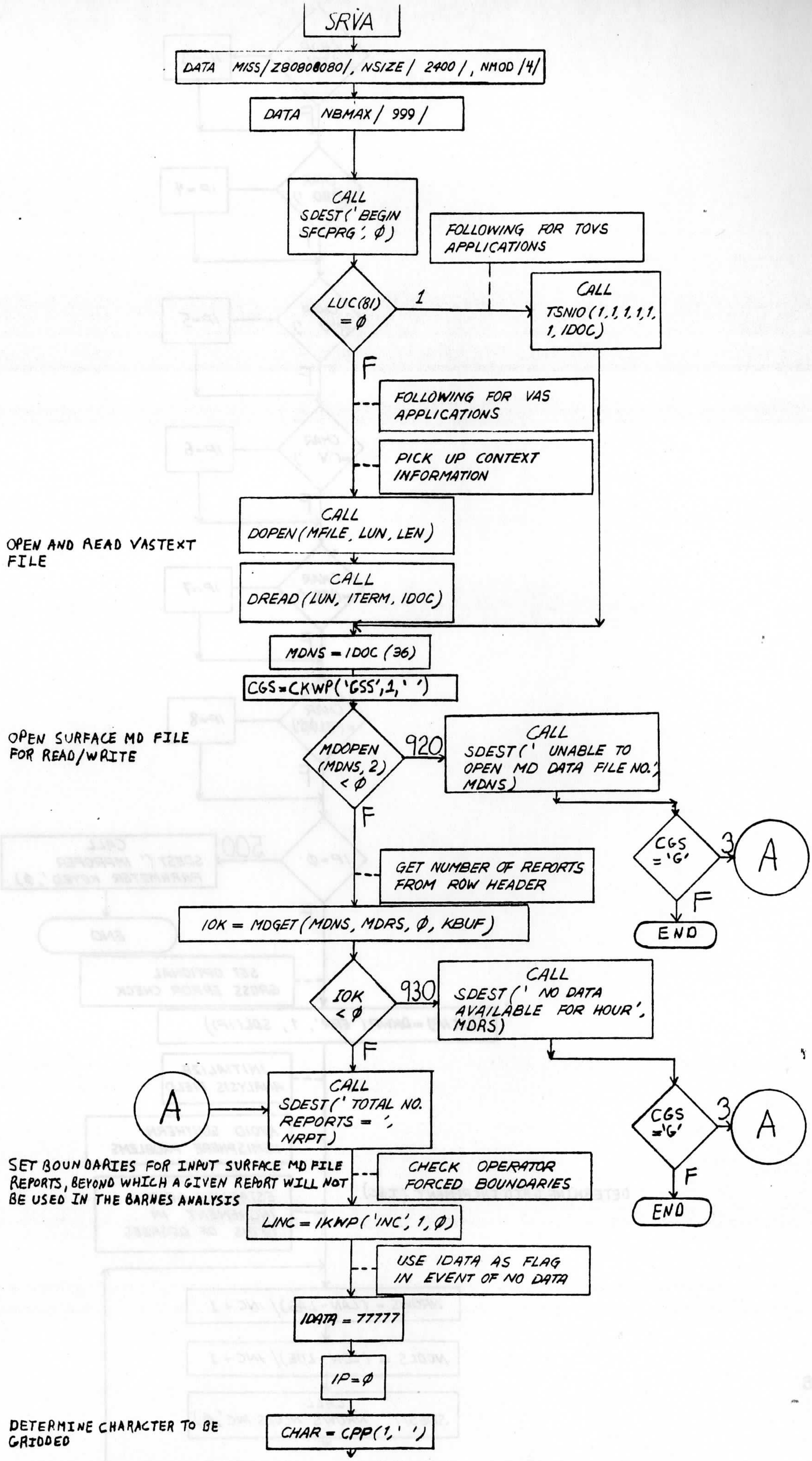
Barnes analysis.

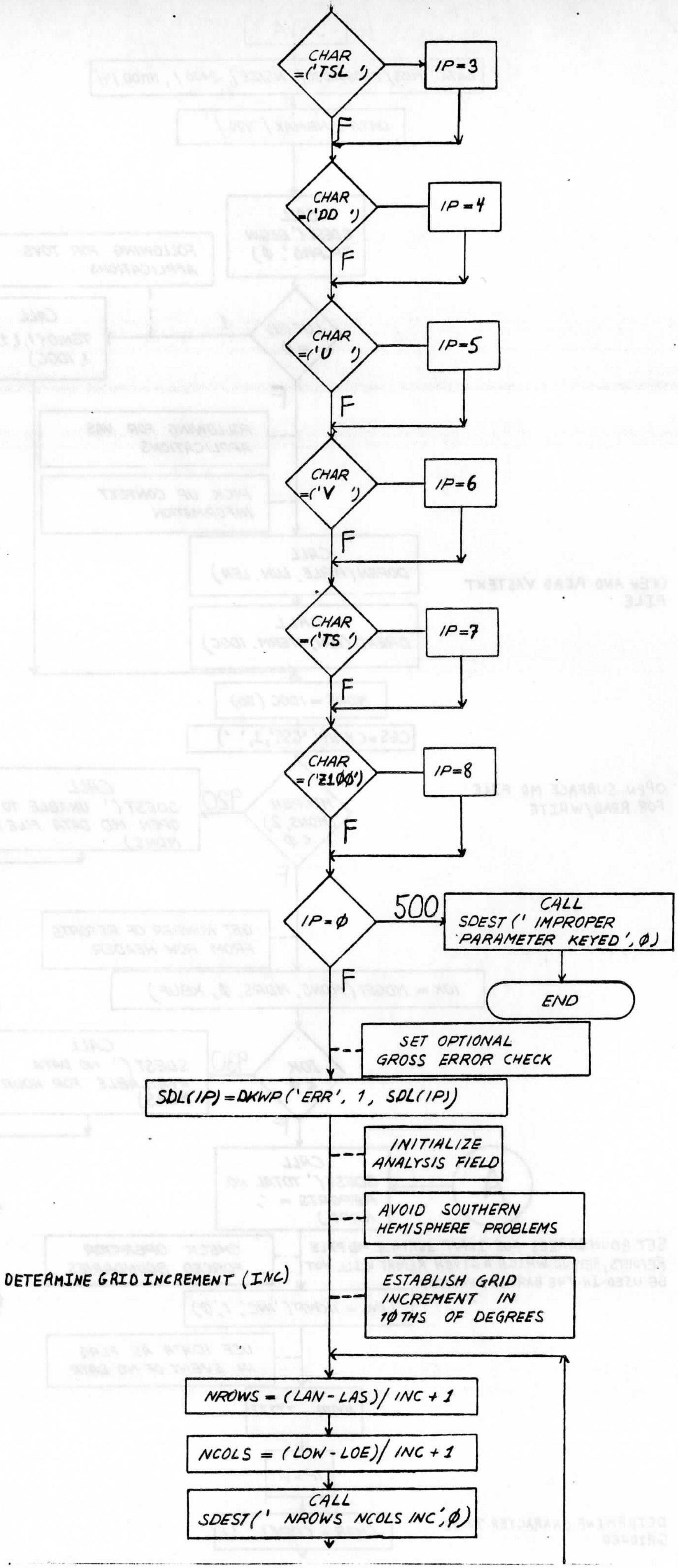
On the other hand, if no guess is desired (no keyin for keyword parameter GSS, or variable CGS = ' '), SRVA simply moves to DO LOOP 148 to read the actual surface data. This option should normally be used in the event there is good data coverage.

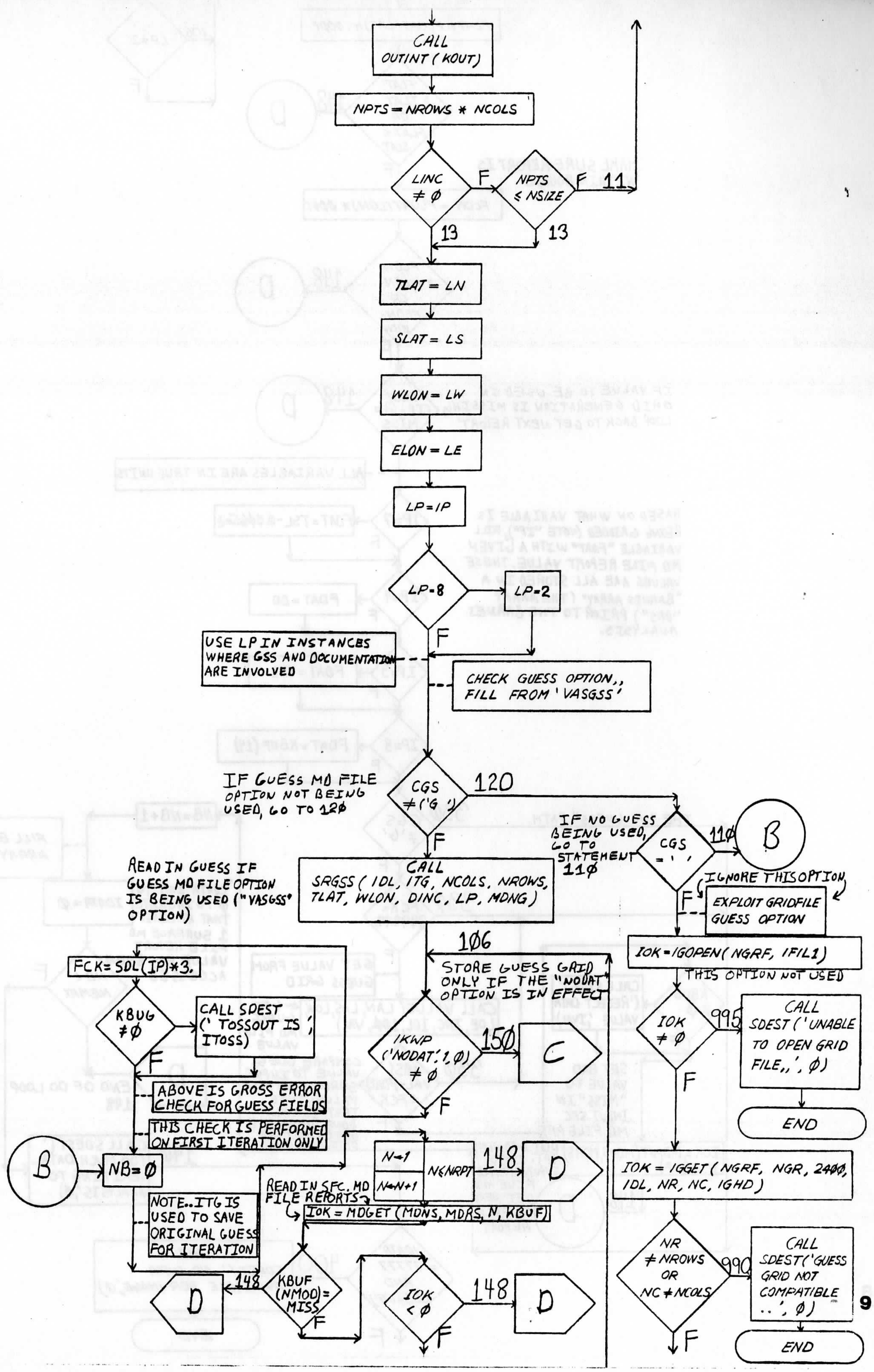
SRVA now enters DO LOOP 148. This loop reads each individual surface report (which contains the 1000 mb height, sea level temperature, dewpoint depression, etc. data) from the surface MD file (MDNS). In addition, it makes sure a given report is within the previously-set LAT/LON bounds, performs a test (only for the "VASGSS" guess option) to screen out very poor surface data, and fills Barnes arrays such as DAS, RW and CL for use in the forthcoming Barnes analysis. Upon completion of DO LOOP 148, the Barnes analysis is set up and executed. The analysis itself is carried out by subroutine FBARN, and results in a uniform latitude-longitude grid of values.

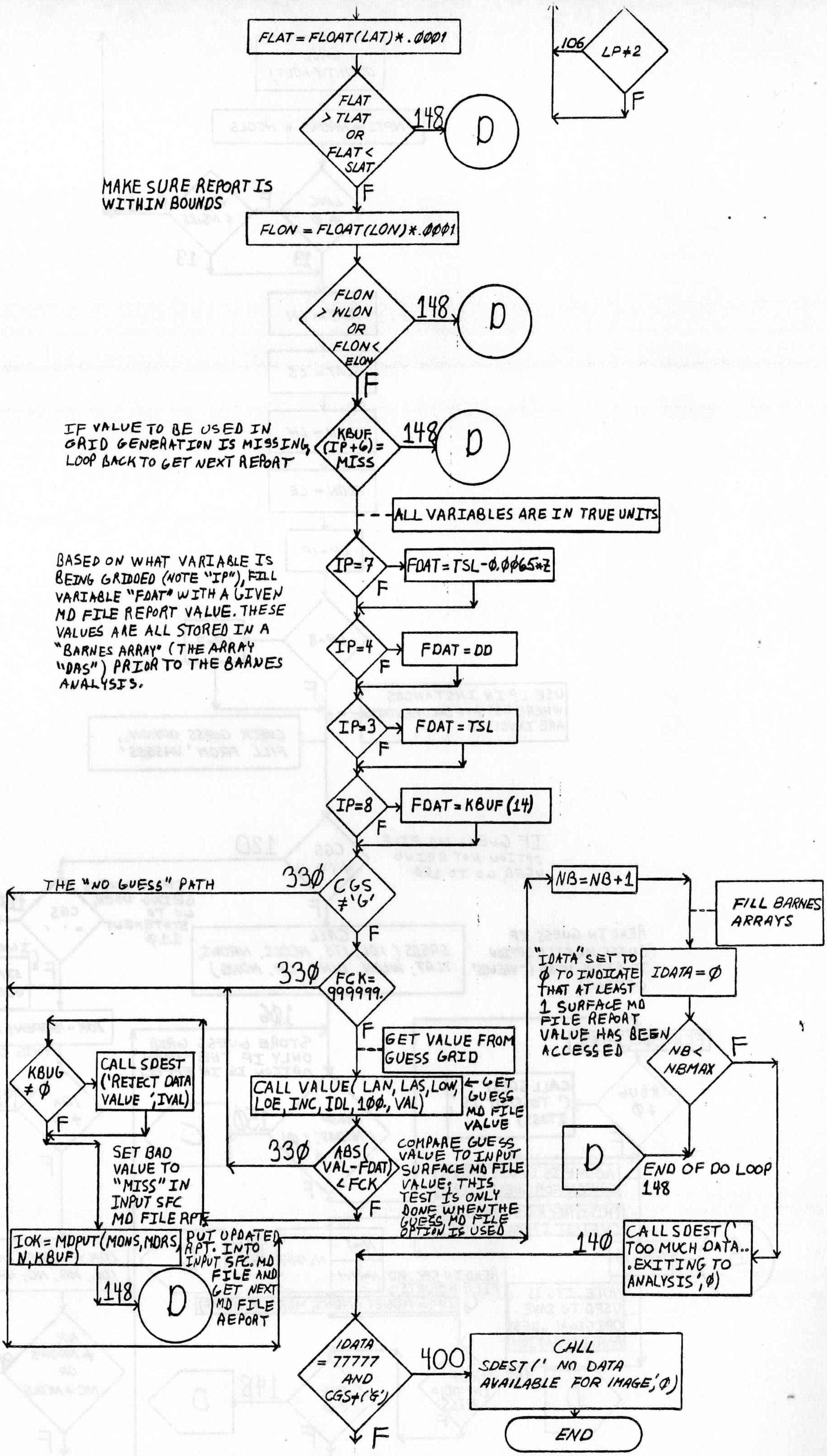
At this point in SRVA, assuming the edit option is on (IKWP('EDIT',1,0).EQ.0), the standard deviation of the actual data used in the grid generation (FDAT) is computed and stored in variable SD. Then, if the difference between a given raw data value and a value derived from the final grid (VAL) is .GE. to the standard deviation, the original surface data report is flagged with variable MISS, and the program returns to statement 110 to read in the surface data again and repeat the Barnes analysis. This time, however, the flagged report(s) will not be taken into account in the Barnes analysis.

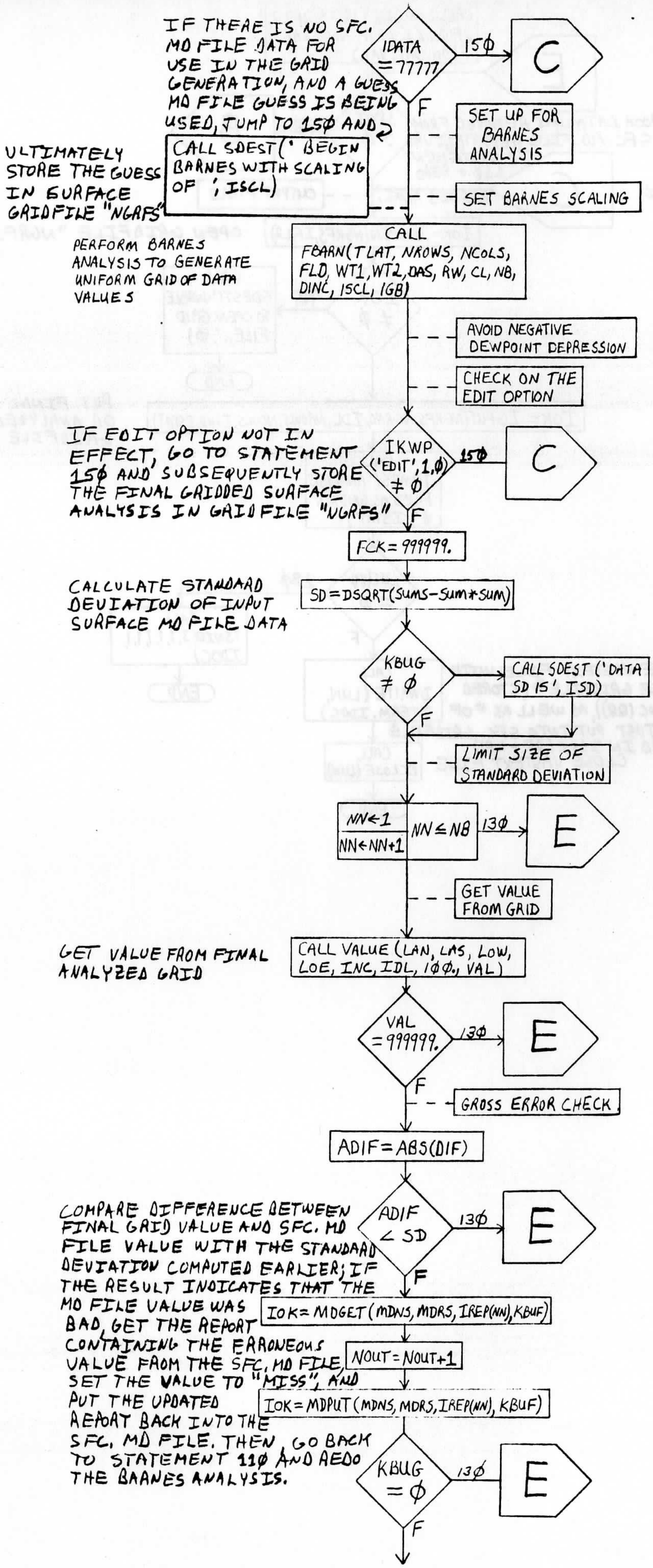
Assuming there are no problems with the Barnes analysis grid and the edit option passes, an output surface gridfile (NGRFS) is opened (IOK=IGOPEN(NGRFS,IFIL2)), and the sea level temperature, dewpoint depression, 1000 mb height, etc. grid is placed within it by function IGPUT. In addition, the array IDOC is updated to show both the output surface gridfile number and the number of the new grid within this gridfile (ISTAT). This information is then subsequently written into the VASTEXT file via subroutine DWRITE.

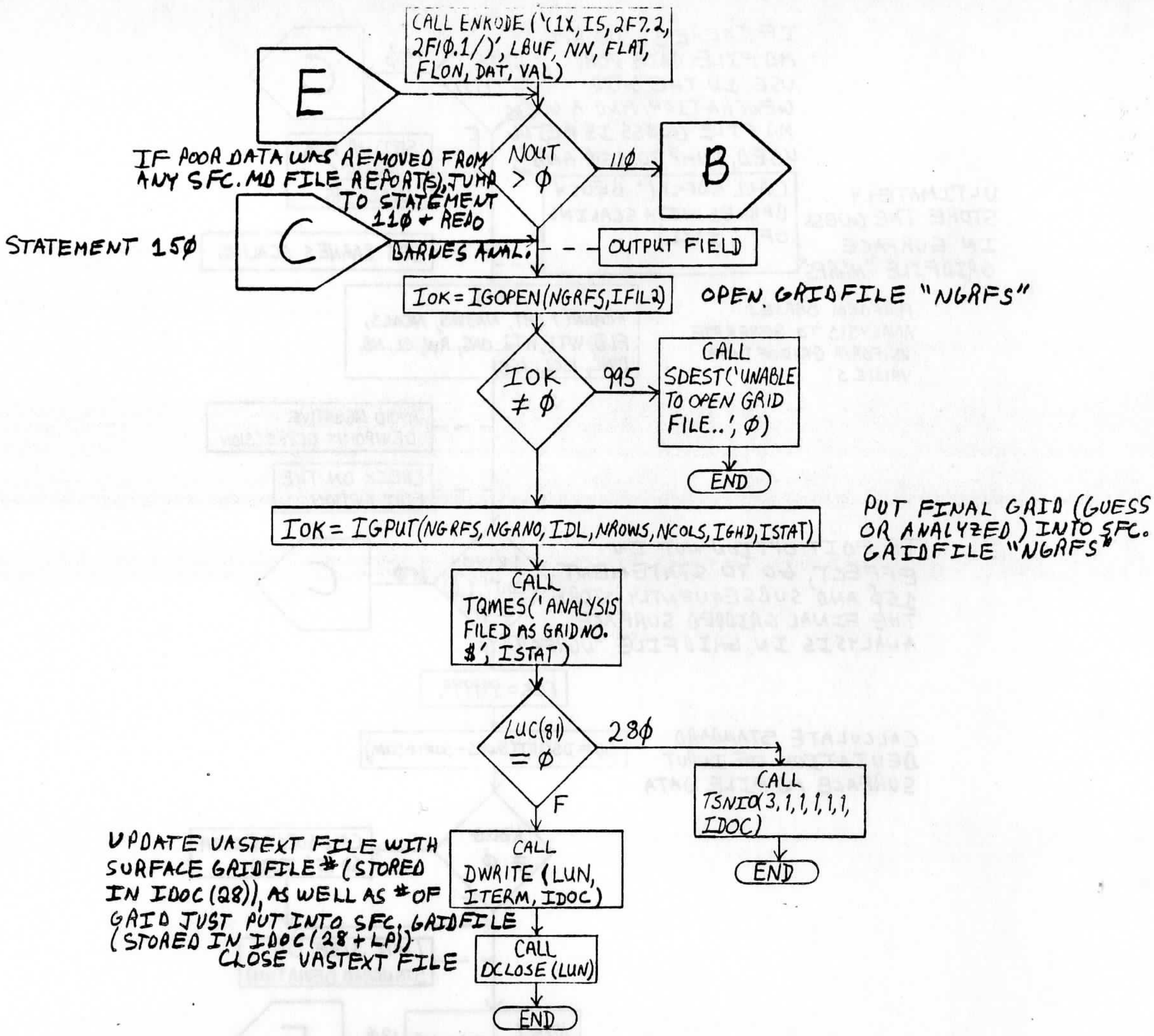

SRVA
DATA MISS/Z80808080/, NSIZE/ 2400 /, NMOD /4/
DATA NBMAX / 999 /
CALL SDEST('BEGIN SFCPRG', Ø)
FOLLOWING FOR TOVS APPLICATIONS
LUC(81) = Ø
1
CALL TSNIO(1,1,1,1,1, 1, IDOC)
F
FOLLOWING FOR VAS APPLICATIONS
PICK UP CONTEXT INFORMATION
OPEN AND READ VASTEXT FILE
CALL DOPEN(MFILE, LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
MDNS = IDOC (36)
CGS=CKWP('GSS',1,' ')
OPEN SURFACE MD FILE FOR READ/WRITE
MDOPEN (MDNS, 2) < Ø
920
CALL SDEST(' UNABLE TO OPEN MD DATA FILE NO.', MDNS)
CGS = 'G'
3
A
F
END
F
GET NUMBER OF REPORTS FROM ROW HEADER
IOK = MDGET(MDNS, MDRS, Ø, KBUF)
IOK < Ø
930
CALL SDEST(' NO DATA AVAILABLE FOR HOUR', MDRS)
CGS = 'G'
3
A
F
END
F
A
CALL SDEST(' TOTAL NO. REPORTS = ', NRPT)
SET BOUNDARIES FOR INPUT SURFACE MD FILE REPORTS, BEYOND WHICH A GIVEN REPORT WILL NOT BE USED IN THE BARNES ANALYSIS
CHECK OPERATOR FORCED BOUNDARIES
LINC = IKWP('INC', 1, Ø)
USE IDATA AS FLAG IN EVENT OF NO DATA
IDATA = 77777
IP=Ø
DETERMINE CHARACTER TO BE GRIDDED
CHAR = CPP(1,' ')


CHAR =('TSL ')
IP=3
F
CHAR =('DD ')
IP=4
F
CHAR =('U ')
IP=5
F
CHAR =('V ')
IP=6
F
CHAR =('TS ')
IP=7
F
CHAR =('Z100')
IP=8
F
IP=0
500
CALL SDEST(' IMPROPER PARAMETER KEYED',0)
END
F
SET OPTIONAL GROSS ERROR CHECK
SDL(IP)=DKWP('ERR', 1, SDL(IP))
INITIALIZE ANALYSIS FIELD
AVOID SOUTHERN HEMISPHERE PROBLEMS
DETERMINE GRID INCREMENT (INC)
ESTABLISH GRID INCREMENT IN 10THS OF DEGREES
NROWS = (LAN-LAS)/ INC + 1
NCOLS = (LOW-LOE)/ INC + 1
CALL SDEST(' NROWS NCOLS INC',0)


CALL OUTINT (KOUT)
NPTS = NROWS * NCOLS
LINC ≠ Ø
F
NPTS ≤ NSIZE
F
11
13
13
TLAT = LN
SLAT = LS
WLON = LW
ELON = LE
LP = IP
LP = 8
LP = 2
F
USE LP IN INSTANCES WHERE GSS AND DOCUMENTATION ARE INVOLVED
CHECK GUESS OPTION,, FILL FROM 'VASGSS'
IF GUESS MD FILE OPTION NOT BEING USED, GO TO 12Ø
CGS ≠ ('G')
120
F
IF NO GUESS BEING USED, GO TO STATEMENT 11Ø
CGS = ' '
11Ø
B
IGNORE THIS OPTION
EXPLOIT GRIDFILE GUESS OPTION
F
READ IN GUESS IF GUESS MD FILE OPTION IS BEING USED ("VASGSS" OPTION)
CALL SRGSS (IDL, ITG, NCOLS, NROWS, TLAT, WLON, DINC, LP, MDNG)
1Ø6
IOK = IGOPEN (NGRF, IFIL1)
THIS OPTION NOT USED
FCK = SDL (IP) * 3.
STORE GUESS GRID ONLY IF THE "NODAT" OPTION IS IN EFFECT
KBUG ≠ Ø
CALL SDEST (' TOSSOUT IS ', ITOSS)
F
IKWP ('NODAT', 1, Ø) ≠ Ø
15Ø
C
F
IOK ≠ Ø
995
CALL SDEST ('UNABLE TO OPEN GRID FILE,, ', Ø)
F
END
ABOVE IS GROSS ERROR CHECK FOR GUESS FIELDS
THIS CHECK IS PERFORMED ON FIRST ITERATION ONLY
B
NB = Ø
N←1
N←N+1
N≤NRPT
148
D
IOK = IGGET (NGRF, NGR, 24ØØ, IDL, NR, NC, IGHD)
NOTE.. ITG IS USED TO SAVE ORIGINAL GUESS FOR ITERATION
READ IN SFC. MD FILE REPORTS
IOK = MDGET (MDNS, MDRS, N, KBUF)
D
148
KBUF (NMOD) = MISS
F
IOK < Ø
148
D
F
NR ≠ NROWS OR NC ≠ NCOLS
99Ø
CALL SDEST ('GUESS GRID NOT COMPATIBLE .. ', Ø)
END
F


FLAT = FLOAT(LAT)*.0001
FLAT > TLAT OR FLAT < SLAT
148
D
MAKE SURE REPORT IS WITHIN BOUNDS
F
FLON = FLOAT(LON)*.0001
FLON > WLON OR FLON < ELON
148
D
F
IF VALUE TO BE USED IN GRID GENERATION IS MISSING, LOOP BACK TO GET NEXT REPORT
KBUF (IP+6) = MISS
148
D
ALL VARIABLES ARE IN TRUE UNITS
BASED ON WHAT VARIABLE IS BEING GRIDDED (NOTE "IP"), FILL VARIABLE "FDAT" WITH A GIVEN MD FILE REPORT VALUE. THESE VALUES ARE ALL STORED IN A "BARNES ARRAY" (THE ARRAY "DAS") PRIOR TO THE BARNES ANALYSIS.
IP=7
FDAT = TSL-0.0065*Z
F
IP=4
FDAT = DD
F
IP=3
FDAT = TSL
F
IP=8
FDAT = KBUF(14)
F
106
LP≠2
F
THE "NO GUESS" PATH
330
CGS ≠'G'
F
330
FCK= 999999.
F
GET VALUE FROM GUESS GRID
CALL VALUE(LAN, LAS, LOW, LOE, INC, IDL, 100., VAL)
← GET GUESS MD FILE VALUE
330
ABS(VAL-FDAT) < FCK
F
COMPARE GUESS VALUE TO INPUT SURFACE MD FILE VALUE. THIS TEST IS ONLY DONE WHEN THE GUESS, MD FILE OPTION IS USED
KBUG ≠ 0
CALL SDEST ('REJECT DATA VALUE ',IVAL)
F
SET BAD VALUE TO "MISS" IN INPUT SFC MD FILE RPT.
IOK = MDPUT(MONS, MDRS, N, KBUF)
PUT UPDATED RPT. INTO INPUT SFC. MD FILE AND GET NEXT MD FILE REPORT
148
D
NB=NB+1
FILL BARNES ARRAYS
"IDATA" SET TO 0 TO INDICATE THAT AT LEAST 1 SURFACE MD FILE REPORT VALUE HAS BEEN ACCESSED
IDATA = 0
NB< NBMAX
F
D
END OF DO LOOP 148
140
CALL SDEST (' TOO MUCH DATA.. .EXITING TO ANALYSIS',0)
IDATA = 77777 AND CGS≠('G')
400
CALL SDEST (' NO DATA AVAILABLE FOR IMAGE',0)
END
F


IF THERE IS NO SFC. MD FILE DATA FOR USE IN THE GRID GENERATION, AND A GUESS MD FILE GUESS IS BEING USED, JUMP TO 15Ø AND
ULTIMATELY STORE THE GUESS IN SURFACE GRIDFILE "NGRFS"
IDATA = 77777
15Ø
C
F
SET UP FOR BARNES ANALYSIS
CALL SDEST(' BEGIN BARNES WITH SCALING OF ', ISCL)
SET BARNES SCALING
PERFORM BARNES ANALYSIS TO GENERATE UNIFORM GRID OF DATA VALUES
CALL FBARN(TLAT, NROWS, NCOLS, FLD, WT1, WT2, DAS, RW, CL, NB, DINC, ISCL, IGB)
AVOID NEGATIVE DEWPOINT DEPRESSION
CHECK ON THE EDIT OPTION
IF EDIT OPTION NOT IN EFFECT, GO TO STATEMENT 15Ø AND SUBSEQUENTLY STORE THE FINAL GRIDDED SURFACE ANALYSIS IN GRIDFILE "NGRFS"
IKWP ('EDIT',1,Ø) ≠ Ø
15Ø
C
F
FCK = 999999.
CALCULATE STANDARD DEVIATION OF INPUT SURFACE MD FILE DATA
SD = DSQRT(SUMS−SUM*SUM)
KBUG ≠ Ø
CALL SDEST ('DATA SD IS', ISD)
F
LIMIT SIZE OF STANDARD DEVIATION
NN←1
NN←NN+1
NN ≤ NB
13Ø
E
GET VALUE FROM GRID
GET VALUE FROM FINAL ANALYZED GRID
CALL VALUE (LAN, LAS, LOW, LOE, INC, IDL, 1ØØ., VAL)
VAL = 999999.
13Ø
E
F
GROSS ERROR CHECK
ADIF = ABS(DIF)
COMPARE DIFFERENCE BETWEEN FINAL GRID VALUE AND SFC. MD FILE VALUE WITH THE STANDARD DEVIATION COMPUTED EARLIER; IF THE RESULT INDICATES THAT THE MD FILE VALUE WAS BAD, GET THE REPORT CONTAINING THE ERRONEOUS VALUE FROM THE SFC. MD FILE, SET THE VALUE TO "MISS", AND PUT THE UPDATED REPORT BACK INTO THE SFC. MD FILE. THEN GO BACK TO STATEMENT 11Ø AND REDO THE BARNES ANALYSIS.
ADIF < SD
13Ø
E
F
IOK = MDGET (MDNS, MDRS, IREP(NN), KBUF)
NOUT = NOUT+1
IOK = MDPUT (MDNS, MDRS, IREP(NN), KBUF)
KBUG = Ø
13Ø
E
F


CALL ENKODE ('(1X, I5, 2F7.2, 2F10.1/)', LBUF, NN, FLAT, FLON, DAT, VAL)
E
NOUT > 0
110
B
IF POOR DATA WAS REMOVED FROM ANY SFC. MD FILE REPORT(S), JUMP TO STATEMENT 110 + REDO
STATEMENT 150
C
BARNES ANAL.
OUTPUT FIELD
IOK = IGOPEN(NGRFS, IFIL2)
OPEN GRIDFILE "NGRFS"
IOK ≠ 0
995
CALL SDEST('UNABLE TO OPEN GRID FILE..', 0)
END
IOK = IGPUT(NGRFS, NGRNO, IDL, NROWS, NCOLS, IGHD, ISTAT)
PUT FINAL GRID (GUESS OR ANALYZED) INTO SFC. GRIDFILE "NGRFS"
CALL TQMES('ANALYSIS FILED AS GRIDNO. $', ISTAT)
LUC(81) = 0
280
CALL TSNIO(3, 1, 1, 1, 1, 1, IDOC)
END
F
UPDATE VASTEXT FILE WITH SURFACE GRIDFILE # (STORED IN IDOC(28)), AS WELL AS # OF GRID JUST PUT INTO SFC. GRIDFILE (STORED IN IDOC(28 + LR)).
CALL DWRITE (LUN, ITERM, IDOC)
CLOSE VASTEXT FILE
CALL DCLOSE (LUN)
END

Subroutines used by SRVA:

1) SDEST-I
2) VALUE-I
3) DOPEN-I
4) ENKODE-I
5) DREAD-I
6) TSNIO-I
7) SRGSS-I
8) OUTINT-I
9) FBARN-I
10) TQMES-I
11) DWRITE-I
12) DCLOSE-I
13) BLKA
14) MOVCW
15) MOVB
16) LTQ
17) CLEANA
18) TQ
19) PRLINX
20) PRCLOS
21) PROPEN
22) PRPRPR
23) LOCK
24) PRWR
25) PRRD
26) UNLOCK
27) PRCL
28) POST
29) ABORT
30) ENCODE
31) ENCODX
32) CONTNT
33) ZECONV
34) II
35) STC
36) ITOC
37) VASGES
38) FILL
39) LINFIL
40) INTER
41) HEAPFY

XRVA

XRVA is the main editing program used in conjunction with VAS retrieval processing. It can be used to delete entire report(s) contained within the cursor, a parameter at a given level in all report(s) contained within the cursor, or a parameter at all levels in all report(s) contained within the cursor. (As a side note, since the cursor size can be varied by using McIDAS program "DF," it is possible to cover either one report or several reports at one time under the cursor.) If the user is working on a surface data MD file, he/she can delete either an entire report or reports, or only a particular parameter value for single or multiple reports, within the cursor. The parameter deleted can be any valid variable which exists in the schema for that particular surface MD file, but it will usually be one of Z100 (1000 mb height), TSL (sea level temperature) or DD (surface dewpoint depression). On the other hand, if the user is working on a retrieval MD file, an entire retrieval can be deleted if keyword parameter NRET is keyed in as the only positional or keyword parameter, or one or more entire retrievals contained within the cursor can be deleted if XRVA is keyed in by itself. In addition, a particular parameter at a keyed-in level for a retrieval or retrievals can be deleted, or a particular parameter at all levels for a retrieval or retrievals can be deleted. Note finally that one must do an "MDU SET" before the execution of XRVA to assure that McIDAS will be operating on the correct surface or retrieval MD file.

After the MD file number listed in User Common is read via function LUC (MDNO=LUC(5)), the VASTEXT file is opened and read, and the cursor line/element position (IL,IE) is determined via subroutine TVSAT. Then, the navigation is initialized, and the MD file whose number was determined earlier is opened for read/write, provided that its number matches either IDOC(36) or (40) (surface and retrieval MD files, respectively, as defined in the VASTEXT file).

Just before statement 2, XRVA checks to see if an entire retrieval with the specified number should be deleted. If this is the case (NRET .NE. 0), the report which exists in row MDR, column NRET of MD file MDNO will have all of its values changed to MISS, after which XRVA will terminate. However, if this is not the case, (NRET .EQ. 0), XRVA jumps to statement 99, where the row header of MD file MDNO is read, which gives the number of retrievals/surface reports in that particular row (variable MREC). At this point, a test to check the MD file the user wishes to edit is performed to see if data exists ("READ TEST RECORD"). Assuming there is no problem with this test, the keys corresponding to the schema for that MD file are read by function MDKEYS. Then, after image information has been accessed (GETFRM) and the plotting routines initialized (INITPL), XRVA checks to see what parameter should be deleted (variable ICHR), and what level (or all levels, if no level is keyed in for the second positional parameter) it should be deleted from (variable ILEV).

If no parameter or level are specified, control passes directly to statement 10, and the process to delete ALL reports at ALL levels under the cursor begins. However, if some parameter (such as Z, T, etc.) is keyed in as the first positional parameter, but nothing is entered as the second positional parameter, DO LOOP 3 sets the array IADD and the variable KTOT, both of which will be used later (in DO LOOP 58) to delete that parameter from all levels in each report which is contained by the cursor. Finally, if both a parameter AND level are keyed in, DO LOOPs 5 and 7 prepare IADD and KTOT such that the given parameter will be deleted at that level from all reports within the cursor.

At this point, XRVA is in the vicinity of statement 10. After the "GO" prompt is displayed on the CRT by function ICURG, the user should place the cursor over the report(s) to be edited and then push the spacebar, which will result in an edit box for the forthcoming deletion(s) being traced out around the outside dimensions of the cursor by subroutine WRBOX. Then, the latitude/longitude coordinates for the four corners of the cursor box are stored in the arrays FLA and FLO, and another "GO" appears on the CRT. The user should then position the cursor over the next report(s) to be edited and push the spacebar again. As few as 1, or as many as 40 edit boxes can be traced out on the video screen (exact number of boxes stored in the variable NX).

The deletion stage of the program begins either when NX=40 or something other than the terminal's spacebar is punched after the "GO" prompt. Punching a character other than the spacebar gives function ICURG a value of 1. Usually, "R END" will be typed by the user to terminate the box-drawing stage of XRVA.

In the next step, since LUC(81) (the VAS sounder area number) is .NE. 0, XRVA enters DO LOOP 70, which carries out the actual deletions. Initially, a given MD file report is read (IOK=MDGET(MDNO,MDR,M,IOUT)) and, provided the report has not been previously marked for deletion via the "mod" flag (NMOD), the program enters DO LOOP 40. Within this loop, the latitude/longitude coordinates of a given MD file entry (retrieval or surface report) are compared with the maximum/minimum latitude/longitude coordinates of each of the NX plotted edit boxes. If a given MD file report is found to be contained within one of the edit boxes, the raster and pixel (TV (television) coordinates) of the retrieval or surface report are calculated via subroutines SATEAR and SATTV and stored in the variables IRAS and IPIC. Then, assuming the MD file report also is within the raster/pixel bounds of the edit box (IRMIN, IRMAX, IPMIN, IPMAX), a red "X" is plotted over the report location by subroutine WRMAR.

Then, if a given parameter at a level or all levels is to be removed from the report (ICHR.NE.' '), the element(s) of the array IOUT corresponding to the data to be deleted is(are) set to the value MISS in DO LOOP 58, after which the updated array is put back into the surface or retrieval MD file via function MDPUT (IOK=MDPUT(MDNO,MDR,M, IOUT)). Following this, XRVA loops back to the beginning of DO LOOP 70 to process the next MD file report.

On the other hand, if the entire MD file report is to be deleted (ICHR .EQ. ' '), the element of IOUT which marks the report for future exemption (the mod flag) is set to the value

999999, after which the updated array is returned to the surface or retrieval MD file via MDPUT.

Finally, DO LOOP 70 concludes, the VASTEXT file is closed, and a message saying " ALL DONE DELETING ..." is printed on the CRT.

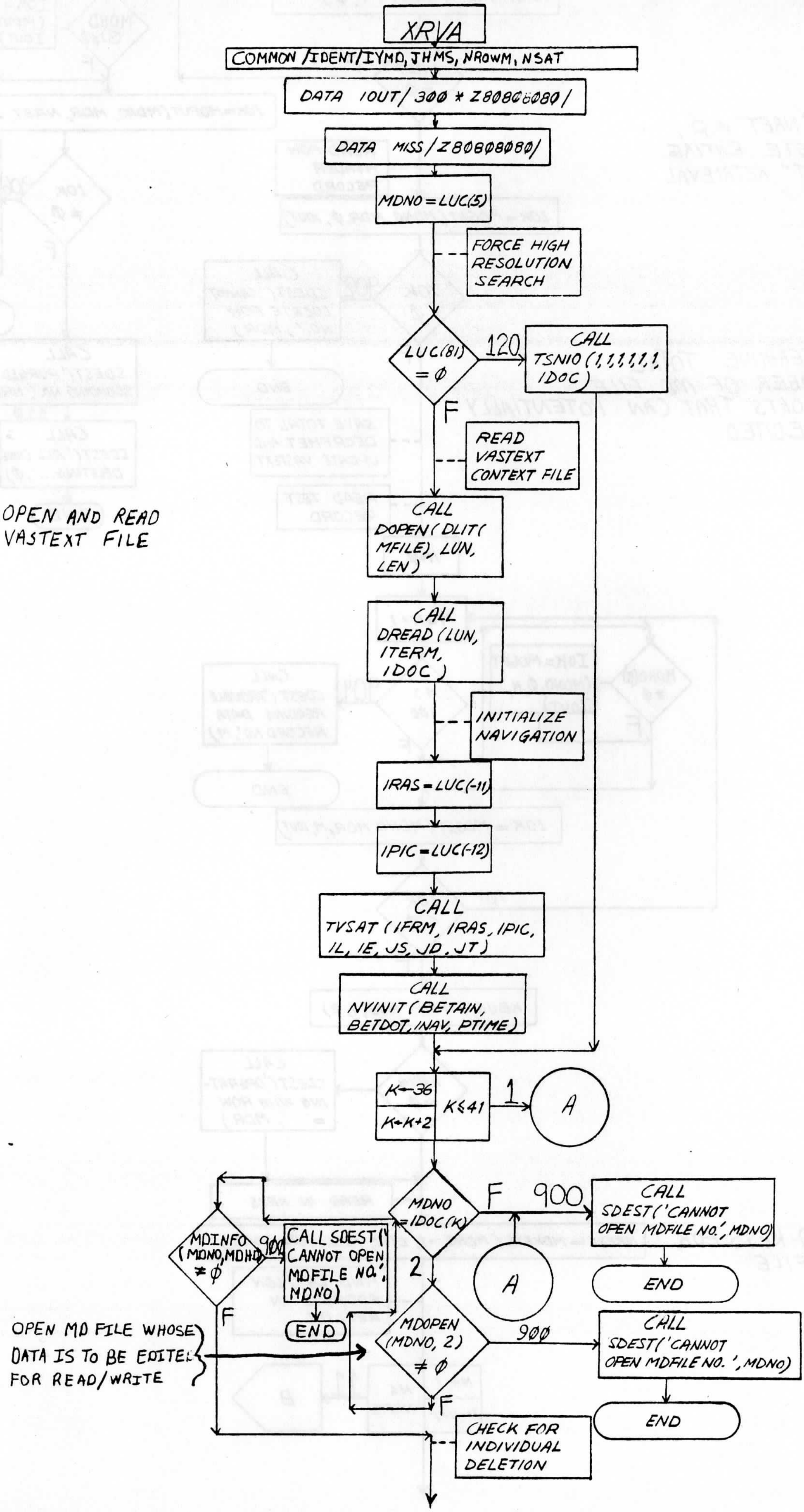
As a final note, a few sample keyins which relate to the previous text will be presented. These examples should help settle any confusion concerning the previous discussion.

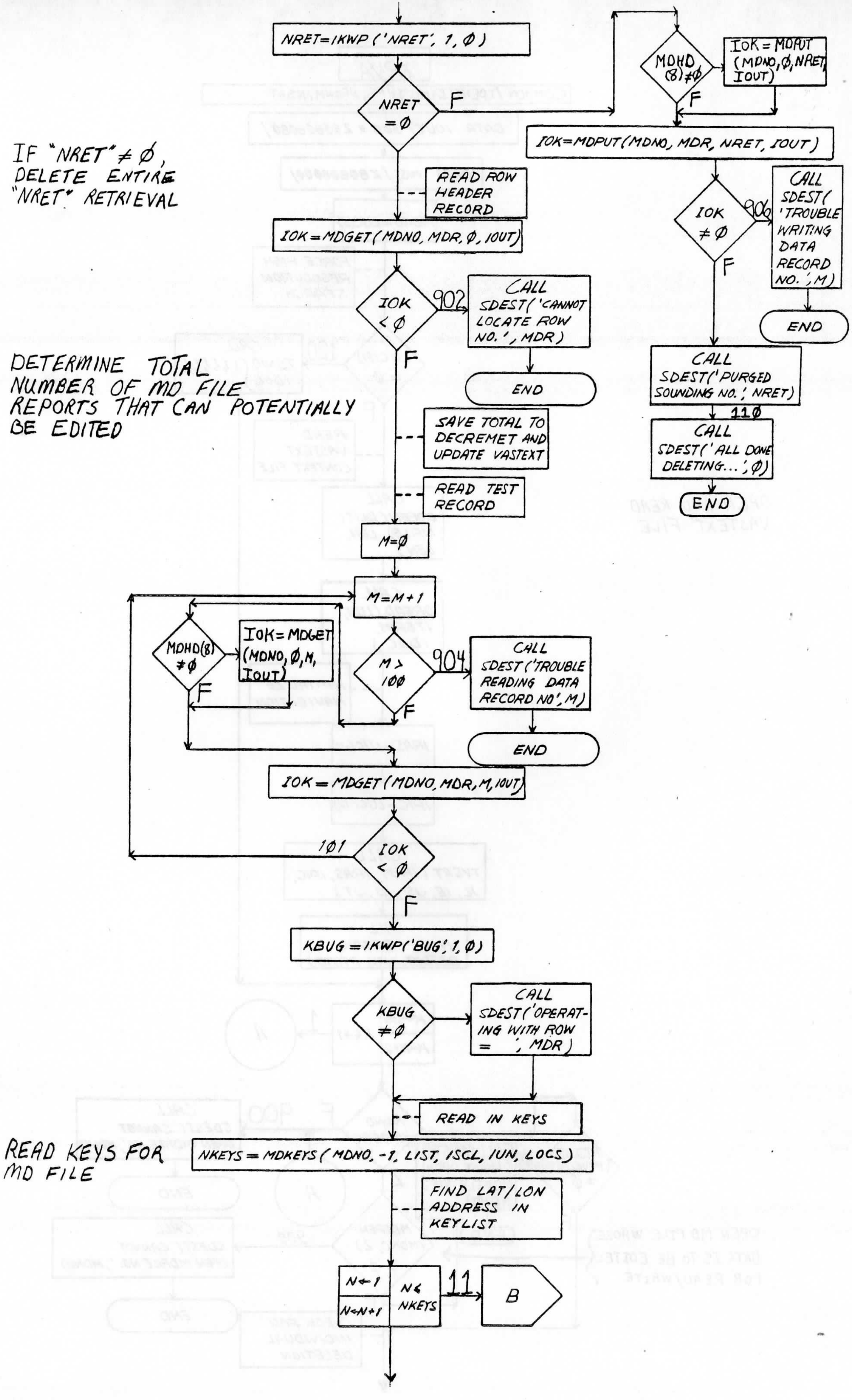
1.) XRVA: Delete entire report(s) that lie within the cursor from MD file whose number was set before XRVA via the "MDU SET."

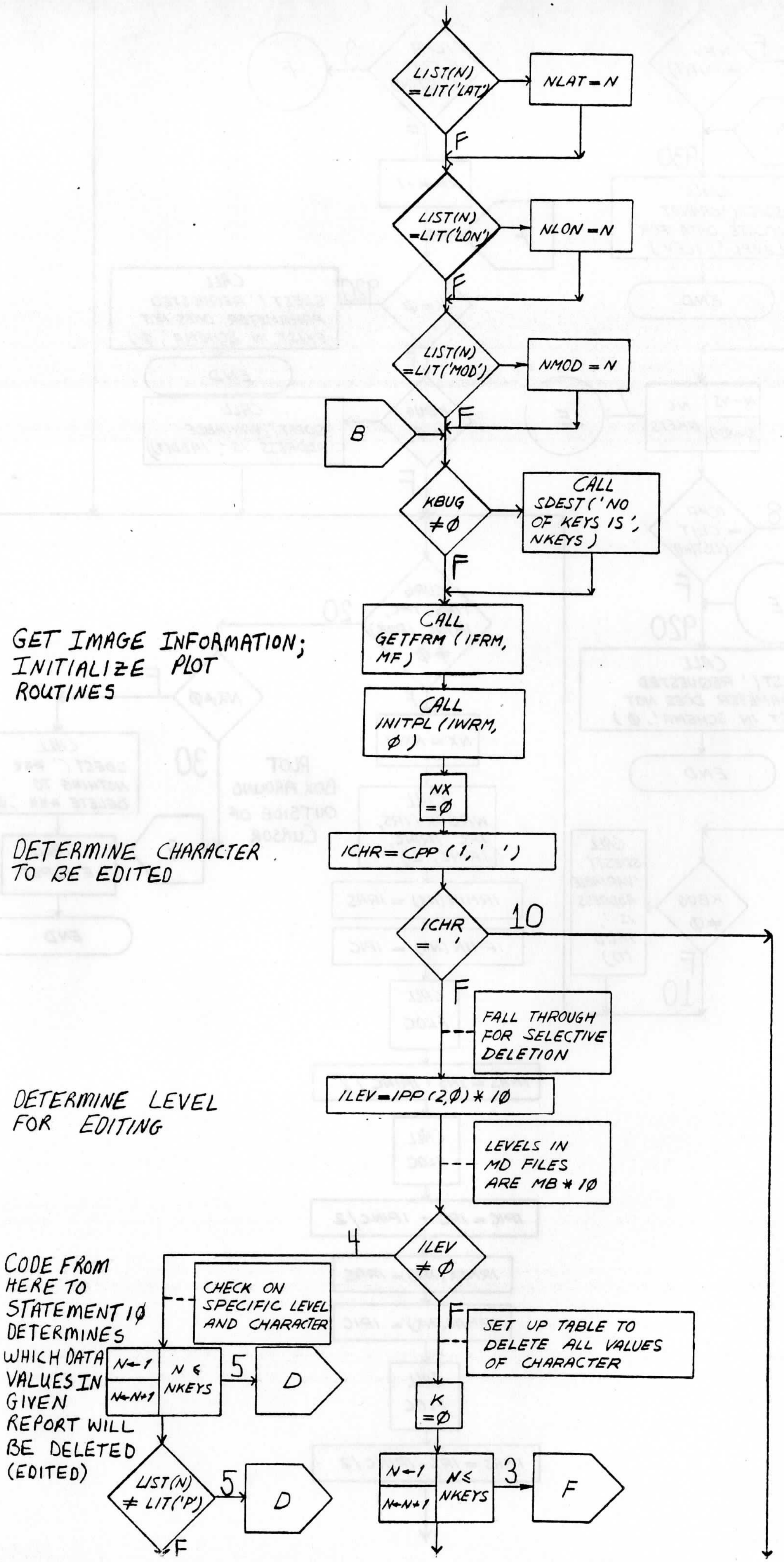
2.) XRVA Z100: Delete all values of 1000 mb height from report(s) located within cursor, with report(s) again being taken from MD file whose number was set via program MDU.

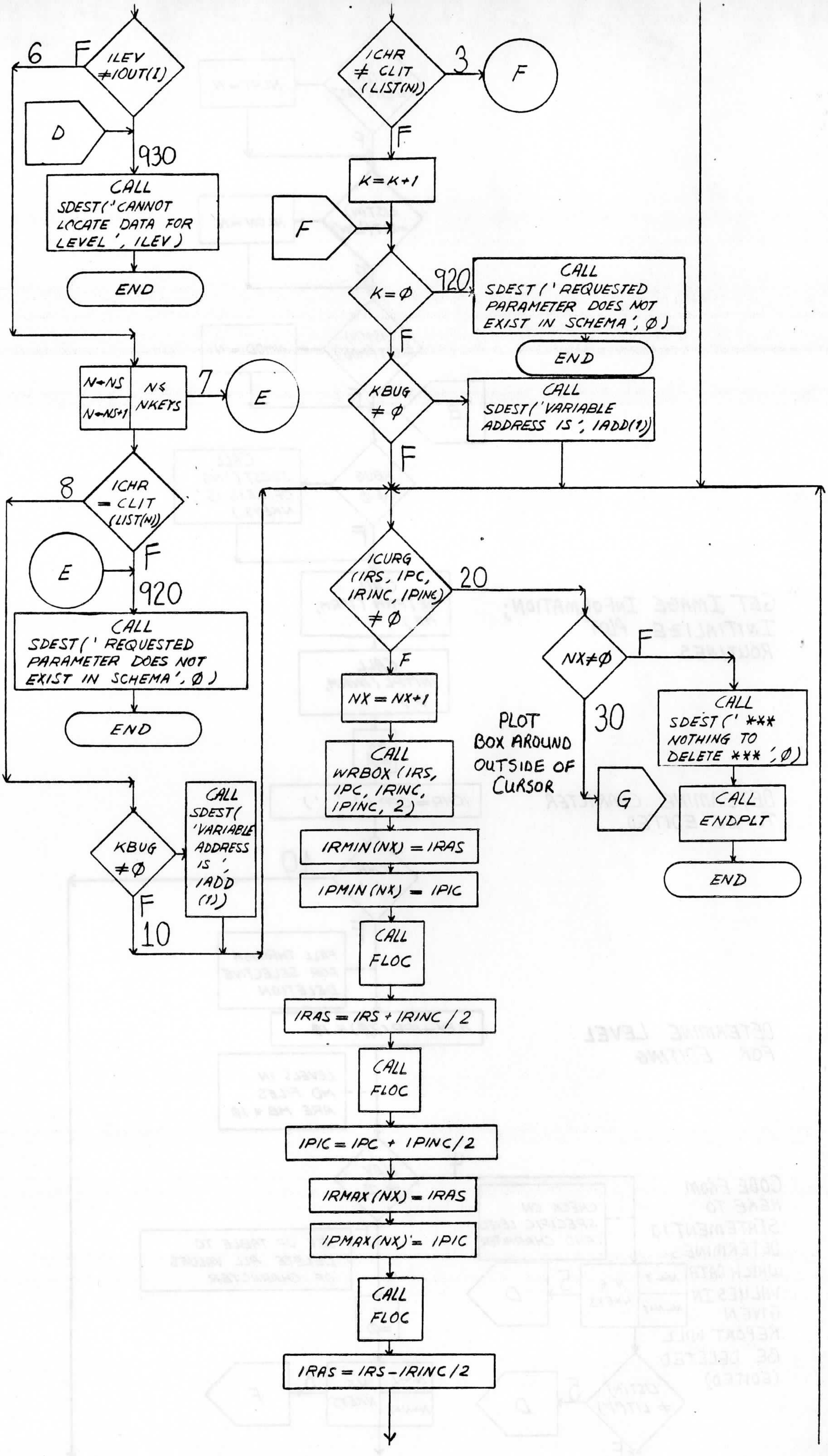
3.) XRVA NRET=105: Delete report #105 (exists in column 105) from row MDR of MD file whose number was set by MDU.

4.) XRVA Z 500: Delete all values of 500 mb temperature from report(s) contained within the cursor (report(s) once again taken from MDU-defined MD file).

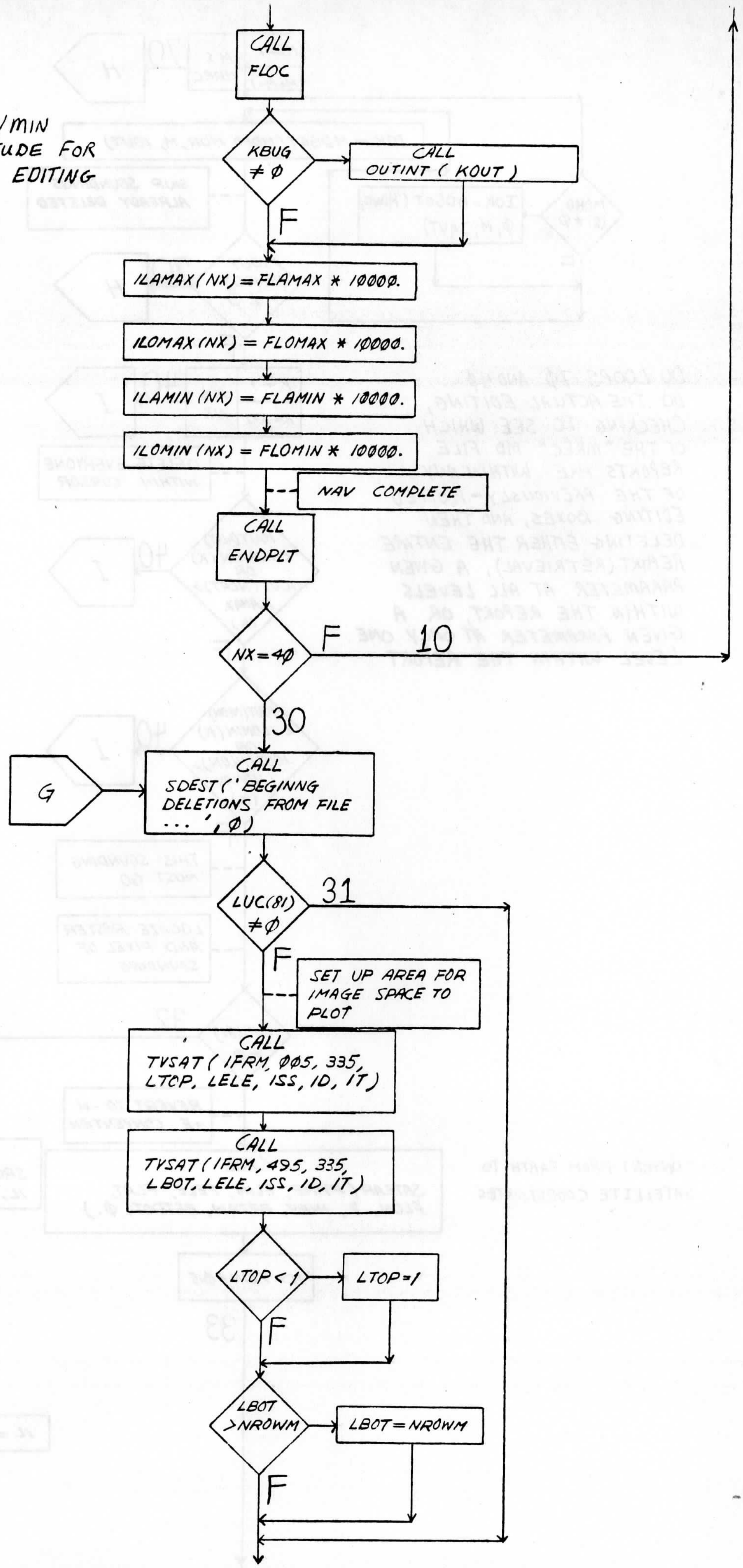

XRVA
COMMON /IDENT/IYMD, JHMS, NROWM, NSAT
DATA IOUT/ 300 * Z8080080/
DATA MISS/Z80808080/
MDNO=LUC(5)
FORCE HIGH RESOLUTION SEARCH
LUC(81) = 0
120
CALL TSNIO (1,1,1,1,1,1, IDOC)
F
READ VASTEXT CONTEXT FILE
OPEN AND READ VASTEXT FILE
CALL DOPEN (DLIT(MFILE), LUN, LEN)
CALL DREAD (LUN, ITERM, IDOC)
INITIALIZE NAVIGATION
IRAS=LUC(-11)
IPIC=LUC(-12)
CALL TVSAT (IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)
CALL NVINIT (BETAIN, BETDOT, INAV, PTIME)
K←36
K←K+2
K≤41
1
A
MDNO =IDOC(K)
F
900
CALL SDEST('CANNOT OPEN MDFILE NO.', MDNO)
END
2
A
MDINFO (MDNO,MDHD) ≠ 0
900
CALL SDEST('CANNOT OPEN MDFILE NO.', MDNO)
F
END
OPEN MD FILE WHOSE DATA IS TO BE EDITED FOR READ/WRITE
MDOPEN (MDNO, 2) ≠ 0
900
CALL SDEST('CANNOT OPEN MDFILE NO.', MDNO)
END
F
CHECK FOR INDIVIDUAL DELETION

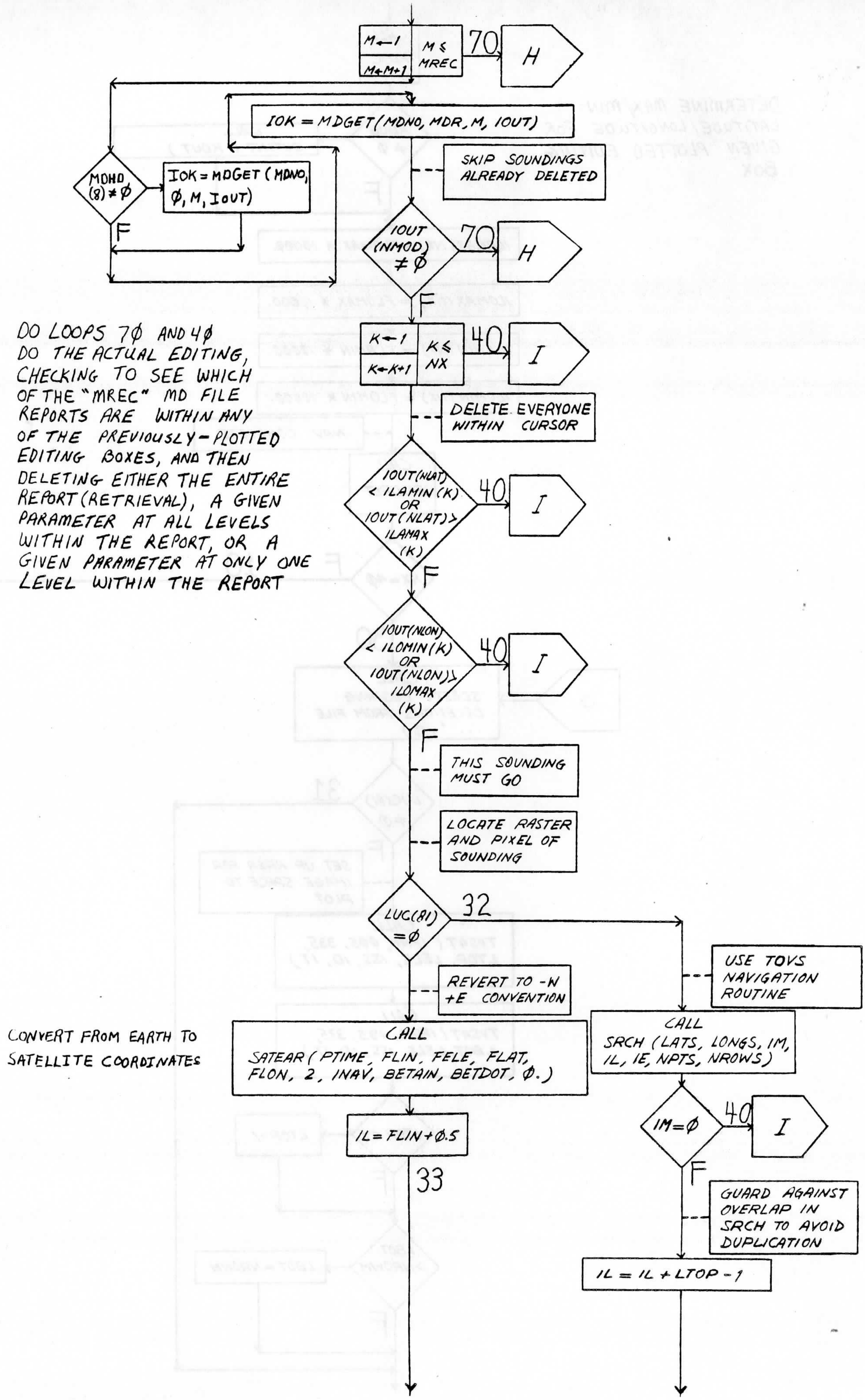

NRET=IKWP('NRET', 1, Ø)
NRET = Ø
F
IF "NRET" ≠ Ø, DELETE ENTIRE "NRET" RETRIEVAL
READ ROW HEADER RECORD
IOK=MDGET(MDNO, MDR, Ø, IOUT)
IOK < Ø
902
CALL SDEST('CANNOT LOCATE ROW NO.', MDR)
END
F
DETERMINE TOTAL NUMBER OF MD FILE REPORTS THAT CAN POTENTIALLY BE EDITED
SAVE TOTAL TO DECREMET AND UPDATE VASTEXT
READ TEST RECORD
M=Ø
M=M+1
MDHD(8) ≠ Ø
IOK=MDGET (MDNO, Ø, M, IOUT)
F
M > 100
904
CALL SDEST('TROUBLE READING DATA RECORD NO', M)
F
END
IOK=MDGET(MDNO, MDR, M, IOUT)
1Ø1
IOK < Ø
F
KBUG=IKWP('BUG', 1, Ø)
KBUG ≠ Ø
CALL SDEST('OPERATING WITH ROW = ', MDR)
READ IN KEYS
READ KEYS FOR MD FILE
NKEYS=MDKEYS(MDNO, -1, LIST, ISCL, IUN, LOCS)
FIND LAT/LON ADDRESS IN KEYLIST
N←1
N←N+1
N ≤ NKEYS
11
B
MDHD (8) ≠ Ø
IOK=MDPUT (MDNO, Ø, NRET, IOUT)
F
IOK=MDPUT(MDNO, MDR, NRET, IOUT)
IOK ≠ Ø
906
CALL SDEST('TROUBLE WRITING DATA RECORD NO.', M)
F
END
CALL SDEST('PURGED SOUNDING NO.', NRET)
11Ø
CALL SDEST('ALL DONE DELETING...', Ø)
END


LIST(N) =LIT('LAT')
NLAT = N
F
LIST(N) =LIT('LON')
NLON = N
F
LIST(N) =LIT('MOD')
NMOD = N
F
B
KBUG ≠ Ø
CALL SDEST('NO OF KEYS IS', NKEYS)
F
GET IMAGE INFORMATION; INITIALIZE PLOT ROUTINES
CALL GETFRM (IFRM, MF)
CALL INITPL (IWRM, Ø)
NX = Ø
DETERMINE CHARACTER TO BE EDITED
ICHR = CPP (1, ' ')
ICHR = ' '
10
F
FALL THROUGH FOR SELECTIVE DELETION
DETERMINE LEVEL FOR EDITING
ILEV = IPP (2,Ø) * 1Ø
LEVELS IN MD FILES ARE MB * 1Ø
4
ILEV ≠ Ø
CODE FROM HERE TO STATEMENT 1Ø DETERMINES WHICH DATA VALUES IN GIVEN REPORT WILL BE DELETED (EDITED)
CHECK ON SPECIFIC LEVEL AND CHARACTER
F
SET UP TABLE TO DELETE ALL VALUES OF CHARACTER
N←1
N←N+1
N ≤ NKEYS
5
D
K = Ø
LIST(N) ≠ LIT('P')
5
D
F
N←1
N←N+1
N ≤ NKEYS
3
F


ILEV ≠ IOUT(I)
6
F
D
930
CALL SDEST('CANNOT LOCATE DATA FOR LEVEL', ILEV)
END
N←NS
N←NS+1
N≤NKEYS
7
E
8
ICHR = CLIT(LIST(N))
F
E
920
CALL SDEST(' REQUESTED PARAMETER DOES NOT EXIST IN SCHEMA', Ø)
END
KBUG ≠ Ø
CALL SDEST('VARIABLE ADDRESS IS', IADD(1))
F
10
ICHR ≠ CLIT(LIST(N))
3
F
F
K=K+1
F
K=Ø
920
CALL SDEST('REQUESTED PARAMETER DOES NOT EXIST IN SCHEMA', Ø)
END
F
KBUG ≠ Ø
CALL SDEST('VARIABLE ADDRESS IS', IADD(1))
F
ICURG(IRS, IPC, IRINC, IPINC) ≠ Ø
20
F
NX=NX+1
CALL WRBOX(IRS, IPC, IRINC, IPINC, 2)
IRMIN(NX) = IRAS
IPMIN(NX) = IPIC
CALL FLOC
IRAS = IRS + IRINC/2
CALL FLOC
IPIC = IPC + IPINC/2
IRMAX(NX) = IRAS
IPMAX(NX) = IPIC
CALL FLOC
IRAS = IRS-IRINC/2
PLOT BOX AROUND OUTSIDE OF CURSOR
NX≠Ø
F
30
G
CALL SDEST(' *** NOTHING TO DELETE ***', Ø)
CALL ENDPLT
END

DETERMINE MAX/MIN LATITUDE/LONGITUDE FOR GIVEN PLOTTED EDITING BOX

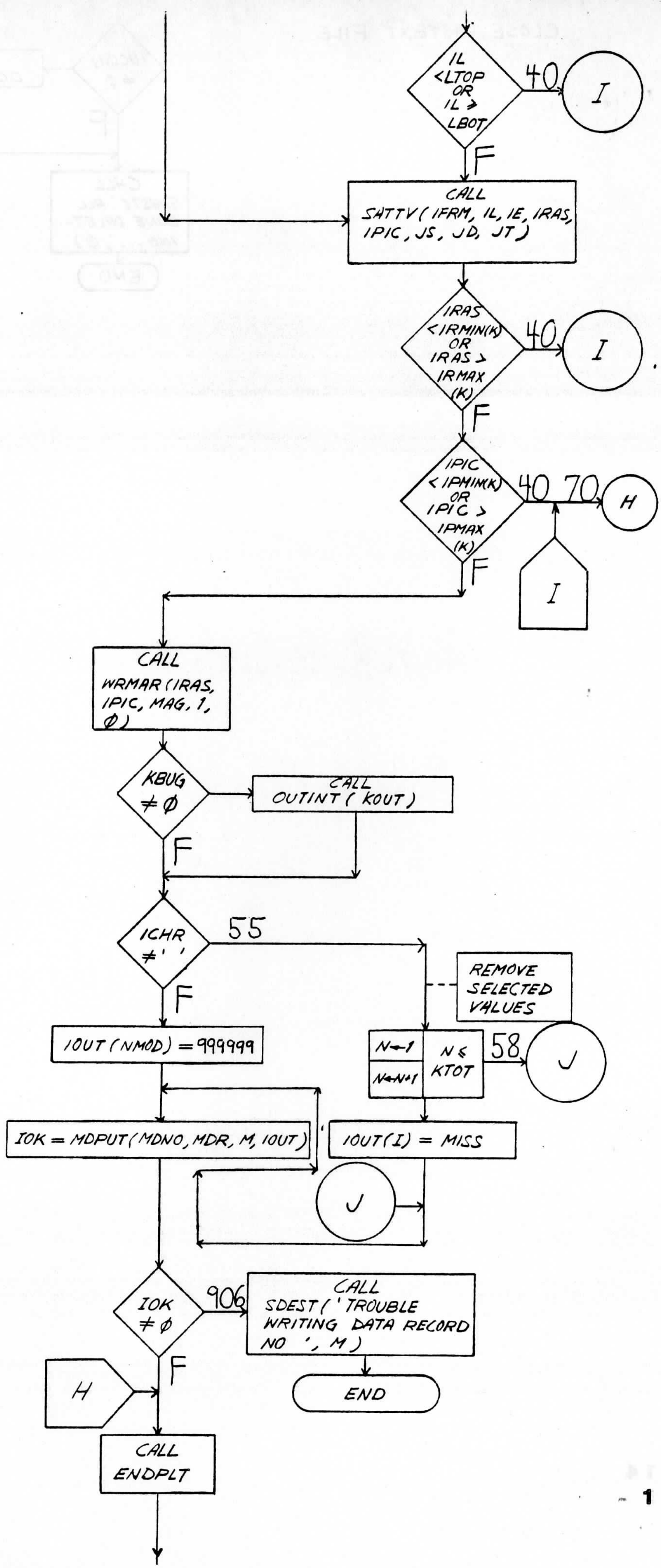



M←1
M←M+1
M ≤ MREC
70
H
IOK = MDGET(MDNO, MDR, M, IOUT)
SKIP SOUNDINGS ALREADY DELETED
MDHD (8) ≠ Ø
IOK = MDGET (MDNO, Ø, M, IOUT)
F
IOUT (NMOD) ≠ Ø
70
H
F
DO LOOPS 7Ø AND 4Ø DO THE ACTUAL EDITING, CHECKING TO SEE WHICH OF THE "MREC" MD FILE REPORTS ARE WITHIN ANY OF THE PREVIOUSLY-PLOTTED EDITING BOXES, AND THEN DELETING EITHER THE ENTIRE REPORT (RETRIEVAL), A GIVEN PARAMETER AT ALL LEVELS WITHIN THE REPORT, OR A GIVEN PARAMETER AT ONLY ONE LEVEL WITHIN THE REPORT
K←1
K←K+1
K ≤ NX
40
I
DELETE EVERYONE WITHIN CURSOR
IOUT(NLAT) < ILAMIN(K) OR IOUT(NLAT) > ILAMAX (K)
40
I
F
IOUT(NLON) < ILOMIN(K) OR IOUT(NLON) > ILOMAX (K)
40
I
F
THIS SOUNDING MUST GO
LOCATE RASTER AND PIXEL OF SOUNDING
LUC(81) = Ø
32
REVERT TO -W +E CONVENTION
USE TOVS NAVIGATION ROUTINE
CONVERT FROM EARTH TO SATELLITE COORDINATES
CALL SATEAR (PTIME, FLIN, FELE, FLAT, FLON, 2, INAV, BETAIN, BETDOT, Ø.)
CALL SRCH (LATS, LONGS, IM, IL, IE, NPTS, NROWS)
IM = Ø
40
I
F
IL = FLIN + Ø.5
33
GUARD AGAINST OVERLAP IN SRCH TO AVOID DUPLICATION
IL = IL + LTOP - 1

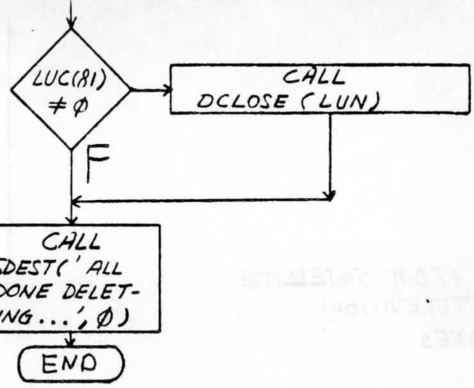
CONVERT FROM SATELLITE TO TV (TELEVISION) COORDINATES

PLOT RED "X" OVER EDITED REPORT

CARRY OUT DELETIONS



CLOSE VASTEXT FILE



Subroutines used by XRVA:

1) DOPEN-I
2) DREAD-I
3) TVSAT-I
4) NVINIT-I
5) TSNIO-I
6) SDEST-I
7) GETFRM-I
8) INITPL-I
9) WRBOX-I
10) FLOC-I
11) OUTINT-I
12) SATEAR-I
13) ENDPLT-I
14) SRCH-I
15) SATTV-I
16) WRMAR-I
17) DCLOSE-I
18) ENCODE
19) ENCODX
20) ABORT
21) CONTNT
22) LTQ
23) ZECONV
24) MOVB
25) CLEANA
26) TQ
27) PRLINX
28) PRCLOS
29) PROPEN
30) PRPRPR
31) LOCK
32) PRWR
33) PRRD
34) UNLOCK
35) PRCL
36) POST
37) BLKA
38) TRMNL
39) TQMES
40) STC
41) ITOC
42) GETNAV
43) EPOCH
44) GETGAM
45) VASNAV
46) DDEST
47) MOVCW
48) CLEANW
49) EDEST
50) MOVW
51) II
52) DWRITE
53) PLOT
54) SATPOS
55) ENPT
56) PENADD
57) PACK
58) SENOUT
59) ATOE
60) TEKPUT
61) PAGE
62) PENMOV
63) BOX
64) PENBEG
65) WALK

VTPZ

VTPZ is the operational VAS retrieval program. For a given retrieval, it generates vertical profiles of temperature and moisture by using a "one-step" physical matrix inversion retrieval algorithm (Smith and Woolf, 1984). VTPZ can be run to generate either one retrieval or a larger area of retrievals, depending on how the program is keyed in on McIDAS. In addition, an option exists to create an image of total precipitable water vapor and also possibly another image of total-totals index simultaneously with the retrievals. These image(s) are stored in separate McIDAS digital areas (of type VISSR). Finally, VTPZ can be run in either of two modes: auto (non-video) or non auto (video). If the user doesn't have access to a video terminal, the program must be run in the auto mode, while video terminal users can run VTPZ in either of the two modes. In this discussion, the operation of VTPZ will be presented from the standpoint of the non-auto mode. Pertinent differences between this mode and the auto mode will be duly noted as the discussion proceeds.

Initially, after a message saying "BEGIN*VTPZ*VERSION OF 3 MAY 84 AT INIT 2" is displayed on the CRT, the number(s) of the digital area(s) into which image(s) of total precipitable water vapor and total-totals index is(are) to be placed is(are) set. Note that an image of total-totals index cannot be created independent of an image of total precipitable water vapor; that is, the user can either create both images, or one image of total precipitable water vapor. This is to be understood throughout the remainder of this discussion. (As a side note, if the user wishes to know the correct number of retrievals made in the retrieval area, he/she should, before executing VTPZ, execute program SPVA as follows: SPVA NRET=0.) After the number of areas to be generated (0.LE. NAREAS .LE. 2) has been calculated, a call to subroutine VRTIO is encountered. This subroutine opens, reads and closes the VASTEXT file, opens the retrieval MD file (MDNR) for read/write, and reads the row header for the same file. The information contained in the VASTEXT file (more explicitly, in the array IDOC) allows VTPZ to set variables JDAY, JTIME, MDNG, MDNR, and MDRR. Note that MDNG, MDNR and MDRR are, respectively, the retrieval guess MD file number, retrieval MD file number, and retrieval MD file row number.

Now, assuming the retrieval MD file row number is .GT. 0 (IF(NARA.EQ.0.AND.MDRR.LE.0) GO TO 1300), the type of guess to be used for the retrievals ("G"= grid (default), or "C"= climatology) is evaluated, and variable IGES is set accordingly to 0 or 1. After the surface option is set, the size (in fields of view (FOVs)) of the box in which retrievals are to be made (NBXS) is determined. If the user does not key in any information about the retrieval box size, it will become 11 FOVs in dimension by default, and variable LDETR will assume a value of 0. However, if the user desires a different retrieval box size, and the VAS data was gathered via the large detectors (resolution = approximately 16 km), keyword parameter BOX should be keyed in with an EVEN integer value ("BOX=6," for example). Assuming BOX is keyed in as 6, the resultant retrieval box size will be 5 X 5 FOVs (the typical box size for retrievals using large-detector

VAS data). In addition, variable LDETR will be set to 1. On the other hand, if the data was gathered via the small detectors (resolution = approximately 8 km), the retrieval box size should be keyed in as an ODD integer value ("BOX=9," for example). With this sample keyin, the resultant retrieval box size will be 9X9 FOVs (NBXS=MINO(NBXS,11)), and LDETR will be set to 0. Note that the range of retrieval box sizes possible with VTPZ is 1 to 11.

Subsequent code determines the retrieval spacing in FOVs in both the line (y) and element (x) directions (INCRL,INCRE), and sets the end/beginning coordinates of the retrieval area to be processed. Note the VASTEXT defaults for the end/beginning coordinates. After the cursor location in TV coordinates (rasters, pixels) has been evaluated (INRAS=LUC(-11), INPIC=LUC(-12)), variable GAMRET (used in retrieval subroutine QVTWR) is determined, and the system date is derived via subroutine GETDAY. At this point, the dimensions of the area (or coordinates of a single retrieval) to be processed are defined. There are numerous different methods of defining areas to be processed, and at least two ways to process only one retrieval. However, the usual case is to key in the ending line/element coordinates of the retrieval area, and let the VASTEXT file default values determine the beginning line/element coordinates (see sample keyin of VTPZ for the non-auto mode within instruction 22 of the sample keyins near the end of chapter 1). Note that a keyin of this type will result in a call to subroutine GETFRM (returns a frame directory relating to the sounder area previously loaded into a given image frame), while subroutine TVSAT will not be called. If the user was running VTPZ in the auto mode on either a video or non-video terminal, there would be no call either to subroutine GETFRM or TVSAT. (Note: LUC(16) = 1 for a video terminal and 0 for a non-video terminal.) Again, see the sample keyin of VTPZ, this time for the auto mode, at the same location in Chapter 1.

Following the setting up of the area to be processed, subroutine VASDAT is called. Since VDAT(1)=-1. when the routine is called, only navigation information is returned. Included in this navigation information is satellite number, Julian date, time the first VAS data in the image was gathered, and line and element resolutions of the image. Next, variables INCIL and INCIE are calculated. INCIL stores the number of satellite coordinate lines separating successive retrieval lines, while INCIE contains the number of satellite coordinate elements separating successive retrieval boxes on any given retrieval line. These two variables are used subsequently to increment the two most important DO LOOPS of the program -- 1260 and 1180. The last lines of code before DO LOOP 190 set several more variables, most notably variables KLINES, KELEMS and KBOXES. The first two deal with retrieval box dimensions, and both have the same magnitude as variable NBXS discussed earlier. KBOXES, on the other hand, holds a value indicating how many retrieval boxes can fit in the x-direction of the area to be processed; that is, on a given retrieval line. For a visual summary of the previous discussion, see Diagram 1 immediately preceeding VTPZ's Level I flowchart. This diagram shows approximately how a small (only 4 retrievals wide) retrieval area looks once it is set up for processing.

Assuming the user wishes to generate at least an image of total precipitable water vapor (NARA .NE. 0), and assuming also that MELE (number of elements in the x-direction of the area to be processed) is not .GT. 660, DO LOOP 190 is now entered. This loop creates and prepares for data insertion up to two digital areas, one to store the image of total precipitable water vapor, the other (if desired) to store an image of total-totals index.

At this point, after a message concerning the opening up of the precipitable water vapor digital area is displayed on the CRT, VTPZ enters DO LOOP 1260. This loop increments each time a given line of retrieval boxes has been completely processed. Note also the beginning, ending and increment values of the loop (ILINE, LLINE and INCIL). Almost immediately after DO LOOP 1260 has been entered, DO LOOP 1180 is encountered. This loop, nested within 1260, increments each time a retrieval box within a given retrieval line has been completely processed.

The first thing done in DO LOOP 1180 is an initialization of the arrays TDAT and IARRAY. TDAT will eventually hold the measured VAS brightness temperature (TBB) data for all 12 bands for each FOV in a given retrieval box being processed. On the other hand, IARRAY, which is used only in the image-generating section of VTPZ, will hold the total precipitable water vapor amount and (if a second image is desired) the total-totals index for each FOV in all the retrieval boxes in a given retrieval line.

After these arrays have been initialized, TBB and other miscellaneous data for each FOV of a given retrieval box is accessed in DO LOOPs 330 and 320. The subroutine which does the actual data-gathering is VASDAT. The input arguments (IL, IE) give the line/element coordinates of the FOV for which the data is to be collected, while the array VDAT is used to return the TBB data itself. The other data is returned via common blocks such as DANGLE, LAST and NAV. After the data for a given FOV has been accessed, several data quality tests are performed, including tests for latitude, satellite zenith angle, land elevation , and whether TBB data exists for that particular FOV.

Assuming these tests pass, VTPZ enters DO LOOP 300. This loop, which runs through all 12 bands for a given FOV, begins with a check to delete extreme TBB values for a given band. Assuming the data for a given band is reasonable, it is stored in the array TDAT, and the spin budget for that particular band is saved in the array NSPIN. Then, if the loop is on band 8, variable MSAM (number of FOVs with good data in a given retrieval box), SELEV (sum of surface elevations for all the good-data FOVs in a given retrieval box) ,and LSTYPE (number of FOVs over land in a given retrieval box) are updated.

Upon the exit of DO LOOPs 330 and 320, and assuming at least one decent FOV of data was found in the retrieval box, the mean surface elevation for the box (SELEV=SELEV/FLOAT(MSAM)) is calculated, and the program proceeds to obtain the first guess for the retrieval.

The subroutine which gathers the first guess is GESPRO. Returned via this subroutine are 40 values of guess temperature and mixing ratio (at levels from 0.1 to 1000 mb), as well as other assorted surface parameters (such as surface pressure, temperature and dewpoint). After the surface mixing ratio is calculated via subroutine WMIX, the first pressure level below the surface (IS) is determined in DO LOOP 340. This level will be used to define the lowest level in the forthcoming retrieval. Note that level IS can never be .GT. 40 and can, in rare situations, be equal to the actual surface pressure. Usually, however, the lowest level of the retrieval will not be equal to the actual surface pressure. After IS has been evaluated, the guess temperature profile is adjusted below 700 (ILO=35) or 500 (ILO=31) mb to take into account the surface temperature, 700 mb being the top level of adjustment for a grid first guess (usually the LFM), and 500 mb being the top level of adjustment for a climatology first guess. See the code immediately preceeding DO LOOP 380 and DO LOOP 380 itself for details.

Following DO LOOP 380, the temperature and mixing ratio arrays for all the retrieval levels from level IS down to level 40 (1000 mb) are set equal to the value found at level IS. This is done in DO LOOP 420. DO LOOP 440 then adjusts the entire guess moisture (mixing ratio) profile (all 40 levels) to take into account the surface mixing ratio value. Note that the final mixing ratio at a given level is constrained to lie between .02 g/kg and the saturation value, inclusive. As the last step in DO LOOP 440, the final guess temperature and mixing ratio values for all 40 retrieval levels are saved in the arrays TG and WG.

The next major step in VTPZ is the calling of subroutine

FILTER. This subroutine generates retrieval box-averaged TBBs in each band 1-12, as well as the expected error of the TBB observations for each band (array ERM), the number of non cloud-contaminated FOVs for the given retrieval box (MSAM), and a new average retrieval box surface elevation which makes use only of the non cloud-contaminated FOVs (see variable IELEV in common block SURF). Upon returning to VTPZ, a check is made to make sure average TBB data was returned in at least 7 of the first 10 VAS bands (DO LOOP 480). Assuming that this check passes and that there is no problem in calculating the surface skin temperature (TSKIN), variables such as ERTSFC and ERWSFC are set. These variables are used in subroutine QVTWR, which does the actual "one-step" retrievals, to define the expected errors in the input surface temperature and mixing ratio, respectively. Thus, the retrieved surface temperature and mixing ratio are constrained to satisfy the input surface values to within these nominal error estimates. Note also the re-calculation of variables TSTA, WSTA and TDSTA using band 7 and 8 TBB information in the event the surface option is not being used. This is done to give a better first guess at the surface in these cases than would be accessed via climatology. Finally, just before the retrieval is performed, DO LOOPs 1000 and 1020 store guess temperature and dewpoint data, respectively, in the arrays TGS and DGS.

Now, the most important section of VTPZ begins -- the retrieval of the temperature and moisture profiles. The subroutine which does the actual retrieval is QVTWR. For further information about the retrieval algorithm, see Smith and Woolf

(1984). If a given retrieval fails, the error flag IFAIL is set to 1 within QVTWR, and VTPZ will move to the next retrieval box to attempt a retrieval there. Assuming, however, that the retrieval is successful, the resultant temperature and moisture (mixing ratio) output is saved in the arrays TG and WG (DO LOOP 1030), after which the retrieval subroutine is called again. This time, however, variable GAMRET holds a value equal to 1/10 the value it held when the first retrieval was done. In effect, this second retrieval "fine tunes" the result from the first retrieval; that is, the user can think of the first retrieval as a first guess for the second retrieval.

Assuming the second retrieval is successful, VTPZ checks to see if the retrieval box had 5 or more non cloud-contaminated FOVs. If there were less than 5 non cloud-contaminated FOVs, the program jumps to statement 490, meaning that this particular retrieval will be used at most only as a contributor to the image(s) of total precipitable water vapor and total-totals index (if desired), and will NOT be saved in the retrieval MD file. Otherwise, if there WERE 5 or more non cloud-contaminated FOVs, the retrieval is satisfactory for inclusion into the retrieval MD file.

Before the retrieval is stored in the retrieval MD file, however, several additional calculations are made. First, a call to subroutine PRECW results in the calculation of total precipitable water vapor down to the lowest level of the retrieval (IS). Following this, dewpoints are calculated from 300 to 1000 mb, the total-totals index is determined, and the retrieval profile heights (subroutine HTV) are computed. Next, preparation of the arrays PST, TST and TDST preceed a call to subroutine SNDANL, whose function is to compute stability indices (such as lifted, K and SWEAT) for the just-completed retrieval. Finally, if the debug option is on (keyword parameter BUG keyed in with a 1 or 2), information concerning the retrieval profile, as well as the total-totals index value computed earlier, will be printed on the CRT (a keyin of "BUG=1" in the VTPZ keyin) or local printer (a keyin of "BUG=2" in the same).

At this point, VTPZ is ready to begin preparing the output buffer; that is, it begins filling the array IRET with retrieval quantities for insertion into the retrieval MD file. The filling of IRET occupies the program for approximately the next 60 lines of code. The data inserted into IRET consists of things such as the user mod flag (IRETD(3) -- a setting to 0 indicates that a good retrieval was performed), retrieval number (IRETD(4)), as well as the averaged TBBs and expected error of the TBB observations for bands 1-12 (DO LOOP 1140). (It should be mentioned at this point that an EQUIVALENCE statement near the beginning of the program equivalences the arrays IRET and IRETD (EQUIVALENCE(IRET(17),IRETD(1))). Other quantities stored in IRET include the total precipitable water vapor amount (IRETD(35)), and the total totals and lifted indices (IRETD(41) and (43), respectively). Next, DO LOOP 1160 stores the retrieval heights and both the retrieval and guess temperature and dewpoint arrays (only at or above the surface). A final input of some surface parameters completes the retrieval MD file output buffer. After the total number of retrievals performed so far (LASRET) is incremented by 1 and placed in IDOC(100) (note: the array IDOC holds current VASTEXT file data), a call to subroutine VRTIO (CALL VRTIO(IRET,LASRET,1)) puts IRET into column LASRET (and row MDRR as specified in the VASTEXT file) of the retrieval MD file.

At this point, if no image(s) of total precipitable water vapor and total-totals index are to be generated (NARA .EQ. 0), DO LOOP 1180 increments one retrieval box spacing, and an attempt to process the next retrieval begins (IF(NARA.EQ.0) GO TO 1180). However, if image(s) of these two quantities is(are) desired (NARA .NE. 0), VTPZ proceeds to DO LOOPs 560 and 540. Together, these two loops generate value(s) of total precipitable water vapor (TPWV) and total-totals index for EACH FOV in a given retrieval box (remember that the earlier retrieved values of TPWV and total-totals index were for an ENTIRE retrieval box, not for each individual FOV). The process is carried out in a scan line-by-scan line fashion within each box. The precipitable water vapor and its image value are calculated first (see subroutine PRECW and variables URET/IURET approximately 30 lines into DO LOOP 540). The final vapor amount is limited to the range 1.-12.cm, inclusive, while the vapor image value, stored in IARRAY(KELE,KLIN,KBOX,1), is limited to the range 20-240, also inclusive. (As a side note, an image value of 0 is pure white, while a value of 255 is pure black.) Then, if a second image of total-totals index is desired (IAREA(2) .NE. 0), and the surface pressure is .GE. 850 mb, a value of total-totals index (TOTL) and a corresponding image value (ITOTL) are calculated, with the image value being stored in IARRAY(KELE,KLIN,KBOX,2). This step then concludes the image-generation section of VTPZ. Note that the program is now at the end of DO LOOP 1180, so control will pass back to the beginning of the loop, where another retrieval will be attempted.

Once a given retrieval line has been completely processed in the retrieval area (DO LOOP 1180 exited), VTPZ checks to see if image (single FOV) total precipitable water vapor and (if desired) total-totals index data was generated previously in DO LOOPs 560 and 540 (IF(NARA.EQ.0) GO TO 1255). Assuming an image of at least one of these quantities is to be outputted to digital area(s) (IF (NARA.EQ.0) GO TO 1255 is NOT true), VTPZ enters DO LOOP 1250. This loop generates one retrieval line image of total precipitable water vapor, and then (if desired) generates a second retrieval line image of total-totals index. Note that the loop processes the data in one retrieval line by completely processing all the FOVs in successive scan lines (DO LOOPs 1220 and 1200), until all the scan lines contained within the retrieval line have been processed.

The data itself is taken from the array IARRAY (recall that the raw data for the image(s) was stored in this array). Then, subroutine PACK stores one scan line's worth of data in the array IABUF, after which subroutine WRITA writes this data into the digital area KARA. At this point, if variable IAGAP (the difference between the retrieval line spacing (in FOVs) and the retrieval box size) is .LE. 0, DO LOOP 1250 is either done (if only a retrieval line image of total precipitable water vapor is desired), or it loops back to generate a second retrieval line image of total totals index. Assuming IAGAP .GT. 0 (meaning a segment of empty scan lines exists between successive retrieval lines), subroutines PACK and WRITA (DO LOOP 1240) fill in the empty image space in the digital area with default image values (the constant MAXWV, which = 255 (white)).

With the conclusion of DO LOOP 1250, VTPZ has completely processed one line of retrieval boxes within the retrieval area. DO LOOP 1260 now loops back to its beginning, incrementing the retrieval line coordinate in the process by the amount INCIL (which will then give the initial line coordinate of the sequence of retrieval boxes to be processed in the next retrieval line).

If all the retrieval lines within the retrieval area have been completely processed, DO LOOP 1260 exits. Then, the arrays IDOC and IRET are updated with the total number of retrievals performed (LASRET), after which a call to subroutine VRTIO updates the row header for the retrieval MD file, as well as the VASTEXT file. If no image(s) were created, this last step concludes VTPZ. However, assuming that at least one image was generated and stored in a digital area, DO LOOP 1265, which closes the digital area(s), is executed, after which VTPZ concludes by printing the message "ALL DONE" on the CRT. This concludes the discussion of the May 3, 1984 version of VTPZ.

Several important changes have been made in VTPZ since the May 3 version described in this manual. Therefore, given the importance of this program, an appendix has been included on the following 2 pages, the details of which describe the major changes in VTPZ up to and including July 27, 1984.

VTPZ: Appendix I

This appendix describes the main changes made to VTPZ between the version discussed in the manual (May 3, 1984) and the version valid July 27, 1984.

1.) A cloud flag has been added (ICLOUD). A non-zero value (cloud contamination) precipitates a test involving band 12 and 8 TBBs, in which a failure condemns the retrieval to use only for image-generating purposes (total precipitable water vapor and (if desired) total totals index).

2.) A section of code which deals simultaneously with low level stratus cloud and temperature inversion situations has been inserted before the call to subroutine QVTWR. This section of code finds cloud pressure (MS), determines cloud top T and W values, interpolates between cloud top and level IS for the guess T and W profiles and, lastly, tests for low level stratus and temperature inversion situations.

3.) Even if a T/W retrieval fails for a given retrieval box, the VAS data for the box can still be used to create total precipitable water vapor and (if desired) total-totals image(s) (unlike the May 3, 1984 version of VTPZ).

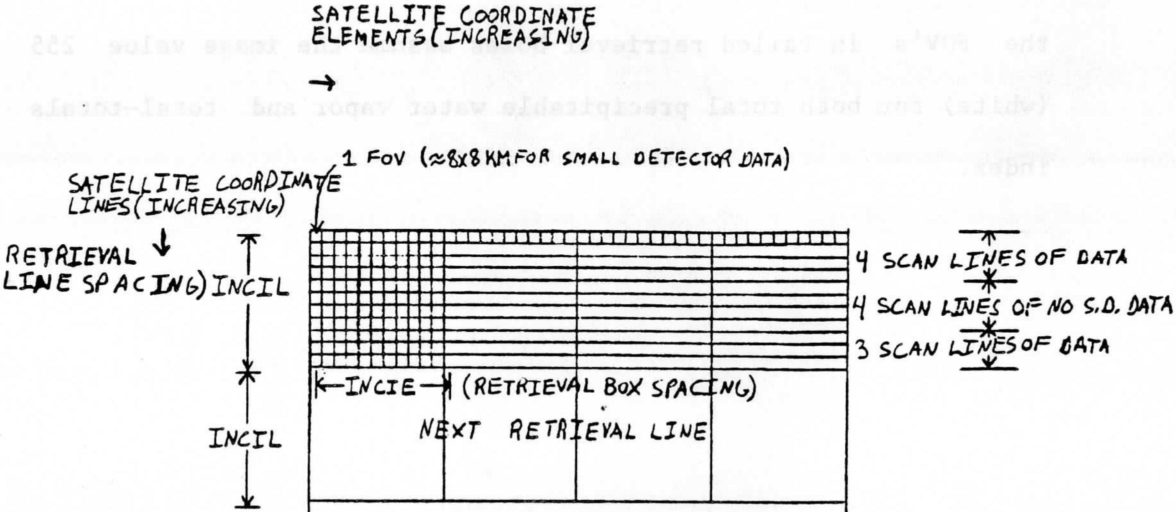
4.) Variable GAMRET is NOT changed between the first and second retrievals (between first and second call to subroutine QVTWR) if the cloud flag (ICLOUD) has been set to 1 in subroutine FILTER.

5.) If no total precipitable water vapor value and (if desired) total-totals index are calculated for a given FOV due to an earlier retrieval failure or severe cloud contamination, a value dependant on the band 8 TBB for that FOV (and limited to the range 155.-255., inclusive) is inserted into the image at that FOV location (The May 3, 1984 version of VTPZ simply let all the FOV's in failed retrieval boxes assume the image value 255 (white) for both total precipitable water vapor and total-totals index.

DIAGRAM 1

TYPICAL RETRIEVAL AREAS

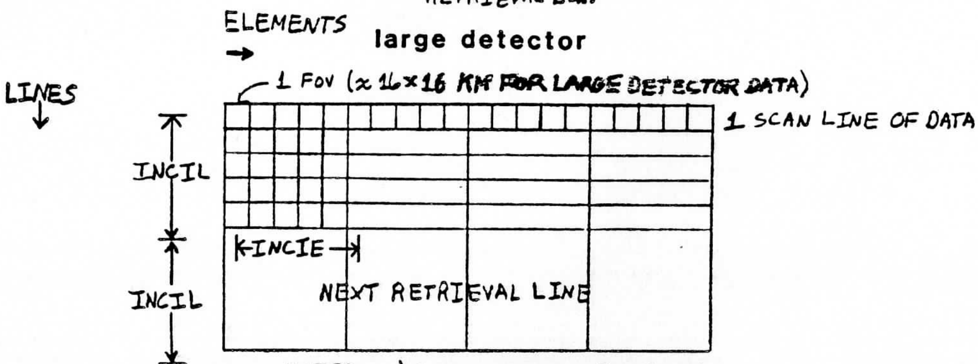
small detector



NOTES: 1.) DIMENSION OF RETRIEVAL BOX = 11 X 11 FOVs

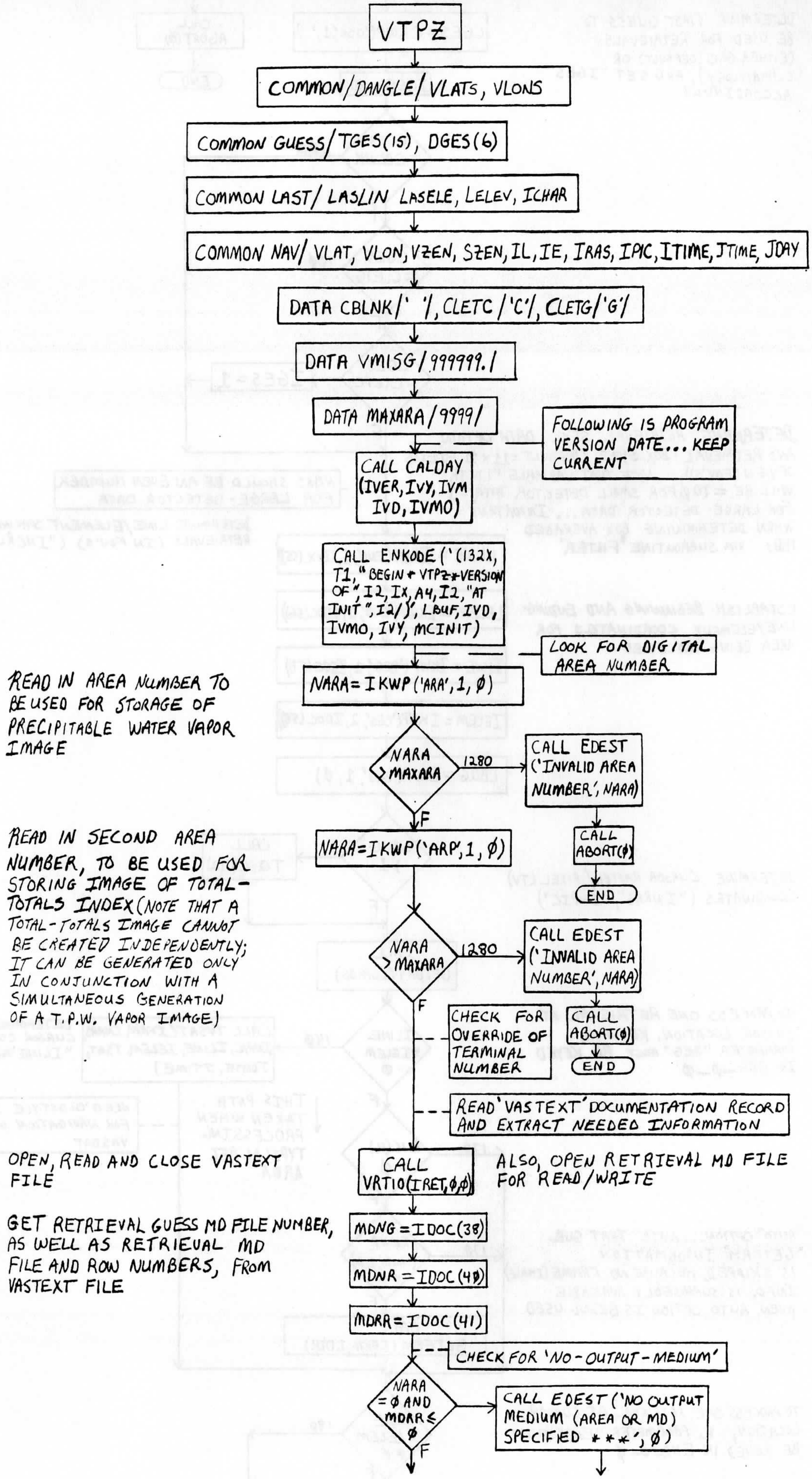
2.) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 121 FOVs. FOR THE FIRST RETRIEVAL LINE, THERE ARE 77 FOVs WHICH HAVE SMALL DETECTOR DATA (4x11 + 3x11), MEANING 44 FOVs HAVE NO SMALL DETECTOR DATA. FOR THE SECOND RETRIEVAL LINE, THE NUMBERS CHANGE TO 55 AND 66, RESPECTIVELY. IT IS POSSIBLE TO HAVE ANYWHERE FROM 4 TO 7 SCAN LINES OF VAS DATA WITHIN A GIVEN 11 X 11 RETRIEVAL BOX.

large detector



NOTES: 1.) DIMENSION OF RETRIEVAL BOX = 5 X 5 FOVs

2.) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 25 FOVs, ALL OF WHICH CONTAIN VAS TBB DATA.


VTPZ
COMMON/DANGLE/VLATS, VLONS
COMMON GUESS/TGES(15), DGES(6)
COMMON LAST/ LASLIN LASELE, LELEV, ICHAR
COMMON NAV/ VLAT, VLON, VZEN, SZEN, IL, IE, IRAS, IPIC, ITIME, JTIME, JDAY
DATA CBLNK/' '/, CLETC/'C'/, CLETG/'G'/
DATA VMISG/999999./
DATA MAXARA/9999/
FOLLOWING IS PROGRAM VERSION DATE... KEEP CURRENT
CALL CALDAY (IVER, IVY, IVM IVD, IVMO)
CALL ENKODE ('(132X, T1, "BEGIN * VTPZ * VERSION OF", I2, IX, A4, I2, "AT INIT", I2/)', LBUF, IVD, IVMO, IVY, MCINIT)
LOOK FOR DIGITAL AREA NUMBER
READ IN AREA NUMBER TO BE USED FOR STORAGE OF PRECIPITABLE WATER VAPOR IMAGE
NARA = IKWP('ARA', 1, Ø)
NARA > MAXARA
1280
CALL EDEST ('INVALID AREA NUMBER', NARA)
CALL ABORT(Ø)
END
F
READ IN SECOND AREA NUMBER, TO BE USED FOR STORING IMAGE OF TOTAL-TOTALS INDEX (NOTE THAT A TOTAL-TOTALS IMAGE CANNOT BE CREATED INDEPENDENTLY; IT CAN BE GENERATED ONLY IN CONJUNCTION WITH A SIMULTANEOUS GENERATION OF A T.P.W. VAPOR IMAGE)
NARA = IKWP('ARP', 1, Ø)
NARA > MAXARA
1280
CALL EDEST ('INVALID AREA NUMBER', NARA)
CALL ABORT(Ø)
END
F
CHECK FOR OVERRIDE OF TERMINAL NUMBER
READ 'VASTEXT' DOCUMENTATION RECORD AND EXTRACT NEEDED INFORMATION
OPEN, READ AND CLOSE VASTEXT FILE
CALL VRTIO(IRET, Ø, Ø)
ALSO, OPEN RETRIEVAL MD FILE FOR READ/WRITE
GET RETRIEVAL GUESS MD FILE NUMBER, AS WELL AS RETRIEVAL MD FILE AND ROW NUMBERS, FROM VASTEXT FILE
MDNG = IDOC(38)
MDNR = IDOC(4Ø)
MDRR = IDOC(41)
CHECK FOR 'NO-OUTPUT-MEDIUM'
NARA = Ø AND MDRR ≤ Ø
CALL EDEST ('NO OUTPUT MEDIUM (AREA OR MD) SPECIFIED ***', Ø)
F

DETERMINE FIRST GUESS TO BE USED FOR RETRIEVALS (EITHER GRID (DEFAULT) OR CLIMATOLOGY), AND SET "IGES" ACCORDINGLY

DETERMINE RETRIEVAL SFC. DATA OPTION, AND RETRIEVAL BOX SIZE (DEFAULT = 11 x 11 FIELDS OF VIEW (FOV'S))... NOTE THAT VARIABLE "LDETH" WILL BE = TO Ø FOR SMALL DETECTOR DATA, = 1 FOR LARGE DETECTOR DATA... IMPORTANT WHEN DETERMINING BOX AVERAGED TBBs VIA SUBROUTINE "FILTER"

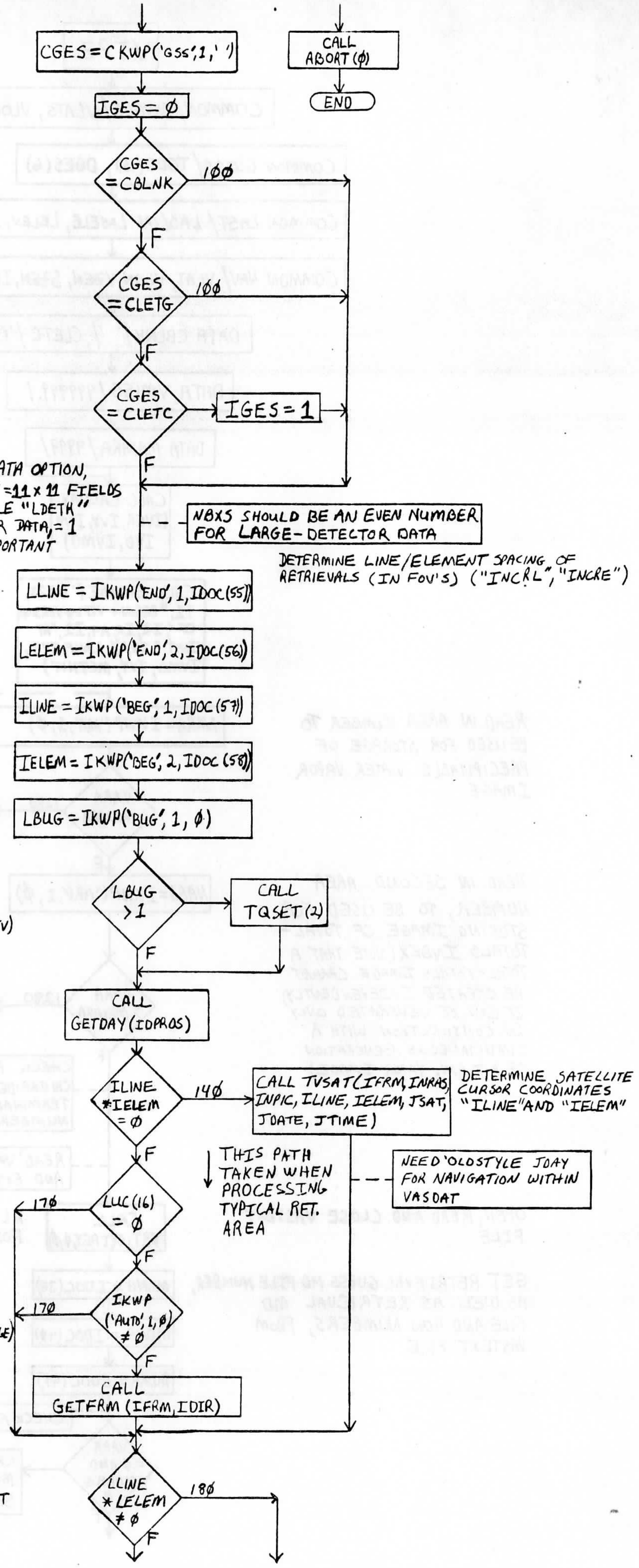
ESTABLISH BEGINNING AND ENDING LINE/ELEMENT COORDINATES FOR AREA BEING PROCESSED

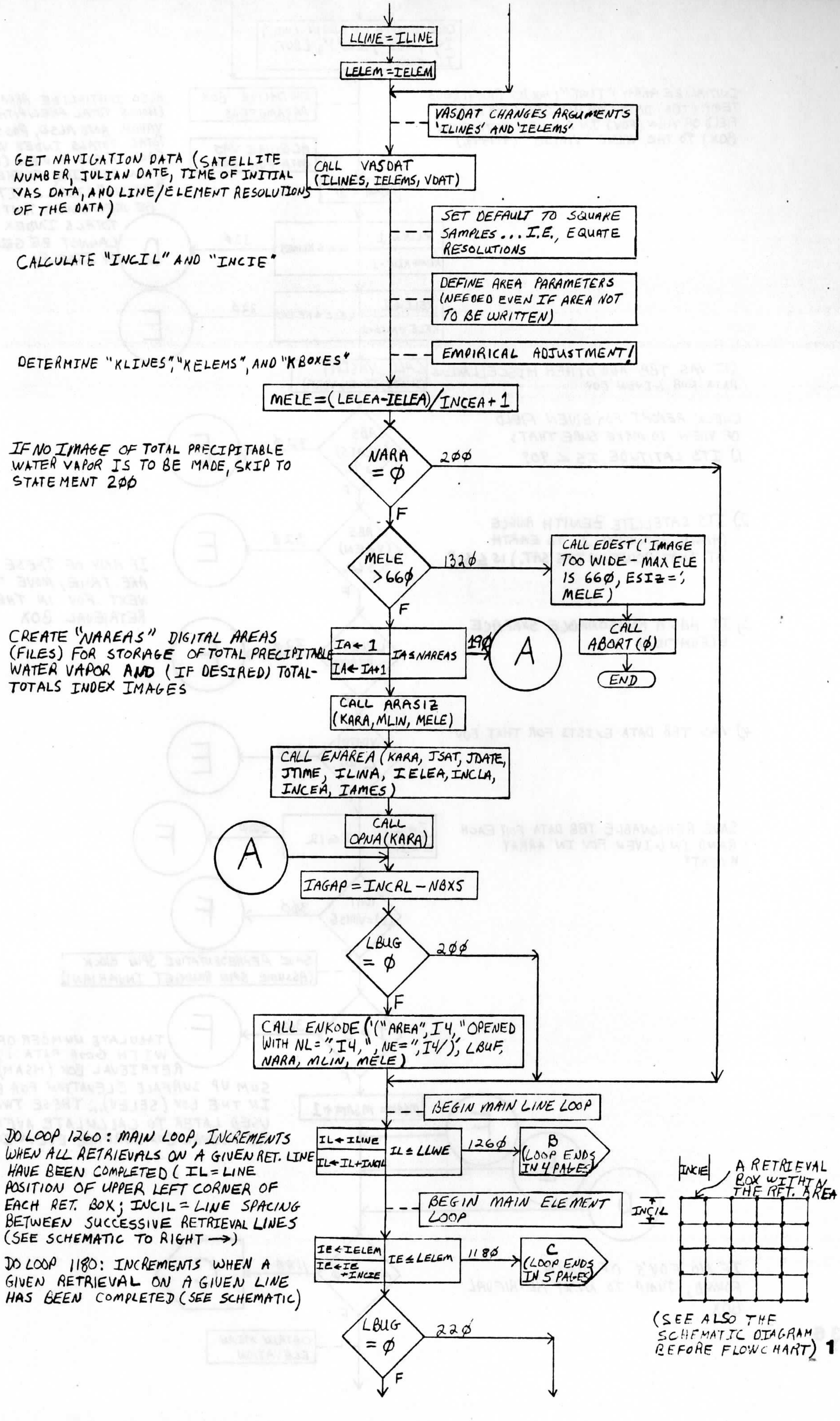
DETERMINE CURSOR RASTER/PIXEL (TV) COORDINATES ("INRAS", "INPIC")

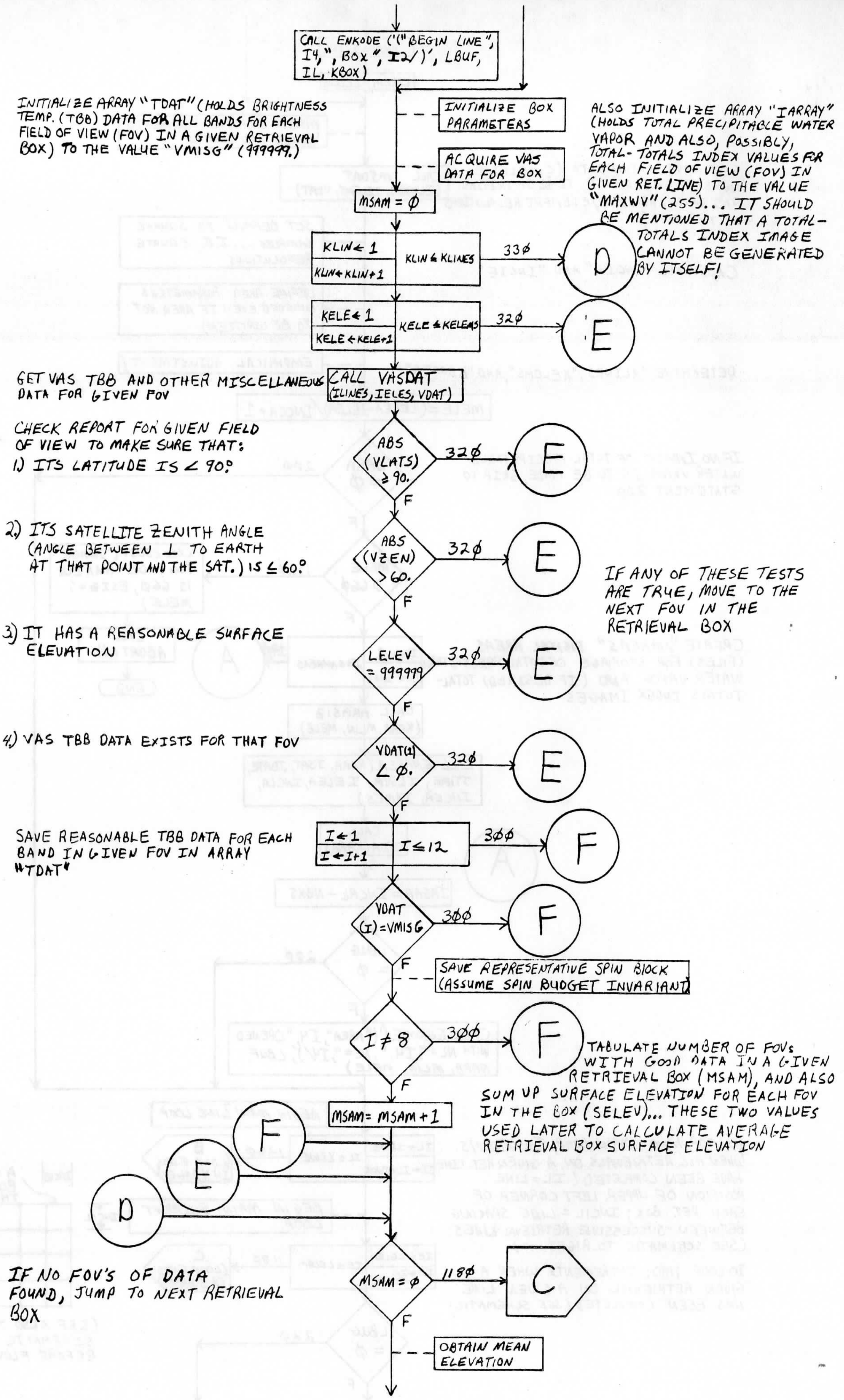
TO PROCESS ONE RETRIEVAL AT CURSOR LOCATION, KEYWORD PARAMETER "BEG" MUST BE KEYED IN BEG_Ø_Ø

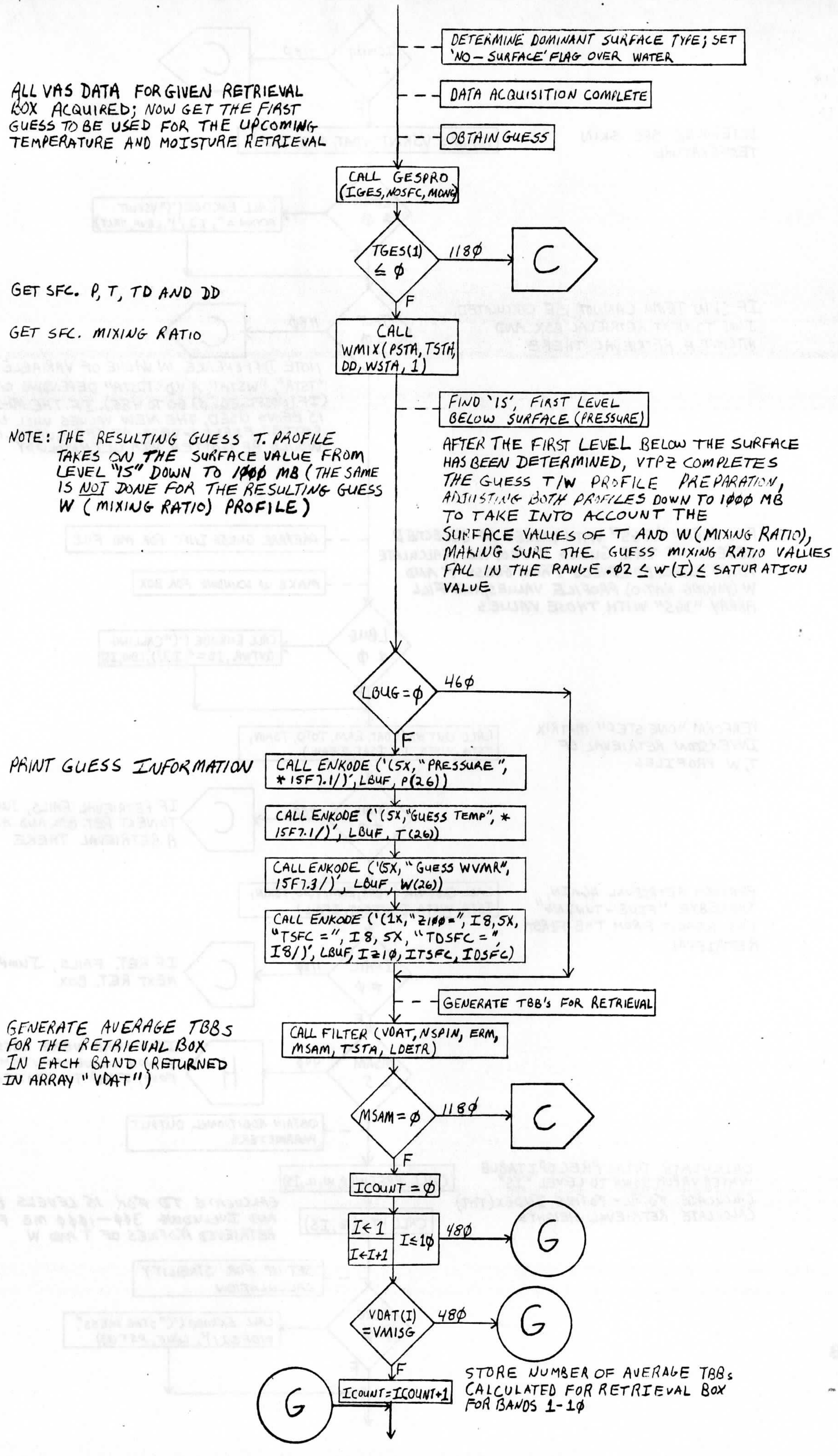
"AUTO" OPTION... NOTE THAT SUB. "GETFRM" INFORMATION IS SKIPPED, BECAUSE NO FRAME (IMAGE) INFO. IS SUPPOSEDLY AVAILABLE WHEN AUTO OPTION IS BEING USED

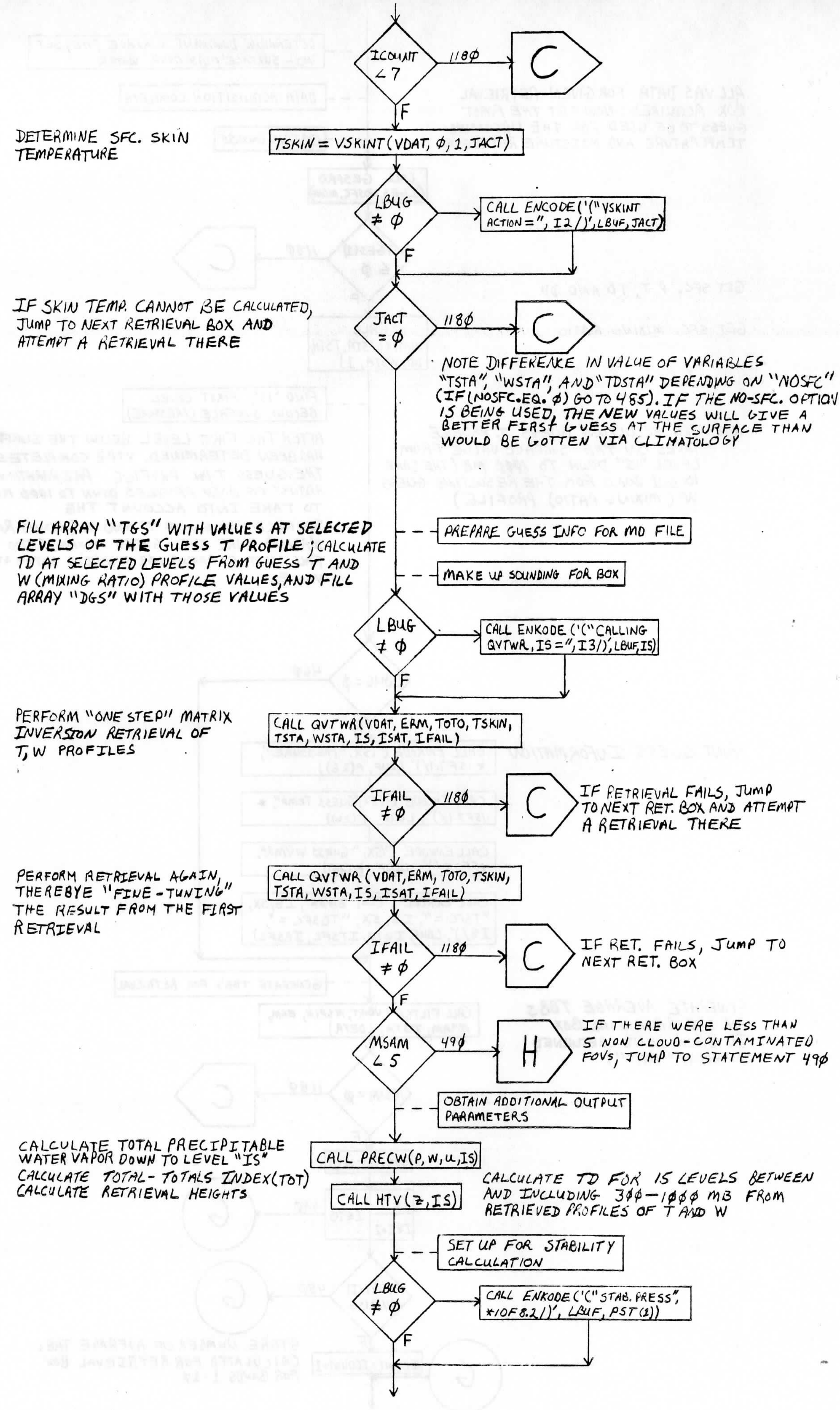
TO PROCESS ONE RETRIEVAL AT CURSOR LOCATION, K. PARAMETER "END" MUST BE KEYED IN END_Ø_Ø

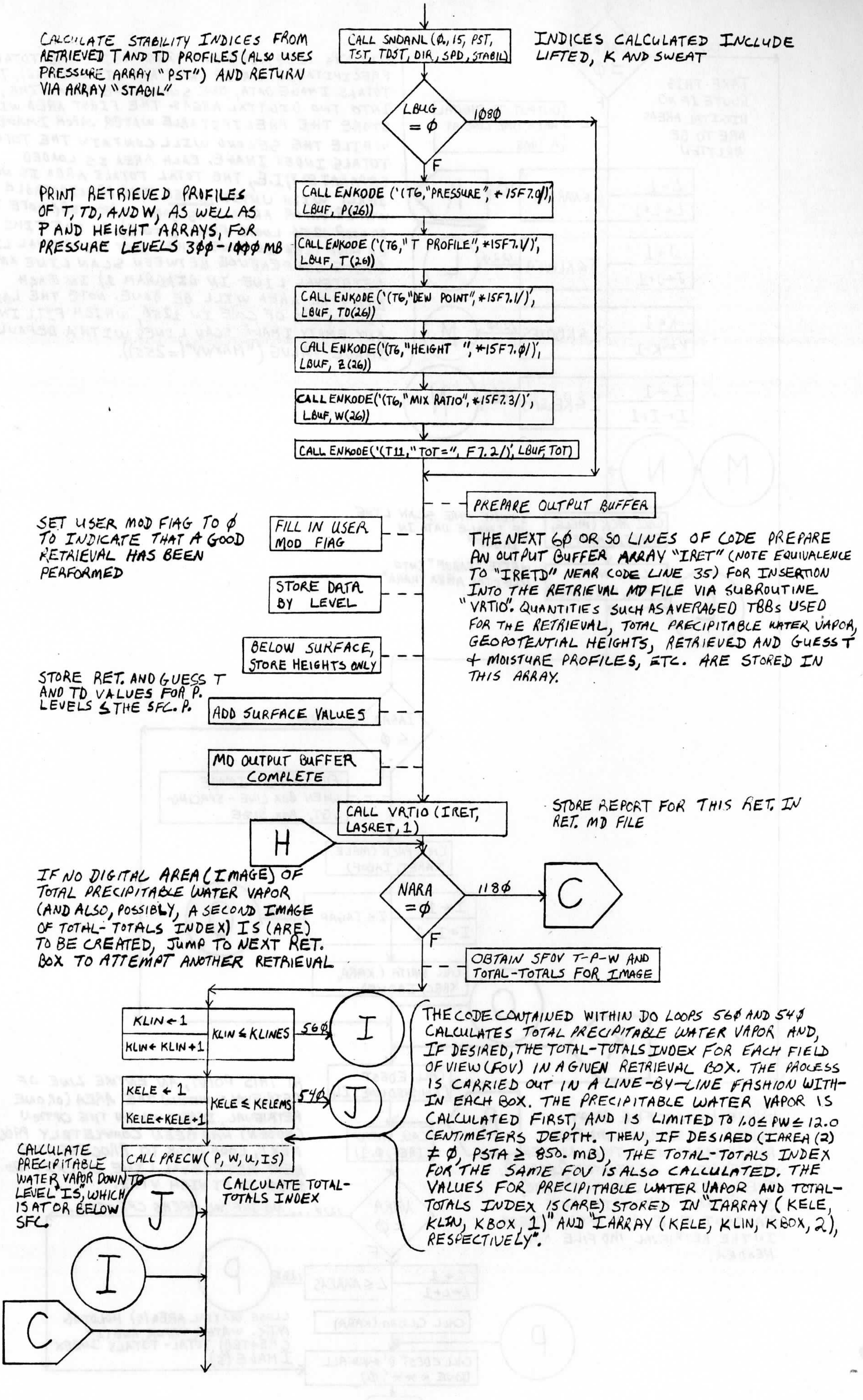


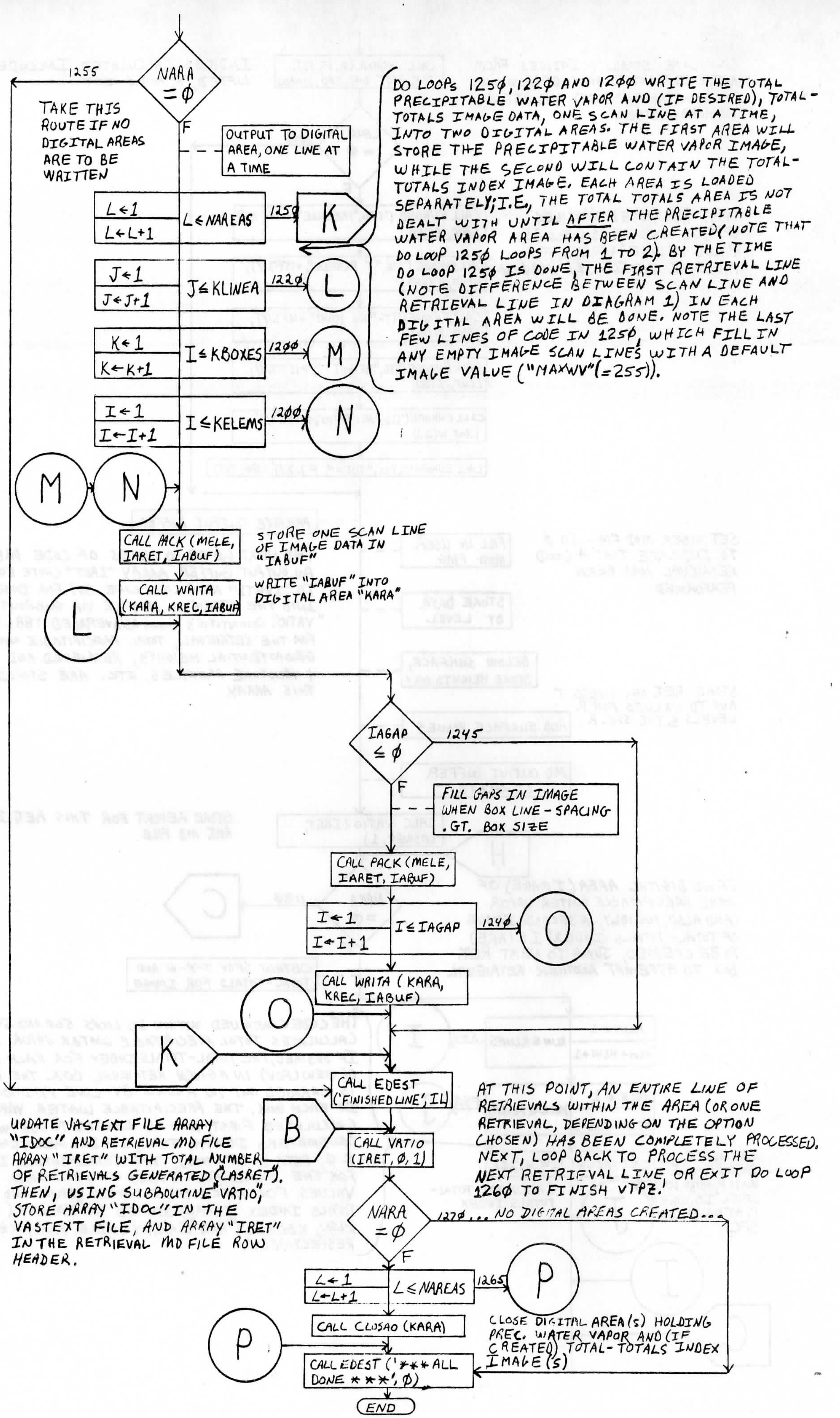

LLINE = ILINE
LELEM = IELEM
VASDAT CHANGES ARGUMENTS 'ILINES' AND 'IELEMS'
GET NAVIGATION DATA (SATELLITE NUMBER, JULIAN DATE, TIME OF INITIAL VAS DATA, AND LINE/ELEMENT RESOLUTIONS OF THE DATA)
CALL VASDAT (ILINES, IELEMS, VDAT)
SET DEFAULT TO SQUARE SAMPLES ... I.E., EQUATE RESOLUTIONS
CALCULATE "INCIL" AND "INCIE"
DEFINE AREA PARAMETERS (NEEDED EVEN IF AREA NOT TO BE WRITTEN)
EMPIRICAL ADJUSTMENT!
DETERMINE "KLINES", "KELEMS", AND "KBOXES"
MELE=(LELEA-IELEA)/INCEA+1
IF NO IMAGE OF TOTAL PRECIPITABLE WATER VAPOR IS TO BE MADE, SKIP TO STATEMENT 200
NARA = Ø
200
F
MELE >66Ø
132Ø
CALL EDEST('IMAGE TOO WIDE - MAX ELE IS 66Ø, ESIZ=', MELE)
F
CALL ABORT(Ø)
END
CREATE "NAREAS" DIGITAL AREAS (FILES) FOR STORAGE OF TOTAL PRECIPITABLE WATER VAPOR AND (IF DESIRED) TOTAL-TOTALS INDEX IMAGES
IA←1
IA≤NAREAS
IA←IA+1
190
A
CALL ARASIZ (KARA, MLIN, MELE)
CALL ENAREA (KARA, JSAT, JDATE, JTIME, ILINA, IELEA, INCLA, INCEA, IAMES)
CALL OPNA(KARA)
A
IAGAP=INCRL-NBXS
LBUG = Ø
200
F
CALL ENKODE ('("AREA", I4, "OPENED WITH NL=", I4, ", NE=", I4/)', LBUF, NARA, MLIN, MELE)
BEGIN MAIN LINE LOOP
DO LOOP 1260: MAIN LOOP, INCREMENTS WHEN ALL RETRIEVALS ON A GIVEN RET. LINE HAVE BEEN COMPLETED (IL=LINE POSITION OF UPPER LEFT CORNER OF EACH RET. BOX; INCIL=LINE SPACING BETWEEN SUCCESSIVE RETRIEVAL LINES (SEE SCHEMATIC TO RIGHT →)
IL←ILINE
IL≤LLINE
IL←IL+INCIL
126Ø
B (LOOP ENDS IN 4 PAGES)
BEGIN MAIN ELEMENT LOOP
DO LOOP 1180: INCREMENTS WHEN A GIVEN RETRIEVAL ON A GIVEN LINE HAS BEEN COMPLETED (SEE SCHEMATIC)
IE←IELEM
IE≤LELEM
IE←IE+INCIE
118Ø
C (LOOP ENDS IN 5 PAGES)
LBUG = Ø
22Ø
F
INCIE
A RETRIEVAL BOX WITHIN THE RET. AREA
INCIL
(SEE ALSO THE SCHEMATIC DIAGRAM BEFORE FLOWCHART)


CALL ENKODE ('("BEGIN LINE", I4, ", BOX ", I2/)', LBUF, IL, KBOX)
INITIALIZE ARRAY "TDAT" (HOLDS BRIGHTNESS TEMP. (TBB) DATA FOR ALL BANDS FOR EACH FIELD OF VIEW (FOV) IN A GIVEN RETRIEVAL BOX) TO THE VALUE "VMISG" (999999.)
INITIALIZE BOX PARAMETERS
ALSO INITIALIZE ARRAY "IARRAY" (HOLDS TOTAL PRECIPITABLE WATER VAPOR AND ALSO, POSSIBLY, TOTAL-TOTALS INDEX VALUES FOR EACH FIELD OF VIEW (FOV) IN GIVEN RET. LINE) TO THE VALUE "MAXWV" (255)... IT SHOULD BE MENTIONED THAT A TOTAL-TOTALS INDEX IMAGE CANNOT BE GENERATED BY ITSELF!
ACQUIRE VAS DATA FOR BOX
MSAM = Ø
KLIN ← 1
KLIN ← KLIN + 1
KLIN ≤ KLINES
33Ø
D
KELE ← 1
KELE ← KELE + 1
KELE ≤ KELEMS
32Ø
E
GET VAS TBB AND OTHER MISCELLANEOUS DATA FOR GIVEN FOV
CALL VASDAT (ILINES, IELES, VDAT)
CHECK REPORT FOR GIVEN FIELD OF VIEW TO MAKE SURE THAT:
1) ITS LATITUDE IS < 90°
ABS (VLATS) ≥ 90.
32Ø
E
F
2) ITS SATELLITE ZENITH ANGLE (ANGLE BETWEEN ⊥ TO EARTH AT THAT POINT AND THE SAT.) IS ≤ 60°
ABS (VZEN) > 60.
32Ø
E
F
IF ANY OF THESE TESTS ARE TRUE, MOVE TO THE NEXT FOV IN THE RETRIEVAL BOX
3) IT HAS A REASONABLE SURFACE ELEVATION
LELEV = 999999
32Ø
E
F
4) VAS TBB DATA EXISTS FOR THAT FOV
VDAT(1) < Ø.
32Ø
E
F
SAVE REASONABLE TBB DATA FOR EACH BAND IN GIVEN FOV IN ARRAY "TDAT"
I ← 1
I ← I + 1
I ≤ 12
3ØØ
F
VDAT (I) = VMISG
3ØØ
F
F
SAVE REPRESENTATIVE SPIN BLOCK (ASSUME SPIN BUDGET INVARIANT)
I ≠ 8
3ØØ
F
TABULATE NUMBER OF FOVs WITH GOOD DATA IN A GIVEN RETRIEVAL BOX (MSAM), AND ALSO SUM UP SURFACE ELEVATION FOR EACH FOV IN THE BOX (SELEV)... THESE TWO VALUES USED LATER TO CALCULATE AVERAGE RETRIEVAL BOX SURFACE ELEVATION
F
MSAM = MSAM + 1
F
E
D
IF NO FOV'S OF DATA FOUND, JUMP TO NEXT RETRIEVAL BOX
MSAM = Ø
118Ø
C
F
OBTAIN MEAN ELEVATION


DETERMINE DOMINANT SURFACE TYPE; SET 'NO-SURFACE' FLAG OVER WATER
ALL VAS DATA FOR GIVEN RETRIEVAL BOX ACQUIRED; NOW GET THE FIRST GUESS TO BE USED FOR THE UPCOMING TEMPERATURE AND MOISTURE RETRIEVAL
DATA ACQUISITION COMPLETE
OBTAIN GUESS
CALL GESPRO (IGES, NOSFC, MONG)
TGES(1) ≤ Ø
118Ø
C
F
GET SFC. P, T, TD AND DD
GET SFC. MIXING RATIO
CALL WMIX (PSTA, TSTH, DD, WSTA, 1)
FIND 'IS', FIRST LEVEL BELOW SURFACE (PRESSURE)
NOTE: THE RESULTING GUESS T. PROFILE TAKES ON THE SURFACE VALUE FROM LEVEL "IS" DOWN TO 1ØØØ MB (THE SAME IS NOT DONE FOR THE RESULTING GUESS W (MIXING RATIO) PROFILE)
AFTER THE FIRST LEVEL BELOW THE SURFACE HAS BEEN DETERMINED, VTPZ COMPLETES THE GUESS T/W PROFILE PREPARATION, ADJUSTING BOTH PROFILES DOWN TO 1ØØØ MB TO TAKE INTO ACCOUNT THE SURFACE VALUES OF T AND W (MIXING RATIO), MAKING SURE THE GUESS MIXING RATIO VALUES FALL IN THE RANGE .Ø2 ≤ W(I) ≤ SATURATION VALUE
LBUG = Ø
46Ø
F
PRINT GUESS INFORMATION
CALL ENKODE ('(5X, "PRESSURE", * 15F7.1/)', LBUF, P(26))
CALL ENKODE ('(5X, "GUESS TEMP", * 15F7.1/)', LBUF, T(26))
CALL ENKODE ('(5X, "GUESS WVMR", 15F7.3/)', LBUF, W(26))
CALL ENKODE ('(1X, "Z1ØØ=", I8, 5X, "TSFC =", I8, 5X, "TDSFC =", I8/)', LBUF, IZ1Ø, ITSFC, IDSFC)
GENERATE TBB's FOR RETRIEVAL
GENERATE AVERAGE TBBs FOR THE RETRIEVAL BOX IN EACH BAND (RETURNED IN ARRAY "VDAT")
CALL FILTER (VDAT, NSPIN, ERM, MSAM, TSTA, LDETR)
MSAM = Ø
118Ø
C
F
ICOUNT = Ø
I←1
I≤1Ø
I←I+1
48Ø
G
VDAT(I) = VMISG
48Ø
G
F
STORE NUMBER OF AVERAGE TBBs CALCULATED FOR RETRIEVAL BOX FOR BANDS 1-1Ø
G
ICOUNT=ICOUNT+1


ICOUNT < 7
1180
C
F
DETERMINE SFC. SKIN TEMPERATURE
TSKIN = VSKINT(VDAT, Ø, 1, JACT)
LBUG ≠ Ø
CALL ENCODE('("VSKINT ACTION=", I2/)', LBUF, JACT)
F
IF SKIN TEMP. CANNOT BE CALCULATED, JUMP TO NEXT RETRIEVAL BOX AND ATTEMPT A RETRIEVAL THERE
JACT = Ø
1180
C
NOTE DIFFERENCE IN VALUE OF VARIABLES "TSTA", "WSTA", AND "TDSTA" DEPENDING ON "NOSFC" (IF(NOSFC.EQ.Ø) GO TO 485). IF THE NO-SFC. OPTION IS BEING USED, THE NEW VALUES WILL GIVE A BETTER FIRST GUESS AT THE SURFACE THAN WOULD BE GOTTEN VIA CLIMATOLOGY
FILL ARRAY "TGS" WITH VALUES AT SELECTED LEVELS OF THE GUESS T PROFILE; CALCULATE TD AT SELECTED LEVELS FROM GUESS T AND W (MIXING RATIO) PROFILE VALUES, AND FILL ARRAY "DGS" WITH THOSE VALUES
PREPARE GUESS INFO FOR MD FILE
MAKE UP SOUNDING FOR BOX
LBUG ≠ Ø
CALL ENKODE('("CALLING QVTWR, IS=", I3/)', LBUF, IS)
F
PERFORM "ONE STEP" MATRIX INVERSION RETRIEVAL OF T, W PROFILES
CALL QVTWA(VDAT, ERM, TOTO, TSKIN, TSTA, WSTA, IS, ISAT, IFAIL)
IFAIL ≠ Ø
1180
C
IF RETRIEVAL FAILS, JUMP TO NEXT RET. BOX AND ATTEMPT A RETRIEVAL THERE
PERFORM RETRIEVAL AGAIN, THEREBYE "FINE-TUNING" THE RESULT FROM THE FIRST RETRIEVAL
CALL QVTWR (VDAT, ERM, TOTO, TSKIN, TSTA, WSTA, IS, ISAT, IFAIL)
IFAIL ≠ Ø
1180
C
IF RET. FAILS, JUMP TO NEXT RET. BOX
F
MSAM < 5
490
H
IF THERE WERE LESS THAN 5 NON CLOUD-CONTAMINATED FOVS, JUMP TO STATEMENT 490
OBTAIN ADDITIONAL OUTPUT PARAMETERS
CALCULATE TOTAL PRECIPITABLE WATER VAPOR DOWN TO LEVEL "IS"
CALCULATE TOTAL-TOTALS INDEX (TOT)
CALCULATE RETRIEVAL HEIGHTS
CALL PRECW(P, W, U, IS)
CALL HTV(Z, IS)
CALCULATE TD FOR 15 LEVELS BETWEEN AND INCLUDING 300–1000 MB FROM RETRIEVED PROFILES OF T AND W
SET UP FOR STABILITY CALCULATION
LBUG ≠ Ø
CALL ENKODE('C("STAB. PRESS", *10F8.2/)', LBUF, PST(1))
F


CALCULATE STABILITY INDICES FROM RETRIEVED T AND TD PROFILES (ALSO USES PRESSURE ARRAY "PST") AND RETURN VIA ARRAY "STABIL"
CALL SNDANL (Ø, 15, PST, TST, TDST, DIR, SPD, STABIL)
INDICES CALCULATED INCLUDE LIFTED, K AND SWEAT
LBUG = Ø
1Ø8Ø
F
PRINT RETRIEVED PROFILES OF T, TD, AND W, AS WELL AS P AND HEIGHT ARRAYS, FOR PRESSURE LEVELS 3ØØ-1ØØØ MB
CALL ENKODE ('(T6,"PRESSURE", *15F7.Ø/)', LBUF, P(26))
CALL ENKODE ('(T6," T PROFILE", *15F7.1/)', LBUF, T(26))
CALL ENKODE ('(T6,"DEW POINT", *15F7.1/)', LBUF, TD(26))
CALL ENKODE ('(T6,"HEIGHT ", *15F7.Ø/)', LBUF, Z(26))
CALL ENKODE('(T6,"MIX RATIO", *15F7.3/)', LBUF, W(26))
CALL ENKODE('(T11, "TOT=", F7.2/)', LBUF, TOT)
PREPARE OUTPUT BUFFER
SET USER MOD FLAG TO Ø TO INDICATE THAT A GOOD RETRIEVAL HAS BEEN PERFORMED
FILL IN USER MOD FLAG
THE NEXT 6Ø OR SO LINES OF CODE PREPARE AN OUTPUT BUFFER ARRAY "IRET" (NOTE EQUIVALENCE TO "IRETD" NEAR CODE LINE 35) FOR INSERTION INTO THE RETRIEVAL MD FILE VIA SUBROUTINE "VRTIO". QUANTITIES SUCH AS AVERAGED TBBs USED FOR THE RETRIEVAL, TOTAL PRECIPITABLE WATER VAPOR, GEOPOTENTIAL HEIGHTS, RETRIEVED AND GUESS T + MOISTURE PROFILES, ETC. ARE STORED IN THIS ARRAY.
STORE DATA BY LEVEL
BELOW SURFACE, STORE HEIGHTS ONLY
STORE RET. AND GUESS T AND TD VALUES FOR P. LEVELS ≤ THE SFC. P.
ADD SURFACE VALUES
MD OUTPUT BUFFER COMPLETE
CALL VRTIO (IRET, LASRET, 1)
STORE REPORT FOR THIS RET. IN RET. MD FILE
H
IF NO DIGITAL AREA (IMAGE) OF TOTAL PRECIPITABLE WATER VAPOR (AND ALSO, POSSIBLY, A SECOND IMAGE OF TOTAL-TOTALS INDEX) IS (ARE) TO BE CREATED, JUMP TO NEXT RET. BOX TO ATTEMPT ANOTHER RETRIEVAL
NARA = Ø
118Ø
C
F
OBTAIN SFOV T-P-W AND TOTAL-TOTALS FOR IMAGE
KLIN ← 1
KLIN ← KLIN + 1
KLIN ≤ KLINES
56Ø
I
THE CODE CONTAINED WITHIN DO LOOPS 56Ø AND 54Ø CALCULATES TOTAL PRECIPITABLE WATER VAPOR AND, IF DESIRED, THE TOTAL-TOTALS INDEX FOR EACH FIELD OF VIEW (FOV) IN A GIVEN RETRIEVAL BOX. THE PROCESS IS CARRIED OUT IN A LINE-BY-LINE FASHION WITHIN EACH BOX. THE PRECIPITABLE WATER VAPOR IS CALCULATED FIRST, AND IS LIMITED TO 1.0 ≤ PW ≤ 12.0 CENTIMETERS DEPTH. THEN, IF DESIRED (IAREA (2) ≠ Ø, PSTA ≥ 850. MB), THE TOTAL-TOTALS INDEX FOR THE SAME FOV IS ALSO CALCULATED. THE VALUES FOR PRECIPITABLE WATER VAPOR AND TOTAL-TOTALS INDEX IS (ARE) STORED IN "IARRAY (KELE, KLIN, KBOX, 1)" AND "IARRAY (KELE, KLIN, KBOX, 2), RESPECTIVELY".
KELE ← 1
KELE ← KELE + 1
KELE ≤ KELEMS
54Ø
J
CALCULATE PRECIPITABLE WATER VAPOR DOWN TO LEVEL "IS", WHICH IS AT OR BELOW SFC.
CALL PRECW(P, W, U, IS)
CALCULATE TOTAL-TOTALS INDEX
J
I
C


NARA = ∅
1255
TAKE THIS ROUTE IF NO DIGITAL AREAS ARE TO BE WRITTEN
F
OUTPUT TO DIGITAL AREA, ONE LINE AT A TIME
DO LOOPS 125∅, 122∅ AND 12∅∅ WRITE THE TOTAL PRECIPITABLE WATER VAPOR AND (IF DESIRED), TOTAL-TOTALS IMAGE DATA, ONE SCAN LINE AT A TIME, INTO TWO DIGITAL AREAS. THE FIRST AREA WILL STORE THE PRECIPITABLE WATER VAPOR IMAGE, WHILE THE SECOND WILL CONTAIN THE TOTAL-TOTALS INDEX IMAGE. EACH AREA IS LOADED SEPARATELY; I.E, THE TOTAL TOTALS AREA IS NOT DEALT WITH UNTIL AFTER THE PRECIPITABLE WATER VAPOR AREA HAS BEEN CREATED (NOTE THAT DO LOOP 125∅ LOOPS FROM 1 TO 2). BY THE TIME DO LOOP 125∅ IS DONE, THE FIRST RETRIEVAL LINE (NOTE DIFFERENCE BETWEEN SCAN LINE AND RETRIEVAL LINE IN DIAGRAM 1) IN EACH DIGITAL AREA WILL BE DONE. NOTE THE LAST FEW LINES OF CODE IN 125∅, WHICH FILL IN ANY EMPTY IMAGE SCAN LINES WITH A DEFAULT IMAGE VALUE ("MAXWV"(=255)).
L←1
L←L+1
L≤NAREAS
125∅
K
J←1
J←J+1
J≤KLINEA
122∅
L
K←1
K←K+1
I≤KBOXES
12∅∅
M
I←1
I←I+1
I≤KELEMS
12∅∅
N
M
N
CALL PACK (MELE, IARET, IABUF)
STORE ONE SCAN LINE OF IMAGE DATA IN "IABUF"
CALL WRITA (KARA, KREC, IABUF)
WRITE "IABUF" INTO DIGITAL AREA "KARA"
L
IAGAP ≤ ∅
1245
F
FILL GAPS IN IMAGE WHEN BOX LINE-SPACING .GT. BOX SIZE
CALL PACK (MELE, IARET, IABUF)
I←1
I←I+1
I≤IAGAP
124∅
O
CALL WRITA (KARA, KREC, IABUF)
O
K
CALL EDEST ('FINISHED LINE', IL)
AT THIS POINT, AN ENTIRE LINE OF RETRIEVALS WITHIN THE AREA (OR ONE RETRIEVAL, DEPENDING ON THE OPTION CHOSEN) HAS BEEN COMPLETELY PROCESSED. NEXT, LOOP BACK TO PROCESS THE NEXT RETRIEVAL LINE, OR EXIT DO LOOP 126∅ TO FINISH VTPZ.
B
UPDATE VASTEXT FILE ARRAY "IDOC" AND RETRIEVAL MD FILE ARRAY "IRET" WITH TOTAL NUMBER OF RETRIEVALS GENERATED (LASRET). THEN, USING SUBROUTINE "VRTIO", STORE ARRAY "IDOC" IN THE VASTEXT FILE, AND ARRAY "IRET" IN THE RETRIEVAL MD FILE ROW HEADER.
CALL VRTIO (IRET, ∅, 1)
NARA = ∅
127∅ ... NO DIGITAL AREAS CREATED ...
F
L←1
L←L+1
L≤NAREAS
1265
P
CALL CLOSAO (KARA)
CLOSE DIGITAL AREA(S) HOLDING PREC. WATER VAPOR AND (IF CREATED) TOTAL-TOTALS INDEX IMAGE(S)
P
CALL EDEST ('***ALL DONE***', ∅)
END

Subroutines used by VTPZ:

1) CALDAY-I	53) READD	105) VASGES
2) ENKODE-I	54) EDEST	106) SURGES
3) VRTIO-I	55) MOVW	107) GETSFV
4) TQSET-I	56) CLEANW	108) IGNAME
5) GETDAY-I	57) TQ	109) PROFIX
6) GETFRM-I	58) OUTINT	110) VASTAU
7) TVSAT-I	59) II	111) PRETAV
8) VASDAT-I	60) SATEAR	112) PREATV
9) ARASIZ-I	61) SATPOS	113) ULMR
10) ENAREA-I	62) ANGGET	114) CO2TAV
11) OPNA-I	63) SOLARP	115) H2OTAV
12) GESPRO-I	64) ANGLES	116) CONTAV
13) WMIX-I	65) SATPOS	117) O3TAV
14) FILTER-I	66) VRTOPO	118) GAMTAV
15) QVTWR-I	67) NVINIT	119) VASRTE
16) PRECW-I	68) GETNAV	120) SOLVEX
17) HTV-I	69) EPOCH	121) SYMVRT
18) SNDANL-I	70) GETGAM	122) EMES
19) PACK-I	71) VASNAV	123) WRTRK
20) WRITA-I	72) DDEST	124) CLOSA
21) EDEST-I	73) PLNKIV	
22) CLOSAO-I	74) LWOPEN	
23) ABORT-I	75) MOVWC	
24) ENCODX	76) LWGET	
25) CONTNT	77) READOF	
26) LTQ	78) RBYTSX	
27) MOVB	79) ZEROS	
28) CLEANA	80) RDTRK	
29) TQ	81) MOVC	
30) PRLINX	82) READDL	
31) PRCLOS	83) LWCLOS	
32) LOCK	84) LWMOP	
33) PRWR	85) LWPO	
34) PRRD	86) LWSO	
35) UNLOCK	87) JMBWTF	
36) PRCL	88) LWNEWF	
37) POST	89) GETTIM	
38) PROPEN	90) WRITDU	
39) BLKA	91) RAOBIN	
40) PRPRPR	92) CSRAOB	
41) ZECONV	93) CSRAOR	
42) DOPEN	94) SGRAOB	
43) DREAD	95) CSRAOS	
44) ENCODE	96) CSRAOM	
45) MOVCW	97) CSRAOZ	
46) DWRITE	98) CSRAOP	
47) DCLOSE	99) CSRAOI	
48) TRMNL	100) MNRAOB	
49) TQMES	101) INTPTW	
50) ITOC	102) EXTEMP	
51) STC	103) STDATM	
52) SDEST	104) CLMGES	

PLVA

PLVA is used to plot either VAS retrieval or surface MD file data on the video screen over the satellite image. Several different types of data can be displayed. For instance, the user can display a certain parameter at a given level, differences between retrieval and first guess for either temperature or dew point at a given level, non-level parameters such as brightness temperature or relative humidity, layer thickness, or winds.

Initially, data such as debug option, graphics and image frame numbers (IWRM and IFRM,respectively), the VASTEXT file, etc. are read in. Then, after variables IRAS and IPIC have been determined, the satellite coordinates (line, element) of the cursor are set via subroutine TVSAT, the navigation is initialized, and the coordinates of the image/sounder area (LLNW,LLSE) are filled from the VASTEXT file via IDOC (25) and (26), respectively. These coordinates are then used to determine the N, S, W and E boundaries for the data to be plotted. Note that these latitude/longitude boundaries can also be included in the program keyin via keyword parameters LAT and LON. Now, the MD file and row from which the user wishes data to be taken and plotted is set (MDNO,MDR), with the default values being the retrieval MD file and row numbers taken from the VASTEXT file (IDOC (40) and (41), respectively).

Following this, the MD file is opened for read/write, and the row header for the file is read. The data in the row header includes, among other things, the number of reports which exist in that particular row. Then, a test is conducted (READ TEST

RECORD) to make sure data actually exists in this row. Note also that the maximum possible number of data entries in a given MD file row is stored in variable MMAX (MMAX=MDHD(5)). (The array MDHD was filled via a previous call to function MDINFO). Following this, the keys for the MD file are read, which becomes vital when the actual data is accessed from the file later on.

Next, after the size of the data to be plotted is determined and the plot package is initialized, the first and second positional parameters are read in via the three statements CCHR=CPP(1,'Z '), ILEV=IPP(2,0) * 10 and CLEV=CPP(2,' '). Note the different values which variable ISUB assumes, depending on which data quantity is being plotted. The information loaded into these three variables will determine both WHAT type of data is to be plotted (Z, T, brightness temperature, etc.), as well as HOW the data should be plotted (either as values or differences). The default values (if neither of the first two positional parameters are keyed in) are Z and 500, respectively, meaning a 500 mb height field will be plotted on the video screen if PLVA is keyed in by itself. In addition, variable IL2 stores the second (lower) level in case a thickness plot is desired. Then, the data accessed previously by function MDKEYS is used in DO LOOP 18 to determine which words in any given MD file report will contain such quantities as latitude, longitude, etc. Next, the number of reports to be read (not necessarily plotted) from row MDR of the MD file is determined and placed in variable MREC.

Up to this point, PLVA has functioned quite similarly for all the possible plotting options. However, as PLVA now enters an implicit DO LOOP (which runs approximately from code line 185 (15 CONTINUE) to 320 (IF(MM.LE.MMAX) GO TO 15)), its operation varies, depending on which parameter is being plotted. Therefore, the discussion will henceforth deal separately with different individual options.

Initially, after the first report has been read (IOK=MDGET(MDNO, MDR, MM, IOUT)), there is an option to skip a given report if it has been previously rejected, or plot all reports, regardless of previous rejection (IF(IOUT(NMOD).NE. 0.AND.IALL.EQ.0) GO TO 70). These two steps are done for EVERY option. Then, if the pressure level (ILEV) has been set to some non-zero integer, and the plot is to consist of a non-difference variable (T, Z, etc.), the words of the array IOUT storing the correct level and character to be plotted are calculated in DO LOOPs 45 and 47. Following this, assuming the retrieval report is within the previously-defined plotting area, the report line/element (FLIN,FELE) and then raster/pixel (IRAS,IPIC) coordinates are determined in succession by subroutines SATEAR and SATTV. Finally, the color in which the data will be plotted is established, and the actual data plot itself (note that ISUB=0, and that the value plotted is stored in variable IDAT) is carried out. The subroutine which does the actual plotting is PLTDIG.

As another case, if the user is plotting relative humidity or brightness temperature values (no level keyed in), the word of the array IOUT which stores the variable to be plotted is again determined in DO LOOP 43, after which the same procedure as above is once again followed.

If thicknesses are to be plotted, the words within IOUT containing the first (upper) pressure level and character (Z) are determined in DO LOOPs 45 and 47, with the same process being repeated for the second (lower) pressure level and character in DO LOOPs 52 and 54. (Note that variable IL2 will have been given a non-zero value (IL2=IKWP('LEV2',1,0)*10) just before DO LOOP 18 was entered.) Following this, assuming the data is not missing for either pressure level and that the retrieval is within the latitude/longitude boundaries, the retrieval raster/pixel coordinates are determined via SATEAR and SATTV, and the thickness to be plotted is calculated (IF(IAD2.NE.0) IDAT=IDAT-IOUT(IAD2)/ISC). Following this, the same steps as above are followed.

On the other hand, if difference plots between retrieval and guess for temperature or dewpoint are desired, DO LOOPs 45 and 47 are again executed, and the raster/pixel coordinates and plotting color are again calculated. Then, the difference quantity is formed immediately beyond statement 60 (IDAT=IDAT-IOUT(IADD+ISUB)/ISC). (The user should note at this point, in regard to DO LOOP 47 and the difference quantity calculation, that variable CCHR will have been reset to T or TD, depending on whether TDIF or DDIF was keyed in, and that variable ISUB will have been given a value of 5, back in the vicinity of statement 20.) Subsequent to the difference quantity calculation, the difference quantity itself is plotted on the video screen.

Finally, PLVA can also be used to plot surface data, such as winds or other variables. The user should key in only the appropriate character positional parameter (no level positional parameter keyed in) after PLVA (PLVA (space) character) to force the program to operate in this mode. For example, when the user desires a plot of surface winds (PLVA WIN), PLVA executes DO LOOP 43, jumps to statement 56, and calculates TV coordinates, color and the data value to be plotted (IDAT). Then, since variable LCHR .EQ. WIN, and variable ISS .NE. 0 (ISS = satellite identification number, and is .NE. 0 for GOES satellites), subroutine DIRADJ is called. This subroutine adjusts the wind direction so it is from its true direction when plotted on the satellite image (Note that true north, for example, is different on the video screen at different image longitude locations along a given image latitude.) Finally, subroutine BARB does the actual plotting.

Essentially, the main point to remember about PLVA is that the user has to worry only about the parameter itself (not the level) when plotting surface data or non-level data (such as relative humidity or brightness temperature), while he/she must worry about both parameter AND level when plotting Z, T, TD, retrieval-guess differences for T or TD, etc. at a given level other than the surface. The only exception to this rule pertains to a surface data plot of Z (height), where the user must key in SFC as the second positional parameter (see code detailing default case when neither first nor second positional parameters have been keyed in).

At this point, a few sample keyins will be presented.

Hopefully, these commands will enable the user to become more familiar with the operation of PLVA.

1.) PLVA Z 500: Plot 500 mb heights, taking data from retrieval MD file and row as listed in VASTEXT file.

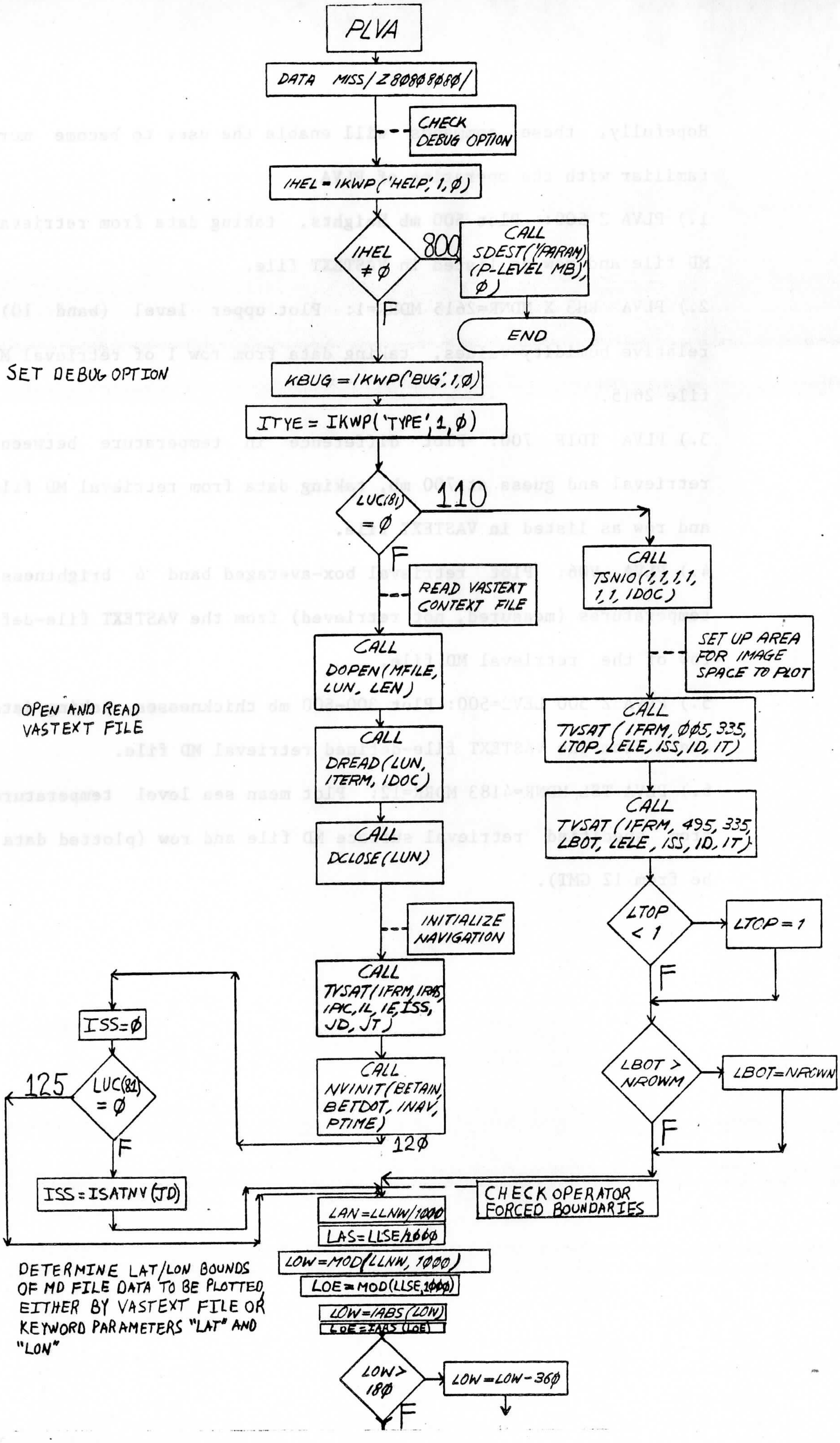
2.) PLVA RH3 X MDNR=2615 MDRR=1: Plot upper level (band 10) relative humidity values, taking data from row 1 of retrieval MD file 2615.

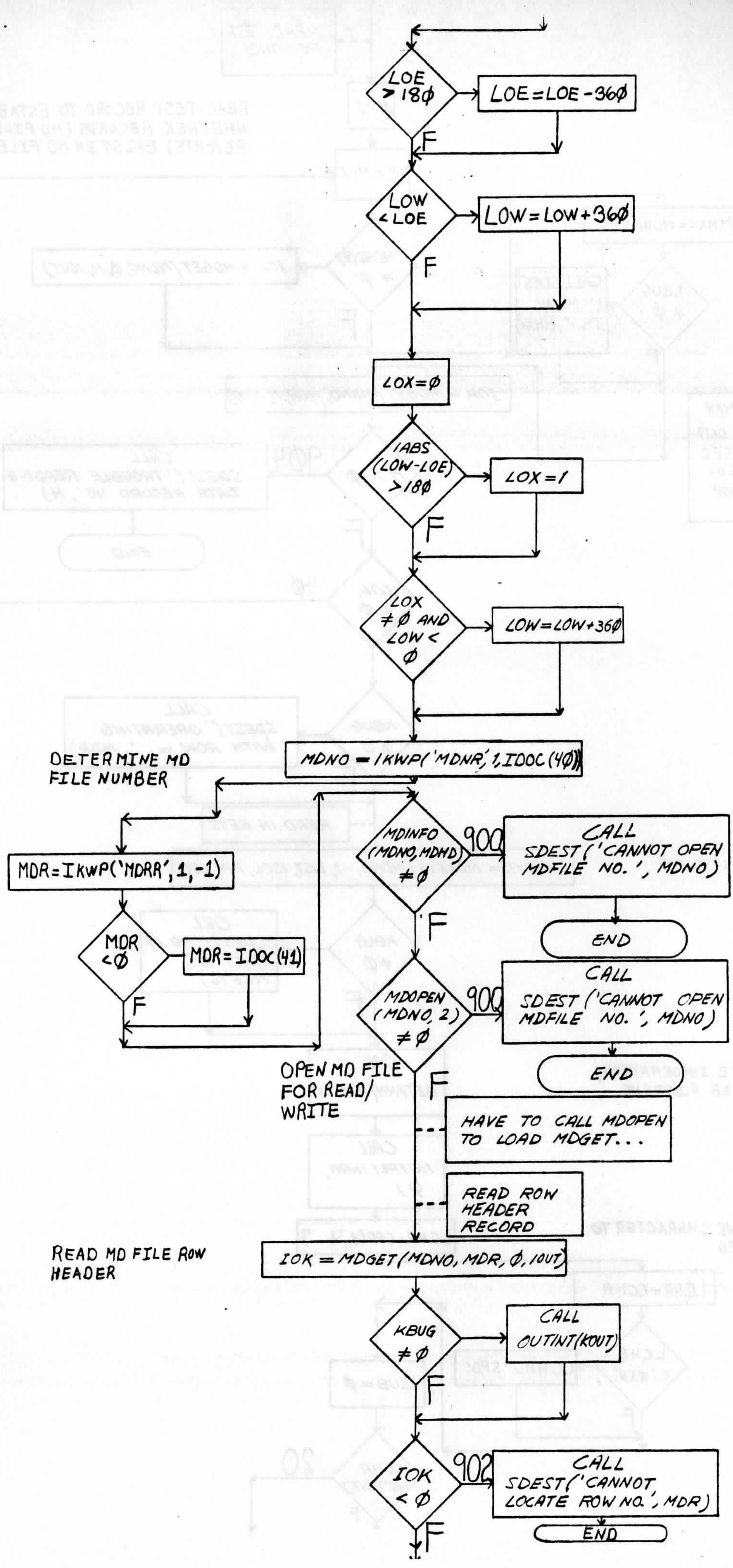
3.) PLVA TDIF 700: Plot difference in temperature between retrieval and guess at 700 mb, taking data from retrieval MD file and row as listed in VASTEXT file.

4.) PLVA V06: Plot retrieval box-averaged band 6 brightness temperatures (measured, not retrieved) from the VASTEXT file-defined row of the retrieval MD file.

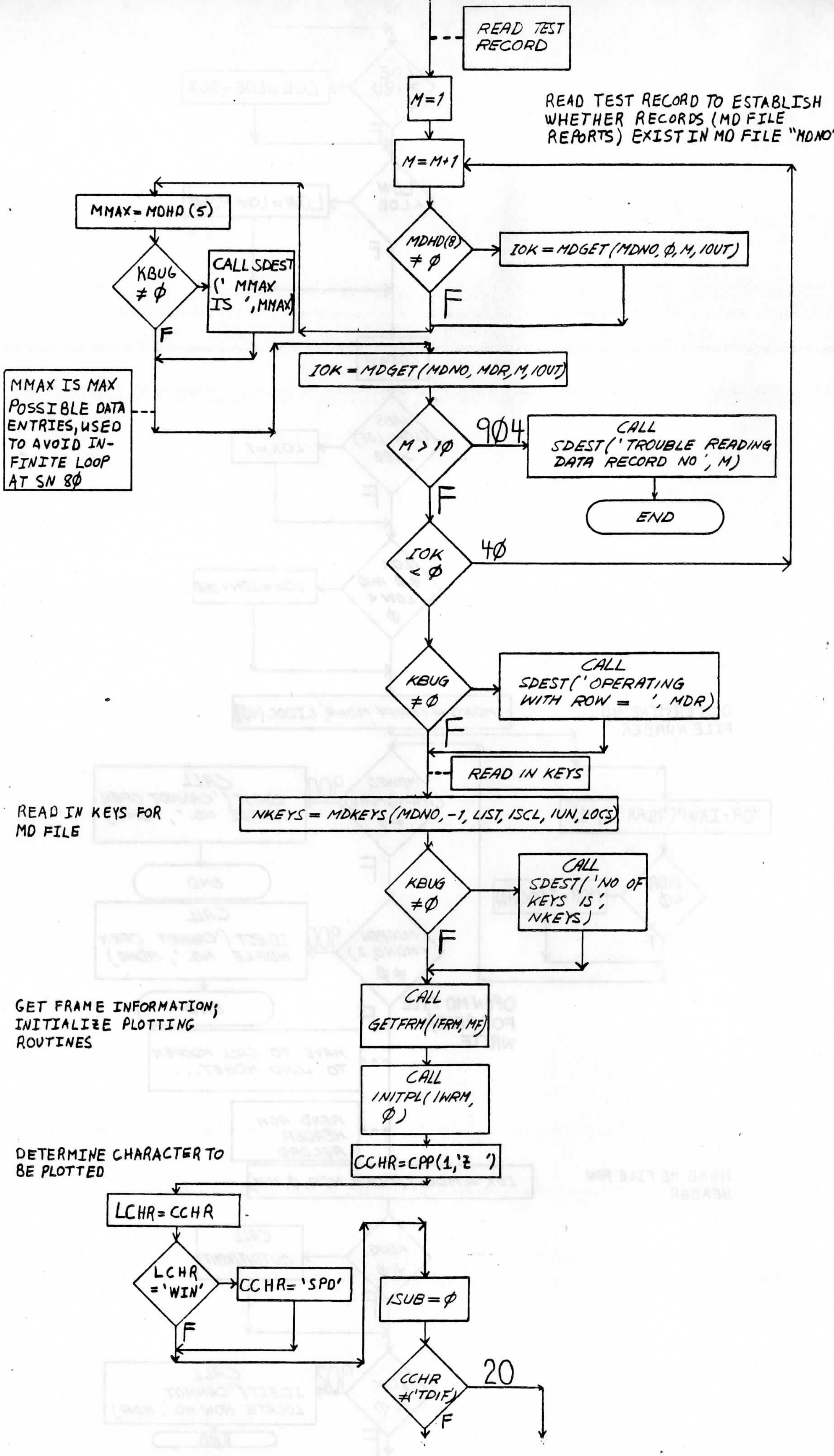
5.) PLVA Z 300 LEV2=500: Plot 300-500 mb thicknesses, taking data again from the VASTEXT file-defined retrieval MD file.

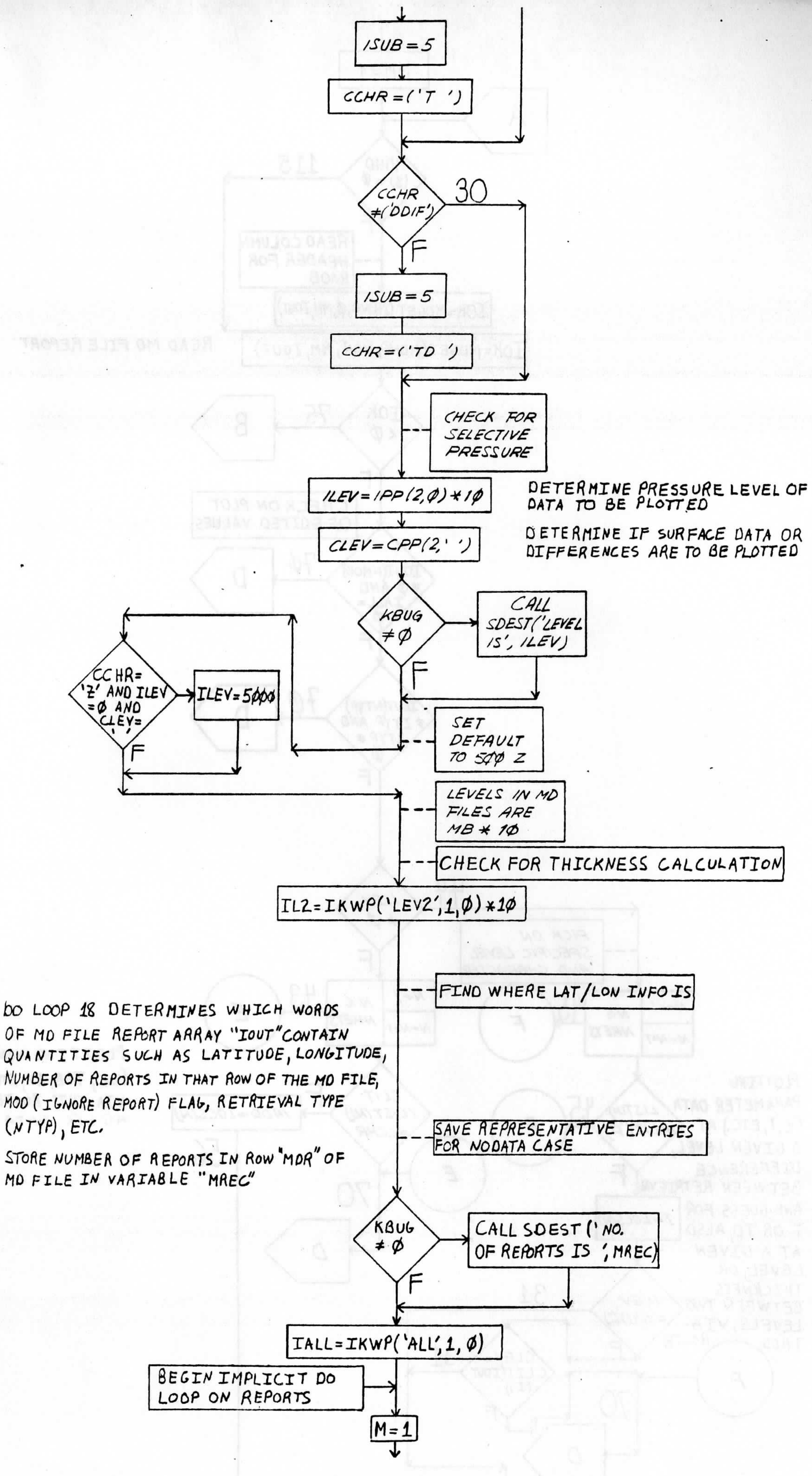
6.) PLVA TSL MDNR=4183 MDRR=12: Plot mean sea level temperature from specified retrieval surface MD file and row (plotted data will be from 12 GMT).

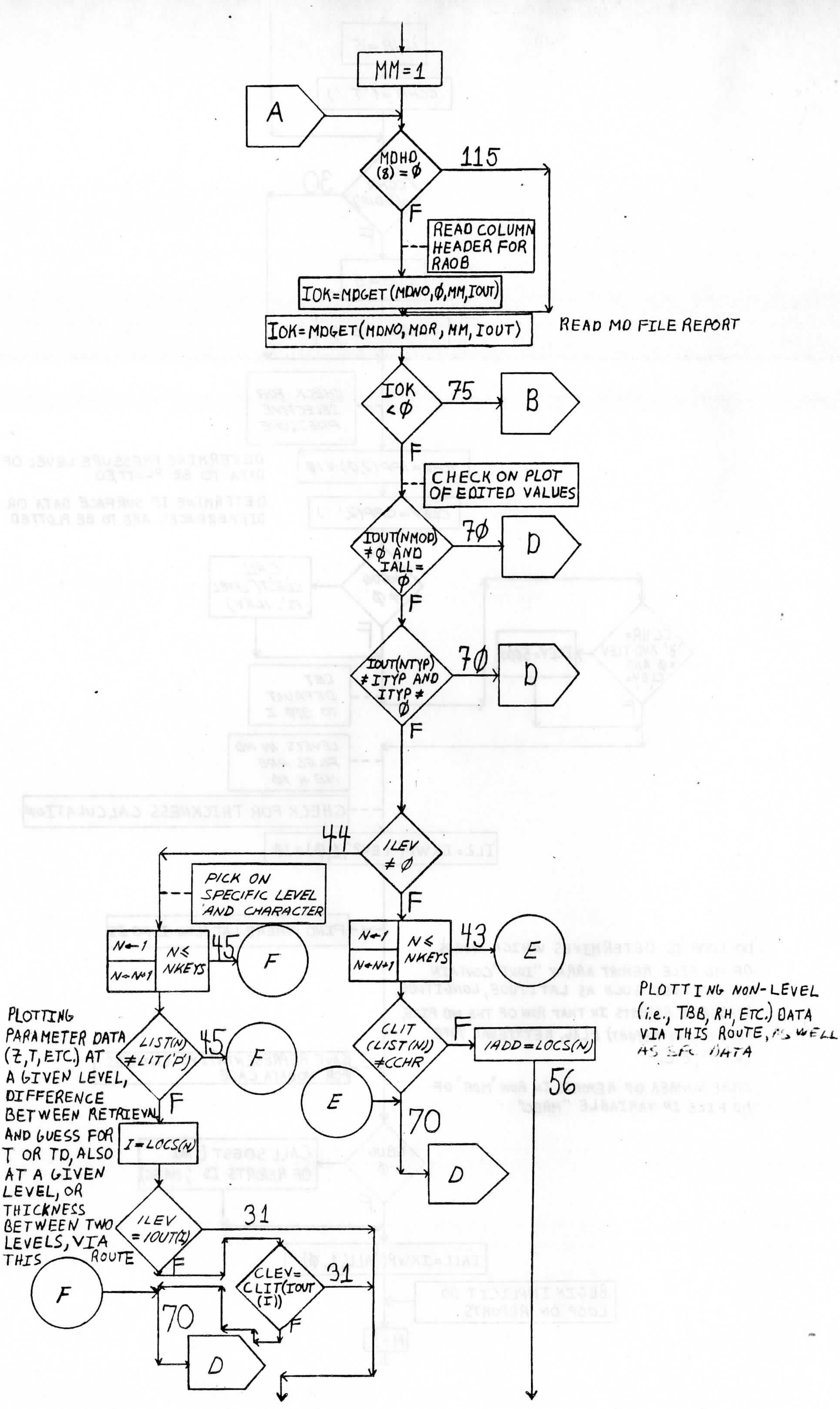

PLVA
DATA MISS/Z80808080/
CHECK DEBUG OPTION
IHEL = IKWP('HELP', 1, 0)
IHEL ≠ 0
800
CALL SDEST('/PARAM) (P-LEVEL MB)', 0)
END
F
SET DEBUG OPTION
KBUG = IKWP('BUG', 1, 0)
ITYE = IKWP('TYPE', 1, 0)
LUC(81) = 0
110
F
READ VASTEXT CONTEXT FILE
CALL TSNIO(1, 1, 1, 1, 1, IDOC)
SET UP AREA FOR IMAGE SPACE TO PLOT
CALL DOPEN(MFILE, LUN, LEN)
OPEN AND READ VASTEXT FILE
CALL DREAD(LUN, ITERM, IDOC)
CALL TVSAT(IFRM, 005, 335, LTOP, LELE, ISS, ID, IT)
CALL TVSAT(IFRM, 495, 335, LBOT, LELE, ISS, ID, IT)
CALL DCLOSE(LUN)
INITIALIZE NAVIGATION
LTOP < 1
LTOP = 1
F
CALL TVSAT(IFRM, IFRS, IPIC, IL, IE, ISS, JD, JT)
ISS = 0
CALL NVINIT(BETAIN, BETDOT, INAV, PTIME)
LUC(81) = 0
125
LBOT > NROWM
LBOT = NROWM
F
120
ISS = ISATNV(JD)
CHECK OPERATOR FORCED BOUNDARIES
LAN = LLNW/1000
LAS = LLSE/1000
LOW = MOD(LLNW, 1000)
LOE = MOD(LLSE, 1000)
LOW = IABS(LOW)
LOE = IABS(LOE)
DETERMINE LAT/LON BOUNDS OF MD FILE DATA TO BE PLOTTED, EITHER BY VASTEXT FILE OR KEYWORD PARAMETERS "LAT" AND "LON"
LOW > 180
LOW = LOW - 360
F

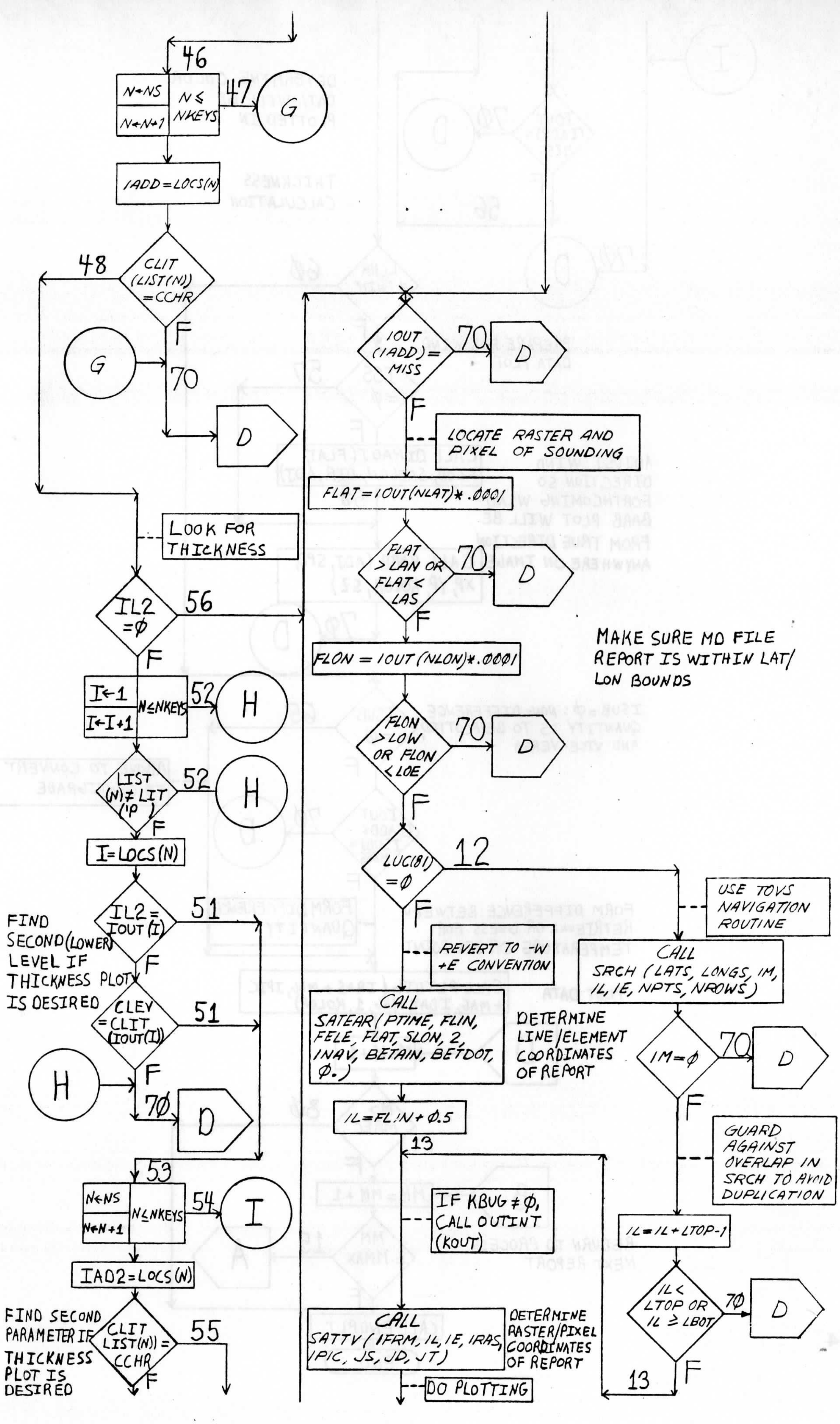


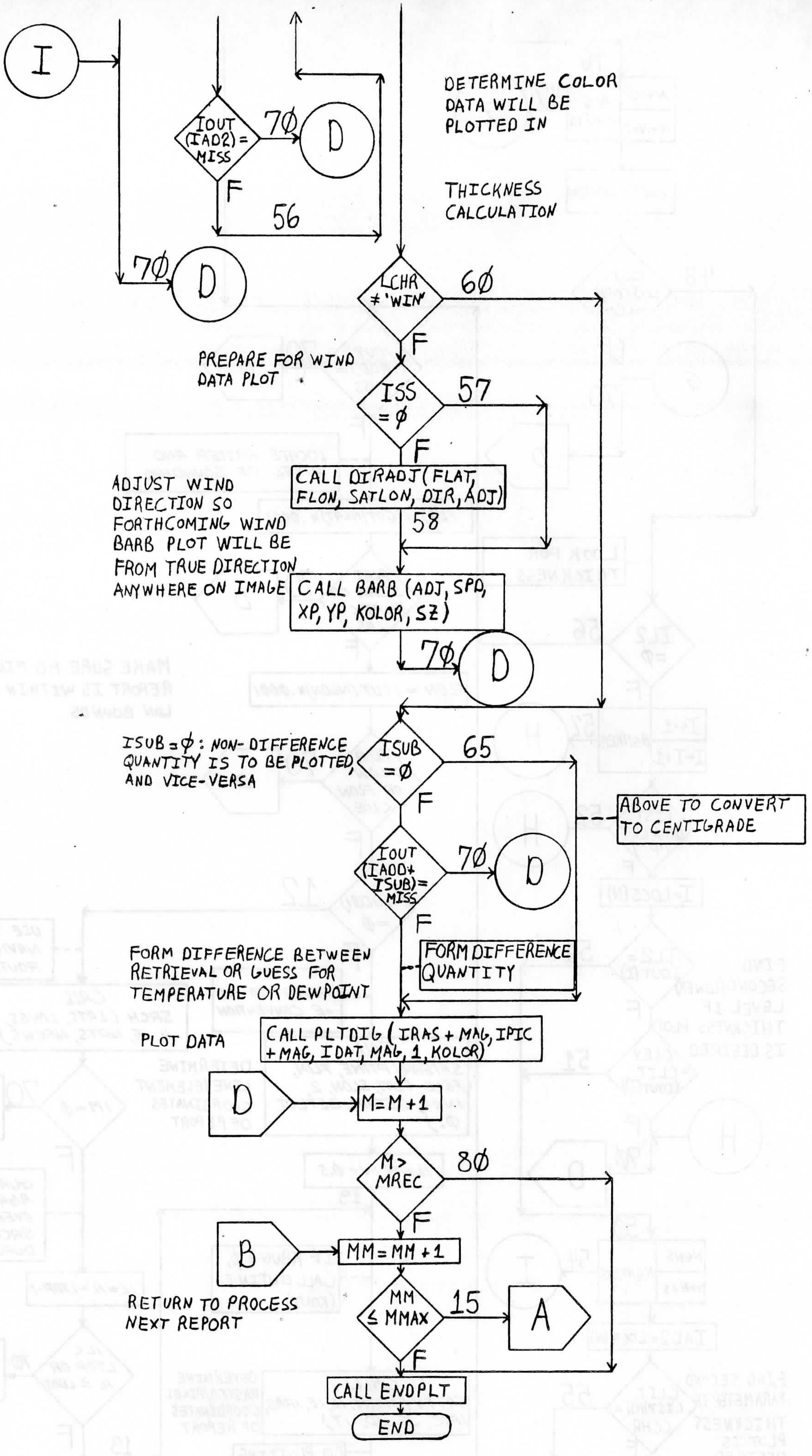
LOE > 18Ø
LOE=LOE-36Ø
F
LOW < LOE
LOW=LOW+36Ø
F
LOX=Ø
IABS (LOW-LOE) >18Ø
LOX=1
F
LOX ≠Ø AND LOW < Ø
LOW=LOW+36Ø
DETERMINE MD FILE NUMBER
MDNO = IKWP('MDNR', 1, IDOC(4Ø))
MDR=IKWP('MDRR', 1, -1)
MDR <Ø
MDR=IDOC(41)
F
MDINFO (MDNO, MDHD) ≠Ø
900
CALL SDEST('CANNOT OPEN MDFILE NO.', MDNO)
END
F
MDOPEN (MDNO, 2) ≠Ø
900
CALL SDEST ('CANNOT OPEN MDFILE NO.', MDNO)
END
OPEN MD FILE FOR READ/WRITE
F
HAVE TO CALL MDOPEN TO LOAD MDGET...
READ ROW HEADER RECORD
READ MD FILE ROW HEADER
IOK = MDGET(MDNO, MDR, Ø, IOUT)
KBUG ≠Ø
CALL OUTINT(KOUT)
F
IOK < Ø
902
CALL SDEST('CANNOT LOCATE ROW NO.', MDR)
END
F


READ TEST RECORD
M=1
READ TEST RECORD TO ESTABLISH WHETHER RECORDS (MD FILE REPORTS) EXIST IN MD FILE "MDNO"
M=M+1
MMAX=MDHD(5)
MDHD(8) ≠ ∅
IOK=MDGET(MDNO,∅,M,IOUT)
F
KBUG ≠ ∅
CALL SDEST (' MMAX IS ',MMAX)
F
IOK=MDGET(MDNO,MDR,M,IOUT)
MMAX IS MAX POSSIBLE DATA ENTRIES, USED TO AVOID INFINITE LOOP AT SN 8∅
M > 1∅
904
CALL SDEST('TROUBLE READING DATA RECORD NO',M)
END
F
IOK < ∅
4∅
KBUG ≠ ∅
CALL SDEST('OPERATING WITH ROW = ', MDR)
F
READ IN KEYS
READ IN KEYS FOR MD FILE
NKEYS = MDKEYS(MDNO,-1,LIST,ISCL,IUN,LOCS)
KBUG ≠∅
CALL SDEST('NO OF KEYS IS', NKEYS)
F
GET FRAME INFORMATION; INITIALIZE PLOTTING ROUTINES
CALL GETFRM(IFRM,MF)
CALL INITPL(IWRM, ∅)
DETERMINE CHARACTER TO BE PLOTTED
CCHR=CPP(1,'Z ')
LCHR=CCHR
LCHR ='WIN'
CCHR='SPD'
F
ISUB=∅
CCHR ≠('TDIF')
20
F


ISUB=5
CCHR=('T ')
CCHR ≠('DDIF')
30
F
ISUB=5
CCHR=('TD ')
CHECK FOR SELECTIVE PRESSURE
ILEV=IPP(2,Ø)*1Ø
DETERMINE PRESSURE LEVEL OF DATA TO BE PLOTTED
CLEV=CPP(2,' ')
DETERMINE IF SURFACE DATA OR DIFFERENCES ARE TO BE PLOTTED
KBUG ≠Ø
CALL SDEST('LEVEL IS', ILEV)
F
CCHR='Z' AND ILEV =Ø AND CLEV=' '
ILEV=5ØØØ
F
SET DEFAULT TO 5ØØ Z
LEVELS IN MD FILES ARE MB * 1Ø
CHECK FOR THICKNESS CALCULATION
IL2=IKWP('LEV2',1,Ø)*1Ø
FIND WHERE LAT/LON INFO IS
DO LOOP 18 DETERMINES WHICH WORDS OF MD FILE REPORT ARRAY "IOUT" CONTAIN QUANTITIES SUCH AS LATITUDE, LONGITUDE, NUMBER OF REPORTS IN THAT ROW OF THE MD FILE, MOD (IGNORE REPORT) FLAG, RETRIEVAL TYPE (NTYP), ETC.
STORE NUMBER OF REPORTS IN ROW "MDR" OF MD FILE IN VARIABLE "MREC"
SAVE REPRESENTATIVE ENTRIES FOR NODATA CASE
KBUG ≠Ø
CALL SDEST ('NO. OF REPORTS IS ', MREC)
F
IALL=IKWP('ALL',1,Ø)
BEGIN IMPLICIT DO LOOP ON REPORTS
M=1


MM=1
A
MDHD (8) = Ø
115
F
READ COLUMN HEADER FOR RAOB
IOK=MDGET(MDNO,Ø,MM,IOUT)
IOK=MDGET(MDNO, MDR, MM, IOUT)
READ MD FILE REPORT
IOK <Ø
75
B
F
CHECK ON PLOT OF EDITED VALUES
IOUT(NMOD) ≠Ø AND IALL = Ø
7Ø
D
F
IOUT(NTYP) ≠ITYP AND ITYP ≠ Ø
7Ø
D
F
44
ILEV ≠Ø
F
PICK ON SPECIFIC LEVEL AND CHARACTER
N←1
N←N+1
N≤ NKEYS
45
F
43
E
PLOTTING NON-LEVEL (i.e., TBB, RH, ETC.) DATA VIA THIS ROUTE, AS WELL AS SFC DATA
PLOTTING PARAMETER DATA (Z,T, ETC.) AT A GIVEN LEVEL, DIFFERENCE BETWEEN RETRIEVAL AND GUESS FOR T OR TD, ALSO AT A GIVEN LEVEL, OR THICKNESS BETWEEN TWO LEVELS, VIA THIS ROUTE
LIST(N) ≠LIT('P')
45
F
CLIT (LIST(N)) ≠CCHR
F
IADD=LOCS(N)
56
E
70
D
I=LOCS(N)
ILEV =IOUT(I)
31
F
CLEV= CLIT(IOUT (I))
31
F
70
D


46
N←NS
N←N+1
N ≤ NKEYS
47
G
IADD=LOCS(N)
48
CLIT (LIST(N)) =CCHR
F
G
70
D
LOOK FOR THICKNESS
IL2 =Ø
56
F
I←1
I←I+1
N≤NKEYS
52
H
LIST (N)≠LIT (IP)
52
H
F
I=LOCS(N)
FIND SECOND(LOWER) LEVEL IF THICKNESS PLOT IS DESIRED
IL2= IOUT (I)
51
F
CLEV =CLIT (IOUT(I))
51
F
H
7Ø
D
53
N←NS
N←N+1
N≤NKEYS
54
I
IAD2=LOCS(N)
FIND SECOND PARAMETER IF THICKNESS PLOT IS DESIRED
CLIT LIST(N)) = CCHR
55
F
IOUT (IADD)= MISS
70
D
F
LOCATE RASTER AND PIXEL OF SOUNDING
FLAT=IOUT(NLAT)*.ØØØ1
FLAT >LAN OR FLAT< LAS
70
D
F
FLON = IOUT(NLON)*.ØØØ1
MAKE SURE MD FILE REPORT IS WITHIN LAT/LON BOUNDS
FLON >LOW OR FLON <LOE
70
D
F
LUC(81) =Ø
12
F
REVERT TO -W +E CONVENTION
CALL SATEAR(PTIME, FLIN, FELE, FLAT, SLON, 2, INAV, BETAIN, BETDOT, Ø.)
DETERMINE LINE/ELEMENT COORDINATES OF REPORT
IL=FLIN+Ø.5
13
IF KBUG≠Ø, CALL OUTINT (KOUT)
CALL SATTV(IFRM, IL, IE, IRAS, IPIC, JS, JD, JT)
DETERMINE RASTER/PIXEL COORDINATES OF REPORT
DO PLOTTING
USE TOVS NAVIGATION ROUTINE
CALL SRCH (LATS, LONGS, IM, IL, IE, NPTS, NROWS)
IM=Ø
70
D
F
GUARD AGAINST OVERLAP IN SRCH TO AVOID DUPLICATION
IL=IL+LTOP-1
IL< LTOP OR IL ≥ LBOT
7Ø
D
13
F


I
IOUT (IAD2) = MISS
70
D
F
56
70
D
DETERMINE COLOR DATA WILL BE PLOTTED IN
THICKNESS CALCULATION
LCHR ≠ 'WIN'
60
F
PREPARE FOR WIND DATA PLOT
ISS = 0
57
F
CALL DIRADJ(FLAT, FLON, SATLON, DIR, ADJ)
ADJUST WIND DIRECTION SO FORTHCOMING WIND BARB PLOT WILL BE FROM TRUE DIRECTION ANYWHERE ON IMAGE
58
CALL BARB (ADJ, SPD, XP, YP, KOLOR, SZ)
70
D
ISUB = 0: NON-DIFFERENCE QUANTITY IS TO BE PLOTTED, AND VICE-VERSA
ISUB = 0
65
F
ABOVE TO CONVERT TO CENTIGRADE
IOUT (IADD+ ISUB)= MISS
70
D
F
FORM DIFFERENCE BETWEEN RETRIEVAL OR GUESS FOR TEMPERATURE OR DEWPOINT
FORM DIFFERENCE QUANTITY
PLOT DATA
CALL PLTDIG (IRAS + MAG, IPIC + MAG, IDAT, MAG, 1, KOLOR)
D
M=M+1
M > MREC
80
F
B
MM = MM + 1
RETURN TO PROCESS NEXT REPORT
MM ≤ MMAX
15
A
F
CALL ENDPLT
END

Subroutines used by PLVA:

1) DOPEN-I
2) DREAD-I
3) DCLOSE-I
4) TVSAT-I
5) NVINIT-I
6) TSNIO-I
7) OUTINT-I
8) GETFRM-I
9) INITPL-I
10) SATEAR-I
11) SRCH-I
12) SATTV-I
13) PLTDIG-I
14) ENDPLT-I
15) SDEST-I
16) DIRADJ-I
17) BARB-I
18) ENCODE
19) ENCODX
20) ABORT
21) CONTNT
22) LTQ
23) ZECONV
24) MOVB
25) CLEANA
26) TQ
27) PRLINX
28) PRCLOS
29) PROPEN
30) PRPRPR
31) LOCK
32) PRWR
33) PRRD
34) UNLOCK
35) PRCL
36) POST
37) BLKA
38) TRMNL
39) TQMES
40) STC
41) ITOC
42) GETNAV
43) EPOCH
44) GETGAM
45) VASNAV
46) DDEST
47) MOVCW
48) CLEANW
49) EDEST
50) MOVW
51) II
52) DWRITE
53) SATPOS
54) PLTDIG
55) QGDASH
56) DSHOFF
57) PLOT
58) DSHON
59) ENPT
60) PENADD
61) PACK
62) SENOUT
63) ATOE
64) TEKPUT
65) PAGE
66) PENMOV
67) BOX
68) PENBEG
69) WALK

BNVA

The purpose of BNVA is to create grids of different meteorological parameters. These parameters can be either from retrieval MD or other types of MD files. Some of the parameters that can be gridded with BNVA include differences between retrieval and guess for temperature (T) and dewpoint (TD) (brightness temperature (TBB) differences between retrieval and guess cannot be gridded as of this time, due to the structure of retrieval MD file reports). In addition, one can also generate grids of T, TD, TBB, relative humidity, thickness, winds or other quantities which exist in the MD file from which the data is being taken. Before the user runs BNVA, the gridfile he/she wishes the results to be stored in must be set with system program IGU to make sure the final grid is stored in the desired gridfile. (The reason for this will become more clear as the discussion proceeds.)

There are two first guess options which can be used in BNVA, in addition to the option of not using ANY first guess. One of these is the grid first guess, while the other is the guess MD file first guess. A guess is advisable if the data to be gridded is sporadic in coverage over the area of the desired grid. In such situations, the generation of a grid without using a first guess could lead to a poor analysis. In addition, note that each guess consists only of heights, temperatures or dewpoints, depending upon which parameter is being gridded, so the user must keep in mind that a first guess cannot be used when generating grids of TBBs, thicknesses, winds, differences between retrieval and guess for T or TD, etc.

Initially, things such as the final output grid number (NGOUT) and the debug option (KBUG) are set, after which the VASTEXT file is opened, read and closed. At this point, several other variables are determined, such as the retrieval MD file and row numbers (MDNO and MDR, respectively), the upper air guess grid file number (NGFG), and the upper air guess MD file number (MDNG). All four of these variables have VASTEXT defaults if no keyword parameter value has been designated. Now, after the parameter to be gridded (ICHR) and the level at which it is to be gridded (ILEV) have been determined, the MD file (if it exists -- checked via function MDINFO) is opened for read/write, the row header is read, a test to make sure that data exists in the MD file is performed ("READ TEST RECORD"), and a call to function MDKEYS reads the keys contained within the schema for that particular MD file. At this point, the elements within the MD file report array for the number of retrievals or reports in a given row of the MD file (variable NCA), latitude, longitude, and mod flag (variable NMOD -- MD file report array element NMOD will indicate whether a given retrieval or report is good) are determined in DO LOOP 2. The variable which will store the number of reports, NRPT, is then filled following DO LOOP 2.

The following lines of code set several variables, including variable ISUB, which equals 5 for T or TD differences between retrieval and guess. Then, keyword parameter LEV2 is evaluated to see if thickness calculations and a resultant grid generation thereof is desired, and variable NGG is filled with the number of the guess grid to be used in the event a guess grid option is desired. Note that variable CLEV is unimportant at this time (5/1/84), since TBB differences between retrieval and guess cannot be done, and surface grids of retrieval minus guess differences for temperature or dewpoint do not require a second keyed-in positional parameter.

From this point forward (IF (NGG.NE.0) GO TO 14), the discussion of BNVA will be broken down into separate discussions for each of the three program options: guess grid, guess MD file or no guess.

First, in the case when the guess grid option is being used, BNVA moves to statement 14 (IF (NGG.NE.0) GO TO 14). Then, after the analysis field (the array IDL, in this case) has been initialized to zero in DO LOOP 12, the guess grid is accessed from the guess gridfile (NGFG) by function IGGET. The user must make sure the guess grid being accessed is no larger than 2400 words; otherwise, IOK will be less than 0, and BNVA will terminate. Following this, the number of gridpoints (NPTS) is calculated, the guess grid data is transferred to the array FLD, the latitude/longitude dimensions of the grid are determined (LAN/TLAT, LAS/SLAT, LOW/WLON and LOE/ELON), and information pertaining to number of rows, number of columns and grid increment is stored in KOUT(2), (3) and (4), respectively. In addition, if KBUG .GE. 0, a call to subroutine OUTINT will display this information on the CRT. Then, after a jump to statement 110 (remember, this is a discussion of the guess grid option), and if keyword parameter NODAT was keyed in .NE. 0, BNVA will jump to statement 150, and the guess grid just accessed will be stored in gridfile NGFG by function IGPUT, after which BNVA will terminate.

Normally, however, NODAT = 0, and the program enters an implicit DO LOOP ("BEGIN IMPLICIT DO LOOP ON REPORTS"), the broad purpose of which is to read data from MD file MDNO and prepare the input data for the subsequent grid generation. The length of this loop is approximately 130 lines of code.

Initially, after a given MD file report is read (IOK=MDGET(MDNO,MDR,NN,KBUF)), and assuming it has not been previously earmarked for rejection by the mod flag (KBUF(NMOD).EQ. 0), DO LOOPs 43, 45, 47, 52 and 54 determine (in terms of level and parameter) which data will be taken from each MD file report and used in the subsequent grid generation. Following these loops, and assuming the raw MD file data is not missing (KBUF(IADD/IAD2) is .NE. MISS), the data to be gridded (nearly in final form) is stored in variable IDAT (IDAT=KBUF(IADD)). The final data value to be gridded will then be calculated and stored in variable DAT.

If the correct data cannot be accessed (in terms of getting an MD file report during the original MDGET, or in terms of matching the pressure level/character during the DO LOOP sequence), the implicit loop will jump to statement 148, increment variable NN, and jump back to statement 15 to access the next MD file data report, meaning THAT particular report will not be used when generating the grid. Other conditions causing a jump to statement 148 include missing data (IF

(KBUF(IADD).EQ.MISS) GO TO 148), or location of the MD file report outside the grid boundaries (IF FLAT.GT. TLAT.OR.FLAT.LT.SLAT) GO TO 148).

However, assuming the raw MD file data is gathered and the value to be gridded is determined with no problems, the final data to be gridded, and the latitude/longitude coordinates of the data, are stored in the arrays DA, STLAT and STLON, respectively. Now, since IGB is .NE. 0, and assuming that keyword parameter EDIT was not keyed in (IKWP('EDIT',1,0).EQ.0), a gross error check is performed which compares the value to be gridded (DAT) with a value taken from the guess grid (VAL).

If a given MD file report fails the gross error test, BNVA jumps to statement 148, again meaning THAT particular MD file report will not be used in the Barnes analysis. However, assuming the test passes, the program jumps to statement 70, where the Barnes arrays DAS, RW and CL are filled. Then, after the variables SUM and SUMS have been updated, and assuming KK .LE. NRPT and that there are less than 5000 data values generated so far, variable NN is incremented and, unless NN .GT. MMAX, BNVA jumps back to statement 15 to look for the next report in row MDR of the MD file.

After all the MD data has been read and prepared for the Barnes analysis, BNVA jumps to statement 140 (usually via the statement IF (KK.GT.NRPT) GO TO 140), where a standard deviation (ISD) is determined both from the data to be gridded and previously-set ERR values (see code immediately preceeding the implicit DO LOOP). Then, a Barnes analysis (subroutine FBARN) is executed, which produces a uniform set of gridpoint values based on the data to be gridded and the guess grid. The Barnes analysis makes only 1 pass if the guess grid or guess MD file option is being used, while 2 passes are made for the no- guess option. The final data from the Barnes analysis is then stored in the array IDL in DO LOOP 155.

Following this, if the edit option is NOT in operation (IKWP ('EDIT',1,0) .EQ. 0), the grid is written into gridfile NGFG in the next empty slot after NGOUT (see description of function IGPUT in appendix IV). At this point, if the pass weights are not to be written into the same gridfile, BNVA is finished. However, if the weights for the one pass of the Barnes analysis are to be written into the gridfile (IKWP('WGT',1,0) .NE. 0), one more call to function IGPUT accomplishes this task, after which BNVA terminates.

On the other hand, if the edit option IS in operation (IKWP('EDIT',1,0).GT.0), a value determined from the final grid (VAL) is compared to the original data value from before the Barnes analysis (DAT), and if the difference is .GE. variable ERR, the element of the array KBUF which indicates rejection (NMOD...the mod flag) is set to a non-zero value. Then, the MD file is updated via function MDPUT to reflect this change. Note that the Barnes analysis grid will not be written to an output gridfile if the edit option is in effect (see the RETURN immediately following DO LOOP 130). In other words, the only purpose of the edit option is to edit the source MD file data.

Returning to the statement IF (NGG.NE.0) GO TO 14, I will now discuss briefly the guess MD file option of BNVA. First, since NGG = 0, the boundaries of the grid (LN, LS, LW and LE) are set by keyed-in values of LAT and(or) LON, or by VASTEXT default values if no keyin of LAT/LON exists. Then, the grid increment (in tenths of degrees) is determined (INC). If the grid increment has been keyed in (LINC .NE. 0), the final number of gridpoints (NPTS) is not restricted (can be .GT. 2400). However, if the grid increment has not been keyed in (LINC .EQ. 0), the total number of gridpoints must be .LE. 2400. Otherwise, the grid increment is increased by 1 (less resultant gridpoints), and control returns to statement 13.

Once the grid increment has been determined, the analysis field is initialized to 0, a jump to statement 105 is taken, variables TLAT, SLAT, WLON and ELON (bounds for using MD file data in the Barnes analysis) are set, and the guess data is read from the guess MD file (MDNG), with the surface option (NOSFC) taken into account, via subroutine ANGSS. Then, assuming keyword parameter NODAT has not been keyed in (IKWP('NODAT',1,0).EQ.0), variables ERR/FCK are set, and the implicit DO LOOP accesses data from the MD file and prepares it in a like manner to that described earlier for the guess grid option. The final data and their latitude/longitude positions are again stored in the arrays DA, STLAT and STLON, respectively. Next, since IGB is .NE. 0, and assuming keyword parameter EDIT was not keyed in, the gross error check is performed, which compares the value to be gridded (DAT) with a value taken this time from the guess MD file (VAL). If a given MD file report fails the gross error test, BNVA jumps to statement 148. On the other hand, if the test does pass, the Barnes arrays are filled, variables SUM and SUMS are updated, and BNVA returns to statement 15 for more MD file data. After all the MD file reports have been read and the data to be gridded determined, BNVA jumps to statement 140 (KK becomes .GT. NRPT), where the standard deviation of the data to be gridded is calculated, and the Barnes analysis is performed. From this point to the end, BNVA via the guess MD file option functions identically to the guess grid option, except the destination gridfile number for the final Barnes analysis grid is given by word 6 of User Common (UC(6)... set before the execution of BNVA via system program IGU, and accessed within BNVA itself by function LUC), instead of the variable NGFG.

Returning once more to the statement IF (NGG.NE.0) GO TO 14, the no guess option of BNVA will now be described. This version should be used only if the MD file data coverage is fairly complete over the grid area.

Beginning with this statement, BNVA progresses with the no guess option precisely as the guess MD file option down through the initialization of the analysis field. Then, after jumping to statement 105 (note that NGG=0 in this case), the bounds for using MD file data in the grid generation (TLAT, SLAT, WLON and ELON) are set, variable CGS is established (CGS=CKWP('GSS',1,' ') and, since CGS will equal ' ', BNVA jumps to statement 110. At this point, keyword parameter NODAT is checked and, assuming NODAT was not keyed in, the program sets variables ERR/FCK and then passes directly into the implicit DO LOOP. Within the loop, a given MD file report is read via function MDGET, and DO LOOPs 43, 45, 47, 52 and 54 again evaluate which particular data will be taken from a given MD file report and used in the grid generation. (Note that any type of MD file data possible with BNVA (see introductory paragraph) can be gridded with the no guess option.) Then, assuming all the raw data exists (KBUF(IADD) and, if needed, (IAD2) .NE. MISS), the data value to be used in the grid is calculated and stored in variable IDAT and then subsequently in variable DAT. IDAT will be calculated via IDAT=KBUF(IADD), for single-level data, IDAT=IDAT-KBUF(IAD2), for thicknesses, or IDAT=IDAT-KBUF(IADD+ISUB), for retrieval-guess differences of T or TD. At this point, the arrays DA, STLAT and STLON are again filled, and BNVA jumps to statement 146 (IF (IGB.EQ.0) GO TO 146) where, if the character to be gridded is wind, the appropriate U or V component is determined (if U or V was keyed in as the character to be gridded, ICHR='SPD', LCHR='U' or 'V'). Following this, the Barnes arrays DAS, RW and CL are filled, and variables SUM and SUMS are updated. Then, after the implicit DO LOOP has been exited, the standard deviation of the data to be gridded is calculated, and the Barnes analysis is performed by subroutine FBARN. From this point to the end, BNVA processes the final gridded data identically to the guess MD file option, with the exception that 2 pass weight arrays can be stored in gridfile NGRF with the no-guess option, as opposed to only 1 with the guess MD file (and guess grid) options, due to the fact the Barnes analysis makes 2 passes for the no-guess option.

Since there might be some confusion concerning the different gridding options possible with BNVA, I will now present a few sample keyins and the associated response of BNVA.

1.) BNVA T 700 19: grid 700 mb retrieval temperatures taken from MD file whose number is defined in the VASTEXT file, using the guess contained in grid #19 of the upper air guess gridfile defined in the VASTEXT file; results will be stored after the last grid in the same upper air guess gridfile.

2.) BNVA Z 500 GSS=G INC=10: grid 500 mb retrieval heights taken from the retrieval MD file defined in the VASTEXT file, using a first guess taken from the upper air guess MD file (whose number is also defined in the VASTEXT file); final grid will have spacing of 1 degree latitude/longitude, and will be stored in the gridfile whose number is defined in word 6 of User Common.

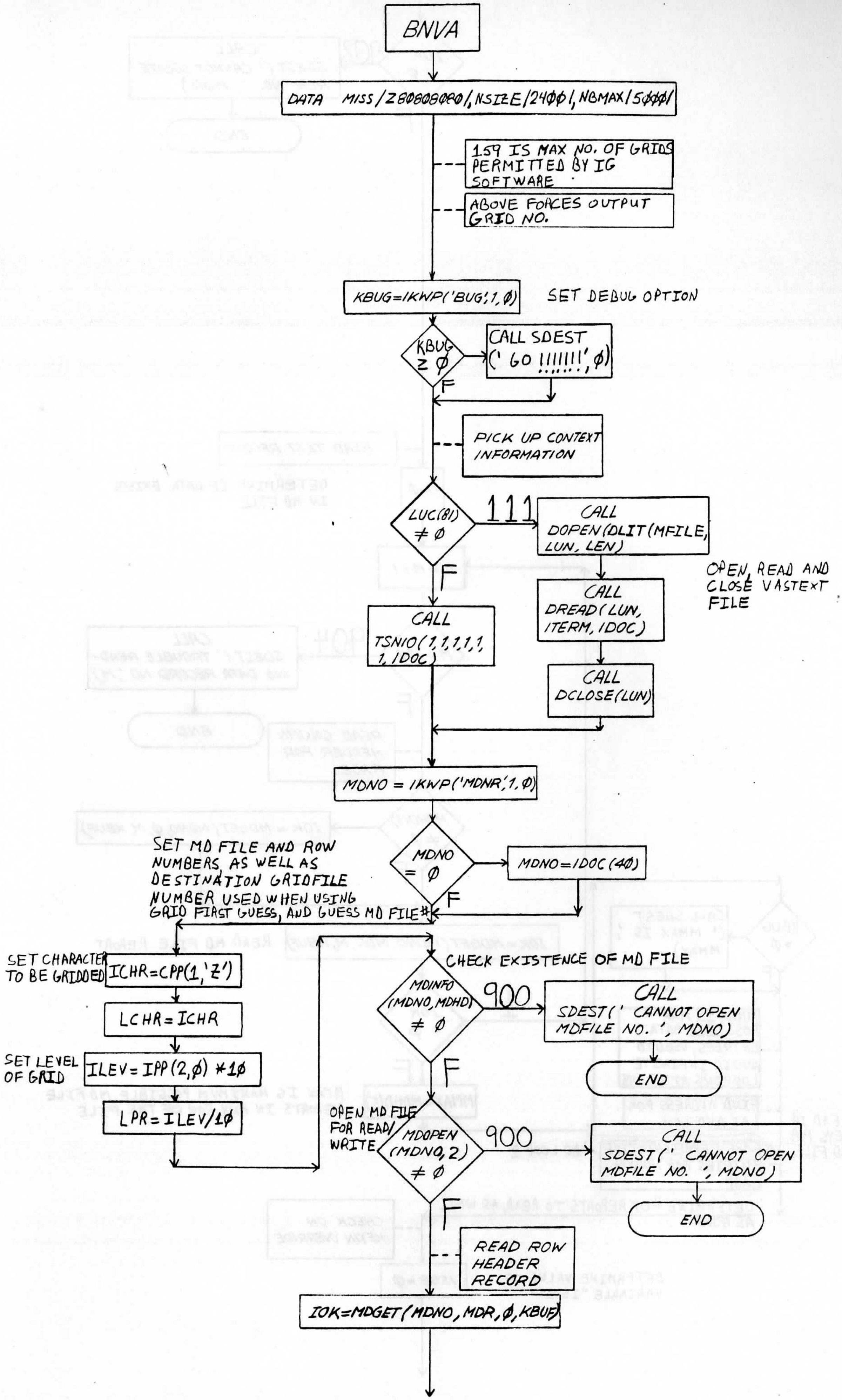
3.) BNVA T 500: grid 500 mb retrieval temperatures, with no first guess, with the data being taken again from the VASTEXT file-defined retrieval MD file; results will be stored in gridfile whose number is defined in word 6 of User Common.

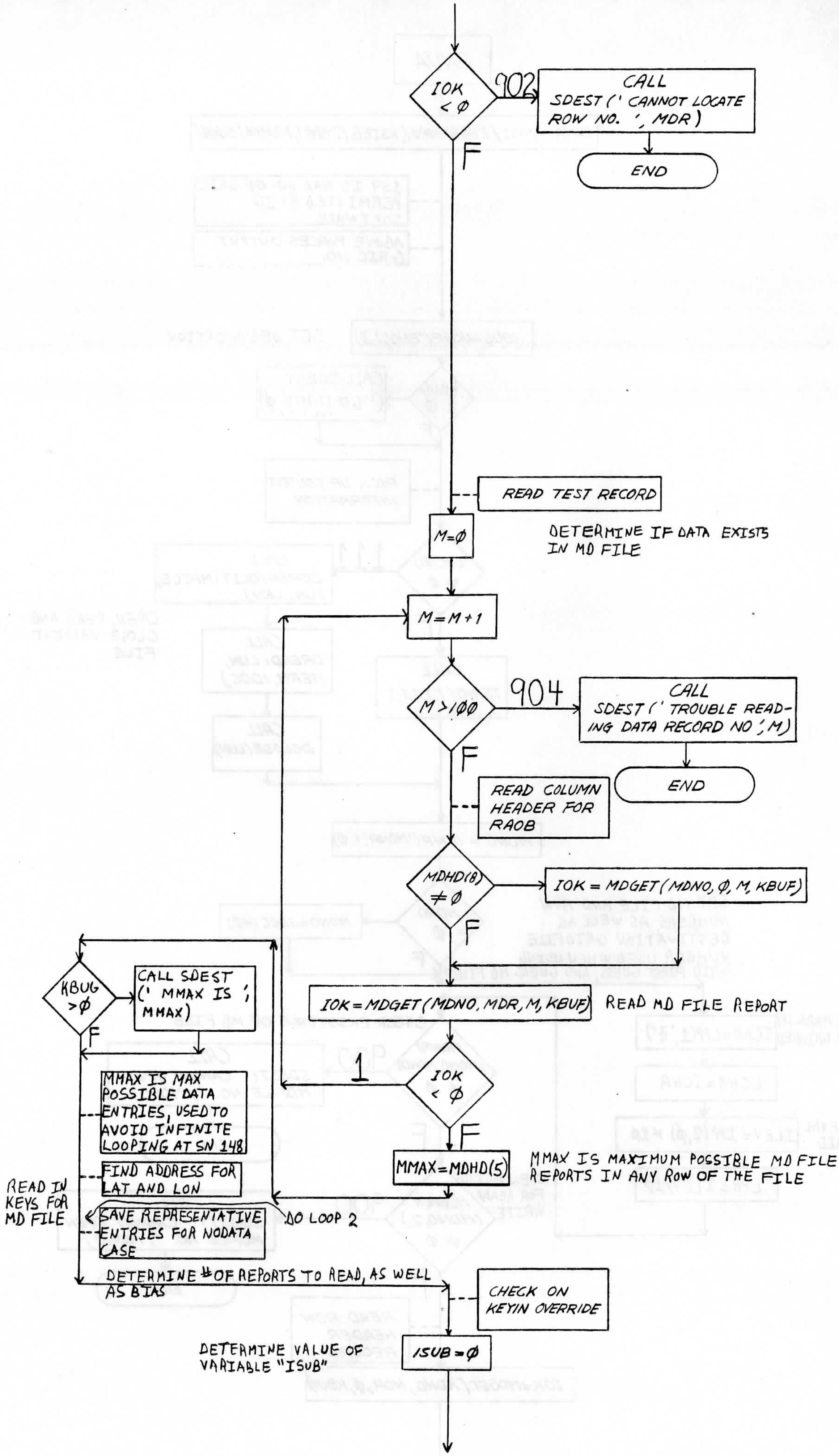
4.) BNVA T 500 MDNR=4184: same as above, except raw data will be taken from MD file 4184.

5.) BNVA Z 500 LEV2=850: grid 850-500 mb thicknesses, using no first guess, with the data being taken from the VASTEXT file-defined retrieval MD file; results stored in gridfile whose number is defined in word 6 of User Common.

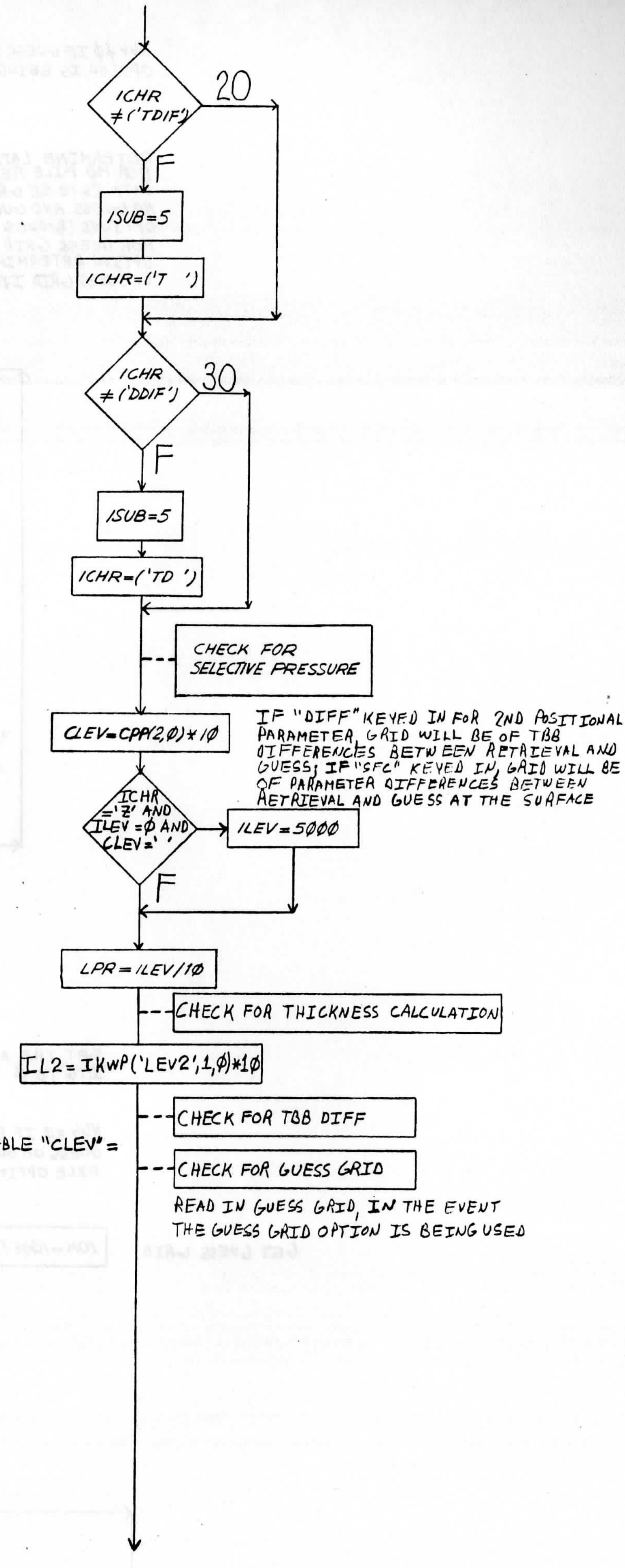
6.) BNVA T: grid surface temperature, using no first guess, with the data being taken from the MD file whose number is defined in the VASTEXT file; store results in gridfile whose number is defined via word 6 of User Common.

7.) BNVA TDIF 700: grid 700 mb retrieval minus guess differences for temperature, with data taken from VASTEXT file-defined retrieval MD file; results stored in gridfile whose number is defined in word 6 of User Common.

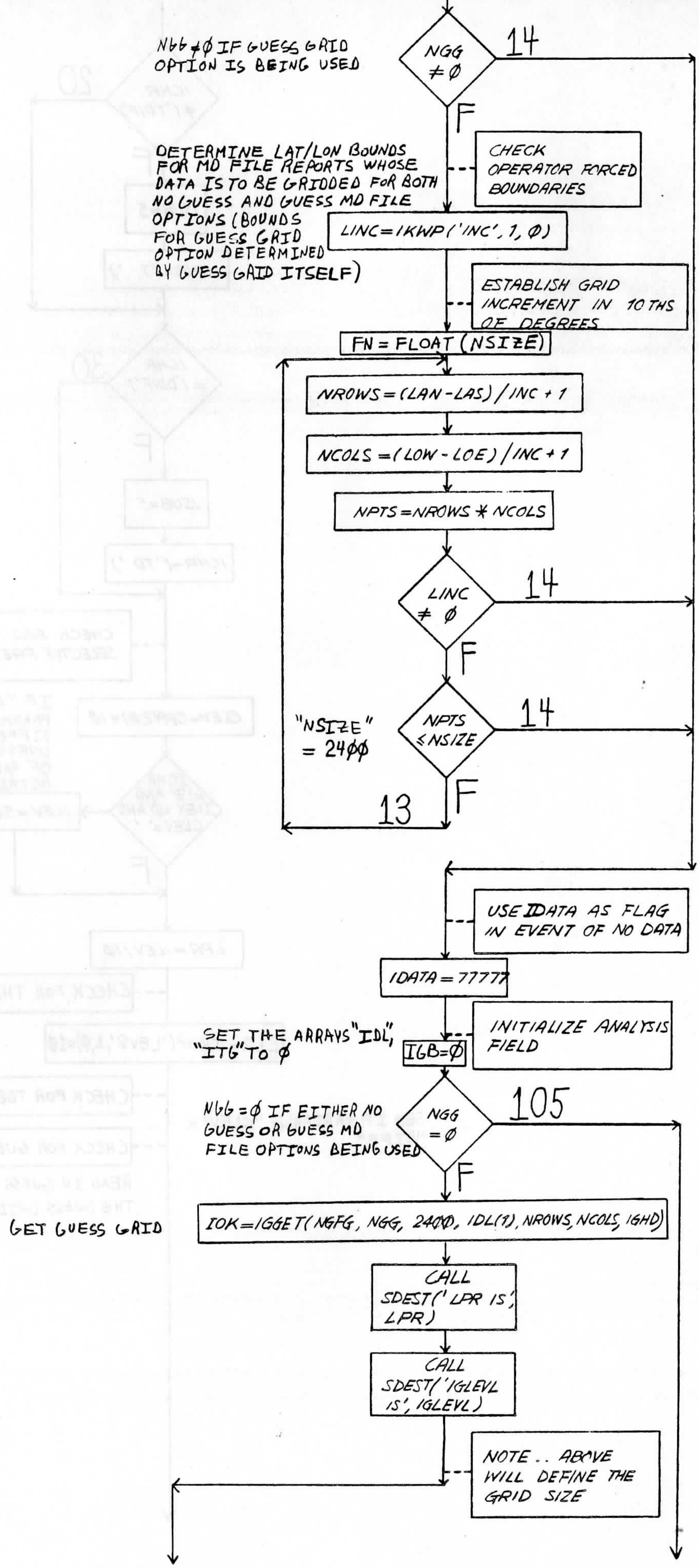

BNVA
DATA MISS/Z80808080/,NSIZE/2400/,NBMAX/5000/
159 IS MAX NO. OF GRIDS PERMITTED BY IG SOFTWARE
ABOVE FORCES OUTPUT GRID NO.
KBUG=IKWP('BUG',1,0)
SET DEBUG OPTION
KBUG ≥ 0
CALL SDEST(' GO !!!!!!!',0)
F
PICK UP CONTEXT INFORMATION
LUC(81) ≠ 0
111
CALL DOPEN(DLIT(MFILE, LUN, LEN)
F
CALL DREAD(LUN, ITERM, IDOC)
OPEN, READ AND CLOSE VASTEXT FILE
CALL TSNIO(1,1,1,1,1, 1, IDOC)
CALL DCLOSE(LUN)
MDNO = IKWP('MDNR',1,0)
SET MD FILE AND ROW NUMBERS, AS WELL AS DESTINATION GRIDFILE NUMBER USED WHEN USING GRID FIRST GUESS, AND GUESS MD FILE #
MDNO = 0
MDNO=IDOC(40)
F
SET CHARACTER TO BE GRIDDED
ICHR=CPP(1,'Z')
CHECK EXISTENCE OF MD FILE
MDINFO (MDNO,MDHD) ≠ 0
900
CALL SDEST(' CANNOT OPEN MDFILE NO.', MDNO)
LCHR=ICHR
END
SET LEVEL OF GRID
ILEV=IPP(2,0)*10
F
LPR=ILEV/10
OPEN MD FILE FOR READ/WRITE
MDOPEN (MDNO,2) ≠ 0
900
CALL SDEST(' CANNOT OPEN MDFILE NO. ', MDNO)
END
F
READ ROW HEADER RECORD
IOK=MDGET(MDNO,MDR,0,KBUF)

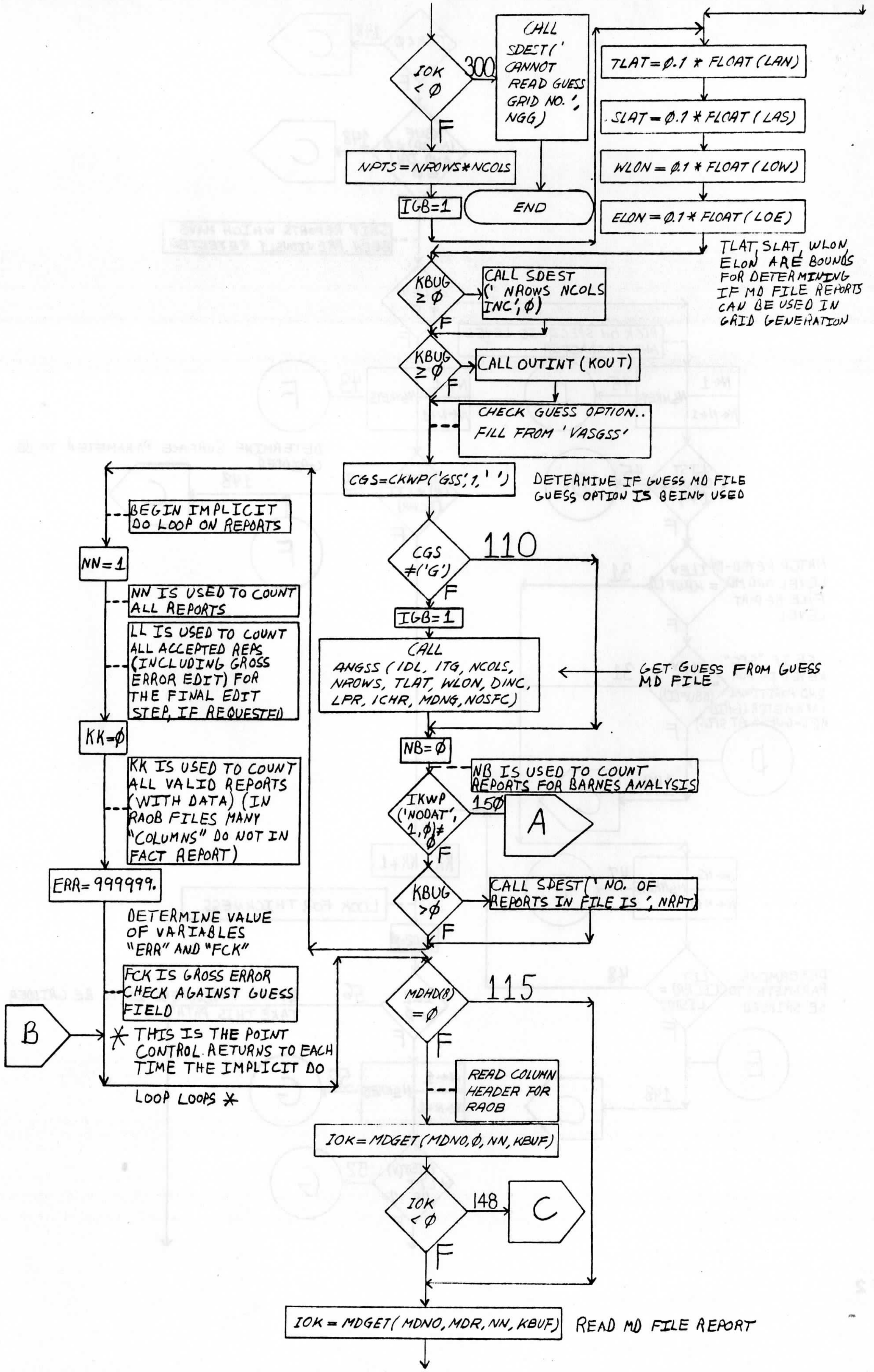


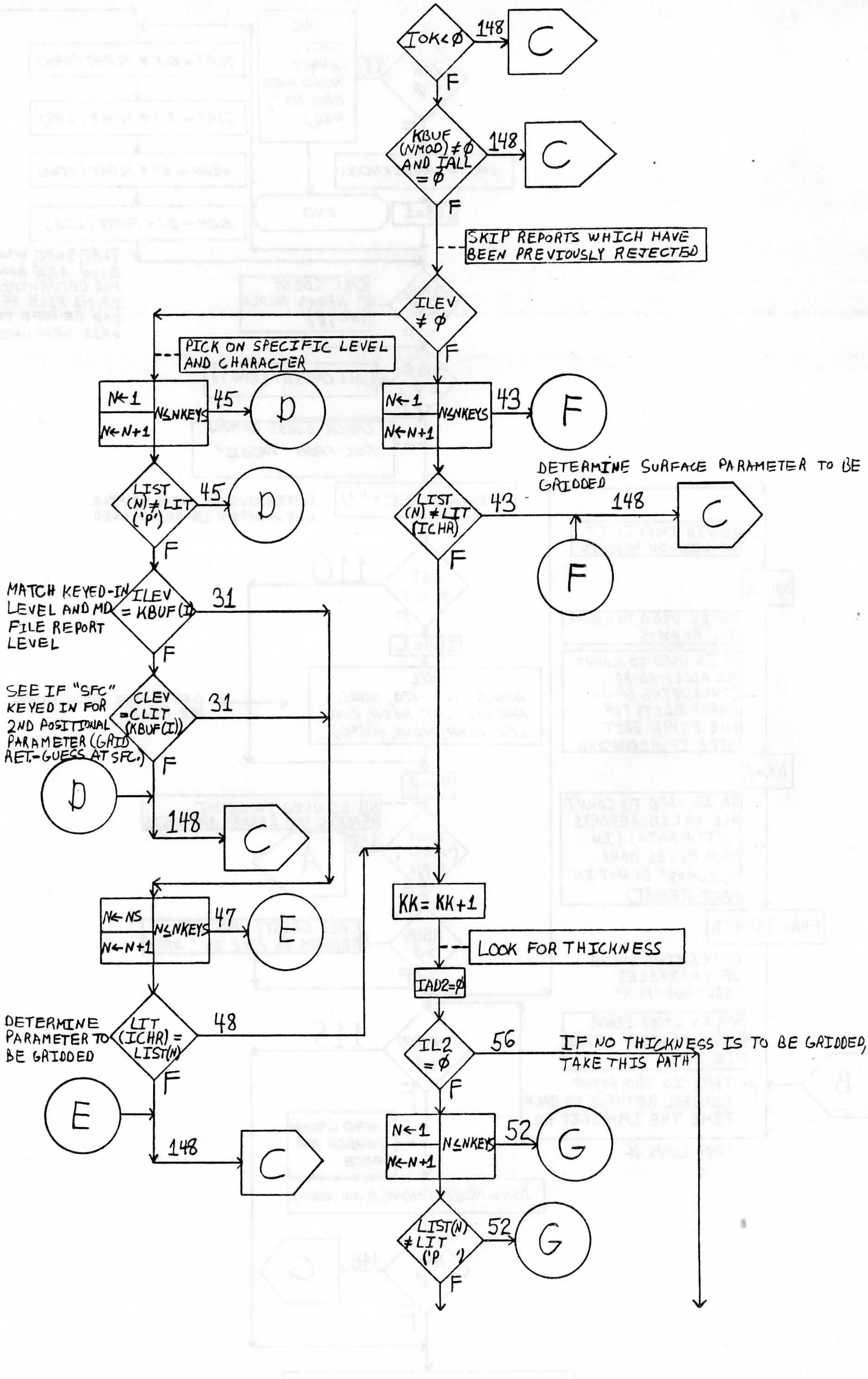
IOK < Ø
902
CALL SDEST (' CANNOT LOCATE ROW NO. ', MDR)
END
F
READ TEST RECORD
M=Ø
DETERMINE IF DATA EXISTS IN MD FILE
M=M+1
M>1ØØ
904
CALL SDEST (' TROUBLE READING DATA RECORD NO ', M)
END
F
READ COLUMN HEADER FOR RAOB
MDHD(8) ≠ Ø
IOK = MDGET(MDNO, Ø, M, KBUF)
F
KBUG >Ø
CALL SDEST (' MMAX IS ', MMAX)
F
IOK=MDGET(MDNO, MDR, M, KBUF)
READ MD FILE REPORT
IOK < Ø
1
F
MMAX IS MAX POSSIBLE DATA ENTRIES, USED TO AVOID INFINITE LOOPING AT SN 148
MMAX=MDHD(5)
MMAX IS MAXIMUM POSSIBLE MD FILE REPORTS IN ANY ROW OF THE FILE
FIND ADDRESS FOR LAT AND LON
READ IN KEYS FOR MD FILE
SAVE REPRESENTATIVE ENTRIES FOR NODATA CASE
DO LOOP 2
DETERMINE # OF REPORTS TO READ, AS WELL AS BIAS
CHECK ON KEYIN OVERRIDE
DETERMINE VALUE OF VARIABLE "ISUB"
ISUB=Ø

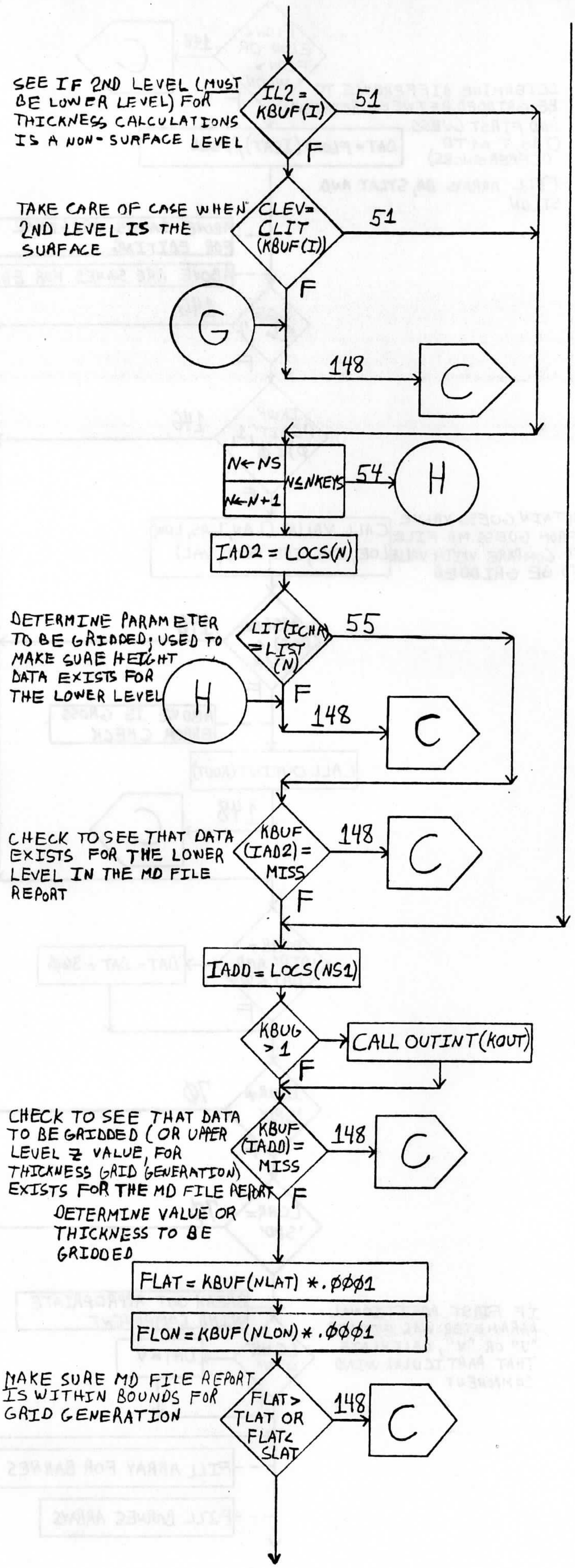


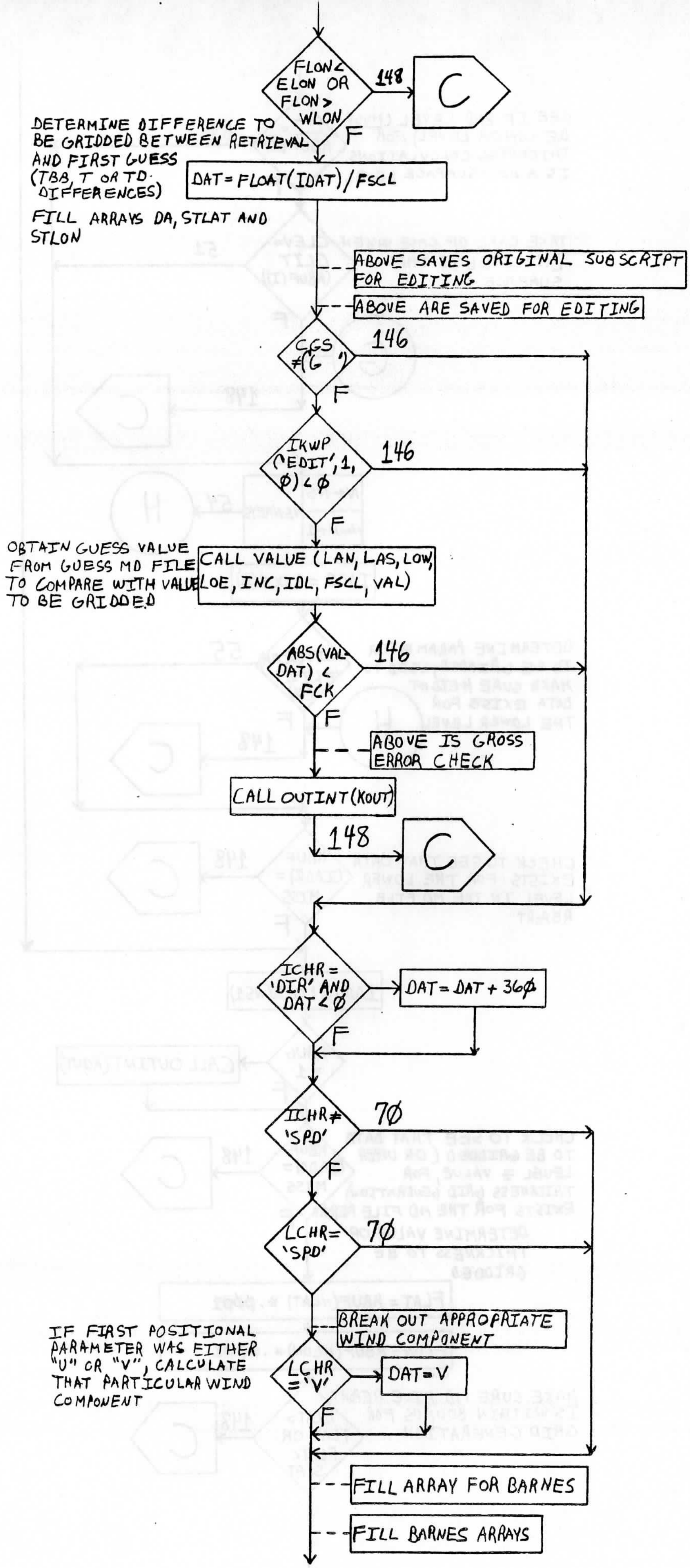
ICHR ≠('TDIF')
20
F
ISUB=5
ICHR=('T ')
ICHR ≠('DDIF')
30
F
ISUB=5
ICHR=('TD ')
CHECK FOR SELECTIVE PRESSURE
CLEV=CPP(2,Ø)*1Ø
IF "DIFF" KEYED IN FOR 2ND POSITIONAL PARAMETER, GRID WILL BE OF TBB DIFFERENCES BETWEEN RETRIEVAL AND GUESS; IF "SFC" KEYED IN, GRID WILL BE OF PARAMETER DIFFERENCES BETWEEN RETRIEVAL AND GUESS AT THE SURFACE
ICHR ='Z' AND ILEV =Ø AND CLEV=' '
ILEV=5ØØØ
F
LPR=ILEV/1Ø
CHECK FOR THICKNESS CALCULATION
IL2=IKWP('LEV2',1,Ø)*1Ø
CHECK FOR TBB DIFF
SEE IF VARIABLE "CLEV" = "DIFF"
CHECK FOR GUESS GRID
READ IN GUESS GRID, IN THE EVENT THE GUESS GRID OPTION IS BEING USED

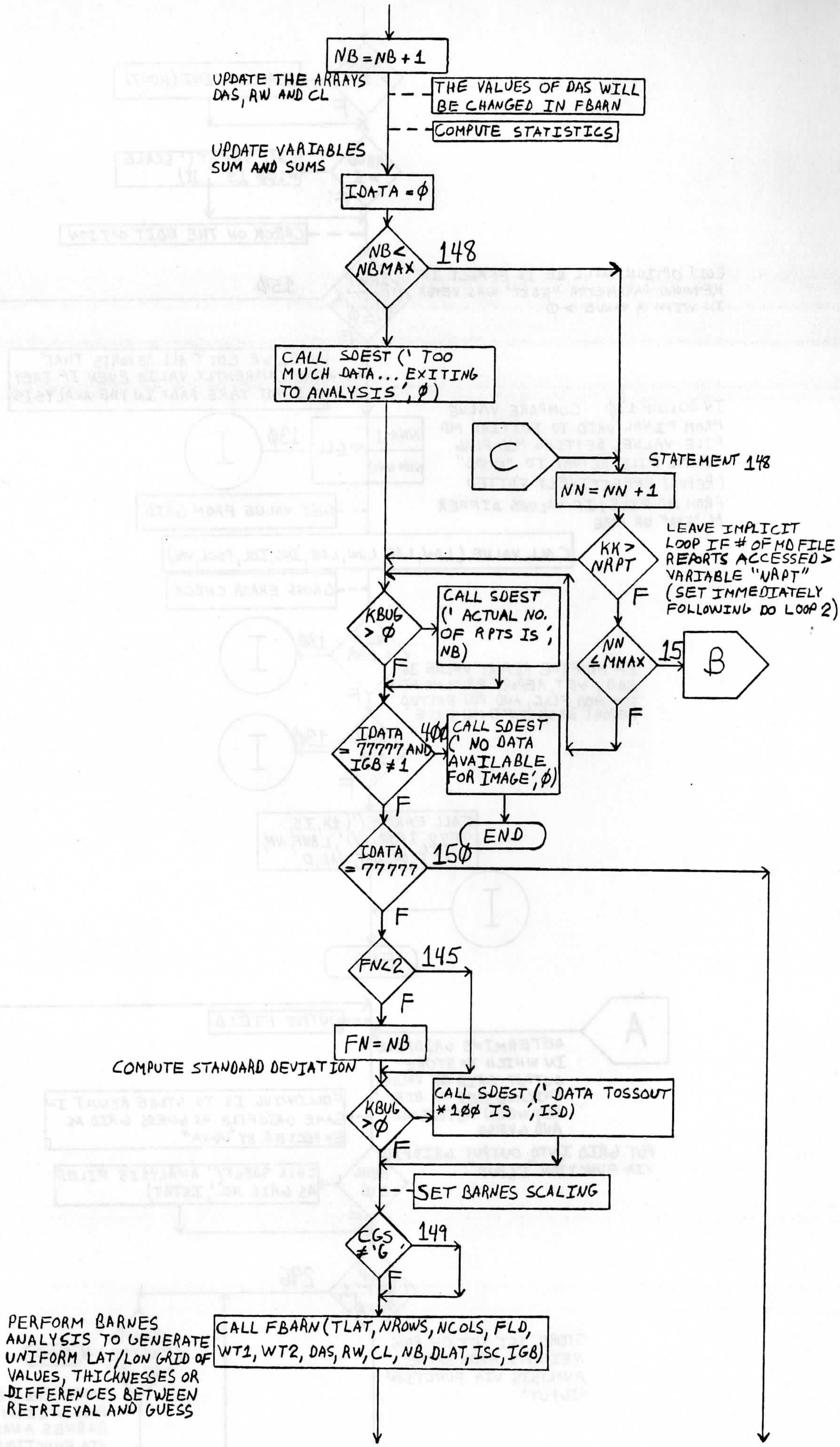

NGG≠Ø IF GUESS GRID OPTION IS BEING USED
NGG ≠ Ø
14
F
DETERMINE LAT/LON BOUNDS FOR MD FILE REPORTS WHOSE DATA IS TO BE GRIDDED FOR BOTH NO GUESS AND GUESS MD FILE OPTIONS (BOUNDS FOR GUESS GRID OPTION DETERMINED BY GUESS GRID ITSELF)
CHECK OPERATOR FORCED BOUNDARIES
LINC=IKWP('INC', 1, Ø)
ESTABLISH GRID INCREMENT IN 10 THS OF DEGREES
FN = FLOAT (NSIZE)
NROWS=(LAN-LAS)/INC+1
NCOLS=(LOW-LOE)/INC+1
NPTS=NROWS * NCOLS
LINC ≠ Ø
14
F
"NSIZE" = 24ØØ
NPTS ≤NSIZE
14
F
13
USE IDATA AS FLAG IN EVENT OF NO DATA
IDATA=77777
INITIALIZE ANALYSIS FIELD
SET THE ARRAYS "IDL", "ITG" TO Ø
IGB=Ø
NGG=Ø IF EITHER NO GUESS OR GUESS MD FILE OPTIONS BEING USED
NGG =Ø
105
F
GET GUESS GRID
IOK=IGGET(NGFG, NGG, 24ØØ, IDL(1), NROWS, NCOLS, IGHD)
CALL SDEST('LPR IS', LPR)
CALL SDEST('IGLEVL IS', IGLEVL)
NOTE .. ABOVE WILL DEFINE THE GRID SIZE

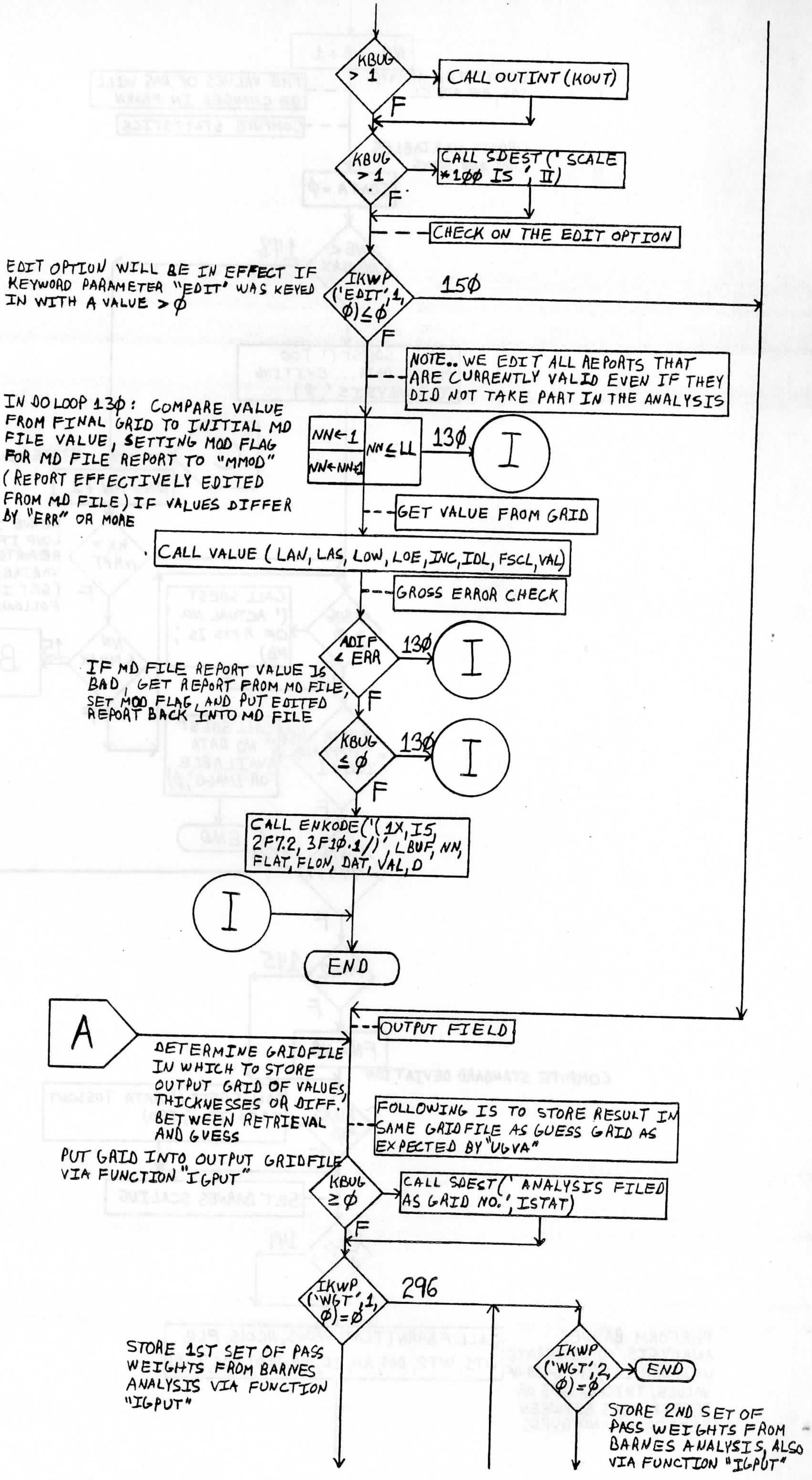

IOK < Ø
300
CALL SDEST(' CANNOT READ GUESS GRID NO. ', NGG)
F
NPTS=NROWS*NCOLS
IGB=1
END
TLAT=Ø.1 * FLOAT(LAN)
SLAT=Ø.1 * FLOAT(LAS)
WLON=Ø.1 * FLOAT(LOW)
ELON=Ø.1 * FLOAT(LOE)
TLAT, SLAT, WLON, ELON ARE BOUNDS FOR DETERMINING IF MD FILE REPORTS CAN BE USED IN GRID GENERATION
KBUG ≥ Ø
CALL SDEST(' NROWS NCOLS INC', Ø)
F
KBUG ≥ Ø
CALL OUTINT(KOUT)
F
CHECK GUESS OPTION.. FILL FROM 'VASGSS'
CGS=CKWP('GSS', 1, ' ')
DETERMINE IF GUESS MD FILE GUESS OPTION IS BEING USED
CGS ≠('G')
110
F
IGB=1
CALL ANGSS(IDL, ITG, NCOLS, NROWS, TLAT, WLON, DINC, LPR, ICHR, MDNG, NOSFC)
GET GUESS FROM GUESS MD FILE
NB=Ø
NB IS USED TO COUNT REPORTS FOR BARNES ANALYSIS
IKWP('NODAT', 1, Ø) ≠ Ø
15Ø
A
F
KBUG > Ø
CALL SDEST(' NO. OF REPORTS IN FILE IS ', NRPT)
F
BEGIN IMPLICIT DO LOOP ON REPORTS
NN=1
NN IS USED TO COUNT ALL REPORTS
LL IS USED TO COUNT ALL ACCEPTED REPS (INCLUDING GROSS ERROR EDIT) FOR THE FINAL EDIT STEP, IF REQUESTED
KK=Ø
KK IS USED TO COUNT ALL VALID REPORTS (WITH DATA) (IN RAOB FILES MANY "COLUMNS" DO NOT IN FACT REPORT)
ERR=999999.
DETERMINE VALUE OF VARIABLES "ERR" AND "FCK"
FCK IS GROSS ERROR CHECK AGAINST GUESS FIELD
B
* THIS IS THE POINT CONTROL RETURNS TO EACH TIME THE IMPLICIT DO LOOP LOOPS *
MDHD(8) = Ø
115
F
READ COLUMN HEADER FOR RAOB
IOK=MDGET(MDNO, Ø, NN, KBUF)
IOK < Ø
148
C
F
IOK = MDGET(MDNO, MDR, NN, KBUF)
READ MD FILE REPORT


IOK<Ø
148
C
F
KBUF (NMOD)≠Ø AND IALL =Ø
148
C
F
SKIP REPORTS WHICH HAVE BEEN PREVIOUSLY REJECTED
ILEV ≠ Ø
F
PICK ON SPECIFIC LEVEL AND CHARACTER
N←1
N←N+1
N≤NKEYS
45
D
N←1
N←N+1
N≤NKEYS
43
F
DETERMINE SURFACE PARAMETER TO BE GRIDDED
LIST (N)≠LIT ('P')
45
D
F
LIST (N)≠LIT (ICHR)
43
148
C
F
F
MATCH KEYED-IN LEVEL AND MD FILE REPORT LEVEL
ILEV = KBUF(I)
31
F
SEE IF "SFC" KEYED IN FOR 2ND POSITIONAL PARAMETER (GRID RET.-GUESS AT SFC.)
CLEV =CLIT (KBUF(I))
31
F
D
148
C
N←NS
N←N+1
N≤NKEYS
47
E
KK=KK+1
LOOK FOR THICKNESS
IAD2=Ø
DETERMINE PARAMETER TO BE GRIDDED
LIT (ICHR)= LIST(N)
48
F
IL2 =Ø
56
IF NO THICKNESS IS TO BE GRIDDED, TAKE THIS PATH
F
E
148
C
N←1
N←N+1
N≤NKEYS
52
G
LIST(N) ≠LIT ('P ')
52
G
F

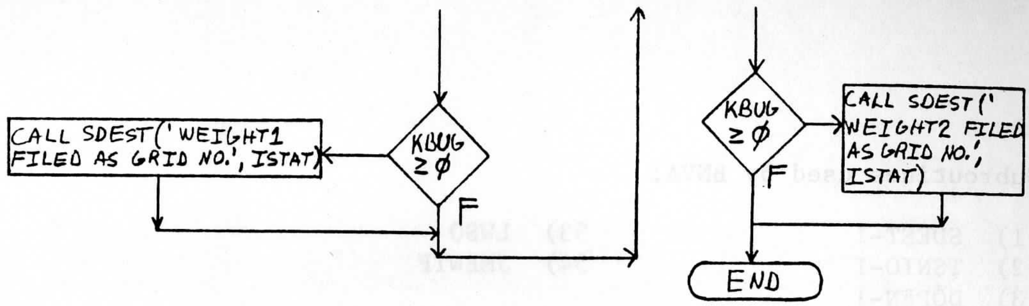

SEE IF 2ND LEVEL (MUST BE LOWER LEVEL) FOR THICKNESS CALCULATIONS IS A NON-SURFACE LEVEL
IL2 = KBUF(I)
51
F
TAKE CARE OF CASE WHEN 2ND LEVEL IS THE SURFACE
CLEV= CLIT (KBUF(I))
51
F
G
148
C
N←NS
N←N+1
N≤NKEYS
54
H
IAD2 = LOCS(N)
DETERMINE PARAMETER TO BE GRIDDED; USED TO MAKE SURE HEIGHT DATA EXISTS FOR THE LOWER LEVEL
LIT(ICHA) =LIST (N)
55
F
H
148
C
CHECK TO SEE THAT DATA EXISTS FOR THE LOWER LEVEL IN THE MD FILE REPORT
KBUF (IAD2) = MISS
148
C
F
IADD = LOCS(NS1)
KBUG > 1
CALL OUTINT(KOUT)
F
CHECK TO SEE THAT DATA TO BE GRIDDED (OR UPPER LEVEL Z VALUE, FOR THICKNESS GRID GENERATION) EXISTS FOR THE MD FILE REPORT
KBUF (IADD)= MISS
148
C
F
DETERMINE VALUE OR THICKNESS TO BE GRIDDED
FLAT = KBUF(NLAT) *.ØØØ1
FLON = KBUF(NLON) *.ØØØ1
MAKE SURE MD FILE REPORT IS WITHIN BOUNDS FOR GRID GENERATION
FLAT> TLAT OR FLAT< SLAT
148
C


FLON < ELON OR FLON > WLON
148
C
F
DETERMINE DIFFERENCE TO BE GRIDDED BETWEEN RETRIEVAL AND FIRST GUESS (TBB, T OR TD DIFFERENCES)
DAT = FLOAT(IDAT)/FSCL
FILL ARRAYS DA, STLAT AND STLON
ABOVE SAVES ORIGINAL SUBSCRIPT FOR EDITING
ABOVE ARE SAVED FOR EDITING
CGS ≠ 'G'
146
F
IKWP ('EDIT', 1, Ø) < Ø
146
F
OBTAIN GUESS VALUE FROM GUESS MD FILE TO COMPARE WITH VALUE TO BE GRIDDED
CALL VALUE (LAN, LAS, LOW, LOE, INC, IDL, FSCL, VAL)
ABS(VAL-DAT) < FCK
146
F
ABOVE IS GROSS ERROR CHECK
CALL OUTINT(KOUT)
148
C
ICHR = 'DIR' AND DAT < Ø
DAT = DAT + 36Ø
F
ICHR ≠ 'SPD'
7Ø
F
LCHR = 'SPD'
7Ø
BREAK OUT APPROPRIATE WIND COMPONENT
IF FIRST POSITIONAL PARAMETER WAS EITHER "U" OR "V", CALCULATE THAT PARTICULAR WIND COMPONENT
LCHR = 'V'
DAT = V
F
FILL ARRAY FOR BARNES
FILL BARNES ARRAYS


NB = NB + 1
UPDATE THE ARRAYS DAS, AW AND CL
THE VALUES OF DAS WILL BE CHANGED IN FBARN
COMPUTE STATISTICS
UPDATE VARIABLES SUM AND SUMS
IDATA = ϕ
NB < NBMAX
148
CALL SDEST (' TOO MUCH DATA... EXITING TO ANALYSIS', ϕ)
C
STATEMENT 148
NN = NN + 1
KK > NRPT
LEAVE IMPLICIT LOOP IF # OF MD FILE REPORTS ACCESSED > VARIABLE "NRPT" (SET IMMEDIATELY FOLLOWING DO LOOP 2)
F
KBUG > ϕ
CALL SDEST (' ACTUAL NO. OF RPTS IS ', NB)
F
NN ≤ MMAX
15
B
F
IDATA = 77777 AND IGB ≠ 1
4ϕϕ
CALL SDEST (' NO DATA AVAILABLE FOR IMAGE', ϕ)
F
END
IDATA = 77777
15ϕ
F
FN < 2
145
F
FN = NB
COMPUTE STANDARD DEVIATION
KBUG > ϕ
CALL SDEST (' DATA TOSSOUT * 1ϕϕ IS ', ISD)
F
SET BARNES SCALING
CGS ≠ 'G
149
F
PERFORM BARNES ANALYSIS TO GENERATE UNIFORM LAT/LON GRID OF VALUES, THICKNESSES OR DIFFERENCES BETWEEN RETRIEVAL AND GUESS
CALL FBARN (TLAT, NROWS, NCOLS, FLD, WT1, WT2, DAS, RW, CL, NB, DLAT, ISC, IGB)



KBUG > 1
CALL OUTINT (KOUT)
F
KBUG > 1
CALL SDEST (' SCALE *1ØØ IS ', II)
F
CHECK ON THE EDIT OPTION
EDIT OPTION WILL BE IN EFFECT IF KEYWORD PARAMETER "EDIT" WAS KEYED IN WITH A VALUE > Ø
IKWP ('EDIT', 1, Ø) ≤ Ø
15Ø
F
NOTE.. WE EDIT ALL REPORTS THAT ARE CURRENTLY VALID EVEN IF THEY DID NOT TAKE PART IN THE ANALYSIS
IN DO LOOP 13Ø: COMPARE VALUE FROM FINAL GRID TO INITIAL MD FILE VALUE, SETTING MOD FLAG FOR MD FILE REPORT TO "MMOD" (REPORT EFFECTIVELY EDITED FROM MD FILE) IF VALUES DIFFER BY "ERR" OR MORE
NN←1
NN≤LL
NN←NN+1
13Ø
I
GET VALUE FROM GRID
CALL VALUE (LAN, LAS, LOW, LOE, INC, IDL, FSCL, VAL)
GROSS ERROR CHECK
ADIF < ERR
13Ø
I
IF MD FILE REPORT VALUE IS BAD, GET REPORT FROM MD FILE, SET MOD FLAG, AND PUT EDITED REPORT BACK INTO MD FILE
F
KBUG ≤ Ø
13Ø
I
F
CALL ENKODE ('(1X, I5, 2F7.2, 3F1Ø.1)', LBUF, NN, FLAT, FLON, DAT, VAL, D
I
END
A
OUTPUT FIELD
DETERMINE GRIDFILE IN WHICH TO STORE OUTPUT GRID OF VALUES, THICKNESSES OR DIFF. BETWEEN RETRIEVAL AND GUESS
FOLLOWING IS TO STORE RESULT IN SAME GRIDFILE AS GUESS GRID AS EXPECTED BY "UGVA"
PUT GRID INTO OUTPUT GRIDFILE VIA FUNCTION "IGPUT"
KBUG ≥ Ø
CALL SDEST (' ANALYSIS FILED AS GRID NO.', ISTAT)
F
IKWP ('WGT', 1, Ø) = Ø
296
STORE 1ST SET OF PASS WEIGHTS FROM BARNES ANALYSIS VIA FUNCTION "IGPUT"
IKWP ('WGT', 2, Ø) = Ø
END
STORE 2ND SET OF PASS WEIGHTS FROM BARNES ANALYSIS, ALSO VIA FUNCTION "IGPUT"


CALL SDEST('WEIGHT1 FILED AS GRID NO.', ISTAT)
KBUG ≥ Ø
F
KBUG ≥ Ø
CALL SDEST('WEIGHT2 FILED AS GRID NO.', ISTAT)
F
END

Subroutines used by BNVA:

1) SDEST-I
2) TSNIO-I
3) DOPEN-I
4) DREAD-I
5) DCLOSE-I
6) OUTINT-I
7) ANGSS-I
8) FBARN-I
9) VALUE-I
10) ENKODE-I
11) BLKA
12) MOVCW
13) MOVB
14) LTQ
15) CLEANA
16) TQ
17) PRLINX
18) PRCLOS
19) PROPEN
20) PRPRPR
21) LOCK
22) PRWR
23) PRRD
24) UNLOCK
25) PRCL
26) POST
27) ABORT
28) II
29) STC
30) ENCODE
31) ENCODX
32) CONTNT
33) ZECONV
34) DWRITE
35) LWCLOS
36) MOVWC
37) LWMOP
38) ITOC
39) VASGES
40) SURGES
41) GETSFV
42) IGNAME
43) WMIX
44) FILL
45) LINFIL
46) INTER
47) HEAPFY
48) LWPO
49) EDEST
50) MOVW
51) CLEANW
52) LWNEWF
53) LWSO
54) JMBWTF

GWVA

Program GWVA is used to produce wind vectors from the previously-calculated height grids generated by program BNVA. In addition, a plot of the resultant wind vectors is included as an option.

The first things done in GWVA are to initialize the plot routines using subroutine INITPL, set the plot option (IPLT), and open/read the VASTEXT file. Assuming the plot option is in effect (IPLT=0), subroutine TVSAT returns variables ISS and JD, which are required for the navigation initialization process, after which the navigation file is opened via subroutine NVINIT. Then, variables MDNO and MDR are set to the retrieval MD file and row numbers, respectively, as listed in the VASTEXT file. After the retrieval MD file is opened for read/write, the wind type to be calculated is determined by reading in the value associated with keyword parameter TYPE. There are four types of wind calculations possible with GWVA: GR (gradient), G (geostrophic), AG (ageostrophic), and IS (isallobaric). If no wind type is keyed in, the program will produce gradient winds by default.

In the following code, information such as color of plotted barbs (variable KOL--default = 1 (red)), grid file (NGRF), and grid number within the gridfile (NGR) are set, after which the height grid from which the winds are to be produced is accessed via function IGGET. Note that the gridfile from which the height grid is to be taken must be set before GWVA is executed. This can be done by using system program IGU ("IGU SET 4190," for example). Further code assigns the grid pressure level (LEVP).

Assuming LEVP has been keyed in, it must be the same as the level of the grid just read from the gridfile (LEVEL), or GWVA will terminate. The value of LEVP will later determine whether the final wind data will be stored in the retrieval MD file (if LEVP is .NE. 0, data stored, and vice-versa). Following this, a check is performed in DO LOOP 14 to see if the keyed-in level is a mandatory level.

At this point, the lat/lon grid boundaries (LAN, LOW, LAS and LOE) and grid increment (INC) are determined, as well as the maximum and minimum latitude and longitude values for which winds can be computed from the height grid (LATMAX, LONMAX, LATMIN and LONMIN).

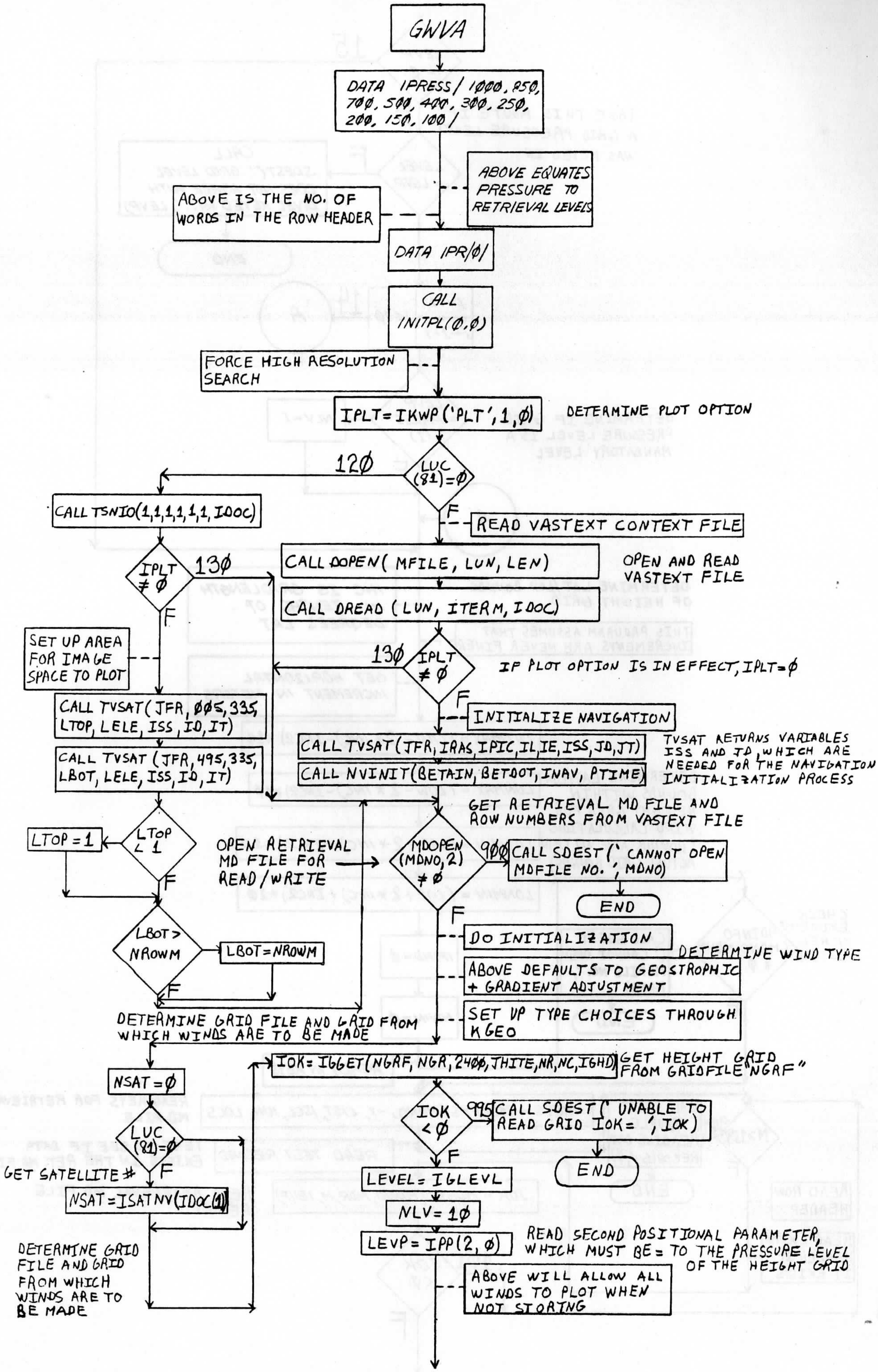
In the next important step, the existence of the retrieval MD file is checked via function MDINFO, after which the keys for the file are read via function MDKEYS. Following a test designed to make sure data exists in the retrieval MD file, during which the row header of the retrieval file corresponding to row MDR is read, program GWVA enters DO LOOP 11. In this loop, the addresses within the keylist corresponding to such things as number of reports within a given retrieval MD file row, retrieval latitude, and retrieval longitude are determined. Some of this information is then used to assign a value to variable LAST.

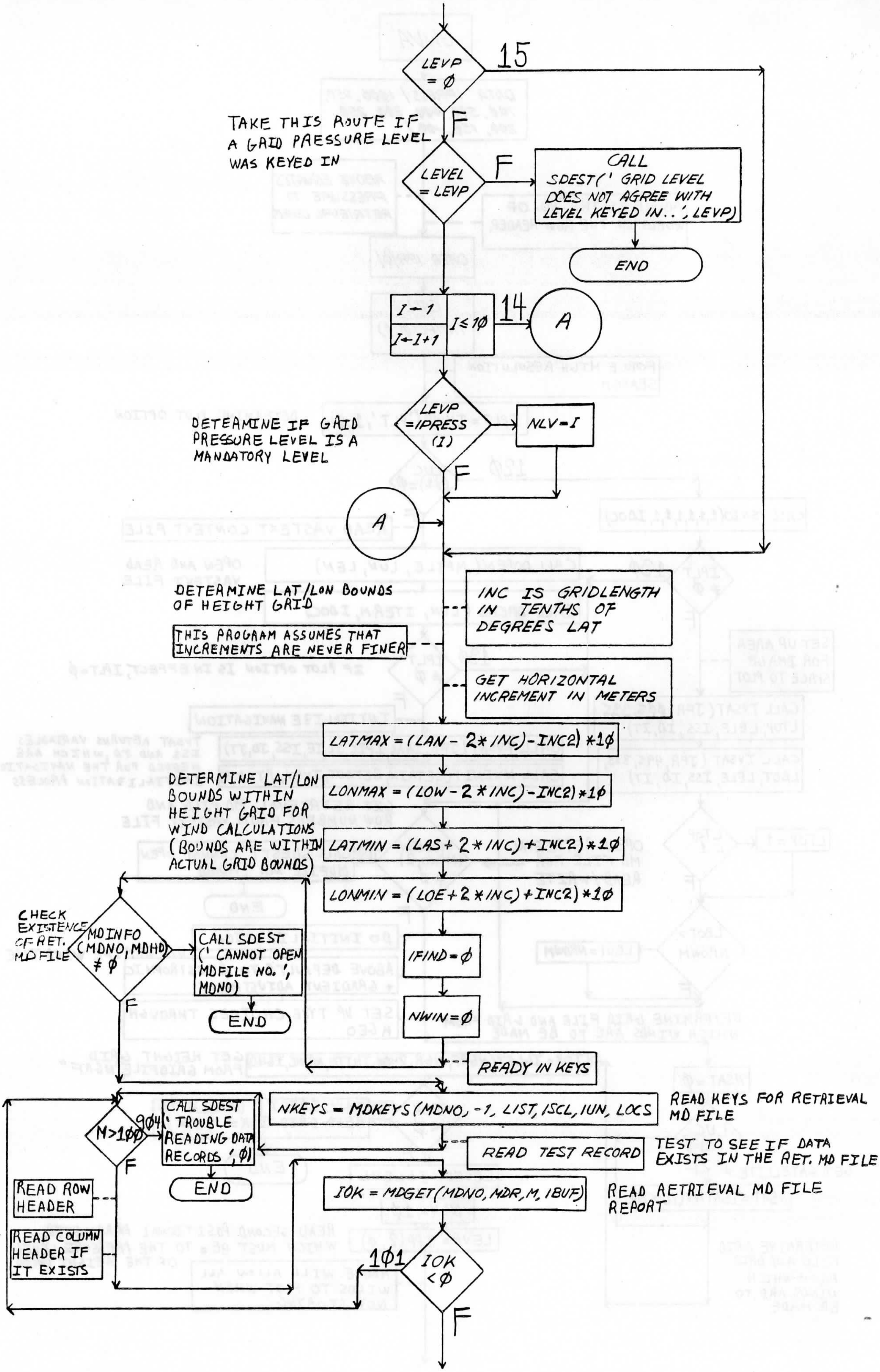
Assuming LAST now has a value .NE. 0, variables DELT and NGR are set. If the user wishes to produce either isallobaric or ageostrophic winds, he/she should key in keyword parameters DELT and NGOLD with non-zero values, since both of these winds require changes of wind over a period of time or, in other words, another earlier height grid. If indeed one of these wind options is desired, the earlier height grid is read by function IGGET ((IF DELT .NE. 0) IOK=IGGET(NGRF,NGR,2400,JHIT2,NR,NC,IGHD)). Finally, GWVA enters DO LOOP 200. The purpose of this large loop is to calculate the desired winds at each retrieval location within the grid boundaries, and plot the results if the plot option is in effect. After the retrieval report is read via function MDGET (IOK=MDGET(MDNO,MDR,N,IBUF)), GWVA enters DO LOOP 110. This loop finds the address within the keyword list which corresponds to surface pressure (to be used subsequently when accessing a given retrieval's surface pressure from the array IBUF), and also computes the value of variable NWIN, which will be used later near the end of GWVA to determine into which words of IBUF the final wind direction and speed will be placed. Note that if NWIN is never set .NE. 0 (i.e., no retrieval pressure level matches the grid pressure level), but the grid pressure level (LEVP) WAS keyed in, no wind will be calculated at that particular retrieval location. In addition, the winds will not be calculated if either the pressure level of the wind is greater than the retrieval surface pressure (i.e., the wind is below ground), or if the retrieval location is outside the previously-determined wind calculation boundaries for the grid (LATMAX, LONMAX, LATMIN, LONMIN).

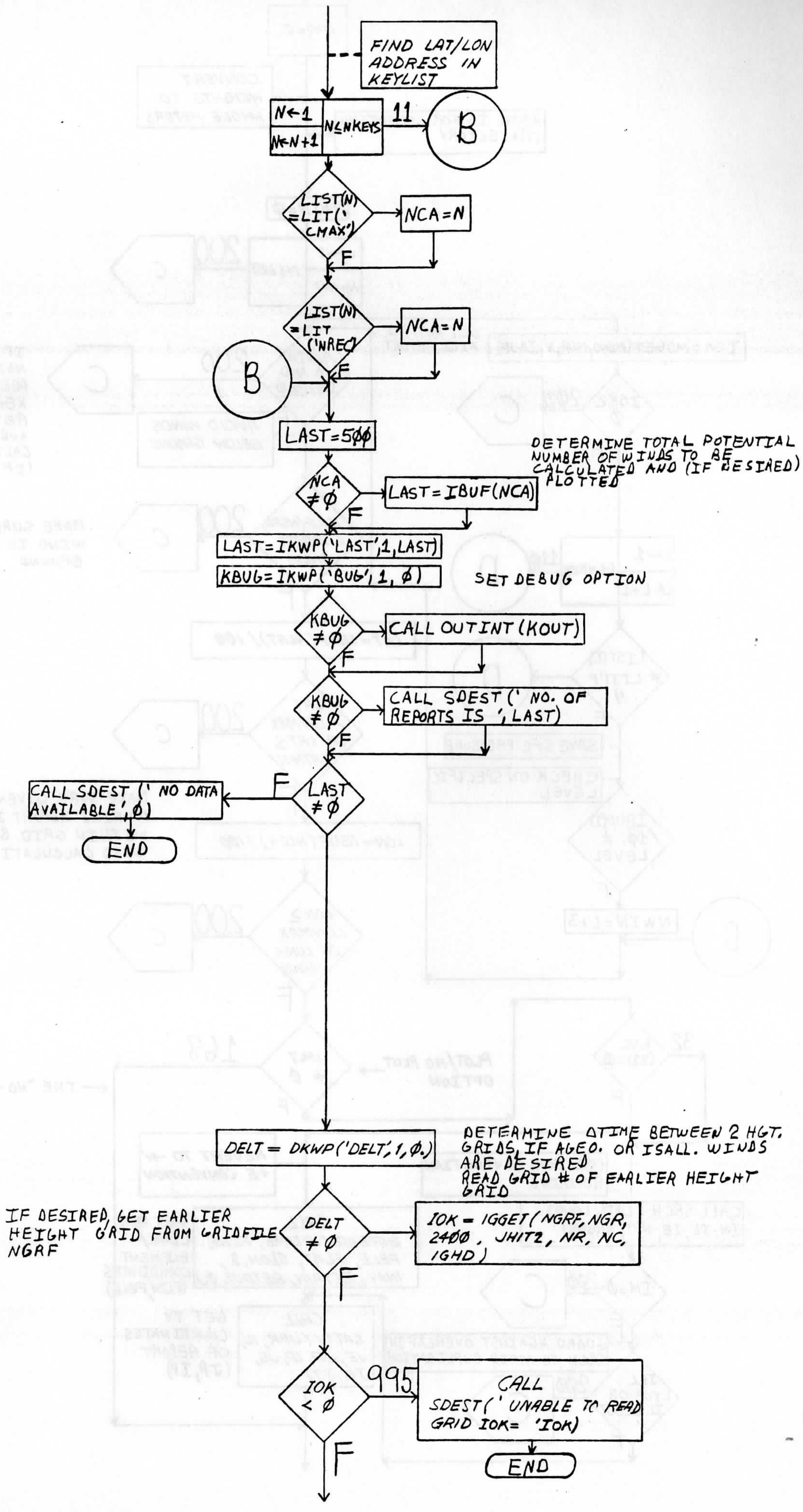
Beyond this point in GWVA, the discussion will be divided into two categories: plot option and no plot option.

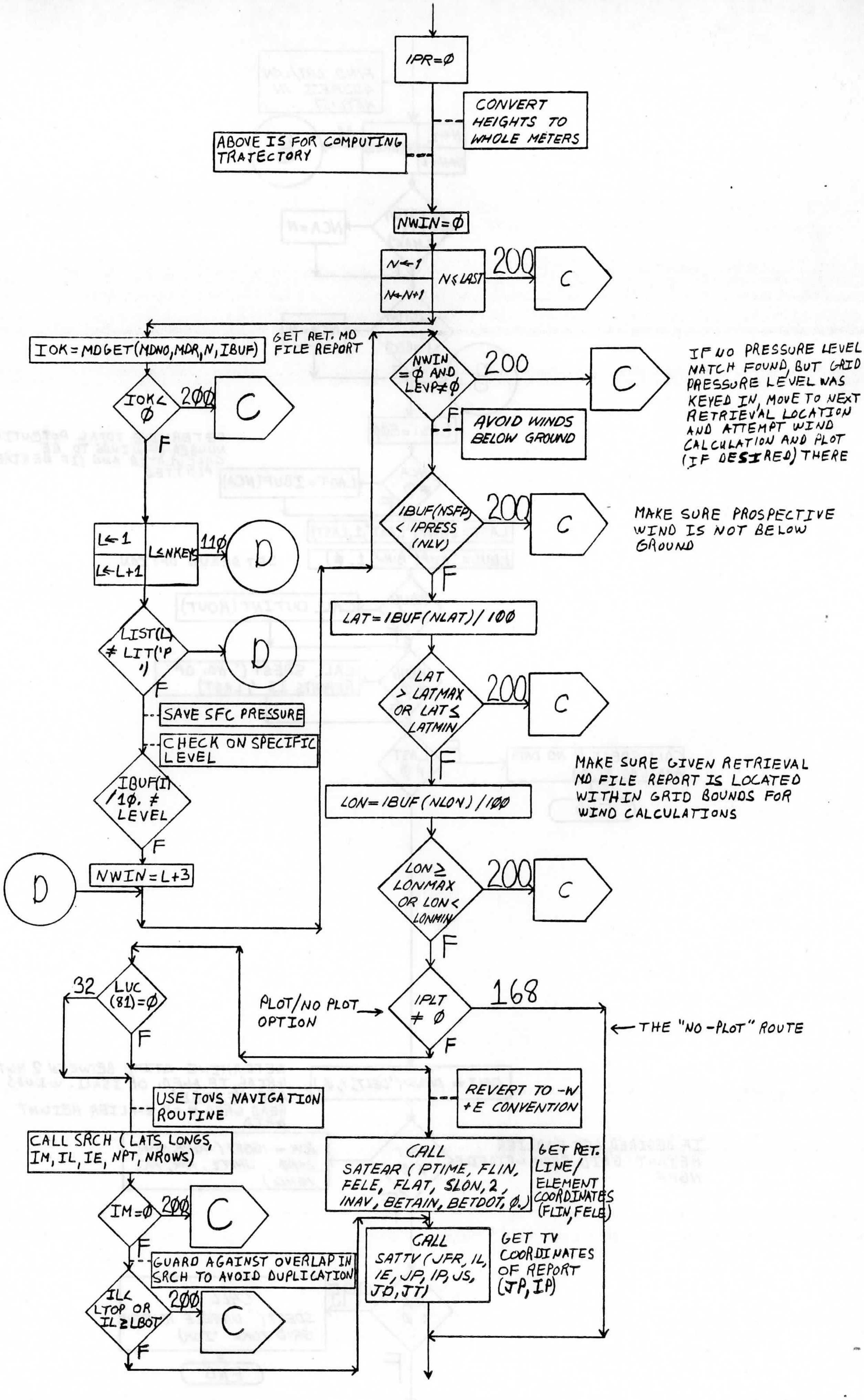
First, if the plot option is in effect (IPLT=0), GWVA calculates TV coordinates (raster (JP), pixel (IP)), first by converting retrieval latitude/longitude to satellite coordinates (FLIN/FELE) via subroutine SATEAR, and then in turn converting these coordinates to the desired TV coordinates via subroutine SATTV. Following this, the desired wind is calculated at grid location (I,J) by subroutine ZWIND. Then, the wind barb itself is plotted on the video (TV) screen by subroutine BARB, after the wind barb's direction has been adjusted by DIRADJ (am assuming here that NSAT is .NE. 0). Finally, if a level was keyed in earlier as the second positional parameter (LEVP .NE. 0), the wind direction and speed are put into the retrieval MD file via function MDPUT (IOK=MDPUT(MDNO,MDR,N,IBUF), after which DO LOOP 200 loops back to process another wind at a new retrieval location. On the other hand, if no level was keyed in, none of the wind results will be put into the retrieval MD file.

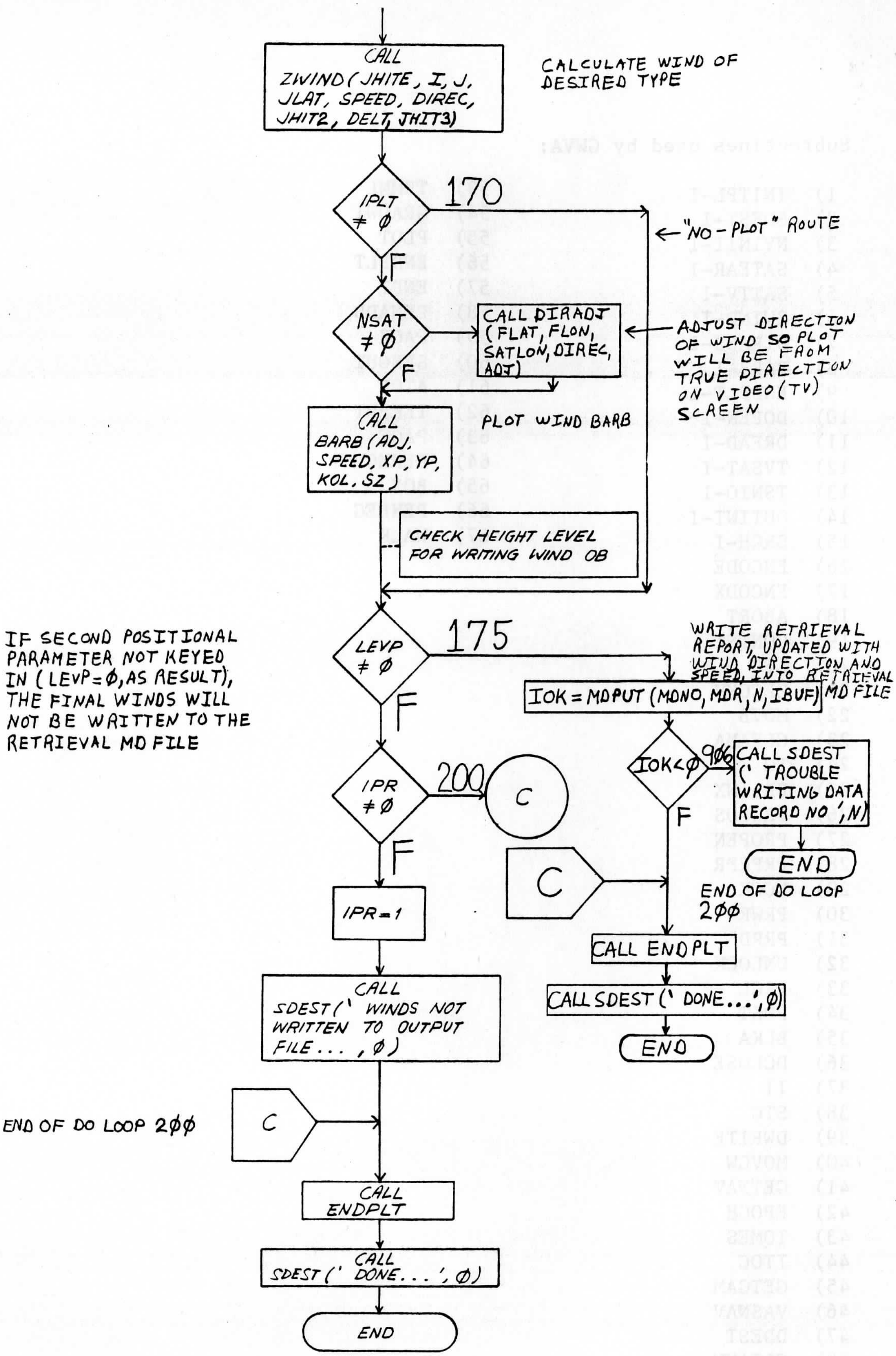
Second, if the plot option is NOT in effect, GWVA immediately calculates the wind at the grid location. Then, after skipping to statement 170 (remember, IPLT .NE. 0), the program proceeds the same as the case described above.


GWVA
DATA IPRESS/ 1ØØØ, 85Ø, 7ØØ, 5ØØ, 4ØØ, 3ØØ, 25Ø, 2ØØ, 15Ø, 1ØØ /
ABOVE EQUATES PRESSURE TO RETRIEVAL LEVELS
ABOVE IS THE NO. OF WORDS IN THE ROW HEADER
DATA IPR/Ø/
CALL INITPL(Ø,Ø)
FORCE HIGH RESOLUTION SEARCH
IPLT=IKWP('PLT',1,Ø)
DETERMINE PLOT OPTION
12Ø
LUC (81)=Ø
F
READ VASTEXT CONTEXT FILE
CALL TSNIO(1,1,1,1,1,1,IDOC)
CALL DOPEN(MFILE, LUN, LEN)
OPEN AND READ VASTEXT FILE
IPLT ≠ Ø
13Ø
CALL DREAD (LUN, ITERM, IDOC)
SET UP AREA FOR IMAGE SPACE TO PLOT
13Ø
IPLT ≠ Ø
IF PLOT OPTION IS IN EFFECT, IPLT=Ø
INITIALIZE NAVIGATION
CALL TVSAT(JFR, ØØ5, 335, LTOP, LELE, ISS, ID, IT)
CALL TVSAT(JFR, IRAS, IPIC, IL, IE, ISS, JD, JT)
TVSAT RETURNS VARIABLES ISS AND JD, WHICH ARE NEEDED FOR THE NAVIGATION INITIALIZATION PROCESS
CALL TVSAT (JFR, 495, 335, LBOT, LELE, ISS, ID, IT)
CALL NVINIT(BETAIN, BETDOT, INAV, PTIME)
GET RETRIEVAL MD FILE AND ROW NUMBERS FROM VASTEXT FILE
LTOP = 1
LTOP < 1
OPEN RETRIEVAL MD FILE FOR READ/WRITE
MDOPEN (MDNO,2) ≠ Ø
9ØØ
CALL SDEST(' CANNOT OPEN MDFILE NO. ', MDNO)
END
LBOT > NROWM
LBOT=NROWM
DO INITIALIZATION
DETERMINE WIND TYPE
ABOVE DEFAULTS TO GEOSTROPHIC + GRADIENT ADJUSTMENT
SET UP TYPE CHOICES THROUGH KGEO
DETERMINE GRID FILE AND GRID FROM WHICH WINDS ARE TO BE MADE
IOK= IGGET(NGRF, NGR, 24ØØ, JHITE, NR, NC, IGHD)
GET HEIGHT GRID FROM GRIDFILE "NGRF"
NSAT=Ø
IOK < Ø
995
CALL SDEST(' UNABLE TO READ GRID IOK= ', IOK)
LUC (81)=Ø
GET SATELLITE #
NSAT=ISATNV(IDOC(1))
END
LEVEL= IGLEVL
NLV= 1Ø
LEVP= IPP(2, Ø)
READ SECOND POSITIONAL PARAMETER, WHICH MUST BE = TO THE PRESSURE LEVEL OF THE HEIGHT GRID
DETERMINE GRID FILE AND GRID FROM WHICH WINDS ARE TO BE MADE
ABOVE WILL ALLOW ALL WINDS TO PLOT WHEN NOT STORING


LEVP = ∅
15
F
TAKE THIS ROUTE IF A GRID PRESSURE LEVEL WAS KEYED IN
LEVEL = LEVP
F
CALL SDEST(' GRID LEVEL DOES NOT AGREE WITH LEVEL KEYED IN..', LEVP)
END
I←1
I≤1∅
I←I+1
14
A
DETERMINE IF GRID PRESSURE LEVEL IS A MANDATORY LEVEL
LEVP =IPRESS (I)
NLV=I
F
A
DETERMINE LAT/LON BOUNDS OF HEIGHT GRID
INC IS GRIDLENGTH IN TENTHS OF DEGREES LAT
THIS PROGRAM ASSUMES THAT INCREMENTS ARE NEVER FINER
GET HORIZONTAL INCREMENT IN METERS
LATMAX = (LAN - 2*INC) - INC2) *1∅
DETERMINE LAT/LON BOUNDS WITHIN HEIGHT GRID FOR WIND CALCULATIONS (BOUNDS ARE WITHIN ACTUAL GRID BOUNDS)
LONMAX = (LOW - 2*INC) - INC2) *1∅
LATMIN = (LAS + 2*INC) + INC2) *1∅
LONMIN = (LOE + 2*INC) + INC2) *1∅
CHECK EXISTENCE OF RET. MD FILE
MDINFO (MDNO, MDHD) ≠ ∅
CALL SDEST (' CANNOT OPEN MDFILE NO. ', MDNO)
END
F
IFIND=∅
NWIN=∅
READY IN KEYS
NKEYS = MDKEYS(MDNO, -1, LIST, ISCL, IUN, LOCS
READ KEYS FOR RETRIEVAL MD FILE
M>1∅∅
9∅4
CALL SDEST (' TROUBLE READING DATA RECORDS ',∅)
END
F
READ TEST RECORD
TEST TO SEE IF DATA EXISTS IN THE RET. MD FILE
READ ROW HEADER
READ COLUMN HEADER IF IT EXISTS
IOK = MDGET(MDNO, MDR, M, IBUF)
READ RETRIEVAL MD FILE REPORT
1∅1
IOK <∅
F


FIND LAT/LON ADDRESS IN KEYLIST
N←1
N←N+1
N≤NKEYS
11
B
LIST(N) =LIT('CMAX')
NCA=N
F
LIST(N) =LIT('NREC')
NCA=N
F
B
LAST=5ØØ
DETERMINE TOTAL POTENTIAL NUMBER OF WINDS TO BE CALCULATED AND (IF DESIRED) PLOTTED
NCA ≠Ø
LAST=IBUF(NCA)
F
LAST=IKWP('LAST',1,LAST)
KBUG=IKWP('BUG',1,Ø)
SET DEBUG OPTION
KBUG ≠Ø
CALL OUTINT(KOUT)
F
KBUG ≠Ø
CALL SDEST('NO. OF REPORTS IS ',LAST)
F
LAST ≠Ø
F
CALL SDEST(' NO DATA AVAILABLE',Ø)
END
DELT = DKWP('DELT',1,Ø.)
DETERMINE ΔTIME BETWEEN 2 HGT. GRIDS, IF AGEO. OR ISALL. WINDS ARE DESIRED
READ GRID # OF EARLIER HEIGHT GRID
IF DESIRED, GET EARLIER HEIGHT GRID FROM GRIDFILE NGRF
DELT ≠Ø
IOK = IGGET(NGRF,NGR, 24ØØ, JHIT2, NR, NC, IGHD)
F
IOK <Ø
995
CALL SDEST(' UNABLE TO READ GRID IOK= 'IOK)
END
F


IPR=Ø
CONVERT HEIGHTS TO WHOLE METERS
ABOVE IS FOR COMPUTING TRAJECTORY
NWIN=Ø
N←1
N←N+1
N≤LAST
200
C
IOK=MDGET(MDNO,MDR,N,IBUF)
GET RET. MD FILE REPORT
IOK<Ø
F
L←1
L←L+1
L≤NKEY
110
D
LIST(L) ≠ LIT('P ')
SAVE SFC PRESSURE
CHECK ON SPECIFIC LEVEL
IBUF(I) /10. ≠ LEVEL
NWIN=L+3
NWIN =Ø AND LEVP≠Ø
IF NO PRESSURE LEVEL MATCH FOUND, BUT GRID PRESSURE LEVEL WAS KEYED IN, MOVE TO NEXT RETRIEVAL LOCATION AND ATTEMPT WIND CALCULATION AND PLOT (IF DESIRED) THERE
AVOID WINDS BELOW GROUND
IBUF(NSFP) < IPRESS (NLV)
MAKE SURE PROSPECTIVE WIND IS NOT BELOW GROUND
LAT=IBUF(NLAT)/100
LAT > LATMAX OR LAT≤ LATMIN
MAKE SURE GIVEN RETRIEVAL MD FILE REPORT IS LOCATED WITHIN GRID BOUNDS FOR WIND CALCULATIONS
LON=IBUF(NLON)/100
LON≥ LONMAX OR LON< LONMIN
32
LUC (81)=Ø
PLOT/NO PLOT OPTION
IPLT ≠ Ø
168
THE "NO-PLOT" ROUTE
USE TOVS NAVIGATION ROUTINE
REVERT TO -W +E CONVENTION
CALL SRCH (LATS, LONGS, IM, IL, IE, NPT, NROWS)
CALL SATEAR (PTIME, FLIN, FELE, FLAT, SLON, 2, INAV, BETAIN, BETDOT, Ø.)
GET RET. LINE/ ELEMENT COORDINATES (FLIN, FELE)
IM=Ø
GUARD AGAINST OVERLAP IN SRCH TO AVOID DUPLICATION
CALL SATTV (JFR, IL, IE, JP, IP, JS, JD, JT)
GET TV COORDINATES OF REPORT (JP, IP)
IL< LTOP OR IL≥LBOT


CALL ZWIND(JHITE, I, J, JLAT, SPEED, DIREC, JHIT2, DELT, JHIT3)
CALCULATE WIND OF DESIRED TYPE
IPLT ≠ Ø
170
"NO-PLOT" ROUTE
F
NSAT ≠ Ø
CALL DIRADJ (FLAT, FLON, SATLON, DIREC, ADJ)
ADJUST DIRECTION OF WIND SO PLOT WILL BE FROM TRUE DIRECTION ON VIDEO (TV) SCREEN
F
CALL BARB (ADJ, SPEED, XP, YP, KOL, SZ)
PLOT WIND BARB
CHECK HEIGHT LEVEL FOR WRITING WIND OB
IF SECOND POSITIONAL PARAMETER NOT KEYED IN (LEVP=Ø, AS RESULT), THE FINAL WINDS WILL NOT BE WRITTEN TO THE RETRIEVAL MD FILE
LEVP ≠ Ø
175
WRITE RETRIEVAL REPORT, UPDATED WITH WIND DIRECTION AND SPEED, INTO RETRIEVAL MD FILE
IOK = MDPUT (MDNO, MDR, N, IBUF)
F
IPR ≠ Ø
200
C
IOK<Ø
906
CALL SDEST (' TROUBLE WRITING DATA RECORD NO ', N)
F
END
F
C
END OF DO LOOP 2ØØ
IPR=1
CALL ENDPLT
CALL SDEST (' DONE...', Ø)
END
CALL SDEST (' WINDS NOT WRITTEN TO OUTPUT FILE... , Ø)
END OF DO LOOP 2ØØ
C
CALL ENDPLT
CALL SDEST (' DONE... ', Ø)
END

Subroutines used by GWVA:

1) INITPL-I
2) SDEST-I
3) NVINIT-I
4) SATEAR-I
5) SATTV-I
6) ZWIND-I
7) DIRADJ-I
8) BARB-I
9) ENDPLT-I
10) DOPEN-I
11) DREAD-I
12) TVSAT-I
13) TSNIO-I
14) OUTINT-I
15) SRCH-I
16) ENCODE
17) ENCODX
18) ABORT
19) CONTNT
20) LTQ
21) ZECONV
22) MOVB
23) CLEANA
24) TQ
25) PRLINX
26) PRCLOS
27) PROPEN
28) PRPRPR
29) LOCK
30) PRWR
31) PRRD
32) UNLOCK
33) PRCL
34) POST
35) BLKA
36) DCLOSE
37) II
38) STC
39) DWRITE
40) MOVCW
41) GETNAV
42) EPOCH
43) TQMES
44) ITOC
45) GETGAM
46) VASNAV
47) DDEST
48) CLEANW
49) EDEST
50) MOVW
51) SATPOS
52) GETFRM
53) TRMNL
54) GRADWI
55) PLOT
56) ENDPLT
57) ENPT
58) PENADD
59) PACK
60) SENOUT
61) ATOE
62) TEKPUT
63) PAGE
64) PENMOV
65) BOX
66) PENBEG
67) WALK

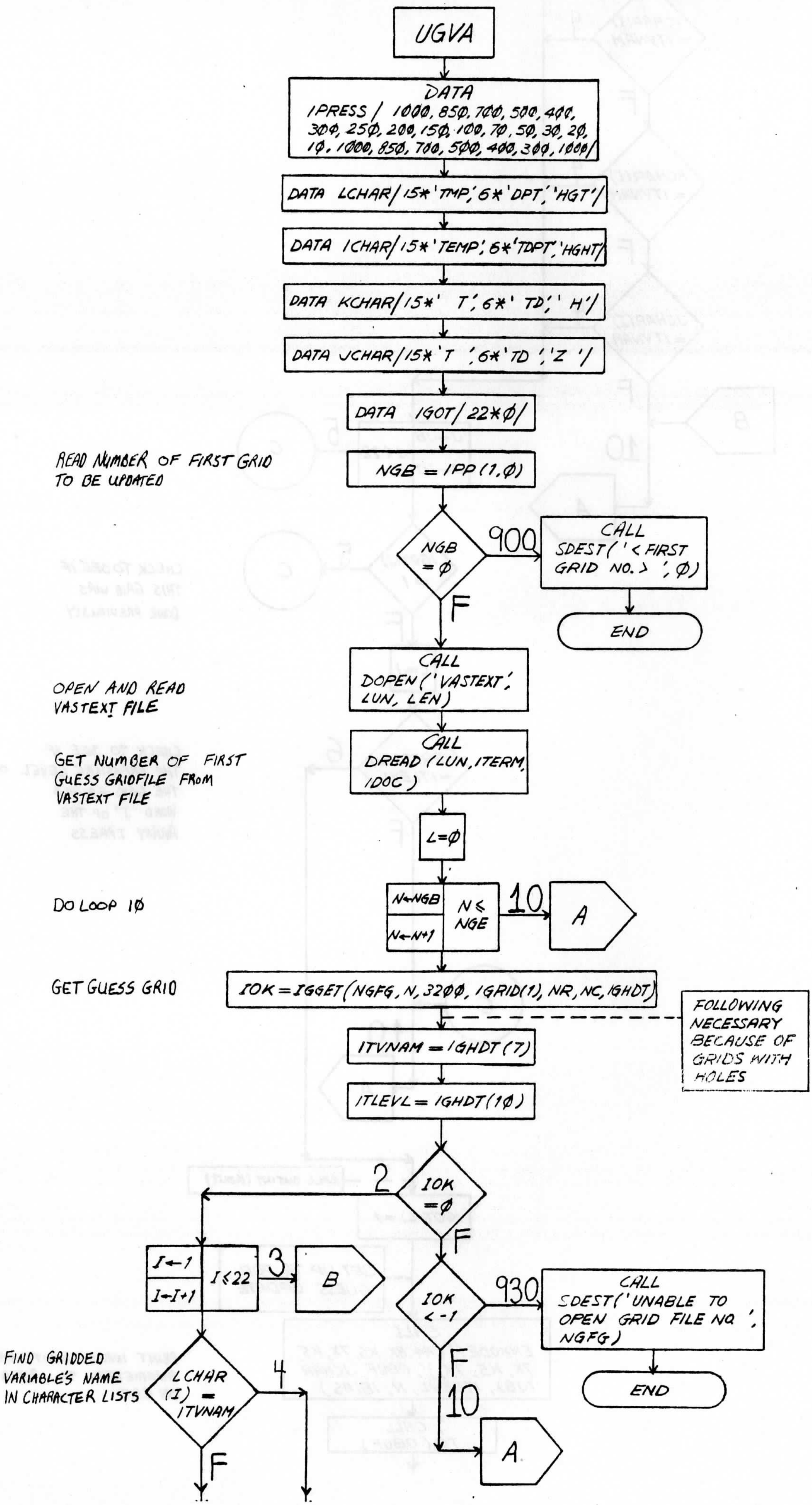
UGVA

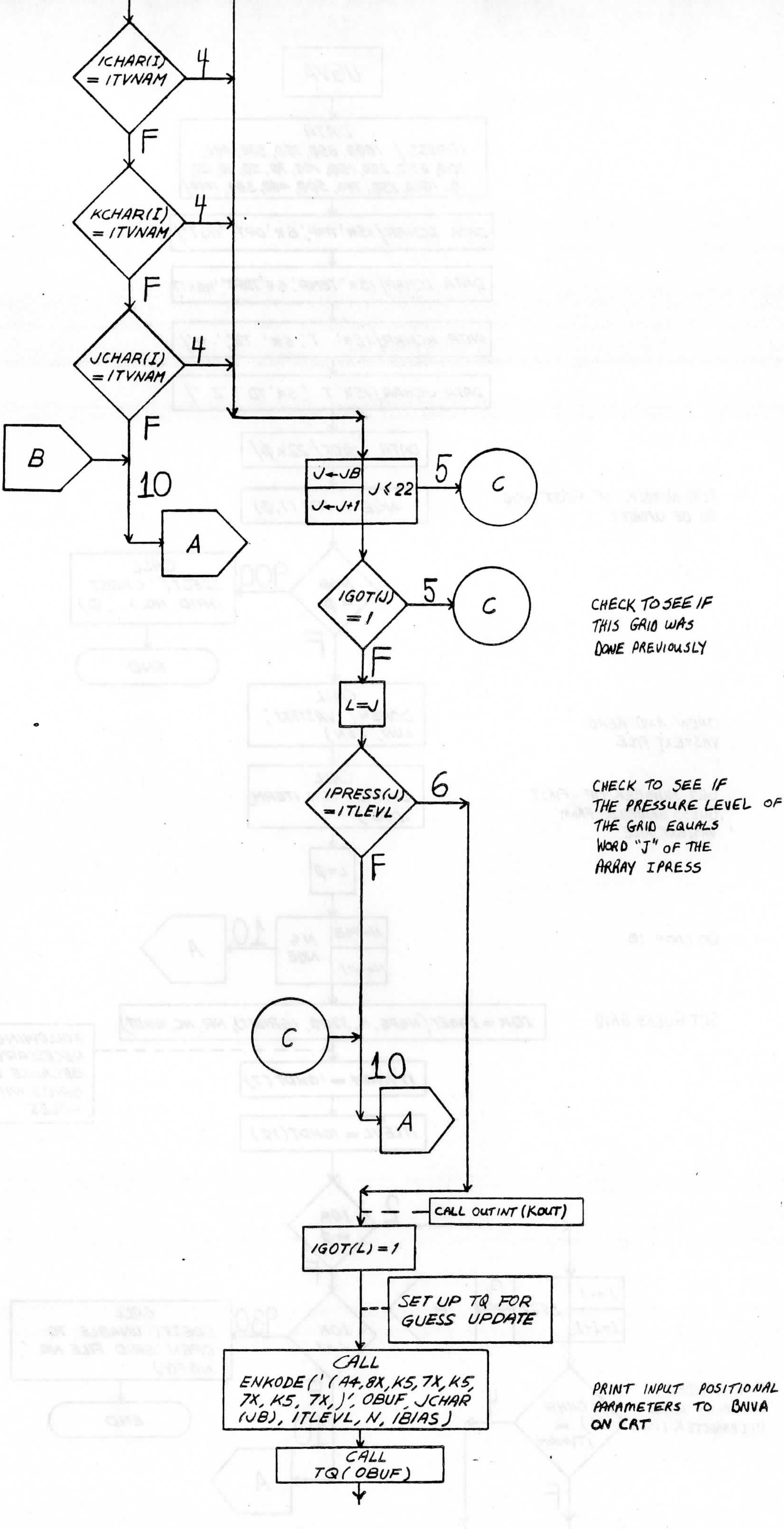
UGVA is used to update the first guess gridfile (NGFG) using VAS retrievals. It is a program that can be used in the event two or more sounder areas of different times have to be processed, because it allows the latter areas to use more up-to- date first guesses. For instance, let's assume we have two sounder areas at 1400 and 1600 Greenwich Mean Time (GMT), and a 1200 GMT first guess. The first guess would be used to process the 1400 GMT retrievals. Then, UGVA could be used to generate grids of 1400 GMT retrieval parameters by updating the original first guess grids with the 1400 GMT retrievals. The resultant grids (normally one of 1000 mb height, 15 of temperature and 6 of dewpoint) could then be used as a first guess for the 1600 GMT retrievals. Essentially, as the reader will soon see, all UGVA does is run BNVA 22 consecutive times, using a given first guess grid for each execution.

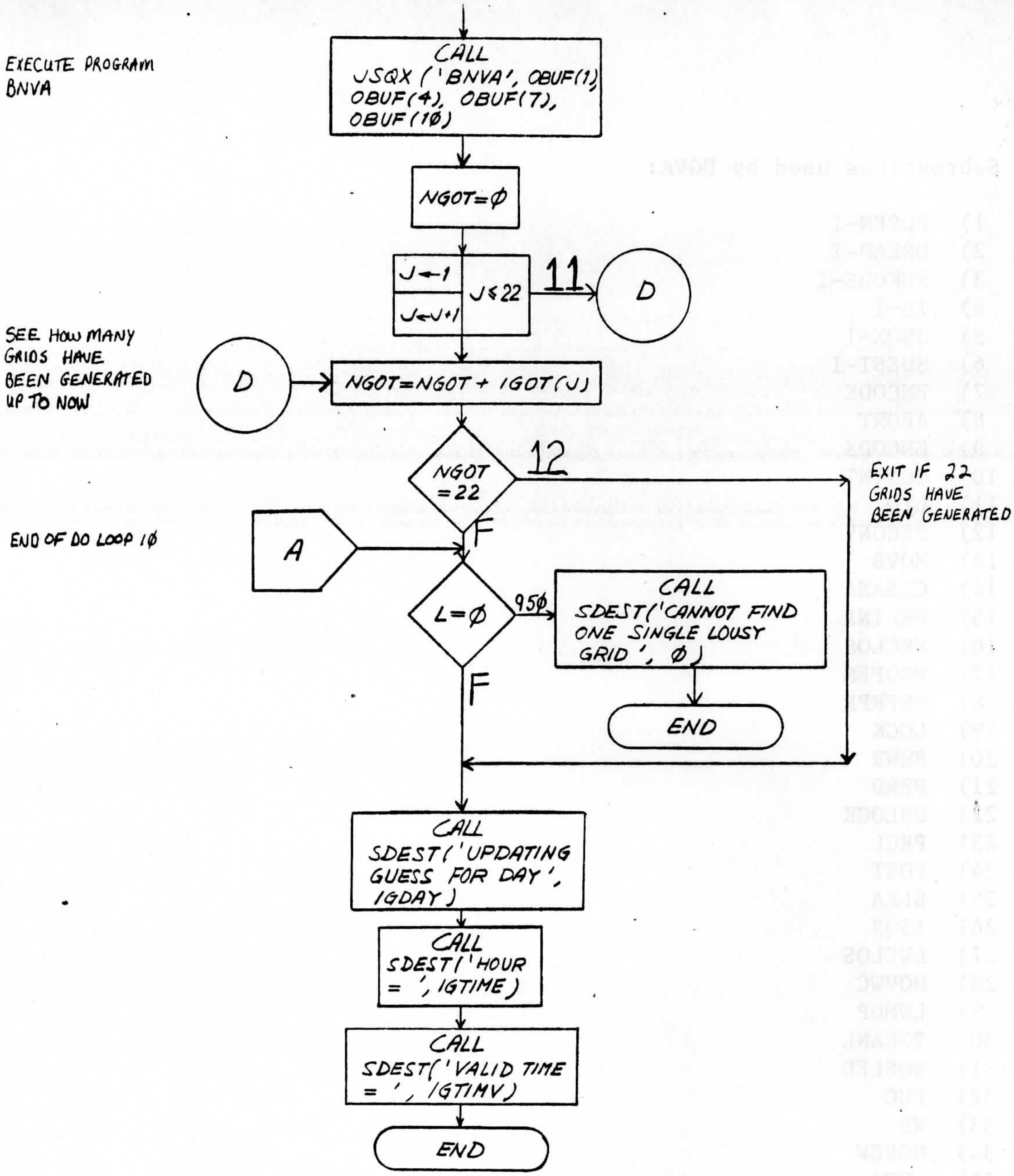
After the first grid to be updated has been determined, the VASTEXT file has been opened and read, and the guess gridfile to be updated has been determined, UGVA enters DO LOOP 10, which encompasses the vast majority of the remainder of the program.

First, a given grid N is read from the guess gridfile via function IGGET. UGVA terminates if there is no such grid file. In addition, the program will move to read the next grid if grid N does not exist or is too big to be read. Assuming the grid is read with no problems, the gridded variable's name (ITVNAM) is found in one of four character lists (LCHAR, ICHAR, KCHAR and JCHAR), and the pressure level of the grid in question is checked in DO LOOP 5 to see if it equals element J of the array IPRESS. UGVA will move to read the next grid if either the gridded variable's name or the pressure level of the grid cannot be found. Assuming that the name and pressure level of the grid ARE found, the grid header is stored in array IGHD, and word L of the array IGOT is set to 1 to prevent UGVA from generating any subsequent grids of the same parameter at the same level. After this, subroutines ENKODE and TQ display on the CRT such information as the name of the gridded variable (JCHAR(JB)), the grid's pressure level (ITLEVL), and the number of the grid within the guess gridfile that is to be updated (N). Grid N will serve as the guess grid in the forthcoming execution of program BNVA.

Next, in the most important step of UGVA, a call to subroutine JSQX results in the execution of program BNVA and the generation of a resultant grid of the specified retrieval parameter at level ITLEVL, using guess grid N. This updated guess grid will then be stored in the first guess gridfile in the first empty slot after 0. Finally, variable NGOT is checked to see how many grids have been generated. If NGOT=22 (hence, 22 grids created), DO LOOP 10 is exited. As mentioned earlier, there will normally be 22 resultant grids, consisting of 15 grids of temperature, 6 grids of dewpoint and 1 grid of 1000 mb height.


UGVA
DATA
IPRESS / 1000, 85Ø, 7ØØ, 500, 400, 3ØØ, 25Ø, 200, 15Ø, 100, 7Ø, 5Ø, 3Ø, 2Ø, 1Ø, 1000, 85Ø, 700, 5ØØ, 400, 3ØØ, 100Ø/
DATA LCHAR/15*'TMP', 6*'DPT', 'HGT'/
DATA ICHAR/15*'TEMP', 6*'TDPT', 'HGHT'/
DATA KCHAR/15*' T', 6*' TD', ' H'/
DATA JCHAR/15*'T ', 6*'TD ', 'Z '/
DATA IGOT/22*Ø/
READ NUMBER OF FIRST GRID TO BE UPDATED
NGB = IPP(1,Ø)
NGB = Ø
900
CALL SDEST('< FIRST GRID NO.> ', Ø)
END
F
OPEN AND READ VASTEXT FILE
CALL DOPEN('VASTEXT', LUN, LEN)
GET NUMBER OF FIRST GUESS GRIDFILE FROM VASTEXT FILE
CALL DREAD (LUN, ITERM, IDOC)
L=Ø
DO LOOP 1Ø
N←NGB
N←N+1
N ≤ NGE
10
A
GET GUESS GRID
IOK = IGGET(NGFG, N, 320Ø, IGRID(1), NR, NC, IGHDT)
FOLLOWING NECESSARY BECAUSE OF GRIDS WITH HOLES
ITVNAM = IGHDT(7)
ITLEVL = IGHDT(1Ø)
IOK = Ø
2
F
I←1
I←I+1
I ≤ 22
3
B
IOK < -1
930
CALL SDEST('UNABLE TO OPEN GRID FILE NO.', NGFG)
END
F
10
A
FIND GRIDDED VARIABLE'S NAME IN CHARACTER LISTS
LCHAR (I) = ITVNAM
4
F


ICHAR(I) = ITVNAM
4
F
KCHAR(I) = ITVNAM
4
F
JCHAR(I) = ITVNAM
4
F
B
10
A
J←JB
J←J+1
J ≤ 22
5
C
IGOT(J) = 1
5
C
CHECK TO SEE IF THIS GRID WAS DONE PREVIOUSLY
F
L=J
IPRESS(J) = ITLEVL
6
CHECK TO SEE IF THE PRESSURE LEVEL OF THE GRID EQUALS WORD "J" OF THE ARRAY IPRESS
F
C
10
A
CALL OUTINT (KOUT)
IGOT(L) = 1
SET UP TQ FOR GUESS UPDATE
CALL ENKODE ('(A4,8X,K5,7X,K5, 7X, K5, 7X,)', OBUF, JCHAR (JB), ITLEVL, N, IBIAS)
PRINT INPUT POSITIONAL PARAMETERS TO BNVA ON CAT
CALL TQ(OBUF)


EXECUTE PROGRAM BNVA
CALL USQX ('BNVA', OBUF(1), OBUF(4), OBUF(7), OBUF(1Ø)
NGOT=Ø
J←1
J←J+1
J≤22
11
D
SEE HOW MANY GRIDS HAVE BEEN GENERATED UP TO NOW
D
NGOT=NGOT + IGOT(J)
NGOT =22
12
EXIT IF 22 GRIDS HAVE BEEN GENERATED
F
END OF DO LOOP 1Ø
A
L=Ø
95Ø
CALL SDEST('CANNOT FIND ONE SINGLE LOUSY GRID', Ø)
END
F
CALL SDEST('UPDATING GUESS FOR DAY', IGDAY)
CALL SDEST('HOUR =', IGTIME)
CALL SDEST('VALID TIME =', IGTIMV)
END

Subroutines used by UGVA:

1) DOPEN-I
2) DREAD-I
3) ENKODE-I
4) TQ-I
5) JSQX-I
6) SDEST-I
7) ENCODE
8) ABORT
9) ENCODX
10) CONTNT
11) LTQ
12) ZECONV
13) MOVB
14) CLEANA
15) PRLINX
16) PRCLOS
17) PROPEN
18) PRPRPR
19) LOCK
20) PRWR
21) PRRD
22) UNLOCK
23) PRCL
24) POST
25) BLKA
26) ISQX
27) LWCLOS
28) MOVWC
29) LWMOP
30) TOKANL
31) SQSLED
32) PUC
33) WD
34) MOVCW
35) LWPO
36) EDEST
37) MOVW
38) CLEANW
39) LWNEWF
40) LWSO
41) JMBWTF

CHAPTER 4

Supplementary VAS Retrieval Software

The software in this chapter should be considered as having an importance secondary to that in Chapter 3. These programs are used less frequently, but can on occasion still make a meaningful contribution to the retrieval process as a whole. The format used in describing the programs in this chapter is identical to that used in Chapter 3.

EXVA

EXVA is used to examine retrieval profiles in terms of temperature and differences in temperature between the retrieval and first guess profiles. The results for each retrieval examined are displayed on the CRT. One or more retrieval profiles can be examined and displayed on the CRT at one time.

Initially, the cursor coordinates (in lines and elements) are determined (CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)), and the VASTEXT file is opened and read. Then, after the navigation has been initialized, the retrieval MD file and row numbers are put, respectively, into the variables MDNO and MDR. These numbers are automatically taken from the VASTEXT file, and correspond to IDOC(40) and (41), respectively. Following this, if the retrieval MD file is successfully opened for reading, the keys for the file (schema VRET) are read, which allows EXVA to determine which words within the retrieval MD file output array (IOUT) for a given retrieval report contain latitude, longitude and surface temperature (NLAT, NLON and NDX, respectively). The values for latitude, longitude and surface temperature will be stored in the same word locations for each retrieval report in the retrieval MD file.

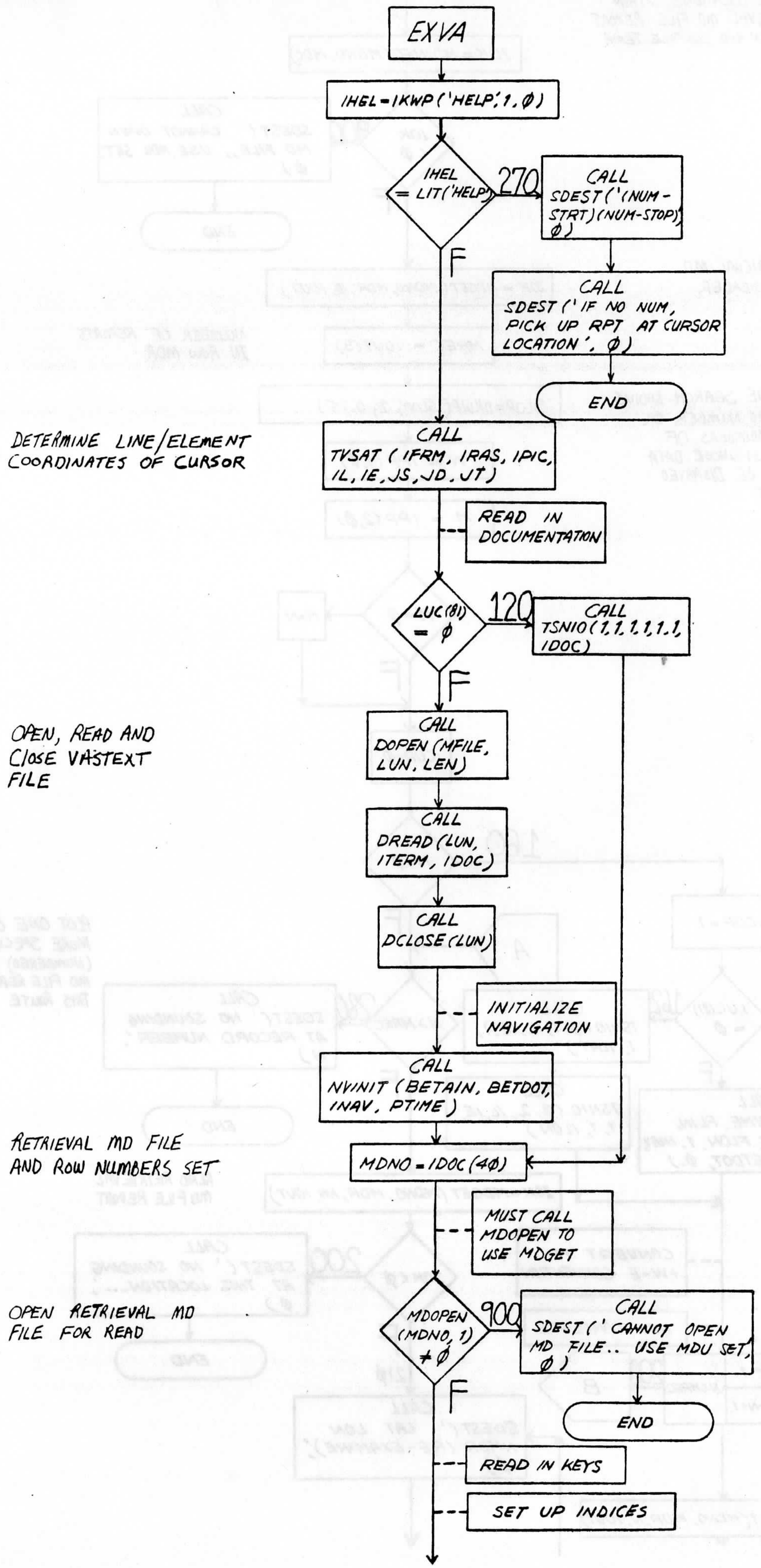
After the existence of the retrieval MD file has been checked with function MDINFO, the row header for row MDR of the file is read, which gives, among other things, the number of retrieval reports contained within that row of the file (stored in variable MREC). In addition, the numbers of the first and last retrievals whose data is to be displayed, if more than one retrieval report is desired, are read into variables N and M. Note also the determination of variable SLOP. This variable is used as a search radius when only one report at(near) the cursor location is to be displayed.

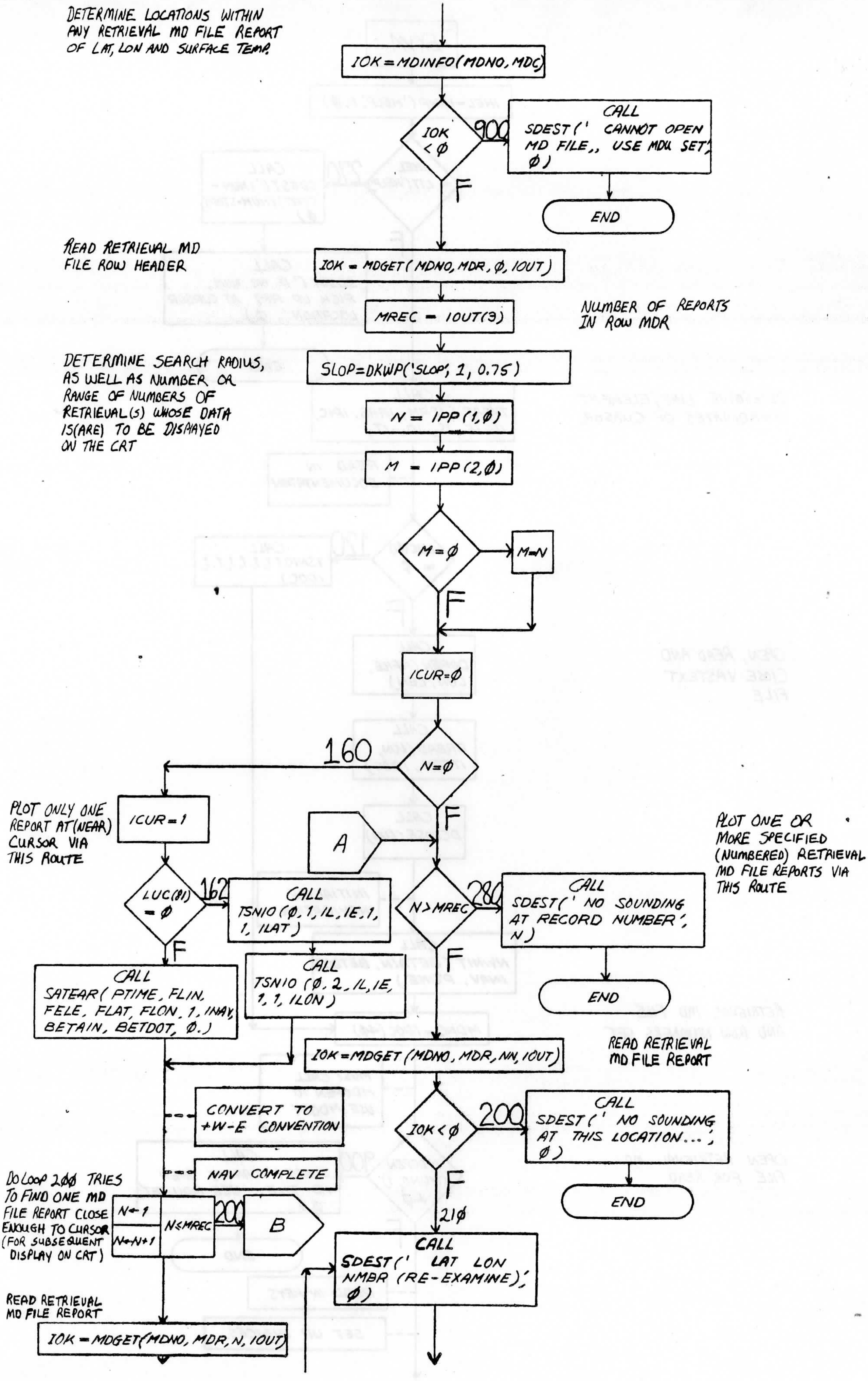
From this point on, the discussion of the remainder of EXVA will consist of three paragraphs. Each paragraph will discuss how EXVA operates for a given keyin. The first paragraph assumes the user has simply keyed in EXVA by itself. The second paragraph assumes that one retrieval number has been keyed in with EXVA, while the third discusses the case when EXVA is keyed in with two retrieval numbers as positional parameters.

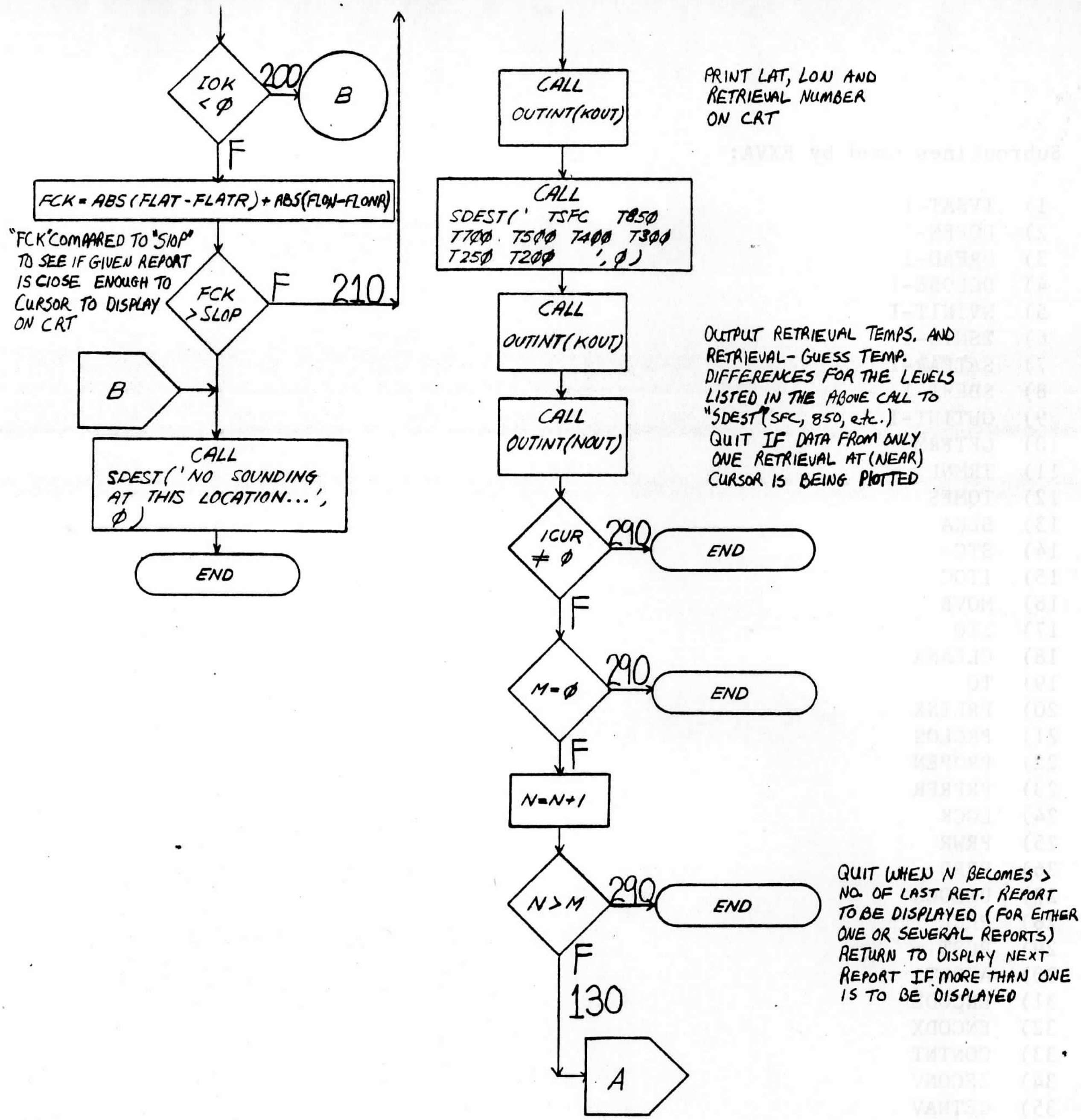
Assuming no retrieval number or numbers have been keyed in (N, M .EQ. 0), only the retrieval report at(near) the cursor location will be presented. In this case, subroutine SATEAR calculates the latitude and longitude of the cursor. Then, DO LOOP 200 goes through each retrieval report in row MDR of the retrieval MD file, looking for any retrieval that is suitably close to the cursor location (note variables FCK and SLOP; SLOP=0.75 degrees, if it was not explicitly keyed in earlier). If no retrievals close enough are found, EXVA terminates. However, if a satisfactory retrieval IS found (FCK .LE. SLOP), its latitude, longitude and number within row MDR are printed on the CRT, as well as surface temperature, 850 mb temperature (T850), T700, T500, T400, T300, T250 and T200. In addition, the temperature differences between the retrieval and first guess for the same levels are also printed. This then concludes EXVA for the no retrieval number option.

If only one retrieval number has been keyed in (N .NE. 0, M=0), the retrieval report corresponding to the number N is read from the retrieval MD file (IOK=MDGET(MDNO, MDR, NN, IOUT)), and the same information as described above is printed on the CRT, which will then bring this particular option of EXVA to a close.

Finally, if two retrieval numbers have been keyed in, the data for all retrievals between and including the two numbers is printed on the CRT, provided of course the number of the retrieval to be displayed (N) does not exceed MREC and that there is a record at each report location in row MDR of the retrieval MD file.


EXVA
IHEL=IKWP('HELP',1,Ø)
IHEL = LIT('HELP')
270
CALL SDEST('(NUM-STRT)(NUM-STOP)', Ø)
CALL SDEST('IF NO NUM, PICK UP RPT AT CURSOR LOCATION', Ø)
END
F
DETERMINE LINE/ELEMENT COORDINATES OF CURSOR
CALL TVSAT (IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)
READ IN DOCUMENTATION
LUC(81) = Ø
120
CALL TSNIO(1,1,1,1,1,1, IDOC)
F
OPEN, READ AND CLOSE VASTEXT FILE
CALL DOPEN(MFILE, LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
CALL DCLOSE(LUN)
INITIALIZE NAVIGATION
CALL NVINIT (BETAIN, BETDOT, INAV, PTIME)
RETRIEVAL MD FILE AND ROW NUMBERS SET
MDNO=IDOC(4Ø)
MUST CALL MDOPEN TO USE MDGET
OPEN RETRIEVAL MD FILE FOR READ
MDOPEN (MDNO,1) ≠ Ø
900
CALL SDEST('CANNOT OPEN MD FILE.. USE MDU SET', Ø)
END
F
READ IN KEYS
SET UP INDICES


DETERMINE LOCATIONS WITHIN ANY RETRIEVAL MD FILE REPORT OF LAT, LON AND SURFACE TEMP.
IOK=MDINFO(MDNO, MDC)
IOK < Ø
900
CALL SDEST(' CANNOT OPEN MD FILE,, USE MDU SET', Ø)
END
F
READ RETRIEVAL MD FILE ROW HEADER
IOK = MDGET(MDNO, MDR, Ø, IOUT)
MREC = IOUT(3)
NUMBER OF REPORTS IN ROW MDR
DETERMINE SEARCH RADIUS, AS WELL AS NUMBER OR RANGE OF NUMBERS OF RETRIEVAL(S) WHOSE DATA IS(ARE) TO BE DISPAYED ON THE CRT
SLOP=DKWP('SLOP', 1, 0.75)
N = IPP(1,Ø)
M = IPP(2,Ø)
M=Ø
M=N
F
ICUR=Ø
160
N=Ø
F
ICUR=1
PLOT ONLY ONE REPORT AT(NEAR) CURSOR VIA THIS ROUTE
A
LUC(81) = Ø
162
CALL TSNIO(Ø, 1, IL, IE, 1, 1, ILAT)
N>MREC
280
CALL SDEST(' NO SOUNDING AT RECORD NUMBER', N)
PLOT ONE OR MORE SPECIFIED (NUMBERED) RETRIEVAL MD FILE REPORTS VIA THIS ROUTE
END
F
CALL TSNIO (Ø, 2, IL, IE, 1, 1, ILON)
CALL SATEAR(PTIME, FLIN, FELE, FLAT, FLON, 1, INAV, BETAIN, BETDOT, Ø.)
F
IOK=MDGET(MDNO, MDR, NN, IOUT)
READ RETRIEVAL MD FILE REPORT
CONVERT TO +W-E CONVENTION
IOK < Ø
200
CALL SDEST(' NO SOUNDING AT THIS LOCATION...', Ø)
NAV COMPLETE
END
F
DO LOOP 200 TRIES TO FIND ONE MD FILE REPORT CLOSE ENOUGH TO CURSOR (FOR SUBSEQUENT DISPLAY ON CRT)
N←1
N≤MREC
N←N+1
200
B
210
CALL SDEST(' LAT LON NMBR (RE-EXAMINE)', Ø)
READ RETRIEVAL MD FILE REPORT
IOK = MDGET(MDNO, MDR, N, IOUT)


IOK < ø
200
B
F
FCK = ABS(FLAT-FLATR) + ABS(FLON-FLONR)
"FCK" COMPARED TO "SLOP" TO SEE IF GIVEN REPORT IS CLOSE ENOUGH TO CURSOR TO DISPLAY ON CRT
FCK > SLOP
F
210
B
CALL SDEST('NO SOUNDING AT THIS LOCATION...', ø)
END
CALL OUTINT(KOUT)
PRINT LAT, LON AND RETRIEVAL NUMBER ON CRT
CALL SDEST(' TSFC T850 T7øø T5øø T4øø T3øø T25ø T2øø ', ø)
CALL OUTINT(KOUT)
OUTPUT RETRIEVAL TEMPS. AND RETRIEVAL-GUESS TEMP. DIFFERENCES FOR THE LEVELS LISTED IN THE ABOVE CALL TO "SDEST" (SFC, 850, etc.)
CALL OUTINT(NOUT)
QUIT IF DATA FROM ONLY ONE RETRIEVAL AT (NEAR) CURSOR IS BEING PLOTTED
ICUR ≠ ø
290
END
F
M = ø
290
END
F
N=N+1
N > M
290
END
QUIT WHEN N BECOMES > NO. OF LAST RET. REPORT TO BE DISPLAYED (FOR EITHER ONE OR SEVERAL REPORTS) RETURN TO DISPLAY NEXT REPORT IF MORE THAN ONE IS TO BE DISPLAYED
F
130
A

Subroutines used by EXVA:

1) TVSAT-I
2) DOPEN-I
3) DREAD-I
4) DCLOSE-I
5) NVINIT-I
6) TSNIO-I
7) SATEAR-I
8) SDEST-I
9) OUTINT-I
10) GETFRM
11) TRMNL
12) TQMES
13) BLKA
14) STC
15) ITOC
16) MOVB
17) LTQ
18) CLEANA
19) TQ
20) PRLINX
21) PRCLOS
22) PROPEN
23) PRPRPR
24) LOCK
25) PRWR
26) PRRD
27) UNLOCK
28) PRCL
29) POST
30) ABORT
31) ENCODE
32) ENCODX
33) CONTNT
34) ZECONV
35) GETNAV
36) EPOCH
37) GETGAM
38) VASNAV
39) DDEST
40) MOVCW
41) CLEANW
42) EDEST
43) MOVW
44) II
45) DWRITE
46) SATPOS

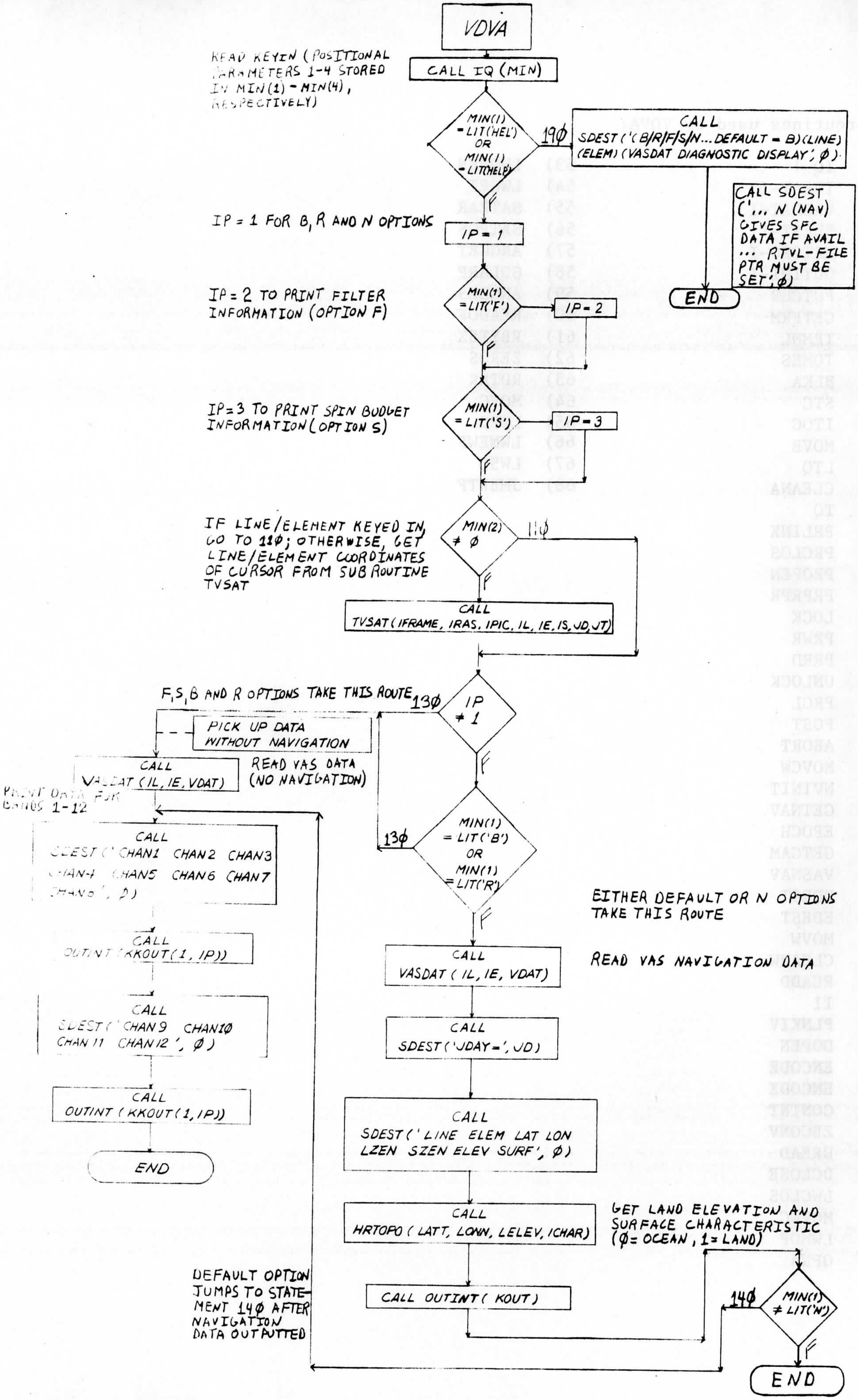
VDVA

The purpose of this program is to pick up VAS data according to the position of the cursor (or keyed in line and element) and the sounder-area pointer (set previously by program VPVA). It is a program that is not vital to the generation of a retrieval or area of retrievals; rather, VDVA can be used to check an individual field of view to make sure no unreasonable values of brightness temperature, spin budget, radiance, etc. exist for any given band.

Eight different general types of keyin possibilities exist for VDVA, depending on which positional parameter(s) is(are) included in the keyin. For VDVA keyed in with one following positional parameter, the five possibilities are as follows: VDVA B (brightness temperature), VDVA R (radiance), VDVA F (filter number), VDVA S (spin budget) and VDVA N (navigation). In order to read the VAS data, all eight options require the cursor position to be known in satellite coordinates (line-IL, element-IE). Normally, the cursor position, in terms of satellite coordinates, will be determined via subroutine TVSAT. However, an option (#6) does exist to key in the line/element coordinates of the field of view to be examined via the second and third positional parameters in the program keyin (VDVA F 2000 9900, for example). If this option is taken, the call to TVSAT will not be executed, because MIN(2) will not be equal to 0 (see "IF(MIN(2).NE.0)GO TO 110").

For the default option (#7--VDVA keyed in by itself), the output consists of brightness temperatures for bands 1-12, with the output for band 11, deleted due to excessive noise, consisting of six 9's. Other output for this option includes variable JD, which consists of a combination of the satellite number + year + Julian date, as well as cursor location data, local (or satellite--LZEN) and solar (SZEN) zenith angles, land elevation, and surface characteristic (0=ocean, 1=land). The output for option B (keyin of "VDVA B") consists only of the brightness temperature data for all twelve bands. The output for option N consists of variable JD, as well as the cursor location data, zenith angle data, land elevation, and surface characteristic. In other words, the output for option N consists of the default option information less the brightness temperature data. Option R returns radiance data only for the twelve bands. Option F returns the filter (sensor) number being used for each band, while the last single positional parameter option, S, simply returns the spin budget for each band.

Finally, if the user keys in VDVA followed by one of the five explicit positional parameters described above (B, N, R, F or S), two zeroes (positional parameters 2 and 3), and then a "1" (#8--a keyin of "VDVA B 0 0 1," for example), the information noted above for options B, N, R, F or S, as well as further supplementary diagnostic information, is printed on the CRT. This diagnostic information always includes the sounder area directory and, in most cases, information pertaining to image to sounder-file (sounder area) coordinate transformations, raw data counts for each band, etc.


VDVA
READ KEYIN (POSITIONAL PARAMETERS 1-4 STORED IN MIN(1) - MIN(4), RESPECTIVELY)
CALL IQ (MIN)
MIN(1) = LIT('HEL') OR MIN(1) = LIT('HELP')
190
CALL SDEST ('(B/R/F/S/N... DEFAULT = B)(LINE)(ELEM) (VASDAT DIAGNOSTIC DISPLAY', Ø)
CALL SDEST ('... N (NAV) GIVES SFC DATA IF AVAIL ... RTVL-FILE PTR MUST BE SET', Ø)
END
F
IP = 1 FOR B, R AND N OPTIONS
IP = 1
IP = 2 TO PRINT FILTER INFORMATION (OPTION F)
MIN(1) = LIT('F')
IP = 2
F
IP = 3 TO PRINT SPIN BUDGET INFORMATION (OPTION S)
MIN(1) = LIT('S')
IP = 3
F
IF LINE/ELEMENT KEYED IN, GO TO 11Ø; OTHERWISE, GET LINE/ELEMENT COORDINATES OF CURSOR FROM SUBROUTINE TVSAT
MIN(2) ≠ Ø
11Ø
F
CALL TVSAT (IFRAME, IRAS, IPIC, IL, IE, IS, JD, JT)
F, S, B AND R OPTIONS TAKE THIS ROUTE
13Ø
IP ≠ 1
PICK UP DATA WITHOUT NAVIGATION
CALL VASDAT (IL, IE, VDAT)
READ VAS DATA (NO NAVIGATION)
PRINT DATA FOR CHNLS 1-12
CALL SDEST (' CHAN1 CHAN2 CHAN3 CHAN4 CHAN5 CHAN6 CHAN7 CHAN8', Ø)
CALL OUTINT (KKOUT(1, IP))
CALL SDEST (' CHAN9 CHAN1Ø CHAN 11 CHAN12 ', Ø)
CALL OUTINT (KKOUT(1, IP))
END
F
13Ø
MIN(1) = LIT('B') OR MIN(1) = LIT('R')
F
EITHER DEFAULT OR N OPTIONS TAKE THIS ROUTE
CALL VASDAT (IL, IE, VDAT)
READ VAS NAVIGATION DATA
CALL SDEST ('JDAY=', JD)
CALL SDEST (' LINE ELEM LAT LON LZEN SZEN ELEV SURF', Ø)
CALL HRTOPO (LATT, LONN, LELEV, ICHAR)
GET LAND ELEVATION AND SURFACE CHARACTERISTIC (Ø= OCEAN, 1= LAND)
DEFAULT OPTION JUMPS TO STATEMENT 14Ø AFTER NAVIGATION DATA OUTPUTTED
CALL OUTINT (KOUT)
14Ø
MIN(1) ≠ LIT('N')
F
END

Subroutines used by VDVA:

1) IQ-I
2) TVSAT-I
3) VASDAT-I
4) SDEST-I
5) HRTOPO-I
6) OUTINT-I
7) PUTCHR
8) GETFRM
9) TRMNL
10) TQMES
11) BLKA
12) STC
13) ITOC
14) MOVB
15) LTQ
16) CLEANA
17) TQ
18) PRLINX
19) PRCLOS
20) PROPEN
21) PRPRPR
22) LOCK
23) PRWR
24) PRRD
25) UNLOCK
26) PRCL
27) POST
28) ABORT
29) MOVCW
30) NVINIT
31) GETNAV
32) EPOCH
33) GETGAM
34) VASNAV
35) DDEST
36) EDEST
37) MOVW
38) CLEANW
39) READD
40) II
41) PLNKIV
42) DOPEN
43) ENCODE
44) ENCODX
45) CONTNT
46) ZECONV
47) DREAD
48) DCLOSE
49) LWCLOS
50) MOVWC
51) LWMOP
52) OPNA
53) LWOPEN
54) LWGET
55) SATEAR
56) SATPOS
57) ANGGET
58) SOLARP
59) ANGLES
60) READOF
61) RBYTSX
62) ZEROS
63) RDTRK
64) MOVC
65) LWPO
66) LWNEWF
67) LWSO
68) JMBWTF

CHAPTER 5

Miscellaneous VAS Retrieval Software

The routines in this chapter are composed of a mixture of rarely (if ever) used retrieval support and deactivated retrieval-processing software. Since any or all of these programs may perhaps be rejuvenated at some future date, it is felt that they too should be included in the manual.

The deactivated software consists of programs SRAD, SRET and VTPW. SRAD and SRET together were used to process routine VAS retrievals up to approximately early April, 1984, while VTPW was used on an experimental basis early in 1984 to generate images of total precipitable water vapor from VAS radiance information (VTPZ performs this function now, in addition to doing the retrievals).

SRAD

This program prepares VAS large or small-detector (approximately 16 and 8 km resolution, respectively) brightness temperatures (TBBs) for SRET, the VAS iterative retrieval program. Initially, before SRAD is keyed in, the cursor should be moved to a position near the northwest (NW) corner of the retrieval area the user wishes to process. The reason for this will become clear later. When the program is keyed in, the first positional parameter should be entered as "GO". Suitable keywords will then be entered after the "GO". (The option of running SRAD with the first positional parameter equal to "CLE" should be used only if the user wishes to update words 50-60 (IDOC(50-60)--bottom row of output as displayed by program LOVA) in the VASTEXT file. This option will rarely be used.) SRAD can be keyed in to process either an area of retrievals or one individual retrieval. This discussion of SRAD will focus on the processing of an area of retrievals, and will be concerned primarily with the "non-auto" option (keyword parameter AUTO not keyed in). Any deviations from this format will be duly noted. Note also that occasionally radiance and TBB will be interchanged. The two, of course, can be related through the Planck function.

Initially, the VASTEXT file and the row header of the retrieval MD file are read, and the first positional parameter is evaluated. Assuming the user has keyed in "GO" for this first parameter, the program jumps to statement 100 (IF CPP(1,' ').NE.'CLE') GO TO 100), where the latitude/longitude bounds which determine whether a given retrieval box of VAS radiance data can be used in the processing of the retrieval area (LN, LS, LW and LE) are set. Note that these values are derived from VASTEXT file values IDOC(25) and (26). Then, several other parameters, such as retrieval type (variable CTYP--a keyin of "TYP=N, C or B/no keyin" for N-star, clear or default retrievals, respectively), surface data option (NOSFC), retrieval box line and element spacings (in fields of view (FOVs)--INCRL,INCRE), retrieval box size (in FOVs--NBXS), etc. are set. Note that the retrieval box size can range from 3*3 to 11*11 FOV's in dimension. However, assuming the box size has NOT been keyed in via keyword parameter SIZ, the default size determined for large detector retrievals will be 5*5 FOVs, while the same for small detector retrievals will be 9*9 FOVs. After the retrieval box size has been determined, the debug option (KBUG) is set. If KBUG is set to 1, 2 or 3 (keyword parameter BUG=1,2 or 3 in keyin), messages detailing program status will be displayed at the corresponding location (either the CRT, local printer or system printer, respectively). Then, the plot option (IPLT) and variables MDNG (retrieval guess MD file) and MDNR (retrieval MD file) are given values. Note that MDNG and MDNR will also get their values from the VASTEXT file.

At this point, the cursor TV coordinates (raster(INRAS), pixel(INPIC)) are accessed from User Common via function LUC, after which SRAD jumps to statement 112 (remember, SRAD is in the non-auto mode), where the initial and last line and element positions for the area of retrievals to be processed is determined. The last line and element positions (SE corner of image--LLINE/LELEM) will usually be keyed in using keyword parameter END as part of the original program keyin. The initial positions (NW corner of image--ILINE/IELEM) normally will NOT be keyed in; as a result, variables ILINE and IELEM will both initially assume the value 0 (DATA statement near code line 70) and will subsequently be determined via subroutine TVSAT. (The user should be able to see now why the cursor had to be positioned near the NW corner of the desired retrieval area before SRAD was executed.) Before the call to TVSAT, however, a call to subroutine VRTIO results in an update to the VASTEXT file, based on the changes made to this point in the array IDOC.

As an aside, assuming for a moment that SRAD is in the auto mode (keyword parameter AUTO keyed in with a value of 1), variables LLINE, LELEM, ILINE and IELEM are gleaned from the VASTEXT file. These values were calculated originally in program IDVA. Then, after the call to VRTIO, SRAD jumps to statement 218, and then to statement 220.

Returning now to the non-auto mode discussion, the single retrieval case will be mentioned briefly. SRAD will perform only one retrieval at the cursor location if the last line and element positions have not been keyed in (LLINE, LELEM =0). If this is the case, ILINE and IELEM will again be determined by TVSAT, after which the program will set the SE and NW line and element positions equal to each other.

Finally, the discussion returns to the original non-auto mode--retrieval area discussion. After TVSAT has been called (just before statement 218), SRAD jumps to statement 220 (IF (LLINE*LELEM.NE.0) GO TO 220). Immediately after this statement, subroutine VASDAT accesses navigation data, including information such as image line and element resolution. Then, after the retrieval box spacings in satellite coordinates (lines (y-direction--INCIL), elements (x-direction--INCIE)) have been determined, subroutine GETFRM returns a directory which contains image/image frame information (including the image magnification), after which SRAD is ready to begin processing the actual TBB data. (For a schematic showing the layout of a typical retrieval area, see Diagram 2 immediately following this discussion.)

The processing is done retrieval line by retrieval line within DO LOOP 2400, moving successively left to right within each line retrieval box by retrieval box (see statement 230: IEP=IEP+INCIE). Assuming IEP .LE. LELEM, the code up to the comments preceeding statement 305 (DO LOOPs 300 and 280, primarily) serves to initialize several of the arrays and variables to be used in the subsequent TBB processing.

Beginning with statement 305, SRAD enters an implicit DO LOOP which runs for approximately the next 45 lines, the main purpose of which is to access VAS brightness temperature data for all available bands for each FOV within a given retrieval box. Subroutine VASDAT does the actual data gathering. If the first FOV's TBB data, which will be at the retrieval box center, is being accessed (N=1), some tests are performed to make sure VASDAT is accessing data from a reasonable location. For instance, if the latitude of this FOV is north or south of the previously-set retrieval area latitude boundaries (LN,LS), the program will move to the next retrieval line within the image to attempt further processing. Other conditions causing problems include longitude greater than the western boundary (in which case SRAD steps one-half retrieval box spacing to the right and tries again), or longitude less than the eastern boundary (in which case SRAD jumps to statement 2390, updates the retrieval MD file row header and the VASTEXT file with the total number of retrievals completed so far (LASRET) via subroutine VRTIO, and again moves to the next retrieval line within the image for further TBB processing). SRAD will also move one-half retrieval box spacing to the right and attempt to access another set of TBB data if the satellite zenith angle (VZEN) is .GT. 60 degrees and LBEG .EQ. 0. LBEG will equal 0 until the TBB data for the first retrieval box in a given retrieval line has been accessed. Assuming there are no problems gathering the TBB data for a given FOV, the TBB information accessed will be collected for each band in the FOV and stored in the array RADS (N,I) in DO LOOP 400. DO LOOP 400 also updates a given element of the TBB array VDAT with the value VMISG if any gross TBB anomalies exist, in addition to tabulating the total number of FOV's in a given retrieval box (MSAM). After DO LOOP 400, variables SELEV, SLAT and SLON are updated with the values for the FOV being processed. The implicit DO LOOP will be exited after variable IBOX has been set to 3 in VASDAT, which will occur when all the radiance information for the retrieval box in question has been gathered.

Following the implicit DO LOOP, and assuming MSAM .GT. 0, the average retrieval box surface elevation, latitude and longitude are stored in the variables SELEV, SLAT and SLON, respectively. Then, a call to subroutine SURGES returns surface information, including variable ITSFC (surface temperature), which is used to determine variable TSTA. Following this, variable TMN (used to screen cloud-contaminated FOVs in a given retrieval box) is determined, and DO LOOPs 580 and 560 evaluate all the FOV's in the retrieval box, summing up for each band the number of FOVs with TBB data, and the TBBs themselves, in the arrays NSAM (or IUSE) and AVG, respectively.

After DO LOOP 580 has been exited, a quality control check involving the band 8 (window) brightness temperature data is done to see if there are at least 3 TBB reports in the retrieval box for this band. If the check fails, the program will step one-half retrieval box spacing to the right and attempt another retrieval. Otherwise, DO LOOP 590 will set IUSE(K)=to 0 for any band K other than band 8 which does not have at least 3 FOVs in the retrieval box with TBB observations.

Up to and including DO LOOP 590, SRAD has executed the same for all three retrieval type options (default, N-star or clear.) From this point on, the function of SRAD depends upon which type of retrieval is being done. As a result, separate discussions will be given for each retrieval type.

If the retrieval type is N-star (used under partly cloudy conditions), SRAD skips to statement 770, where clear column radiances for the retrieval box are inferred mathematically in each band from the mixture of clear, partly cloudy and cloudy radiances by subroutine NSTAR (MacMillin, 1978). Failure in NSTAR (IFAIL.NE.0) results in a new retrieval attempt one-half retrieval box spacing to the right of the failed attempt.

Otherwise, if subroutine NSTAR is successful, SRAD enters DO LOOP 830, in which the expected error of the brightness temperature observations for each band 1-12 (the array EX) are computed. After this DO LOOP, the surface skin temperature (TS) is calculated. Then, the output buffer (the array IRET-- note equivalence of to the array IRETD as stated back near code line 55) containing, among other things, the final brightness temperature and expected error of the brightness temperature observations for each band (DO LOOP 2203), is filled. Following this, a call to subroutine VRTIO (CALL VRTIO(IRET,LASRET,1)) stores IRET in row MDRR, column LASRET of the retrieval MD file. Then, if the plot option is in effect (IPLT .NE. 0), the retrieval box band 8 TBB is plotted on the video screen at the retrieval location in KOLOR=3 (normally yellow) via subroutine VASDIG.

Finally, SRAD jumps back to statement 230 and increments one retrieval box spacing to the right to process another retrieval box, unless one of three things happens. First, if the number of the just-completed retrieval (LASRET) equals the maximum number of retrievals allowed (MAXRET), SRAD jumps to statement 2390, where the retrieval MD file row header is rewritten, the VASTEXT file is updated and, following a jump to statement 2500, the program terminates. Second, if the process of incrementing to the right after a successful retrieval causes variable IEP to exceed LELEM (the rightmost element coordinate (x-direction) within the retrieval area for which retrievals can be processed), SRAD jumps to statement 2385, where the row header and VASTEXT file are again rewritten and updated, respectively, via VRTIO, after which SRAD moves to the next retrieval line to process that line's TBB data. For the third and final case, if the previous retrieval was the last one possible in the sounder area, SRAD concludes in a similar fashion to the second case above, except that instead of returning back to the beginning of DO LOOP 2400, SRAD exits 2400 and terminates. This third case concludes the discussion of SRAD in the N-STAR mode.

Returning now to the end of DO LOOP 590, if the retrieval type is clear or default, the maximum brightness temperature and the FOV in which the maximum brightness temperature occurs for a given retrieval box is computed for bands 6 and 8 in DO LOOP 600. Variables JSAV6 and JSAV store the FOV number(s) in which the maximum band 6 and 8 TBBs occur. After DO LOOP 600, if the maximum band 8 TBB is less than the previously-calculated threshold temperature TMN, or if either JSAV or JSAV6 = 0, SRAD moves one-half retrieval box spacing to the right and attempts a new retrieval.

Assuming that the threshold test passes, and that JSAV6/JSAV both have values, SRAD sets up "screening variables" in bands 4,5,6,8 and 9 (RLO4,RLO5, etc.) to delete cloud contaminated TBBs in bands 3,4,5,6,7,8,9,10 and 12 for all the FOVs in a given retrieval box. The actual screening is then done in DO LOOP 720.

Within this loop, different elements of the array LC are set to 0 to indicate cloud contamination in various VAS bands, depending on how the TBB's in bands 4, 5, 6 and 8 for a given FOV compare to the screening variables RL04, RL05, RL06, RL08 and RL09. In addition, as part of DO LOOP 720, DO LOOP 700 recalculates the arrays NSAM and AVG with the clearest screened radiance information. Following DO LOOP 720, DO LOOP 760 calculates the average retrieval box brightness temperature for each band, with the results being placed in the array VDAT.

At this point, if the retrieval type is clear (FORCE=10.--see statement 105), or if the retrieval type is default (FORCE=0.) and there are 5 or more clear band 7 FOVs, SRAD jumps to statement 790, where the expected error of the brightness temperature observations for each band are computed, after which the surface skin temperature is calculated via function VSKINT. Finally, the output buffer and plot option sequence is completed in the same fashion as the previously-discussed N-star retrieval path, except that the band 8 brightness temperature is plotted in KOLOR=2 (normally blue-green) instead of KOLOR=3.

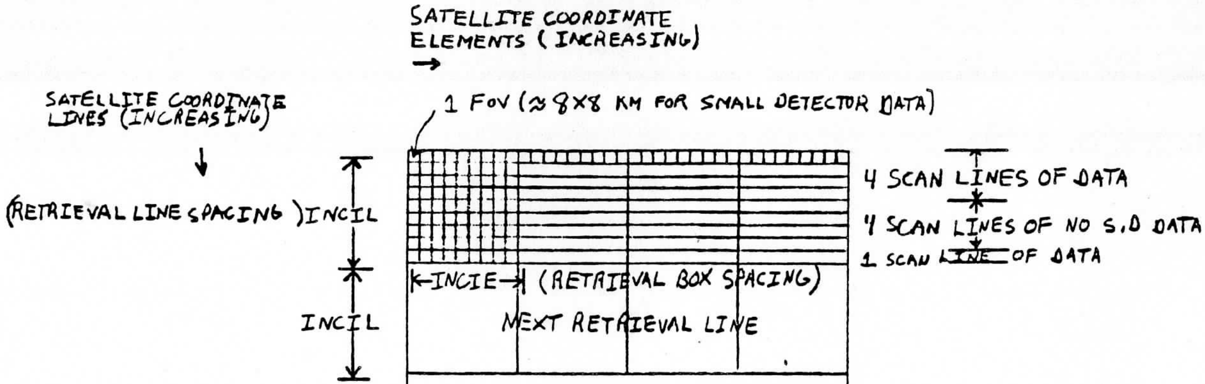
However, if the retrieval type is default and there are LESS THAN five clear band 7 FOVs, the retrieval path will hereafter follow the previously-described N-STAR retrieval path. In other words, the default retrieval path is identical to the clear path, as long as there are at least 5 clear band 7 FOVs within the retrieval box, but default retrievals will become N-star retrievals if there are .LT. 5 clear band 7 FOVs.

This concludes the discussion of program SRAD. A summary of the VAS retrieval algorithm is contained in Smith (1983).

DIAGRAM 2

TYPICAL RETRIEVAL AREAS

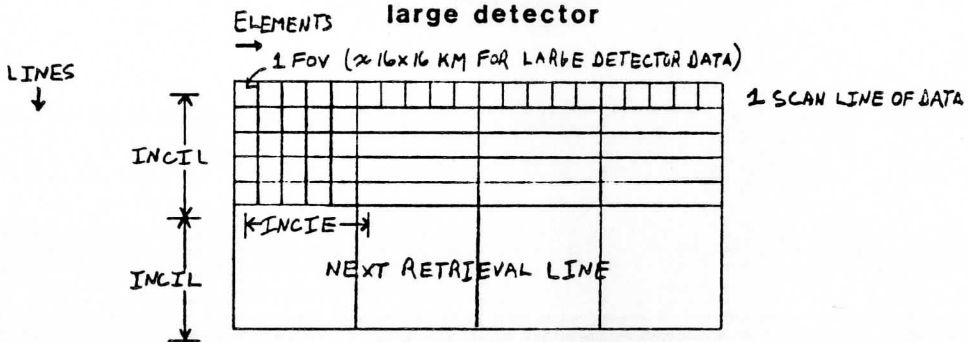
small detector



NOTES: 1.) DIMENSION OF RETRIEVAL BOX = 9x9 FOVS (NOT 11x11 AS IN VTPZ AND VTPW)

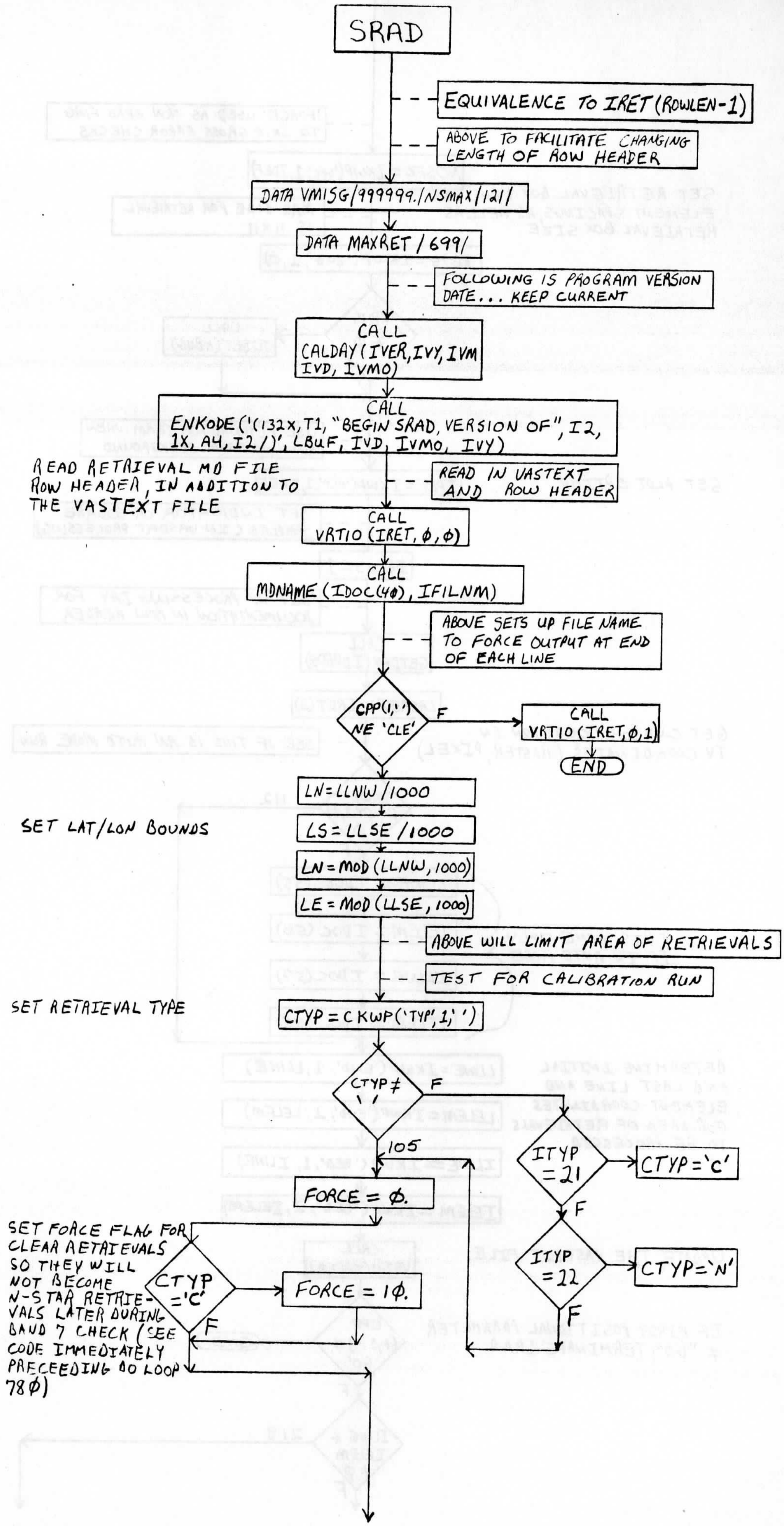
2.) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 81 FOVS. FOR THE FIRST RETRIEVAL LINE, THERE ARE 45 FOVS WHICH HAVE SMALL DETECTOR DATA (4x9 +1x9), MEANING THERE ARE 36 FOVS WHICH HAVE NO SMALL DETECTOR DATA. FOR THE NEXT RETRIEVAL LINE, THE NUMBERS ARE AGAIN 45 AND 36, RESPECTIVELY. IT IS ONLY POSSIBLE TO HAVE EITHER 4 OR 5 SCAN LINES OF VAS DATA WITHIN A GIVEN 9x9 RETRIEVAL BOX.

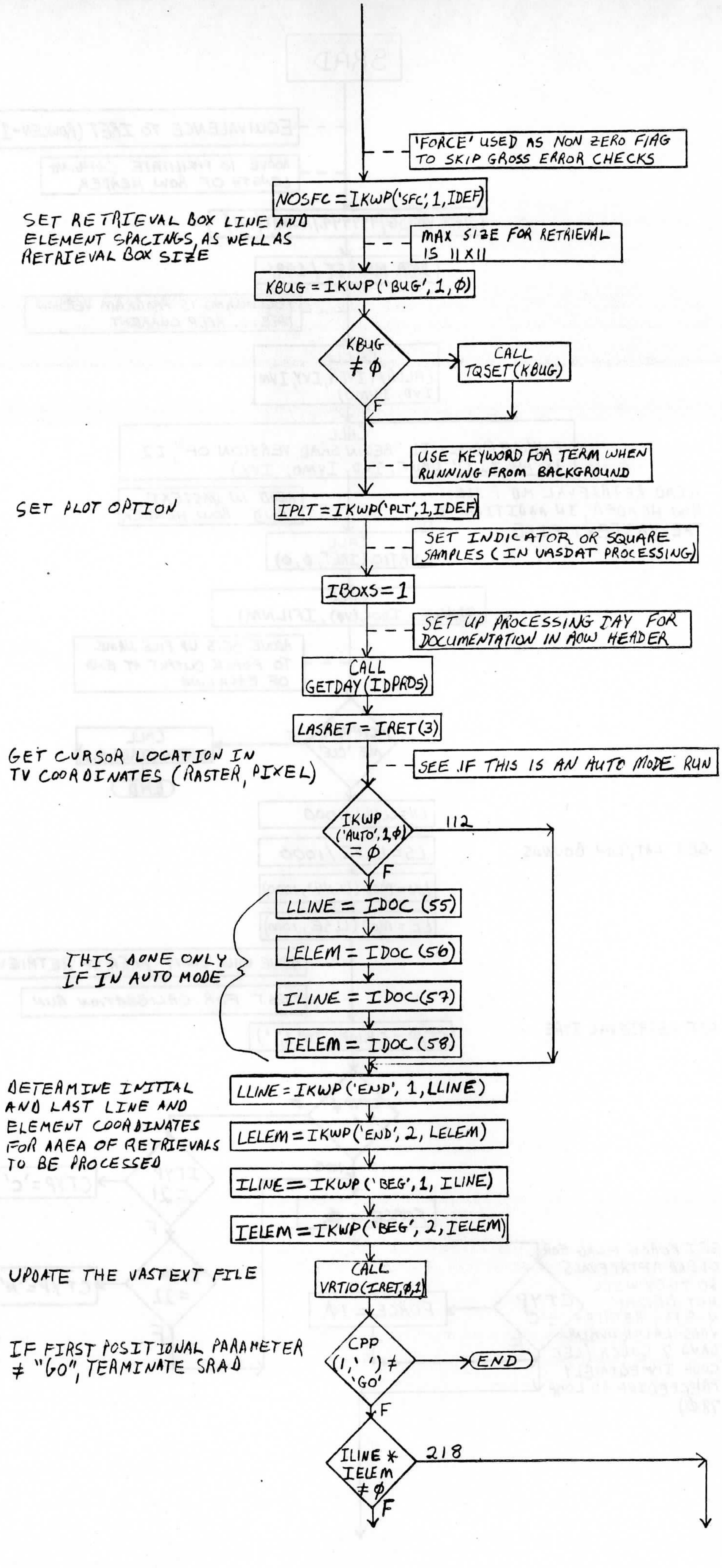
large detector



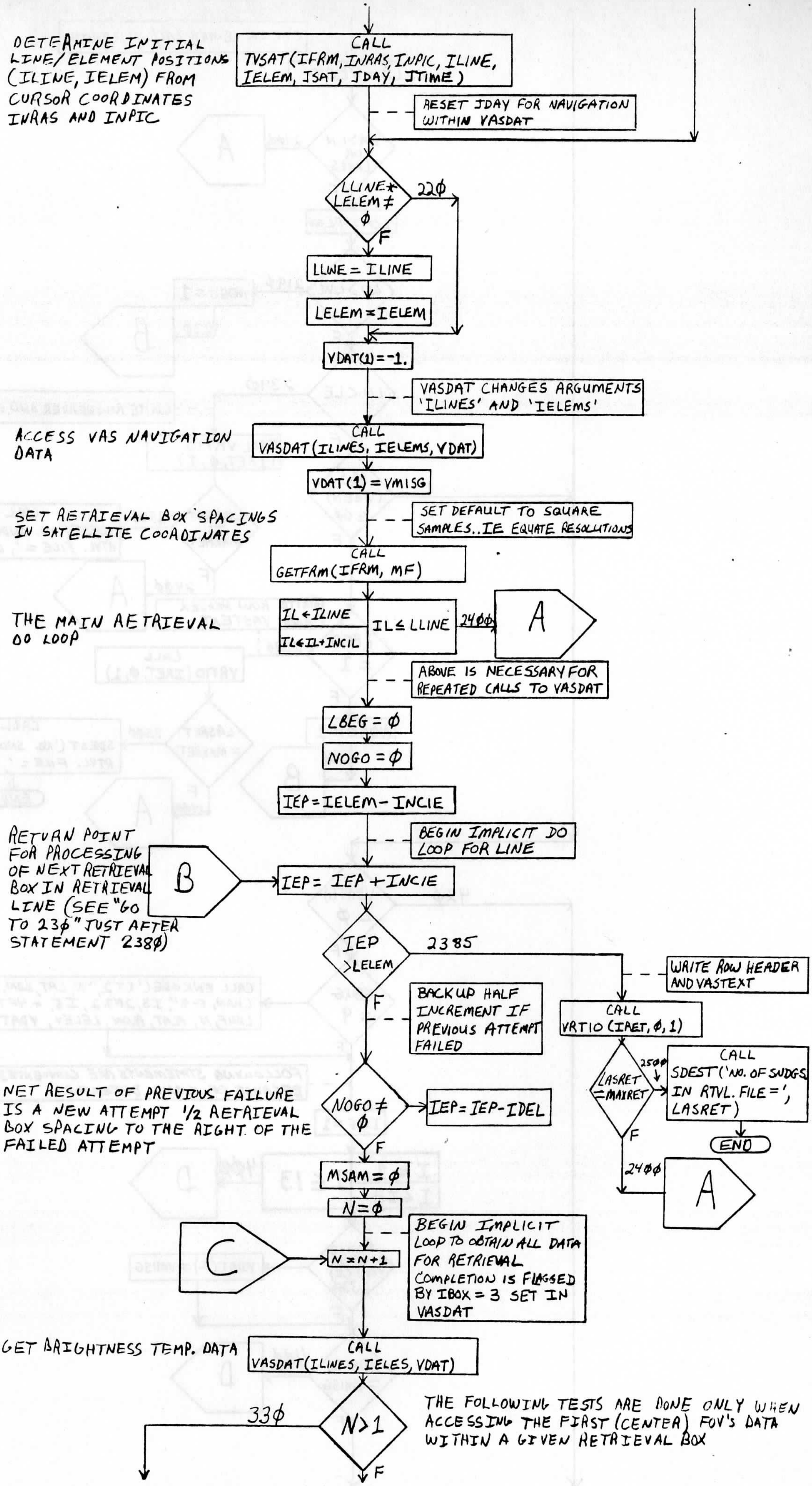
NOTES: 1.) DIMENSION OF RETRIEVAL BOX = 5x5 FOVS

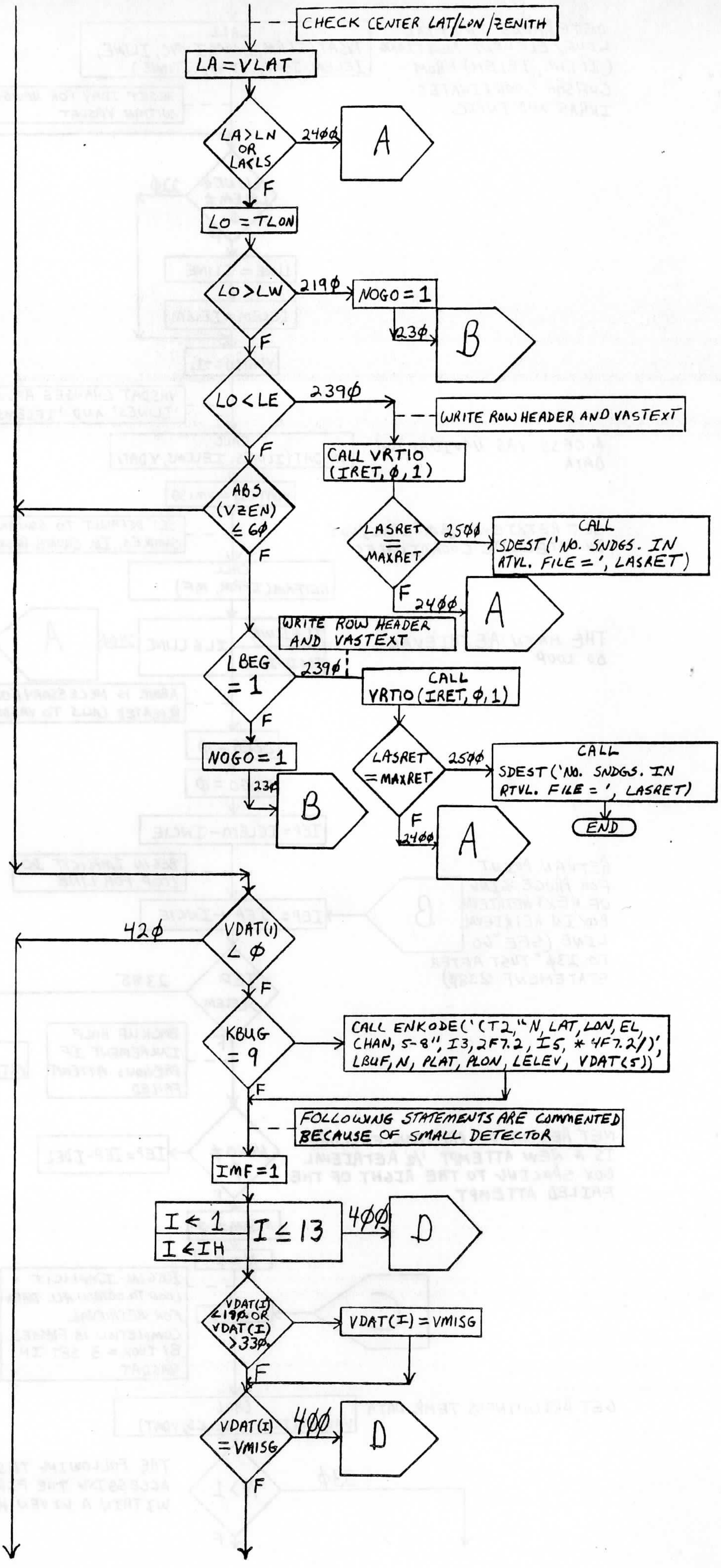
2.) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 25 FOVS, ALL OF WHICH CONTAIN VAS TBB DATA.

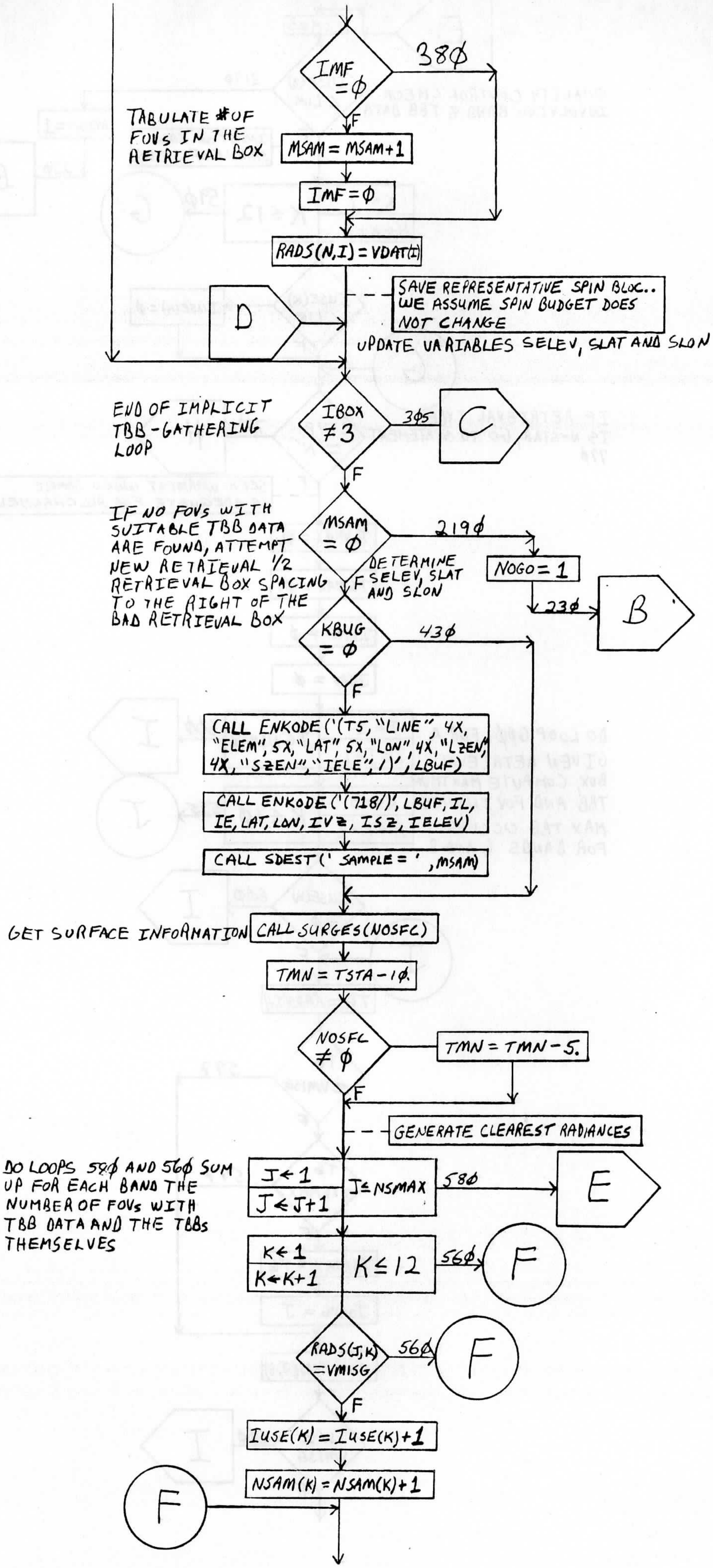

SRAD
EQUIVALENCE TO IRET (ROWLEN-1)
ABOVE TO FACILITATE CHANGING LENGTH OF ROW HEADER
DATA VMISG/999999./NSMAX/121/
DATA MAXRET / 699/
FOLLOWING IS PROGRAM VERSION DATE... KEEP CURRENT
CALL CALDAY (IVER, IVY, IVM IVD, IVMO)
CALL ENKODE ('(132X, T1, "BEGIN SRAD, VERSION OF", I2, 1X, A4, I2/)', LBUF, IVD, IVMO, IVY)
READ RETRIEVAL MD FILE ROW HEADER, IN ADDITION TO THE VASTEXT FILE
READ IN VASTEXT AND ROW HEADER
CALL VRTIO (IRET, ϕ, ϕ)
CALL MDNAME (IDOC(4ϕ), IFILNM)
ABOVE SETS UP FILE NAME TO FORCE OUTPUT AT END OF EACH LINE
CPP(1,' ') NE 'CLE'
F
CALL VRTIO (IRET, ϕ, 1)
END
LN = LLNW / 1000
SET LAT/LON BOUNDS
LS = LLSE / 1000
LN = MOD (LLNW, 1000)
LE = MOD (LLSE, 1000)
ABOVE WILL LIMIT AREA OF RETRIEVALS
TEST FOR CALIBRATION RUN
SET RETRIEVAL TYPE
CTYP = CKWP ('TYP', 1, ' ')
CTYP ≠ ' '
F
105
ITYP = 21
CTYP = 'C'
F
FORCE = ϕ.
SET FORCE FLAG FOR CLEAR RETRIEVALS SO THEY WILL NOT BECOME N-STAR RETRIE-VALS LATER DURING BAND 7 CHECK (SEE CODE IMMEDIATELY PRECEEDING DO LOOP 78ϕ)
CTYP = 'C'
FORCE = 1ϕ.
F
ITYP = 22
CTYP = 'N'
F

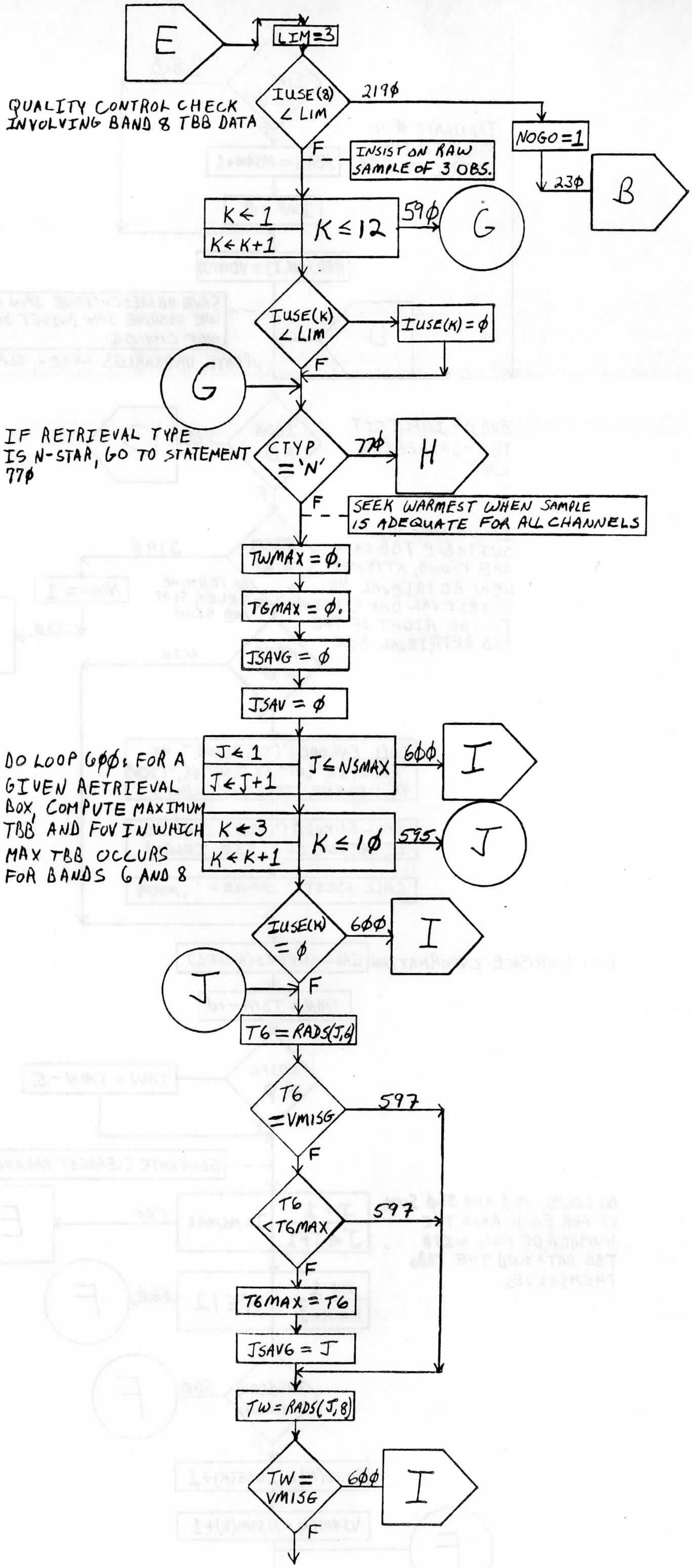


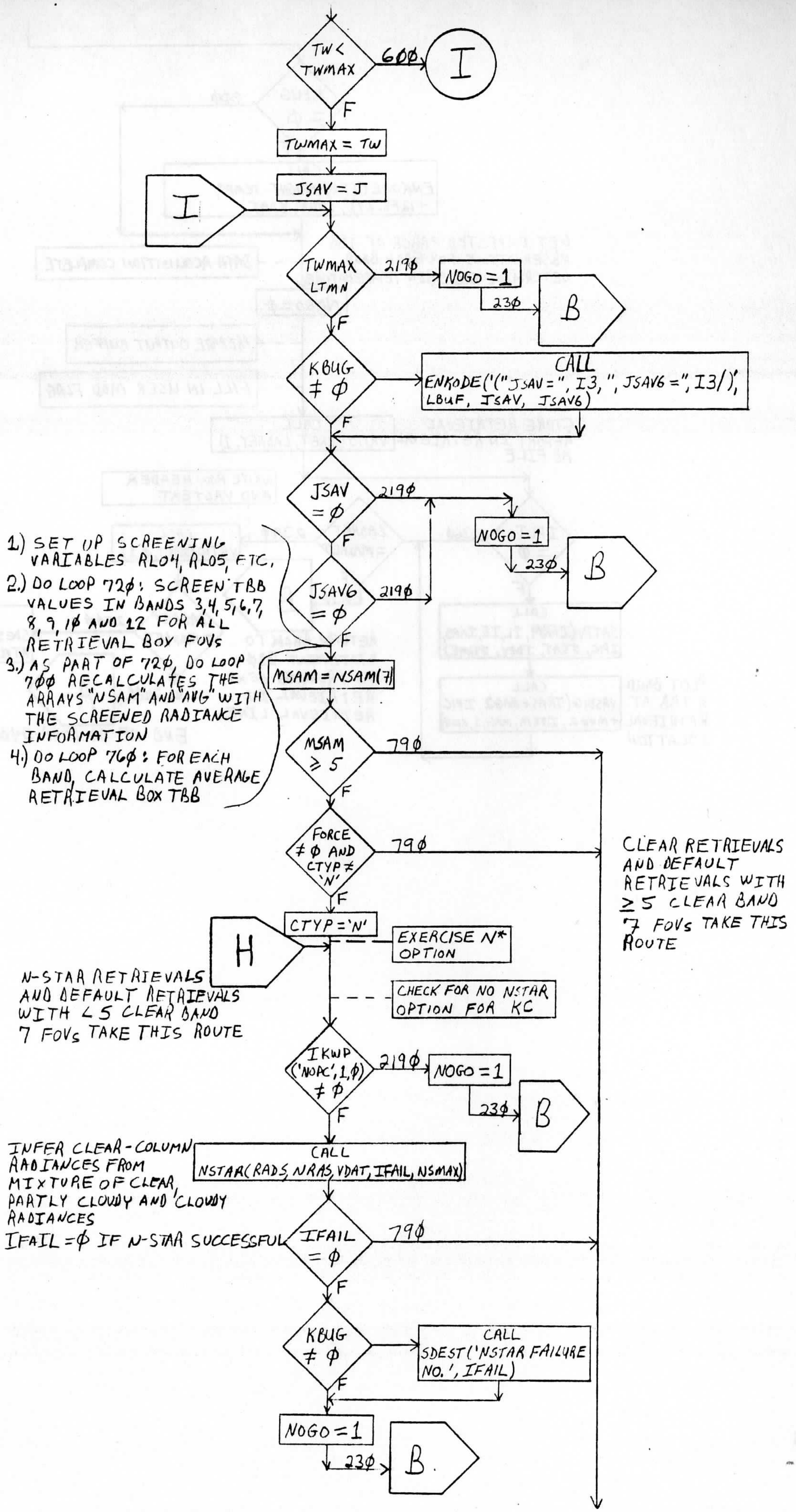
'FORCE' USED AS NON ZERO FLAG TO SKIP GROSS ERROR CHECKS
NOSFC = IKWP('SFC', 1, IDEF)
SET RETRIEVAL BOX LINE AND ELEMENT SPACINGS, AS WELL AS RETRIEVAL BOX SIZE
MAX SIZE FOR RETRIEVAL IS 11 X 11
KBUG = IKWP('BUG', 1, ϕ)
KBUG ≠ ϕ
CALL TQSET(KBUG)
F
USE KEYWORD FOR TERM WHEN RUNNING FROM BACKGROUND
SET PLOT OPTION
IPLT = IKWP('PLT', 1, IDEF)
SET INDICATOR OR SQUARE SAMPLES (IN VASDAT PROCESSING)
IBOXS = 1
SET UP PROCESSING DAY FOR DOCUMENTATION IN ROW HEADER
CALL GETDAY(IDPRDS)
LASRET = IRET(3)
GET CURSOR LOCATION IN TV COORDINATES (RASTER, PIXEL)
SEE IF THIS IS AN AUTO MODE RUN
IKWP ('AUTO', 1, ϕ) = ϕ
112
F
LLINE = IDOC (55)
LELEM = IDOC (56)
ILINE = IDOC (57)
IELEM = IDOC (58)
THIS DONE ONLY IF IN AUTO MODE
DETERMINE INITIAL AND LAST LINE AND ELEMENT COORDINATES FOR AREA OF RETRIEVALS TO BE PROCESSED
LLINE = IKWP('END', 1, LLINE)
LELEM = IKWP('END', 2, LELEM)
ILINE = IKWP('BEG', 1, ILINE)
IELEM = IKWP('BEG', 2, IELEM)
UPDATE THE VASTEXT FILE
CALL VRTIO(IRET, ϕ, 1)
IF FIRST POSITIONAL PARAMETER ≠ "GO", TERMINATE SRAD
CPP (1, ' ') ≠ 'GO'
END
F
ILINE * IELEM ≠ ϕ
218
F

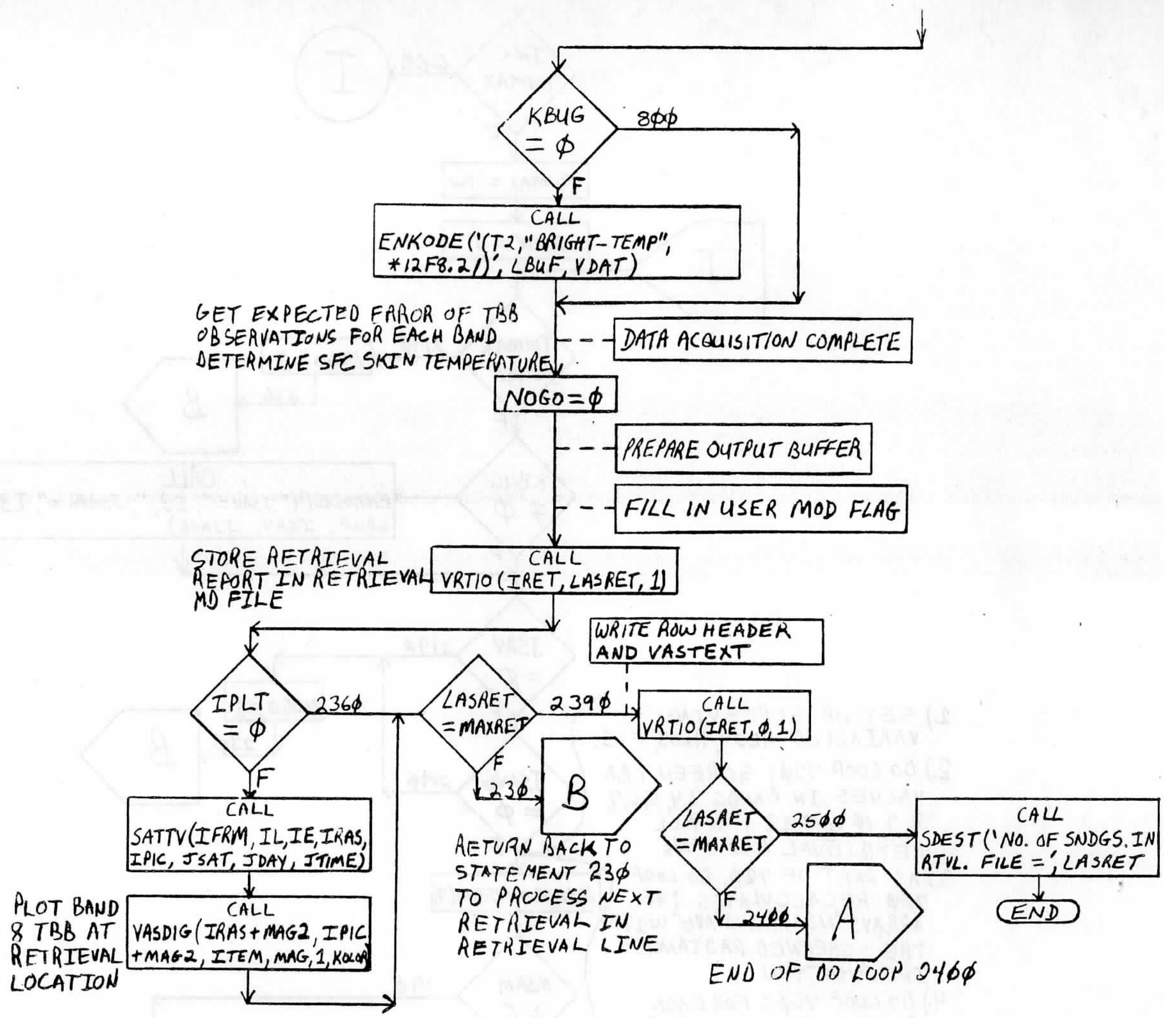

DETERMINE INITIAL LINE/ELEMENT POSITIONS (ILINE, IELEM) FROM CURSOR COORDINATES INRAS AND INPIC
CALL TVSAT(IFRM, INRAS, INPIC, ILINE, IELEM, ISAT, JDAY, JTIME)
RESET JDAY FOR NAVIGATION WITHIN VASDAT
LLINE ≠ LELEM ≠ ϕ
22ϕ
F
LLINE = ILINE
LELEM = IELEM
VDAT(1) = -1.
VASDAT CHANGES ARGUMENTS 'ILINES' AND 'IELEMS'
ACCESS VAS NAVIGATION DATA
CALL VASDAT(ILINES, IELEMS, VDAT)
VDAT(1) = VMISG
SET RETRIEVAL BOX SPACINGS IN SATELLITE COORDINATES
SET DEFAULT TO SQUARE SAMPLES..IE EQUATE RESOLUTIONS
CALL GETFRM(IFRM, MF)
THE MAIN RETRIEVAL DO LOOP
IL ← ILINE
IL ≤ LLINE
IL ← IL+INCIL
24ϕϕ
A
ABOVE IS NECESSARY FOR REPEATED CALLS TO VASDAT
LBEG = ϕ
NOGO = ϕ
IEP = IELEM - INCIE
BEGIN IMPLICIT DO LOOP FOR LINE
RETURN POINT FOR PROCESSING OF NEXT RETRIEVAL BOX IN RETRIEVAL LINE (SEE "GO TO 23ϕ" JUST AFTER STATEMENT 238ϕ)
B
IEP = IEP + INCIE
IEP > LELEM
2385
WRITE ROW HEADER AND VASTEXT
CALL VRTIO (IRET, ϕ, 1)
BACKUP HALF INCREMENT IF PREVIOUS ATTEMPT FAILED
F
LASRET = MAXRET
25ϕϕ
CALL SDEST('NO. OF SNDGS IN RTVL. FILE =', LASRET)
END
NET RESULT OF PREVIOUS FAILURE IS A NEW ATTEMPT 1/2 RETRIEVAL BOX SPACING TO THE RIGHT OF THE FAILED ATTEMPT
NOGO ≠ ϕ
IEP = IEP - IDEL
F
24ϕϕ
A
MSAM = ϕ
N = ϕ
BEGIN IMPLICIT LOOP TO OBTAIN ALL DATA FOR RETRIEVAL COMPLETION IS FLAGGED BY IBOX = 3 SET IN VASDAT
C
N = N+1
GET BRIGHTNESS TEMP. DATA
CALL VASDAT(ILINES, IELES, VDAT)
33ϕ
N > 1
F
THE FOLLOWING TESTS ARE DONE ONLY WHEN ACCESSING THE FIRST (CENTER) FOV'S DATA WITHIN A GIVEN RETRIEVAL BOX


CHECK CENTER LAT/LON/ZENITH
LA=VLAT
LA>LN OR LA<LS
2400
A
F
LO=TLON
LO>LW
2190
NOGO=1
230
B
F
LO<LE
2390
WRITE ROW HEADER AND VASTEXT
CALL VRTIO (IRET, Ø, 1)
F
ABS (VZEN) ≤ 6Ø
LASRET = MAXRET
2500
CALL SDEST('NO. SNDGS. IN ATVL. FILE =', LASRET)
F
2400
A
F
WRITE ROW HEADER AND VASTEXT
LBEG = 1
2390
CALL VRTIO(IRET, Ø, 1)
F
LASRET = MAXRET
2500
CALL SDEST('NO. SNDGS. IN ATVL. FILE =', LASRET)
END
NOGO=1
230
B
F
2400
A
420
VDAT(I) < Ø
F
KBUG = 9
CALL ENKODE('(T2,"N, LAT, LON, EL, CHAN, 5-8", I3, 2F7.2, I5, * 4F7.2/)', LBUF, N, PLAT, PLON, LELEV, VDAT(5))
F
FOLLOWING STATEMENTS ARE COMMENTED BECAUSE OF SMALL DETECTOR
IMF=1
I ← 1
I ← IH
I ≤ 13
400
D
VDAT(I) < 19Ø OR VDAT(I) > 33Ø
VDAT(I)=VMISG
F
VDAT(I) = VMISG
400
D
F


IMF = ϕ
380
F
TABULATE # OF FOVs IN THE RETRIEVAL BOX
MSAM = MSAM+1
IMF = ϕ
RADS(N,I) = VDAT(I)
SAVE REPRESENTATIVE SPIN BLOC.. WE ASSUME SPIN BUDGET DOES NOT CHANGE
D
UPDATE VARIABLES SELEV, SLAT AND SLON
END OF IMPLICIT TBB-GATHERING LOOP
IBOX ≠ 3
3ϕ5
C
F
IF NO FOVs WITH SUITABLE TBB DATA ARE FOUND, ATTEMPT NEW RETRIEVAL 1/2 RETRIEVAL BOX SPACING TO THE RIGHT OF THE BAD RETRIEVAL BOX
MSAM = ϕ
219ϕ
NOGO = 1
23ϕ
B
DETERMINE SELEV, SLAT AND SLON
F
KBUG = ϕ
43ϕ
F
CALL ENKODE('(T5, "LINE", 4X, "ELEM", 5X, "LAT", 5X, "LON", 4X, "LZEN", 4X, "SZEN", "IELE", /)', LBUF)
CALL ENKODE('(7I8/)', LBUF, IL, IE, LAT, LON, IVZ, ISZ, IELEV)
CALL SDEST(' SAMPLE = ', MSAM)
GET SURFACE INFORMATION
CALL SURGES(NOSFC)
TMN = TSTA - 1ϕ.
NOSFC ≠ ϕ
TMN = TMN - 5.
F
GENERATE CLEAREST RADIANCES
DO LOOPS 58ϕ AND 56ϕ SUM UP FOR EACH BAND THE NUMBER OF FOVs WITH TBB DATA AND THE TBBs THEMSELVES
J ← 1
J ← J+1
J ≤ NSMAX
58ϕ
E
K ← 1
K ← K+1
K ≤ 12
56ϕ
F
RADS(J,K) = VMISG
56ϕ
F
F
IUSE(K) = IUSE(K)+1
NSAM(K) = NSAM(K)+1
F


E
LIM=3
IUSE(8) < LIM
219∅
QUALITY CONTROL CHECK INVOLVING BAND 8 TBB DATA
NOGO=1
F
INSIST ON RAW SAMPLE OF 3 OBS.
23∅
B
K←1
K←K+1
K≤12
59∅
G
IUSE(K) < LIM
IUSE(K)=∅
F
G
IF RETRIEVAL TYPE IS N-STAR, GO TO STATEMENT 77∅
CTYP ='N'
77∅
H
F
SEEK WARMEST WHEN SAMPLE IS ADEQUATE FOR ALL CHANNELS
TWMAX=∅.
TGMAX=∅.
JSAVG=∅
JSAV=∅
DO LOOP 6∅∅: FOR A GIVEN RETRIEVAL BOX, COMPUTE MAXIMUM TBB AND FOV IN WHICH MAX TBB OCCURS FOR BANDS 6 AND 8
J←1
J←J+1
J≤NSMAX
6∅∅
I
K←3
K←K+1
K≤1∅
595
J
IUSE(K) =∅
6∅∅
I
J
F
T6=RADS(J,6)
T6 =VMISG
597
F
T6 <T6MAX
597
F
T6MAX=T6
JSAVG=J
TW=RADS(J,8)
TW= VMISG
6∅∅
I
F


TW < TWMAX
600
I
F
TWMAX = TW
JSAV = J
I
TWMAX LTMN
2190
NOGO = 1
230
B
F
KBUG ≠ 0
CALL ENKODE('("JSAV=", I3, ", JSAV6=", I3/)', LBUF, JSAV, JSAV6)
F
JSAV = 0
2190
NOGO = 1
230
B
F
JSAV6 = 0
2190
F
1.) SET UP SCREENING VARIABLES RLO4, RLO5, ETC.
2.) DO LOOP 720: SCREEN TBB VALUES IN BANDS 3,4,5,6,7, 8, 9, 10 AND 12 FOR ALL RETRIEVAL BOX FOVS
3.) AS PART OF 720, DO LOOP 700 RECALCULATES THE ARRAYS "NSAM" AND "AVG" WITH THE SCREENED RADIANCE INFORMATION
4.) DO LOOP 760: FOR EACH BAND, CALCULATE AVERAGE RETRIEVAL BOX TBB
MSAM = NSAM(7)
MSAM ≥ 5
790
F
FORCE ≠ 0 AND CTYP ≠ 'N'
790
CLEAR RETRIEVALS AND DEFAULT RETRIEVALS WITH ≥ 5 CLEAR BAND 7 FOVS TAKE THIS ROUTE
F
CTYP = 'N'
EXERCISE N* OPTION
H
N-STAR RETRIEVALS AND DEFAULT RETRIEVALS WITH < 5 CLEAR BAND 7 FOVS TAKE THIS ROUTE
CHECK FOR NO NSTAR OPTION FOR KC
IKWP ('NOPC', 1, 0) ≠ 0
2190
NOGO = 1
230
B
F
INFER CLEAR-COLUMN RADIANCES FROM MIXTURE OF CLEAR, PARTLY CLOUDY AND CLOUDY RADIANCES
CALL NSTAR(RADS, NRAS, VDAT, IFAIL, NSMAX)
IFAIL = 0 IF N-STAR SUCCESSFUL
IFAIL = 0
790
F
KBUG ≠ 0
CALL SDEST('NSTAR FAILURE NO.', IFAIL)
F
NOGO = 1
230
B


KBUG = ϕ
8ϕϕ
F
CALL
ENKODE('(T2,"BRIGHT-TEMP", *12F8.2/)', LBUF, VDAT)
GET EXPECTED ERROR OF TBB OBSERVATIONS FOR EACH BAND
DETERMINE SFC SKIN TEMPERATURE
DATA ACQUISITION COMPLETE
NOGO=ϕ
PREPARE OUTPUT BUFFER
FILL IN USER MOD FLAG
STORE RETRIEVAL REPORT IN RETRIEVAL MD FILE
CALL
VRTIO(IRET, LASRET, 1)
WRITE ROW HEADER AND VASTEXT
IPLT = ϕ
236ϕ
F
LASRET = MAXRET
239ϕ
F
23ϕ
B
CALL
VRTIO(IRET, ϕ, 1)
RETURN BACK TO STATEMENT 23ϕ TO PROCESS NEXT RETRIEVAL IN RETRIEVAL LINE
CALL
SATTV(IFRM, IL, IE, IRAS, IPIC, JSAT, JDAY, JTIME)
PLOT BAND 8 TBB AT RETRIEVAL LOCATION
CALL
VASDIG(IRAS+MAG2, IPIC +MAG2, ITEM, MAG, 1, KOLOR)
LASRET = MAXRET
25ϕϕ
F
24ϕϕ
A
END OF DO LOOP 24ϕϕ
CALL
SDEST('NO. OF SNDGS. IN RTVL. FILE =', LASRET
END

Subroutines used by SRAD:

1) CALDAY-I
2) ENKODE-I
3) VRTIO-I
4) MDNAME-I
5) TQSET-I
6) GETDAY-I
7) TVSAT-I
8) VASDAT-I
9) GETFRM-I
10) SDEST-I
11) SURGES-I
12) NSTAR-I
13) SATTV-I
14) VASDIG-I
15) ENCODX
16) CONTNT
17) LTQ
18) ZECONV
19) MOVB
20) CLEANA
21) TQ
22) PRLINX
23) PRCLOS
24) PROPEN
25) PRPRPR
26) LOCK
27) PRWR
28) PRRD
29) UNLOCK
30) PRCL
31) POST
32) BLKA
33) ABORT
34) DOPEN
35) ENCODE
36) DREAD
37) DCLOSE
38) LWCLOS
39) MOVWC
40) LWMOP
41) LWPO
42) EDEST
43) MOVW
44) MOVCW
45) CLEANW
46) II
47) STC
48) DWRITE
49) TRMNL
50) TQMES
51) ITOC
52) NVINIT
53) GETNAV
54) EPOCH
55) GETGAM
56) VASNAV
57) DDEST
58) READD
59) OUTINT
60) PLNKIV
61) OPNA
62) LWOPEN
63) LWGET
64) SATEAR
65) SATPOS
66) ANGGET
67) SOLARP
68) ANGLES
69) READOF
70) RBYTSX
71) ZEROS
72) RDTRK
73) MOVC
74) IGNAME
75) RORDER
76) FILCLR
77) FILCLD
78) INITPL
79) QGDASH
80) DSHOFF
81) PLOT
82) DSHON
83) ENDPLT
84) ENPT
85) PENADD
86) PACK
87) SENOUT
88) ATOE
89) TEKPUT
90) LWNEWF
91) LWSO
92) PAGE
93) PENMOV
94) BOX
95) PENBEG
96) WALK
97) JMBWTF
98) GETSFV
99) VRTOPO

SRET

SRET uses the VAS brightness temperature (TBB) measurements prepared by SRAD and retrieves vertical profiles of temperature and moisture. The profiles are retrieved using the iterative retrieval method described in Smith (1983).

Initially, after the program version date has been evaluated and displayed on the CRT via subroutines CALDAY and ENKODE, and the VASTEXT file and retrieval MD file row header have been read via subroutine VRTIO, important parameters such as the type of guess (CGES-- statement 10), surface data analysis option (NOSFC), and retrieval type (CTYP--forced ("F"), default (no keyin or "B"), clear ("C"), or N-star ("N")) are determined. Note that VASTEXT file defaults IDOC(51) and (54), respectively, will be used for the guess type and surface analysis option if keyword parameters GSS and SFC are not entered as part of the keyin. This will result in a grid first guess, since IDOC(51) was set to 0 previously by program IDVA. In this discussion, I will concern myself mainly with how SRET runs in the "default" mode, since this is the retrieval type used most often, and I will also assume that the surface analysis option is being used. Other parameters determined during the pre-retrieval stage include the debug option (LCRT/KBUG) and the plot option (IPLT). Note the call to TQSET after LCRT has been set. TQSET will set the debug information output destination to 1 (CRT), 2 (local printer) or 3 (system printer), depending on the value of LCRT. Then, assuming a plot of the retrieval results is desired (plot will be of retrieved total-totals index), the cursor position in satellite coordinates (line (IL), element (IE)) is determined by subroutine TVSAT, the navigation file is opened via subroutine NVINIT, and subroutine GETFRM returns a 64-word directory pertaining to the VAS image, which includes things such as image frame number, image magnification, etc. Finally, the guess and retrieval MD file numbers (MDNG and MDNR, respectively) are gleaned from the VASTEXT file (IDOC(38) and (40), respectively), the temperature/moisture profile enhancement option (IENH) is set, and the numbers of the first and last retrieval(s) to be performed (NB and LASRET, respectively) are determined. The number of the last retrieval will be either the keyed-in second positional parameter value, or the number of reports indicated in the retrieval MD file row header (IRETD(3)).

At this point, the main retrieval DO LOOP (2400) is entered. This loop runs from the first through the last retrieval. For each retrieval, the data prepared by SRAD is read in by subroutine VRTIO, including the expected error of the brightness temperature observations, as well as the averaged brightness temperatures themselves, both for each VAS band. Then, assuming the user mod flag .LT. 0 (retrieval not done yet), DO LOOP 700 stores these values in the arrays EX and VDAT, respectively. In addition, the array VRAD stores computed Planck radiances for each band. Note also the array IUSE. This array contains "use" flags which indicate whether a given VAS band should be used in the retrieval (a value of 1) or not used because it is missing or has been intentionally set aside (a value of 0).

Now, after other quantities, such as satellite zenith angle

(VZEN), retrieval latitude/longitude (VLAT,VLON), the variable NOGO, etc. are set, the guess temperature and mixing ratio profiles for the retrieval are accessed via subroutine GESPRO. Assuming the guess is obtained without any problems (TGES(1) is NOT .LT. 0), if the guess surface temperature (TSTA) is more than 15 degrees warmer than variable TCK (approximately equal to the surface skin temperature as measured by the VAS), cloud contamination is indicated. If this is the case, the retrieval fails, and SRET jumps to statement 2200, where IRETD(3) (the user mod flag) is set to 7777. Then, a jump is taken to statement 2360 (remember, NOGO=1), where the array IRET (containing essentially only guess profiles) is put back into the retrieval MD file via subroutine VRTIO, and SRET moves on to process the next retrieval. Otherwise, assuming the cloud contamination test passes, the first pressure level below the surface is calculated in DO LOOP 400 (variable IS...maximum value = 40), and the lower portion of the guess temperature profile is adjusted to take into account the surface temperature (provided pressure level IS is .GE. 851 mb for a grid or profile first guess, and .GE. 501 mb for a climatology first guess (IGES=1)). Then, after some guess temperature and dewpoint data is prepared for the retrieval MD file in the arrays TGS and DGS, the guess information to this point (if KBUG is .GT. 0) is displayed on a given display device (=1 for CRT, 2 for local printer, 3 for system printer) via subroutine ENKODE. At this point, the surface dewpoint, dewpoint depression, pressure and mixing ratio are determined and (or) put into the variables TDSTA, DD, PSTA and WSTA, respectively.

The following 25 lines or so of code set a cloud flag (IFLG) either to 99 or 0, based on tests involving type of retrieval first guess, surface option, band 12 and 8 TBBs, surface skin temperature (TS), etc. The tests are used to determine if a cloud contamination problem exists for a given retrieval. If no cloud contamination problem is found, IFLG will be set to 0, while a cloud contamination problem will cause IFLG to assume a value of 99. If the flag is set to 99, the surface skin temperature calculated by SRAD (TS) will not be believed, and the temperature enhancement will not be done during the subsequent retrieval process. One of the tests, which is done only if the band 12 TBB value is to be used for the retrieval (IUSE(12) =1), compares the band 12 TBB value, which can be contaminated by excessive reflected sunlight from scattered cloud tops, with the surface skin temperature. If the band 12 TBB is more than 5 degrees Celsius warmer than TS, SRET will jump to 660, IFLG will be henceforth equal to 99, and the retrieval will be classified as overcast. A subsequent test compares the surface skin temperature to a threshold variable TMN. If the skin temperature is too cold, again indicating a cloud problem, IFLG remains equal to 99. If all the tests pass, and assuming the retrieval type is neither clear nor overcast (remember, this discussion deals with retrievals of type default), IFLG is set to 0, and the retrieval will subsequently be classified as clear.

Following the cloud contamination tests, and for cases where IFLG = 99, the pressure level immediately below the first level colder than the surface skin temperature (LS) is calculated in DO LOOP 920. Then, statement 980 (NLS=MIN0(IS,LS)) calculates variable NLS, which is used to define the lower boundary level for calculating radiance via the radiative transfer equation (RTE) in the following relative humidity (RH) calculations (DO LOOP 1100). In summary, for cloudy cases where IFLG = 99, NLS will be the minimum of the first level below the surface (IS) and the just calculated variable LS (result will usually be LS, the cloud tops), while for clear cases where IFLG = 0, NLS will be the minimum of the first level below the surface and the number 40 (LS set equal to NL=40 just before DO LOOP 920).

The RH computations in DO LOOP 1100 involve only bands 7, 9 and 10, which are the VAS bands affected primarily by water vapor absorption. Before DO LOOP 1100 is entered, however, a mixing ratio array (S) is partially filled, resulting in values for the upper 25 retrieval levels of .001 g/kg. The subsequent net result of DO LOOP 1100 is 3 RH values based on bands 7, 9 and 10 that are assumed to be valid at 800, 600 and 300 mb, respectively, and represent layer RH values for the layers 1000-800, 800-600 and 600-300 mb, respectively. The RH values are stored in the array IHUM. In addition, the final value returned for each band layer will be between (or equal to) .10 and 1 (10-100%).

In the code following DO LOOP 1100, but preceeding DO LOOP 1300, assuming IHUM(3) .NE. MISG, several miscellaneous operations are performed, affecting variables such as IHUM(1), IHUM(2) and RHS (surface relative humidity (RH)). Then, DO LOOP 1180 fills the array HUM with the final IHUM values.

At this point, DO LOOP 1300 is entered, in which the nearly-completed retrieval mixing ratio profile (the mixing ratio array evaluated here (w) is updated during the forthcoming temperature retrieval section) is determined from a linear interpolation involving the 3 layer-RH values evaluated earlier, the surface RH value, and saturated mixing ratio values calculated from the guess temperature profile. After this loop has been completed, the transmittances for bands 7-10 are determined via subroutine VASTAU, after which the total precipitable water vapor (based on the nearly-completed retrieval moisture profile) is calculated by subroutine PRECW.

Now, SRET begins the temperature profile retrieval process. First, the number of possible temperature profile iterations (ITLIM) is set. The default number of iterations are 3 for the grid or profile first guesses, and 10 for the climatology first guess. Following this, the variables IPASS and ITER are initialized to 0. IPASS will be used to keep track of the number of ITERATIVE PASSES, while ITER will keep track of the number of temperature iterations WITHIN A GIVEN ITERATIVE PASS. Then, DO LOOP 1380 sets the values of variables KC1 and KC2, depending on whether a given band 1-6 (CO2 absorption bands) is to be used for the forthcoming temperature retrieval (note the value of element I or J of the array IUSE, and remember that a 0 indicates missing data or that the band has been intentionally set aside). These variables will form the loop bounds for DO LOOP 1520. Following 1380, since ICB will almost always be .NE. 1, a test will be conducted at statement 1385. In this test, if the lower bounds for the CO2 bands being used has been set .GE. 3, and the upper bounds has been set .LE. 3, SRET will jump to statement 2200, where the user mod flag (IRETD(3)) will be set to 7777 (meaning "edit the report from the retrieval MD file") and, since NOGO is .NE. 0, a jump to 2360 will be taken, where the incomplete, non-retrieved report will be placed into the retrieval MD file via subroutine VRTIO. Then, DO LOOP 2400 will attempt to process the next retrieval.

As a side note, from this point (statement 1390) all the way to statement 1951, the reader must pay strict attention to the current value of the variables ITER and IPASS. This is so because this section of code is somewhat complex, and constant familiarity with both of these variables is a must for a fuller understanding of the code. In addition, the user should concurrently follow the outline immediately after this discussion (labelled "SRET SUMMARY FROM STATEMENT 1390 TO STATEMENT 1951"), as it should help settle any confusion he/she may have. This outline summarizes the different possible paths SRET may take in the code between statements 1390 and 1951.

Returning now to the main discussion, assuming the test at 1385 passes, SRET enters DO LOOP 1520. For the first major step in this loop, the transmittances for all CO2 bands are determined through all 40 retrieval levels. The first time through the loop, variable IA, the lowest pressure level to which radiative transfer calculations will be done, is calculated. For clear retrievals (IFLG .EQ. 0), this level will simply be equal to IS, the first pressure level below the surface. For cloudy retrievals, however, DO LOOP 1463 re-evaluates IA, assigning it the minimum of 2 values: the level immediately below the first level colder than the cloud tops, and the level IS. Note that IA is only determined once per retrieval (INIT set to 1, so SRET will henceforth jump to 1470).

After IA has been set, subroutine VASRTE calculates the radiance for a given band, also returning quantities such as the TBB calculated from the guess profile for that band, as well as the derivative quantities needed to calculate the weighting functions at each retrieval level for a given band. The weighting function calculations are then carried out in DO LOOP 1480. The final step in loop 1520 involves the calculation of the difference between the satellite- measured brightness temperature (VDAT(ICV)) and the brightness temperature calculated from the guess profile (TBB) (result known as the residual and stored in the array DELT). This information will eventually be used when evaluating the iterative form of the radiative transfer equation used to retrieve the temperature profile. Note that DELT values will NOT be calculated if either IUSE or KUSE(KC) equals 0.

Immediately following DO LOOP 1520, a data quality check is performed (only for the first temperature iteration, when ITER = 0, and for clear retrieval (IFLG=0) cases in which bands 5 and 6 are being used) to determine if there are problems with band 6. (As a side note, it was found empirically that the residual value for band 6 was occasionally erratic, compared to the band 4 and 5 residuals. Normally, the residuals were supposed to behave in a smoothly monotonic fashion (smoothly increasing/decreasing successively through bands 4-6-5), but they were found to periodically deviate from this behavior. Consequently, this test was included to screen out the band 6 data when it was found to have problems.) The test involves several of the above-mentioned DELT values (AB5, AB6 and AB4). If AB6 .GE. AB5 and AB4, band 6 will no longer be used in the retrieval (IUSE(6)=0). After this test has been completed, another test evaluates the value of the variable CCK (DELT(5)-DELT(6)). If the absolute value of this variable is .GT. 2.5, and variable RHCK .GT. 0.95, a cloud problem is indicated, and IFLG is reset to 1, indicating a cloudy retrieval.

At this point (statement 1540), band 6 (weighting function peak at approximately 850 mb) is deleted from the retrieval process if the retrieval surface pressure is .LE. 850 mb. The subsequent code before DO LOOP 1560 sets aside one band (either 5 or 6) for later use in determining whether the retrieval has been successful, and also evaluates different cases for skipping DO LOOP 1560, including declared clear retrieval type (CTYP=C), no surface option-climatology first guess, or IFLG equal to 0 (default retrieval type determined as being clear).

Within DO LOOP 1560, suspected cloudy retrievals are checked, and those bands whose observed TBB values are more than 0.25 degrees Celsius cooler than the TBB derived from the guess temperature profile are deleted from the retrieval (KUSE(J)=0). In addition, note how the variable ITYP is set to overcast (23) if any band is found to be cloud-contaminated. ITYP is a retrieval-type flag which is stored in each retrieval report, and was determined initially in program SRAD. After DO LOOP 1560 has been completed, a check is performed to see if band 4 (weighting function peak at approximately 450 mb) was deleted from the retrieval process due to cloud contamination (KUSE(4) .EQ. 0). If it was deleted, no retrieval is performed, and SRET jumps to statement 2200, where the same steps as discussed previously (approximately 3 pages back) are repeated.

Assuming the band 4 data is acceptable (KUSE(4).EQ.1), SRET moves to statement 1670 via an intermediate jump to statement 1662. After this, the program enters DO LOOP 1720, whose purpose is to retrieve the actual temperature profile by correcting the guess profile with the measured VAS radiances in each band. Within DO LOOP 1720, DO LOOP 1700 calculates the numerator and denominator of the temperature correction term to be added to the guess temperature at a given pressure level. Following DO LOOP 1720, the new iterated temperature for pressure level I is calculated and stored in the array T(I) via the equation T(I)=T(I)+FNUM/FDEN. In this equation, T(I) is the guess temperature at a given pressure level, while FNUM/FDEN is the temperature correction term. Note that this equation is exactly Equation 3 in Smith (1983).

After the temperature retrieval, if the first pressure level below the surface (IS) is greater than 850 mb, the lower levels of the retrieved temperature profile from 850 mb down to level IS are blended with the surface air temperature in DO LOOP 1780. Then, the temperature at all levels from IS down to 1000 mb (level 40) is set equal to TSTA in DO LOOP 1820, after which the retrieval mixing ratio, saturated mixing ratio and dewpoint for each pressure level between 300 and 1000 mb inclusive is calculated and stored in the arrays W, S and TD, respectively, in DO LOOP 1860. Finally, after a jump to statement 1885 (remember, SRET is still on the FIRST iterative pass), a check is performed to see if the temperature profile satisfies the convergence criterion. The actual test occurs after DO LOOP 1920 (IF(SUM.LT.0.025) GO TO 1940).

If the profile satisfies the convergence criterion after the first temperature iteration, which is possible in unusual cases, the program skips to statement 1940. Otherwise, assuming the convergence criterion is not satisfied, SRET returns to statement 1390. Note that ITER has been incremented in the process and is now equal to 1. SRET is now ready to begin the second temperature iteration of the first iterative pass.

Within DO LOOP 1520 this second time, only radiances for the CO2 bands being used (subroutine VASRTE), as well as weighting functions and DELT values, are recomputed (ITER=1, IPASS=0, INIT=1). No transmittance recalculations (subroutine VASTAU) are done. Then, since ITER .GT. 0, SRET jumps to statement 1580 and then eventually to 1670, after which a second temperature correction to the result of the first temperature iteration (DO LOOP 1720) is carried out. Following this DO LOOP, SRET proceeds like the first temperature iteration down through the rechecking of the convergence criterion. If the criterion fails a second time, control transfers once again to statement 1390, where the same steps as above are done again. The temperature iterations on the first iterative pass repeat until either the retrieved temperature profile satisfies the convergence criterion, or the number of iterations (ITER) becomes greater than or equal to the iteration limit (ITLIM).

Assuming the convergence criterion is satisfied before ITER becomes .GE. ITLIM, SRET skips to statement 1940, where level NLS is redefined using the new retrieved temperature profile via DO LOOP 1942 (am assuming IENH .EQ. 0). This level will be used in the upcoming water vapor profile enhancement. Once NLS has been determined, transmittances for bands 7-10 are recalculated (VASTAU), a water vapor profile enhancement is performed via subroutine VWRET, and the total precipitable water vapor for the now enhanced retrieval mixing ratio profile is calculated. Finally, the dewpoints for levels 26-40 (300-1000 mb) are determined once again in DO LOOP 1940. Then, IPASS is set equal to 1, meaning one iterative pass has now been completed, and SRET once again returns to statement 1390.

At this point, to simplify the discussion which follows, I will assume that the retrieval is clear, that is, that IFLG has been set to 0. After following through the clear case, the user hopefully will be able to track through the cloudy (IFLG .NE. 0) case separately by him/herself.

During the SECOND iterative pass of SRET, assuming the original guess was NOT a climatology first guess, DO LOOP 1520 recomputes transmittances for bands 4 and 5 only. Then, the radiance, weighting function and DELT values are calculated again, and SRET skips to statement 1580 (ITER .GT. 0). At this point, within DO LOOP 1661, if the absolute value of the DELT value for any CO2 band is .GT. 2.5, SRET jumps to statement 2200, and the retrieval is aborted. Assuming this problem does not occur, another convergence check is performed in DO LOOP 1665 (remember, IPASS=1 now) to see whether the absolute value of the difference between the observed and calculated brightness temperature for each CO2 band (DELT(I)) is .LE. to the expected noise of the brightness temperature observation for that band.

If all the bands pass this second convergence test, the retrieval process is done, and SRET jumps to statement 1951, where the post - retrieval process (calculating heights, stability indices, preparing the output array to be stored in the retrieval MD file, etc.) begins.

On the other hand, if any CO2 band fails the test, a temperature profile enhancement is attempted via subroutine VTRET (remember, IFLG=0). If the enhancement is successful (IOK set to +1 by VTRET), SRET jumps to statement 1730, where the lower portion of the temperature profile is again adjusted, and the mixing ratio, saturated mixing ratio and dewpoint arrays (W, S and TD) are all recalculated. IPASS is then incremented to 2, and the program returns once again to statement 1390. SRET is now on the THIRD iterative pass.

On the other hand, if the afore-mentioned temperature enhancement fails (IOK set to -1), SRET jumps to statement 1882, where IPASS is incremented to 2, and then jumps back to statement 1668, where a second temperature profile enhancement is attempted. Success on the second enhancement causes a jump to 1730 again and an eventual jump to 1390, while failure causes a repeat jump to 1882.

The temperature enhancements will continue until either a successful enhancement is accomplished (and a corresponding jump taken to statement 1390), or the variable IPASS is incremented to 4 (indicating a third failed temperature enhancement).

At this point, SRET will be at one of two places. It will be at statement 1390, beginning either the third (IPASS=2), fourth (IPASS=3) or fifth (IPASS=4) iterative pass, if a successful temperature enhancement was accomplished BEFORE a third temperature enhancement failure. On the other hand, if there was a third consecutive temperature enhancement failure, SRET will be at statement 1882. These two locations in the SRET SUMMARY are, respectively, "ITERATIVE PASS #3, #4 or #5" (the letter "E"), or the letter "D," subheading 2b2b.

At this point, it is felt that any further discussion of SRET, in terms of the retrieval process, is too cumbersome. As a result, the user is invited to examine the SRET SUMMARY, together with SRET itself, to get a complete understanding of the remaining retrieval process. I, however, will now turn my attention to the code beginning at statement 1951, which is the initial statement in the post-retrieval portion of SRET.

Even at this late point, it is possible for the retrieval to fail. The retrieval will fail if the absolute DELT value for band KSAV (either band 5 or 6; see DO LOOP 1920 and code immediately preceeding DO LOOP 1560) is greater than the variable GTST. Assuming this check passes, subroutine HTX calculates heights from the retrieved temperature and moisture profiles. Then, after the arrays PST, TST and TDST are filled, a call to subroutine SNDANL calculates several different stability indices, such as lifted index, total totals index, etc. Next, the heights computed via subroutine HTX (the array S) are read into the array U, and two thickness values (850-500 and 850-200 mb) are calculated and stored in variables DELZ1 and DELZ2, respectively. Finally, an averaged total-totals index is computed and stored in variable TOTLS/ITOTLS, the variable NOGO is set to 0, meaning a successful retrieval was performed, and SRET is ready to prepare an output retrieval report array for insertion into the retrieval MD file.

The final stage of the post-retrieval processing consists mainly of inserting the retrieval data into a buffer array IRETD (note equivalence of to the array IRET near code line 50), which is then inserted into column NN of the retrieval MD file via subroutine VRTIO. Some of the retrieval quantities placed in IRETD include user mod flag (=0 now), total precipitable water vapor, surface skin temperature, total totals index, relative humidities calculated using bands 7, 9 and 10, and geopotential heights. In addition, many of the retrieval temperatures and dewpoints are also stored. Finally, surface values of pressure, temperature, dewpoint, etc. are placed into IRETD to complete the output array.

Lastly, NRET (number of reports) is incremented by 1 and, if the plot option is in effect (IPLT=1), the total totals index is plotted at the retrieval location in one of two colors, which are usually blue-green or magenta, corresponding to clear (KOLOR=2) or overcast (KOLOR=1, IFLG .NE. 0) retrievals, respectively. After this, the final data is put into the retrieval MD file by VRTIO, and SRET returns to attempt the next retrieval.

SRET SUMMARY FROM STATEMENT 1390 TO STATEMENT 1951

ITERATIVE PASS #1 (IPASS=0)

A.) TEMPERATURE ITERATION #1: transmittance, lower level for radiative transfer, radiance, weighting functions (WF), DELT, band 6 test, cloudy channel test, temperature retrieval (T. ret.), convergence

successful? go to 1940

otherwise, go to 1390 and do TEMPERATURE ITERATION #2

B.) TEMPERATURE ITERATION #2: radiances, WF, DELT, T. ret., convergence

successful? go to 1940

otherwise, go to 1390 and do TEMPERATURE ITERATION #3

Temperature iterations are done like TEMPERATURE ITERATION #2 until convergence satisfied, or number of TEMPERATURE ITERATION becomes .GE. ITLIM (iteration limit). In either case, SRET will end up at statement 1940.

.

.

.

C.) 1940: first moisture enhancement

total precipitable water vapor (TPWV) calculations

set IPASS to 1 and go to 1390 for SECOND ITERATIVE PASS

ITERATIVE PASS #2 (IPASS=1) (assume retrieval is clear (IFLG=0) from this point on)

D.) transmittance for bands 4 and 5 only, radiance, WF, DELT, test DELT values (DO LOOP 1661) and quit retrieval if too poor; otherwise, test DELT values again (DO LOOP 1665)

1.) if all CO2 band DELT values small enough, retrieval processing done, and begin post-retrieval code at 1951

2.) otherwise, if failure in any CO2 band: first temperature profile enhancement

2a.) if temperature enhancement successful, go to 1730, adjust lower retrieval levels, get mixing ratios (W), saturated mixing ratios (S) and dewpoints (TD), set IPASS to 2 and go to 1390 to begin ITERATIVE PASS #3

2b.) if not successful, go to 1882, set IPASS to 2, and do second temperature enhancement

2b1.) if second temperature enhancement successful, do same as 2a above, but IPASS will = 3 when jump is taken to statement 1390; that is, SRET will be on ITERATIVE PASS #4

2b2.) if temperature enhancement fails, go to 1882, increment IPASS to 3, and do third temperature enhancement

2b2a.) if third temperature enhancement successful, do same as 2a above, but IPASS will = 4 when jump is taken to statement 1390; that is, SRET will be on ITERATIVE PASS #5

2b2b.) if temperature enhancement fails, go to 1882, increment IPASS to 4, set IOK to 0, raise ITLIM by 3, go to 1670, do T. ret., low level adjustments, get W, S and TD, check convergence criterion again successful? go to 1940 and eventually end up at 1951

unsuccessful? return to statement 1390, radiances, WF, DELT, 1661 test, T. ret., low level adjustments, W, S and TD, convergence.

Temperature iterations done until convergence criterion satisfied, or number of TEMPERATURE ITERATION becomes .GE. ITLIM (the iteration limit). In either case, SRET eventually ends up at statement 1940.

.

.

.

1940: second moisture enhancement

TPWV calculations

jump to 1951 to begin post-retrieval code

ITERATIVE PASS #3, #4 or #5 (IPASS=2, 3 or 4)

E.) at statement 1390, no transmittance, WF, DELT, 1661 test

1.) if IPASS = 2 or 3, do 1665 test:

a.) if all bands converged, retrieval processing done, and begin post-retrieval processing at 1951

b.) otherwise, if a failure in any CO2 band, do another temperature enhancement

b1.) if temperature enhancement successful: low level adjustment, W, S and TD, increment IPASS to 3 or 4

b1a.) if IPASS=3, go to 1390, radiance, WF, DELT, 1661 test, 1665 band convergence test

b1a1.) successful? go to 1951 and begin post-retrieval code

b1a2.) unsuccessful? another temperature enhancement

b1a2a.) if temperature enhancement successful, low level adjustment, W, S and TD, increment IPASS to 4, go to 1390, radiance, WF, DELT, 1661 test, jump to 1940 and then end up at 1951

b1a2b.) if temperature enhancement unsuccessful, increment IPASS to 4, set IOK to 0, raise ITLIM by 3, go to 1670, T. ret., low level adjustments, W, S and TD, convergence criterion again

b1a2b1.) successful? go to 1940 and end up at 1951

b1a2b2.) unsuccessful? return to 1390, radiances, WF, DELT, 1661 test, T. ret., low level adjustments, W, S and TD, convergence criterion test

Temperature iterations are done until convergence criterion satisfied, or the ITERATION

NUMBER becomes .GE. ITLIM (iteration limit). In either case, SRET eventually ends up at statement 1940 and then 1951.

b1b.) if IPASS=4, go to 1390, radiance, WF, DELT, 1661 test, jump to 1940, and then end up eventually at 1951

b2.) if temperature enhancement unsuccessful, increment IPASS to 3 or 4

b2a.) if IPASS=3, another temperature enhancement

b2a1.) if temperature enhancement successful, low level adjustments, W, S and TD, increment IPASS to 4, go to 1390, radiance, WF, DELT, 1661 test, jump to 1940, and then end up eventually at 1951

b2a2.) if unsuccessful, increment IPASS to 4, IOK=0, ITLIM raised by 3, go to 1670, T. ret., low level adjustments, W, S and TD, convergence criterion again

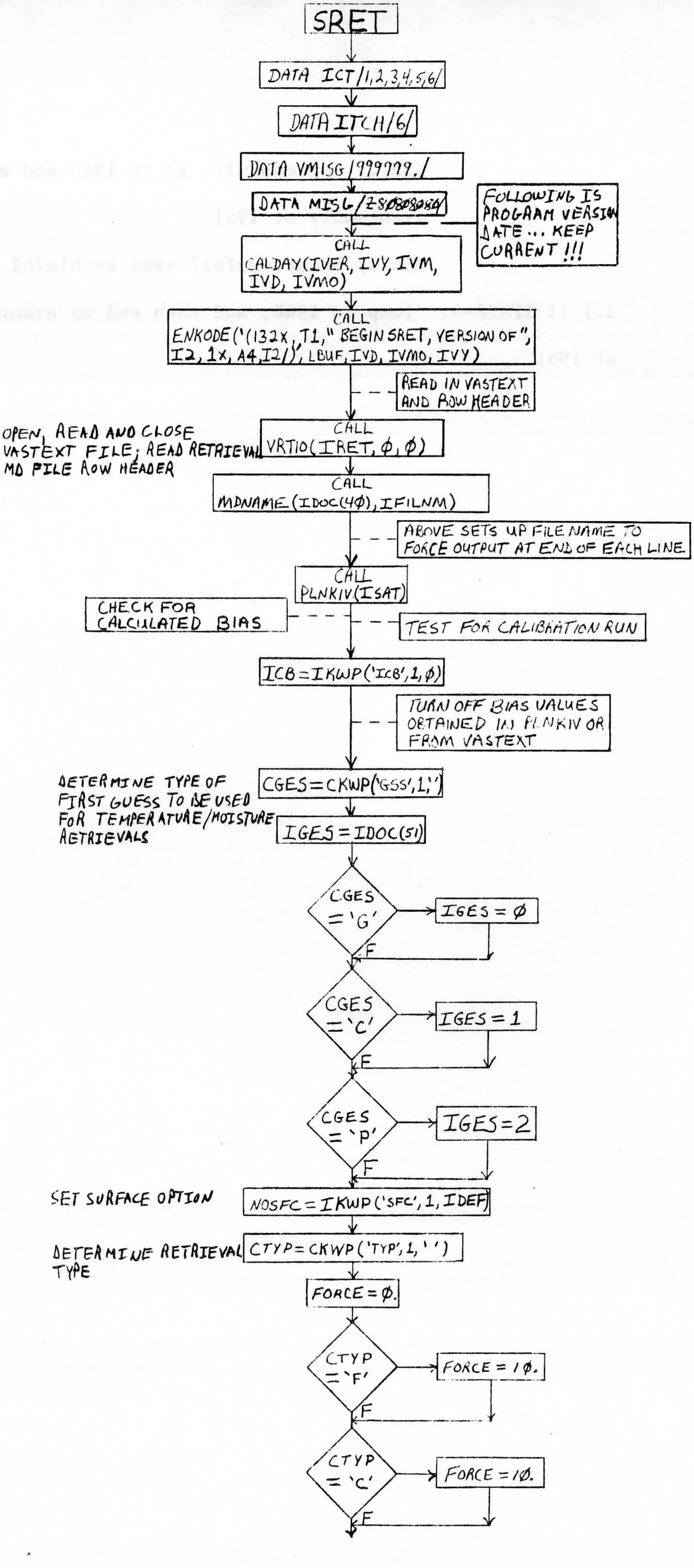
b2a2a.) successful? go to 1940 and end up at 1951

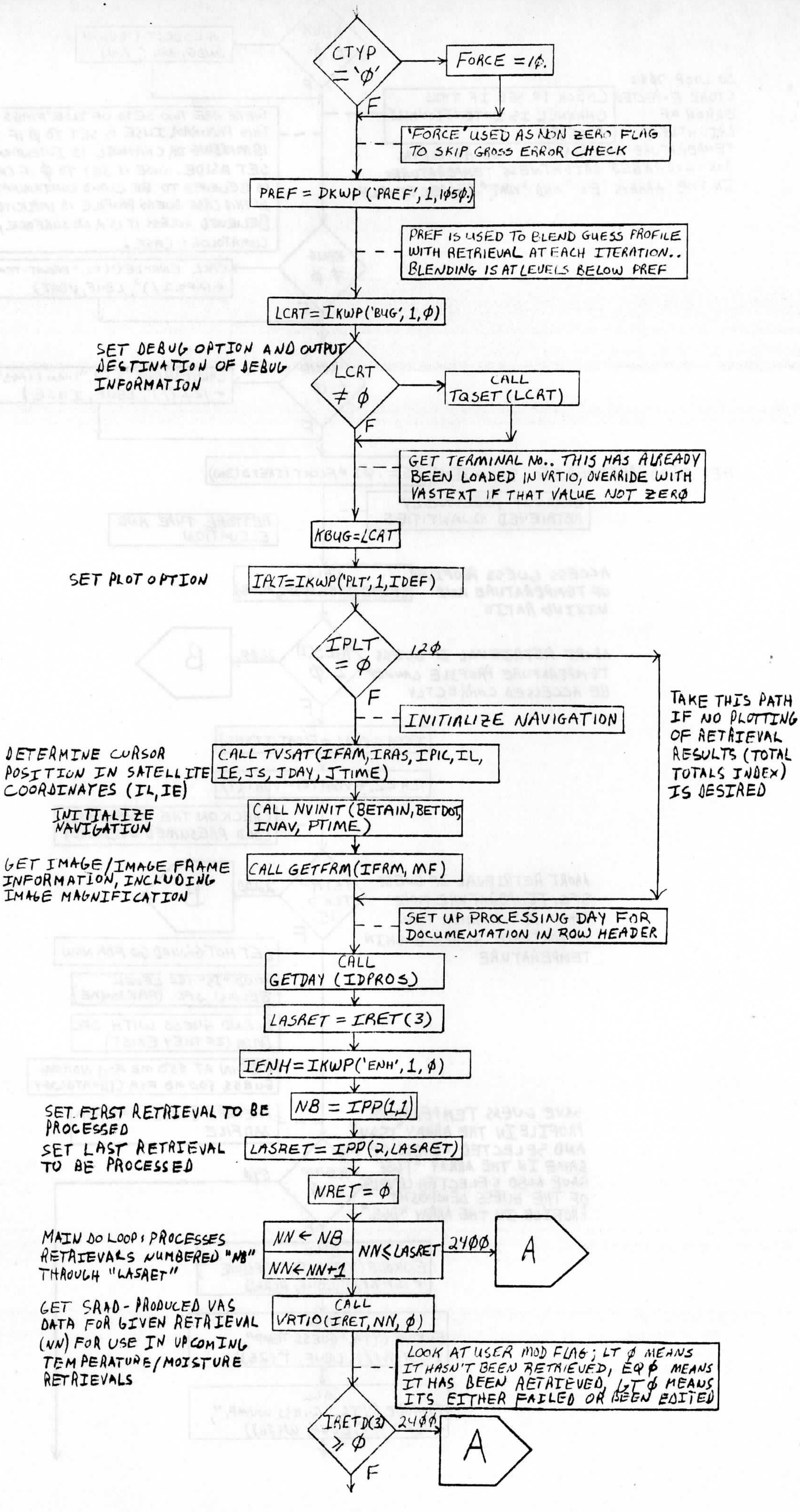
b2a2b.) unsuccessful? same as b1a2b2 above

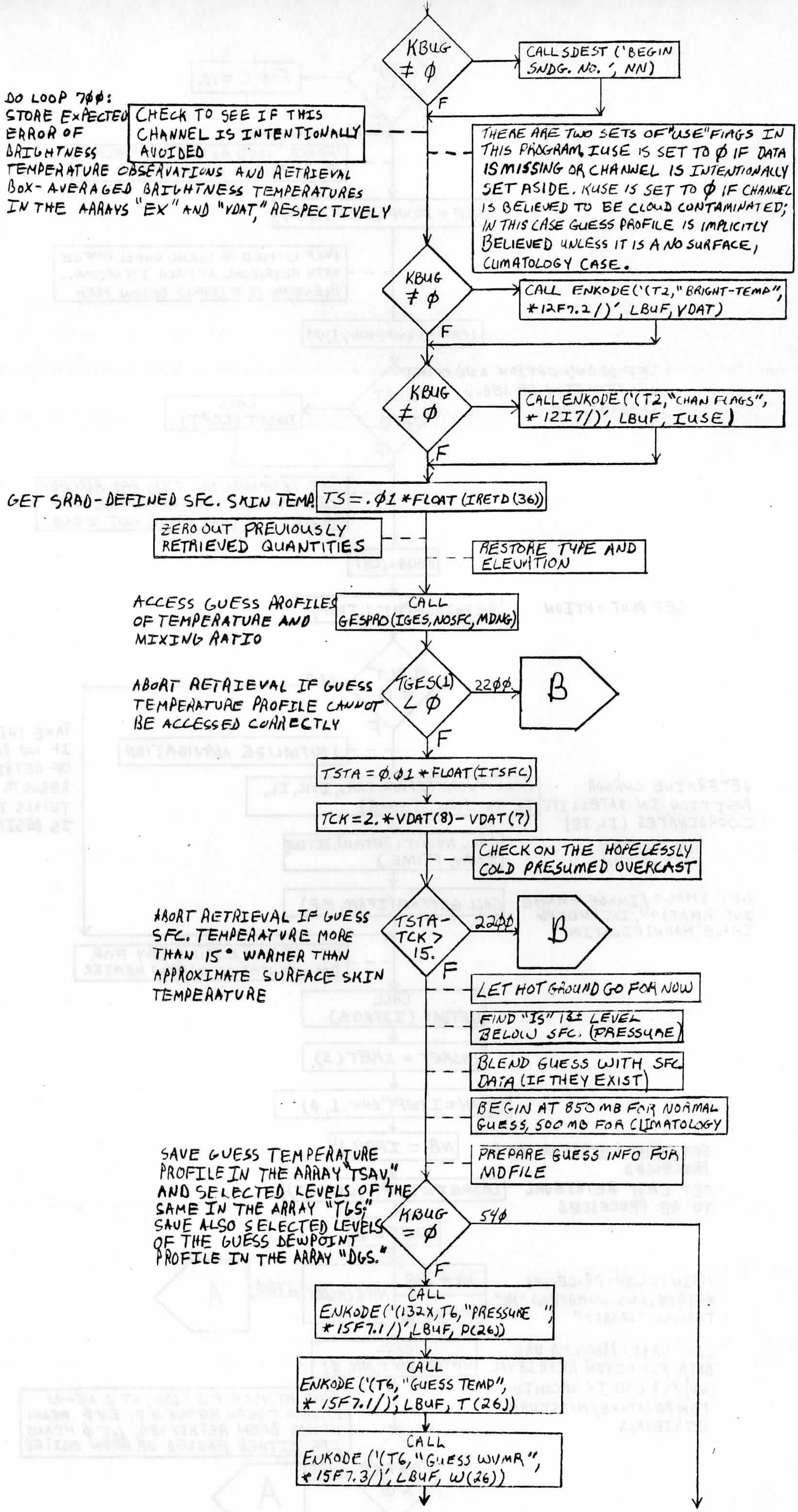
b2b.) if IPASS=4, set IOK=0, ITLIM raised by 3, go to 1670, T. ret., low level adjustments, W, S and TD, convergence criterion again b2b1.) successful? go to 1940 and end up eventually at 1951

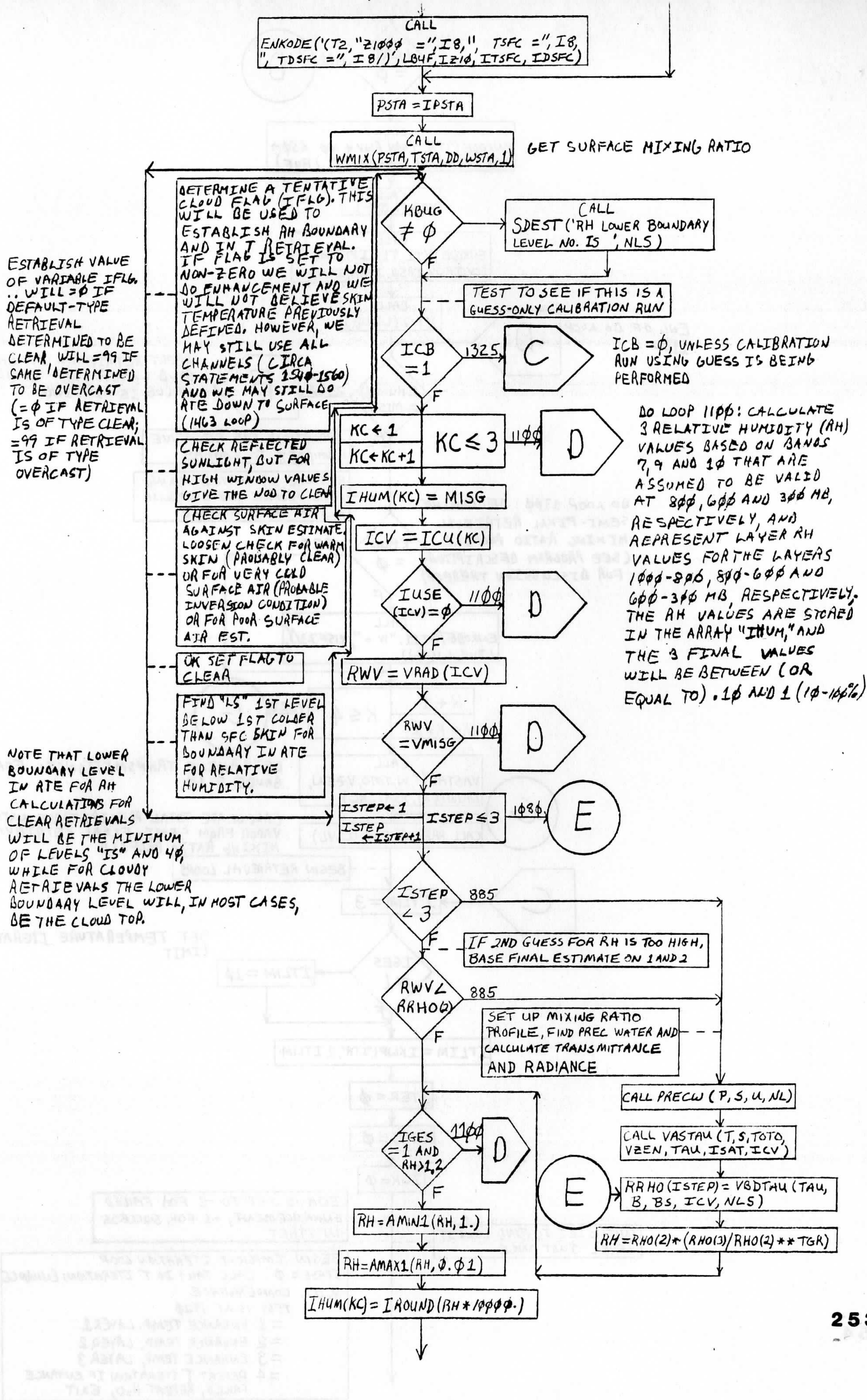
b2b2.) unsuccessful? same as b1a2b2 above

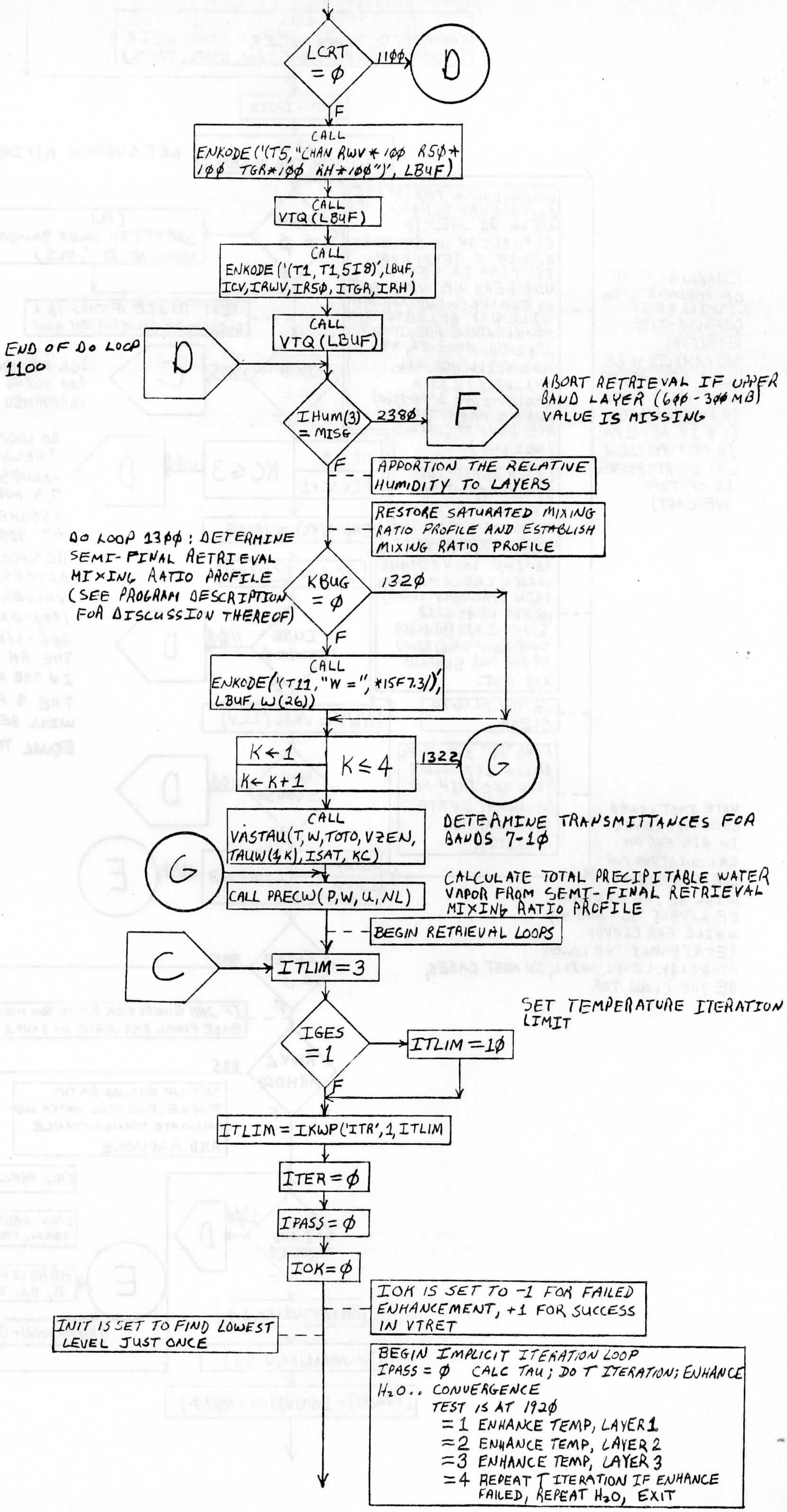
2.) if IPASS=4, jump to 1940, and then end up eventually at 1951

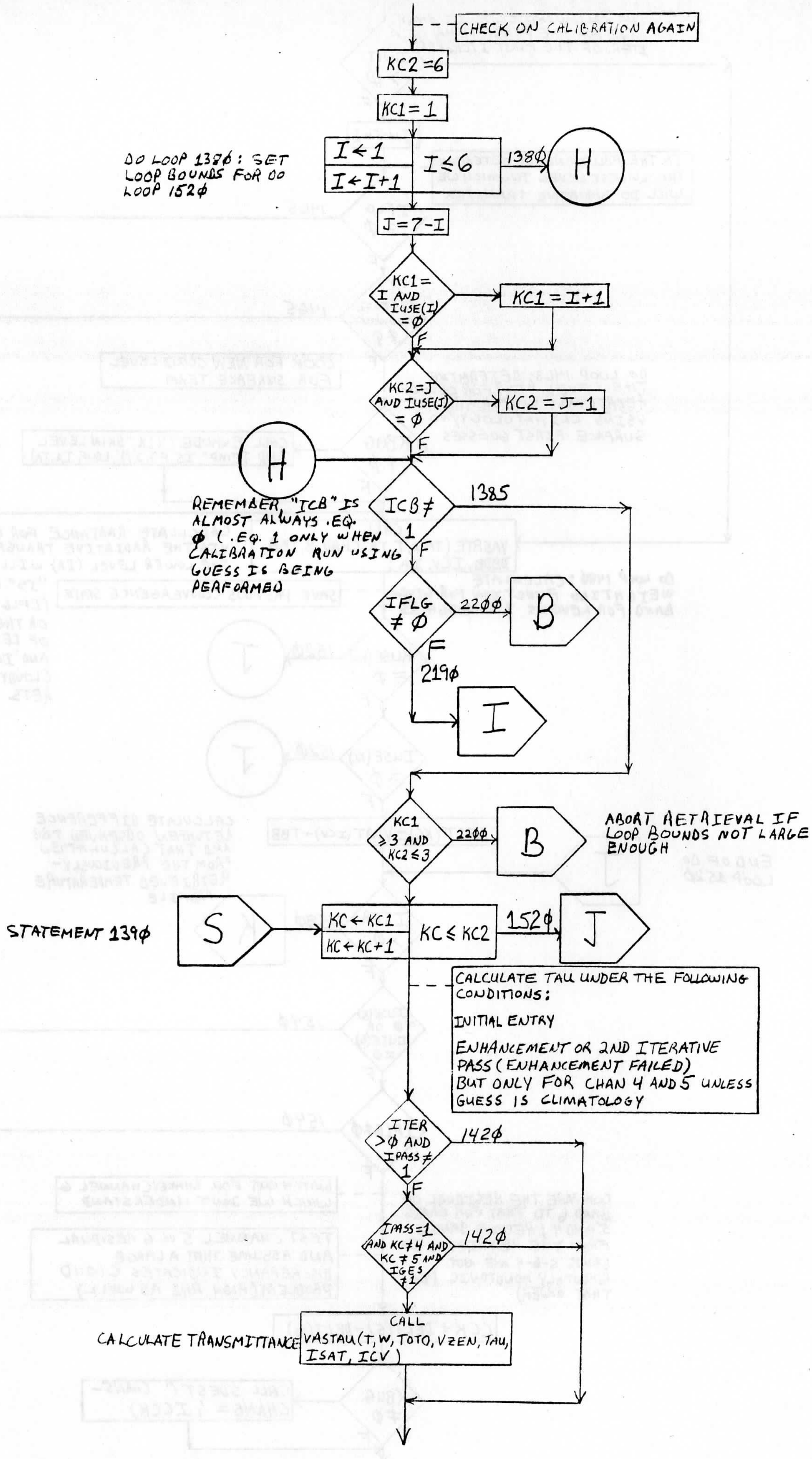

SRET
DATA ICT/1,2,3,4,5,6/
DATA ITCH/6/
DATA VMISG/999999./
DATA MISG/Z80808080/
FOLLOWING IS PROGRAM VERSION DATE ... KEEP CURRENT !!!
CALL CALDAY(IVER, IVY, IVM, IVD, IVMO)
CALL ENKODE('(132X, T1," BEGIN SRET, VERSION OF", I2, 1X, A4, I2)', LBUF, IVD, IVMO, IVY)
READ IN VASTEXT AND ROW HEADER
OPEN, READ AND CLOSE VASTEXT FILE; READ RETRIEVAL MD FILE ROW HEADER
CALL VRTIO(IRET, Ø, Ø)
CALL MDNAME(IDOC(4Ø), IFILNM)
ABOVE SETS UP FILE NAME TO FORCE OUTPUT AT END OF EACH LINE
CALL PLNKIV(ISAT)
CHECK FOR CALCULATED BIAS
TEST FOR CALIBRATION RUN
ICB=IKWP('ICB',1,Ø)
TURN OFF BIAS VALUES OBTAINED IN PLNKIV OR FROM VASTEXT
DETERMINE TYPE OF FIRST GUESS TO BE USED FOR TEMPERATURE/MOISTURE RETRIEVALS
CGES=CKWP('GSS',1,' ')
IGES=IDOC(51)
CGES = 'G'
IGES = Ø
F
CGES = 'C'
IGES = 1
F
CGES = 'P'
IGES=2
F
SET SURFACE OPTION
NOSFC = IKWP('SFC', 1, IDEF)
DETERMINE RETRIEVAL TYPE
CTYP = CKWP('TYP', 1, ' ')
FORCE = Ø.
CTYP = 'F'
FORCE = 1Ø.
F
CTYP = 'C'
FORCE = 1Ø.
F

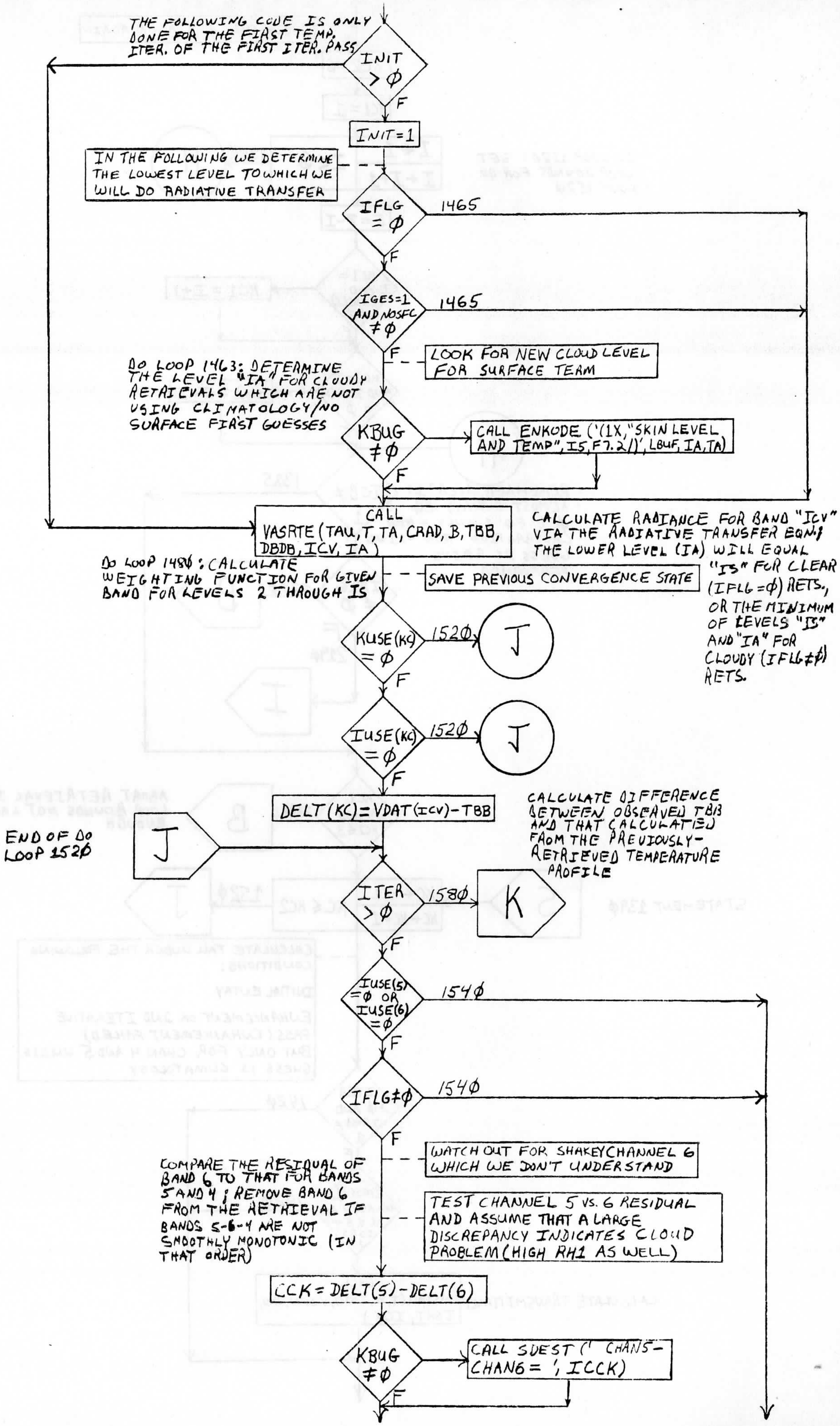

CTYP = 'ϕ'
FORCE = 1ϕ.
F
'FORCE' USED AS NON ZERO FLAG TO SKIP GROSS ERROR CHECK
PREF = DKWP('PREF', 1, 1ϕ5ϕ.)
PREF IS USED TO BLEND GUESS PROFILE WITH RETRIEVAL AT EACH ITERATION.. BLENDING IS AT LEVELS BELOW PREF
LCRT = IKWP('BUG', 1, ϕ)
SET DEBUG OPTION AND OUTPUT DESTINATION OF DEBUG INFORMATION
LCRT ≠ ϕ
CALL TQSET (LCRT)
F
GET TERMINAL NO.. THIS HAS ALREADY BEEN LOADED IN VRTIO, OVERRIDE WITH VASTEXT IF THAT VALUE NOT ZERϕ
KBUG = LCAT
SET PLOT OPTION
IPLT = IKWP('PLT', 1, IDEF)
IPLT = ϕ
12ϕ
F
INITIALIZE NAVIGATION
TAKE THIS PATH IF NO PLOTTING OF RETRIEVAL RESULTS (TOTAL TOTALS INDEX) IS DESIRED
DETERMINE CURSOR POSITION IN SATELLITE COORDINATES (IL, IE)
CALL TVSAT(IFAM, IRAS, IPIC, IL, IE, JS, JDAY, JTIME)
INITIALIZE NAVIGATION
CALL NVINIT (BETAIN, BETDOT, INAV, PTIME)
GET IMAGE/IMAGE FRAME INFORMATION, INCLUDING IMAGE MAGNIFICATION
CALL GETFRM (IFRM, MF)
SET UP PROCESSING DAY FOR DOCUMENTATION IN ROW HEADER
CALL GETDAY (IDPROS)
LASRET = IRET(3)
IENH = IKWP('ENH', 1, ϕ)
SET FIRST RETRIEVAL TO BE PROCESSED
NB = IPP(1,1)
SET LAST RETRIEVAL TO BE PROCESSED
LASRET = IPP(2, LASRET)
NRET = ϕ
MAIN DO LOOP; PROCESSES RETRIEVALS NUMBERED "NB" THROUGH "LASRET"
NN ← NB
NN ≤ LASRET
2400
A
NN ← NN+1
GET SAAD-PRODUCED VAS DATA FOR GIVEN RETRIEVAL (NN) FOR USE IN UPCOMING TEMPERATURE/MOISTURE RETRIEVALS
CALL VRTIO(IRET, NN, ϕ)
LOOK AT USER MOD FLAG; LT ϕ MEANS IT HASN'T BEEN RETRIEVED, EQ ϕ MEANS IT HAS BEEN RETRIEVED, GT ϕ MEANS ITS EITHER FAILED OR BEEN EDITED
IRETD(3) ≥ ϕ
24ϕϕ
A
F

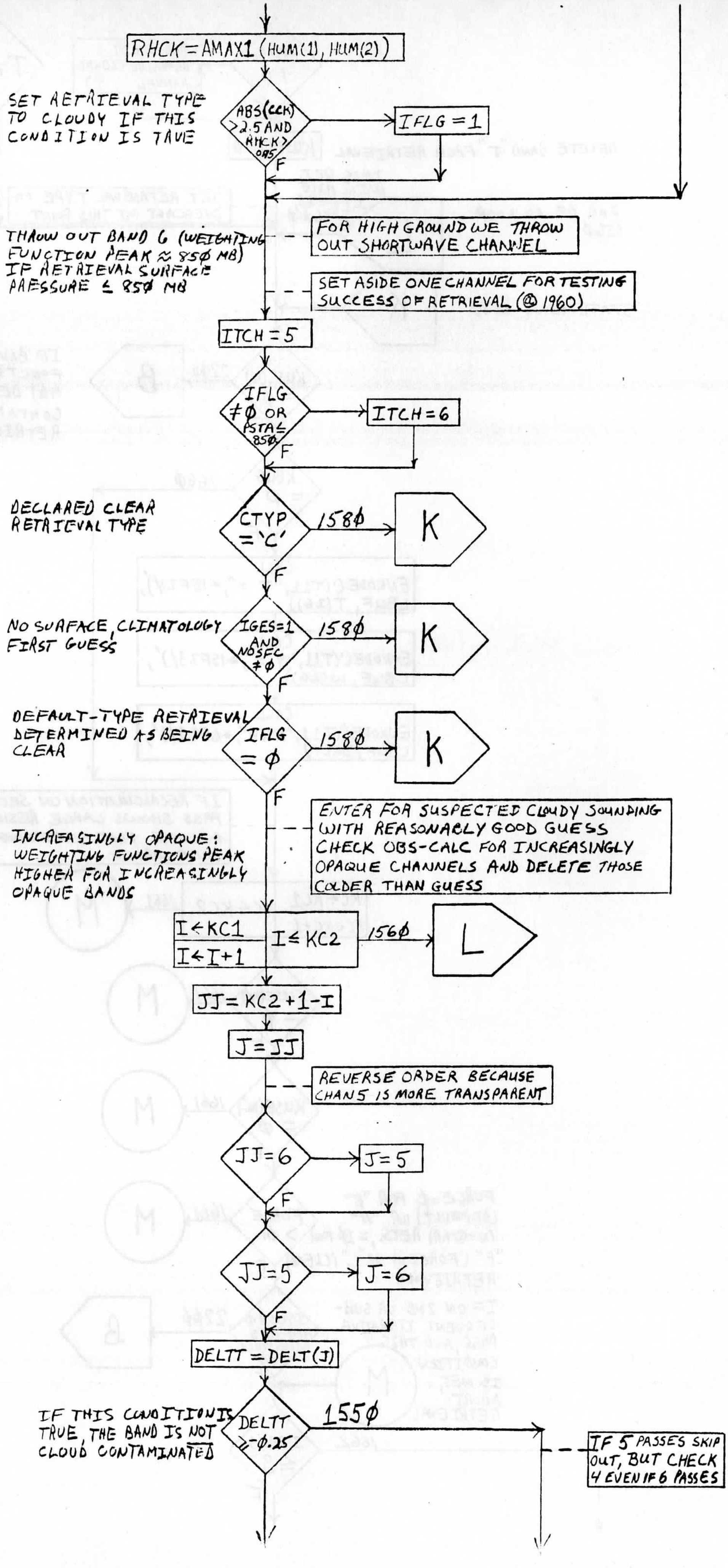

KBUG ≠ Ø
CALL SDEST ('BEGIN SNDG. NO. ', NN)
F
DO LOOP 7ØØ: STORE EXPECTED ERROR OF BRIGHTNESS TEMPERATURE OBSERVATIONS AND RETRIEVAL BOX-AVERAGED BRIGHTNESS TEMPERATURES IN THE ARRAYS "EX" AND "VDAT," RESPECTIVELY
CHECK TO SEE IF THIS CHANNEL IS INTENTIONALLY AVOIDED
THERE ARE TWO SETS OF "USE" FLAGS IN THIS PROGRAM. IUSE IS SET TO Ø IF DATA IS MISSING OR CHANNEL IS INTENTIONALLY SET ASIDE. KUSE IS SET TO Ø IF CHANNEL IS BELIEVED TO BE CLOUD CONTAMINATED; IN THIS CASE GUESS PROFILE IS IMPLICITLY BELIEVED UNLESS IT IS A NO SURFACE, CLIMATOLOGY CASE.
KBUG ≠ Ø
CALL ENKODE ('(T2, "BRIGHT-TEMP", *12F7.2/)', LBUF, VDAT)
F
KBUG ≠ Ø
CALL ENKODE ('(T2, "CHAN FLAGS", *12I7/)', LBUF, IUSE)
F
GET SRAD-DEFINED SFC. SKIN TEMP.
TS = .Ø1 * FLOAT (IRETD (36))
ZERO OUT PREVIOUSLY RETRIEVED QUANTITIES
RESTORE TYPE AND ELEVATION
ACCESS GUESS PROFILES OF TEMPERATURE AND MIXING RATIO
CALL GESPRO (IGES, NOSFC, MDNG)
ABORT RETRIEVAL IF GUESS TEMPERATURE PROFILE CANNOT BE ACCESSED CORRECTLY
TGES(1) < Ø
22ØØ
B
F
TSTA = Ø.Ø1 * FLOAT (ITSFC)
TCK = 2. * VDAT(8) - VDAT(7)
CHECK ON THE HOPELESSLY COLD PRESUMED OVERCAST
ABORT RETRIEVAL IF GUESS SFC. TEMPERATURE MORE THAN 15° WARMER THAN APPROXIMATE SURFACE SKIN TEMPERATURE
TSTA-TCK > 15.
22ØØ
B
F
LET HOT GROUND GO FOR NOW
FIND "IS" 1ST LEVEL BELOW SFC. (PRESSURE)
BLEND GUESS WITH SFC. DATA (IF THEY EXIST)
BEGIN AT 850 MB FOR NORMAL GUESS, 500 MB FOR CLIMATOLOGY
PREPARE GUESS INFO FOR MDFILE
SAVE GUESS TEMPERATURE PROFILE IN THE ARRAY "TSAV," AND SELECTED LEVELS OF THE SAME IN THE ARRAY "TGS." SAVE ALSO SELECTED LEVELS OF THE GUESS DEWPOINT PROFILE IN THE ARRAY "DGS."
KBUG = Ø
54Ø
F
CALL ENKODE ('(132X, T6, "PRESSURE ", *15F7.1/)' LBUF, P(26))
CALL ENKODE ('(T6, "GUESS TEMP", *15F7.1/)', LBUF, T(26))
CALL ENKODE ('(T6, "GUESS WVMR", *15F7.3/)', LBUF, W(26))

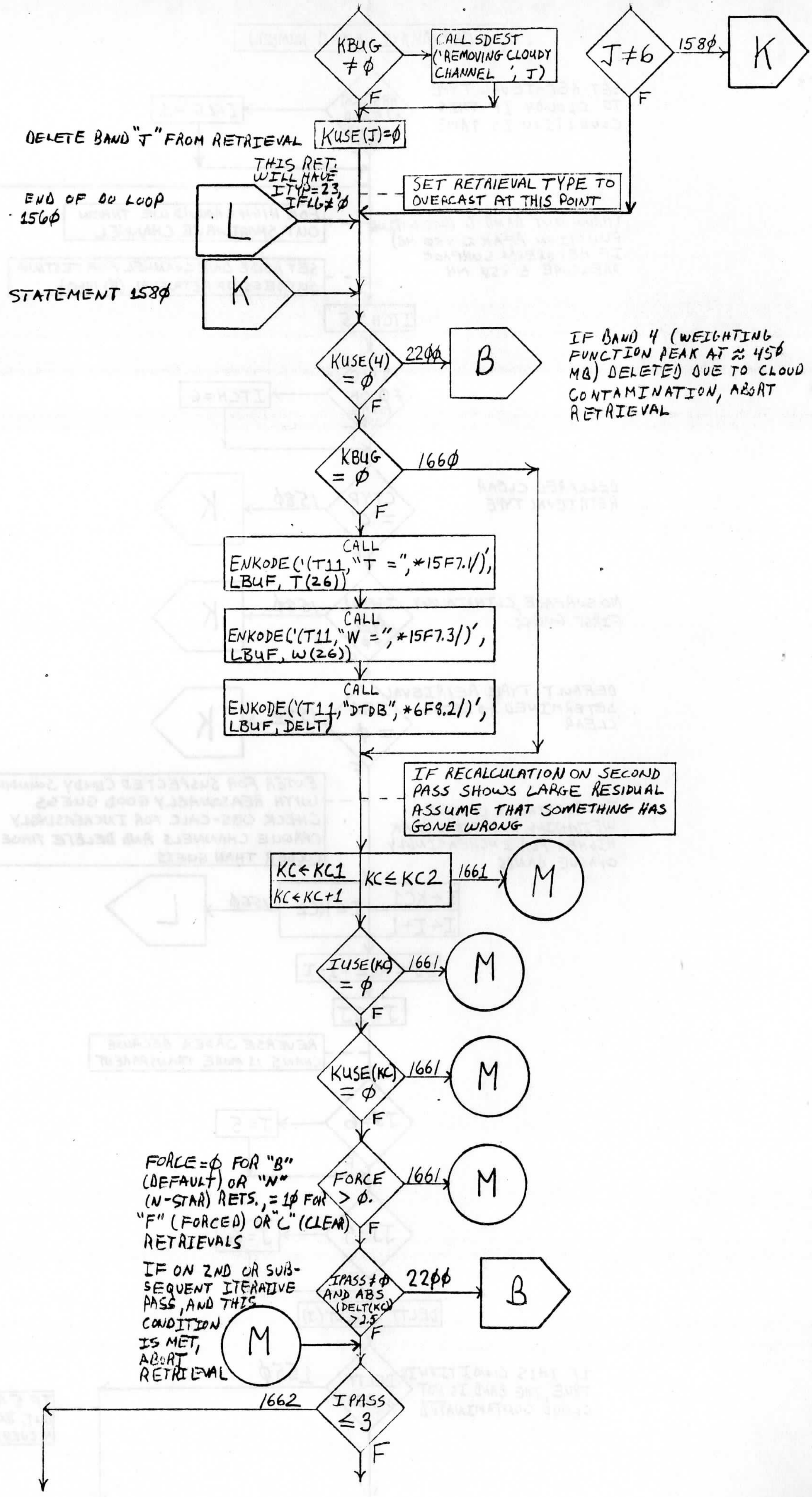

CALL
ENKODE('(T2,"Z1000 =",I8,", TSFC =",I8,", TDSFC =",I8/)',LB4F,IZ10,ITSFC,IDSFC)
PSTA = IPSTA
CALL
WMIX(PSTA,TSTA,DD,WSTA,1)
GET SURFACE MIXING RATIO
DETERMINE A TENTATIVE CLOUD FLAG (IFLG). THIS WILL BE USED TO ESTABLISH RH BOUNDARY AND IN T RETRIEVAL. IF FLAG IS SET TO NON-ZERO WE WILL NOT DO ENHANCEMENT AND WE WILL NOT BELIEVE SKIN TEMPERATURE PREVIOUSLY DEFINED. HOWEVER, WE MAY STILL USE ALL CHANNELS (CIRCA STATEMENTS 1540-1560) AND WE MAY STILL DO RTE DOWN TO SURFACE (1463 LOOP)
ESTABLISH VALUE OF VARIABLE IFLG. .. WILL = 0 IF DEFAULT-TYPE RETRIEVAL DETERMINED TO BE CLEAR, WILL = 99 IF SAME DETERMINED TO BE OVERCAST (= 0 IF RETRIEVAL IS OF TYPE CLEAR; = 99 IF RETRIEVAL IS OF TYPE OVERCAST)
KBUG ≠ 0
F
CALL
SDEST('RH LOWER BOUNDARY LEVEL NO. IS ',NLS)
TEST TO SEE IF THIS IS A GUESS-ONLY CALIBRATION RUN
ICB = 1
1325
C
F
ICB = 0, UNLESS CALIBRATION RUN USING GUESS IS BEING PERFORMED
CHECK REFLECTED SUNLIGHT, BUT FOR HIGH WINDOW VALUES GIVE THE NOD TO CLEAR
CHECK SURFACE AIR AGAINST SKIN ESTIMATE. LOOSEN CHECK FOR WARM SKIN (PROBABLY CLEAR) OR FOR VERY COLD SURFACE AIR (PROBABLE INVERSION CONDITION) OR FOR POOR SURFACE AIR EST.
OK SET FLAG TO CLEAR
FIND "LS" 1ST LEVEL BELOW 1ST COLDER THAN SFC SKIN FOR BOUNDARY IN RTE FOR RELATIVE HUMIDITY.
NOTE THAT LOWER BOUNDARY LEVEL IN RTE FOR RH CALCULATIONS FOR CLEAR RETRIEVALS WILL BE THE MINIMUM OF LEVELS "IS" AND 40, WHILE FOR CLOUDY RETRIEVALS THE LOWER BOUNDARY LEVEL WILL, IN MOST CASES, BE THE CLOUD TOP.
KC ← 1
KC ← KC+1
KC ≤ 3
1100
D
DO LOOP 1100: CALCULATE 3 RELATIVE HUMIDITY (RH) VALUES BASED ON BANDS 7, 9 AND 10 THAT ARE ASSUMED TO BE VALID AT 800, 600 AND 300 MB, RESPECTIVELY, AND REPRESENT LAYER RH VALUES FOR THE LAYERS 1000-800, 800-600 AND 600-300 MB, RESPECTIVELY. THE RH VALUES ARE STORED IN THE ARRAY "IHUM," AND THE 3 FINAL VALUES WILL BE BETWEEN (OR EQUAL TO) .10 AND 1 (10-100%)
IHUM(KC) = MISG
ICV = ICU(KC)
IUSE (ICV) = 0
1100
D
F
RWV = VRAD(ICV)
RWV = VMISG
1100
D
F
ISTEP ← 1
ISTEP ← ISTEP+1
ISTEP ≤ 3
1080
E
ISTEP < 3
885
F
IF 2ND GUESS FOR RH IS TOO HIGH, BASE FINAL ESTIMATE ON 1 AND 2
RWV < RRHO(2)
885
F
SET UP MIXING RATIO PROFILE, FIND PREC WATER AND CALCULATE TRANSMITTANCE AND RADIANCE
CALL PRECW (P, S, U, NL)
CALL VASTAU (T, S, TOTO, VZEN, TAU, ISAT, ICV)
RRHO(ISTEP) = VBDTAU (TAU, B, BS, ICV, NLS)
RH = RHO(2)*(RHO(1)/RHO(2)**TGR)
E
IGES = 1 AND RH > 1.2
1100
D
F
RH = AMIN1(RH, 1.)
RH = AMAX1(RH, 0.01)
IHUM(KC) = IROUND(RH*1000.)

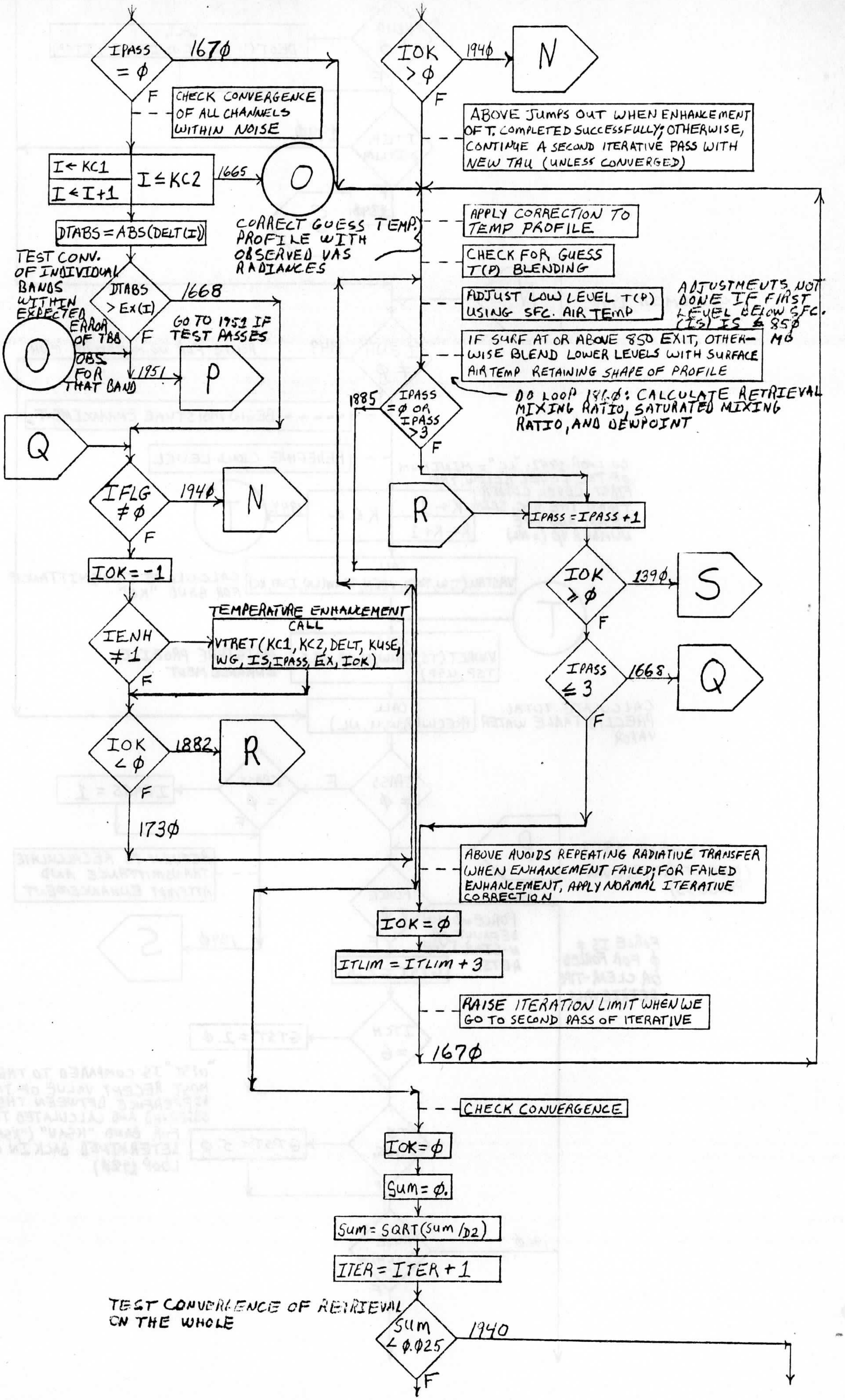

LCRT = ϕ
1100
D
F
CALL ENKODE('(T5, "CHAN RWV * 100 R50 * 100 TGR*100 RH*100")', LBUF)
CALL VTQ(LBUF)
CALL ENKODE('(T1, T1,5I8)', LBUF, ICV, IRWV, IR5ϕ, ITGR, IRH)
CALL VTQ(LBUF)
END OF DO LOOP 1100
D
IHum(3) = MISG
2380
F
ABORT RETRIEVAL IF UPPER BAND LAYER (600-300 MB) VALUE IS MISSING
F
APPORTION THE RELATIVE HUMIDITY TO LAYERS
RESTORE SATURATED MIXING RATIO PROFILE AND ESTABLISH MIXING RATIO PROFILE
DO LOOP 13ϕϕ: DETERMINE SEMI-FINAL RETRIEVAL MIXING RATIO PROFILE (SEE PROGRAM DESCRIPTION FOR DISCUSSION THEREOF)
KBUG = ϕ
132ϕ
F
CALL ENKODE('(T11, "W =", *15F7.3)', LBUF, W(26))
K←1
K≤4
K←K+1
1322
G
CALL VASTAU(T, W, TOTO, VZEN, TAUW(1,K), ISAT, KC)
DETERMINE TRANSMITTANCES FOR BANDS 7-1ϕ
G
CALL PRECW(P, W, U, NL)
CALCULATE TOTAL PRECIPITABLE WATER VAPOR FROM SEMI-FINAL RETRIEVAL MIXING RATIO PROFILE
BEGIN RETRIEVAL LOOPS
C
ITLIM=3
SET TEMPERATURE ITERATION LIMIT
IGES =1
ITLIM=1ϕ
F
ITLIM = IKWP('ITR', 1, ITLIM)
ITER=ϕ
IPASS=ϕ
IOK=ϕ
IOK IS SET TO -1 FOR FAILED ENHANCEMENT, +1 FOR SUCCESS IN VTRET
INIT IS SET TO FIND LOWEST LEVEL JUST ONCE
BEGIN IMPLICIT ITERATION LOOP
IPASS = ϕ CALC TAU; DO T ITERATION; ENHANCE H2O.. CONVERGENCE TEST IS AT 192ϕ
=1 ENHANCE TEMP, LAYER 1
=2 ENHANCE TEMP, LAYER 2
=3 ENHANCE TEMP, LAYER 3
=4 REPEAT T ITERATION IF ENHANCE FAILED, REPEAT H2O, EXIT


CHECK ON CALIBRATION AGAIN
KC2 = 6
KC1 = 1
DO LOOP 138Ø: SET LOOP BOUNDS FOR DO LOOP 152Ø
I ← 1
I ← I+1
I ≤ 6
138Ø
H
J = 7-I
KC1 = I AND IUSE(I) = Ø
KC1 = I+1
F
KC2 = J AND IUSE(J) = Ø
KC2 = J-1
F
H
REMEMBER "ICB" IS ALMOST ALWAYS .EQ. Ø (.EQ. 1 ONLY WHEN CALIBRATION RUN USING GUESS IS BEING PERFORMED
ICB ≠ 1
1385
F
IFLG ≠ Ø
22ØØ
B
F
219Ø
I
KC1 ≥ 3 AND KC2 ≤ 3
22ØØ
B
ABORT RETRIEVAL IF LOOP BOUNDS NOT LARGE ENOUGH
STATEMENT 139Ø
S
KC ← KC1
KC ← KC+1
KC ≤ KC2
152Ø
J
CALCULATE TAU UNDER THE FOLLOWING CONDITIONS:
INITIAL ENTRY
ENHANCEMENT OR 2ND ITERATIVE PASS (ENHANCEMENT FAILED)
BUT ONLY FOR CHAN 4 AND 5 UNLESS GUESS IS CLIMATOLOGY
ITER > Ø AND IPASS ≠ 1
142Ø
F
IPASS=1 AND KC≠4 AND KC≠5 AND IGES ≠1
142Ø
CALCULATE TRANSMITTANCE
CALL VASTAU(T, W, TOTO, VZEN, TAU, ISAT, ICV)

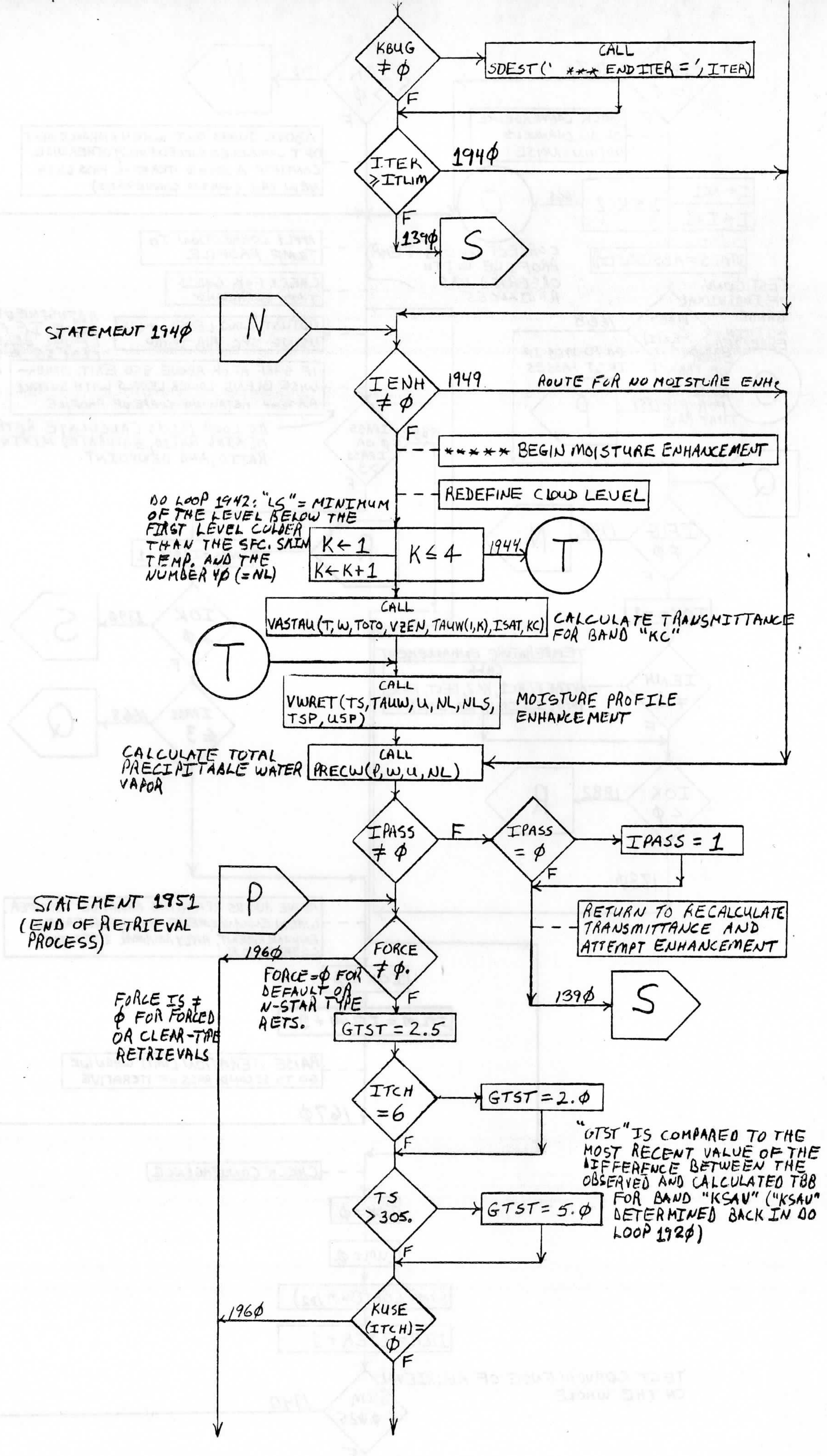

THE FOLLOWING CODE IS ONLY DONE FOR THE FIRST TEMP. ITER. OF THE FIRST ITER. PASS
INIT > Ø
F
INIT=1
IN THE FOLLOWING WE DETERMINE THE LOWEST LEVEL TO WHICH WE WILL DO RADIATIVE TRANSFER
IFLG = Ø
1465
F
IGES=1 AND NOSFC ≠ Ø
1465
F
LOOK FOR NEW CLOUD LEVEL FOR SURFACE TEAM
DO LOOP 1463: DETERMINE THE LEVEL "IA" FOR CLOUDY RETRIEVALS WHICH ARE NOT USING CLIMATOLOGY/NO SURFACE FIRST GUESSES
KBUG ≠ Ø
CALL ENKODE ('(1X,"SKIN LEVEL AND TEMP",I5,F7.2/)', LBUF, IA, TA)
F
CALL VASRTE (TAU, T, TA, CRAD, B, TBB, DBDB, ICV, IA)
CALCULATE RADIANCE FOR BAND "ICV" VIA THE RADIATIVE TRANSFER EQN; THE LOWER LEVEL (IA) WILL EQUAL "IS" FOR CLEAR (IFLG = Ø) RETS., OR THE MINIMUM OF LEVELS "IS" AND "IA" FOR CLOUDY (IFLG ≠ Ø) RETS.
DO LOOP 148Ø: CALCULATE WEIGHTING FUNCTION FOR GIVEN BAND FOR LEVELS 2 THROUGH IS
SAVE PREVIOUS CONVERGENCE STATE
KUSE(KC) = Ø
152Ø
J
F
IUSE(KC) = Ø
152Ø
J
F
DELT(KC)=VDAT(ICV)-TBB
CALCULATE DIFFERENCE BETWEEN OBSERVED TBB AND THAT CALCULATED FROM THE PREVIOUSLY-RETRIEVED TEMPERATURE PROFILE
END OF DO LOOP 152Ø
J
ITER > Ø
158Ø
K
F
IUSE(5) = Ø OR IUSE(6) = Ø
154Ø
F
IFLG≠Ø
154Ø
F
WATCH OUT FOR SHAKEY CHANNEL 6 WHICH WE DON'T UNDERSTAND
COMPARE THE RESIDUAL OF BAND 6 TO THAT FOR BANDS 5 AND 4; REMOVE BAND 6 FROM THE RETRIEVAL IF BANDS 5-6-4 ARE NOT SMOOTHLY MONOTONIC (IN THAT ORDER)
TEST CHANNEL 5 VS. 6 RESIDUAL AND ASSUME THAT A LARGE DISCREPANCY INDICATES CLOUD PROBLEM (HIGH RH1 AS WELL)
CCK = DELT(5)-DELT(6)
KBUG ≠ Ø
CALL SDEST (' CHAN5-CHAN6 = ', ICCK)
F

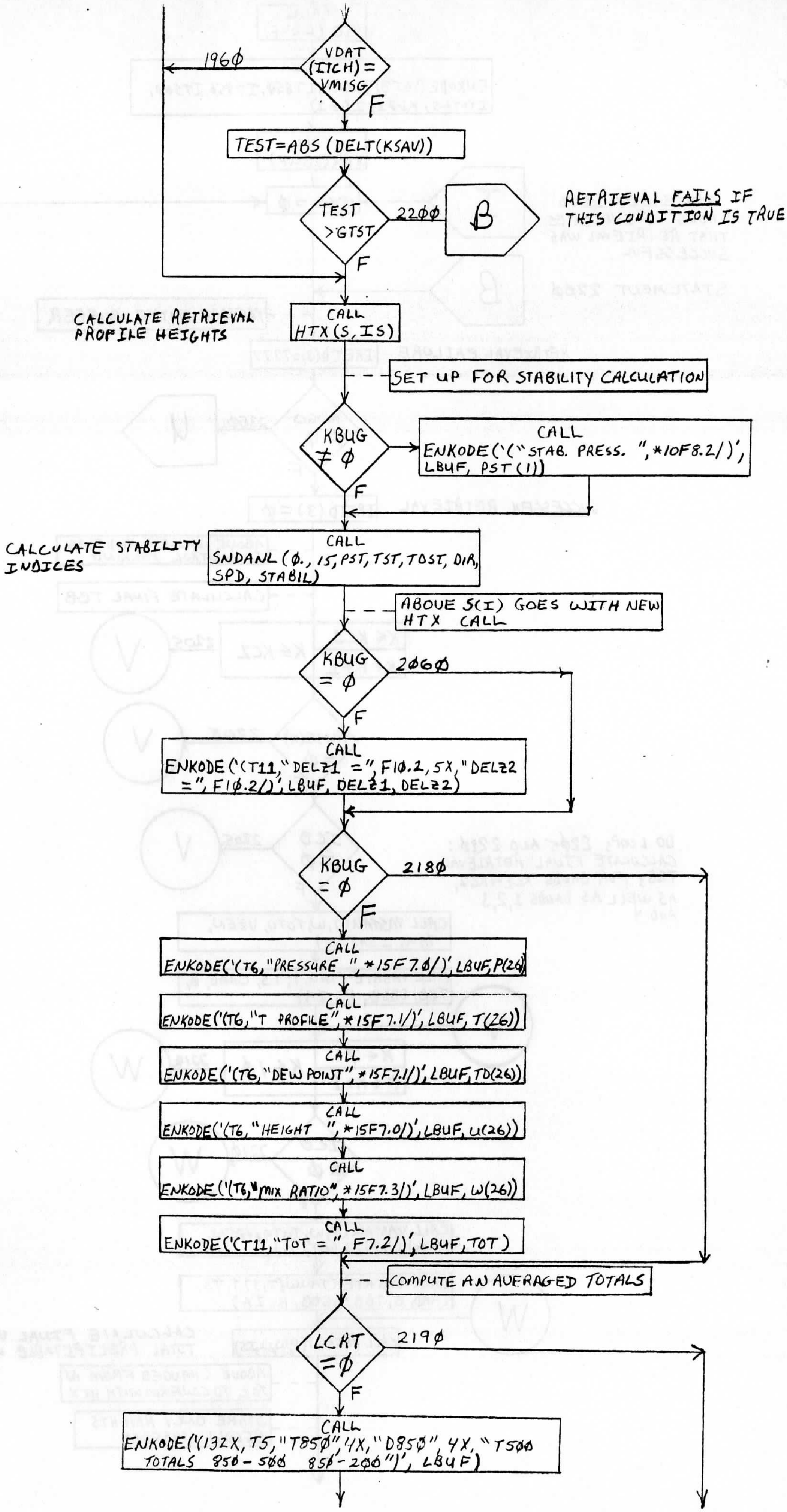

RHCK=AMAX1 (HUM(1), HUM(2))
SET RETRIEVAL TYPE TO CLOUDY IF THIS CONDITION IS TRUE
ABS(CCK) >2.5 AND RHCK> .95
IFLG=1
F
THROW OUT BAND 6 (WEIGHTING FUNCTION PEAK ≈ 85∅ MB) IF RETRIEVAL SURFACE PRESSURE ≤ 85∅ MB
FOR HIGH GROUND WE THROW OUT SHORTWAVE CHANNEL
SET ASIDE ONE CHANNEL FOR TESTING SUCCESS OF RETRIEVAL (@ 1960)
ITCH=5
IFLG ≠∅ OR PSTA≤ 85∅
ITCH=6
F
DECLARED CLEAR RETRIEVAL TYPE
CTYP ='C'
158∅
K
F
NO SURFACE, CLIMATOLOGY FIRST GUESS
IGES=1 AND NOSFC ≠∅
158∅
K
F
DEFAULT-TYPE RETRIEVAL DETERMINED AS BEING CLEAR
IFLG =∅
158∅
K
F
ENTER FOR SUSPECTED CLOUDY SOUNDING WITH REASONABLY GOOD GUESS CHECK OBS-CALC FOR INCREASINGLY OPAQUE CHANNELS AND DELETE THOSE COLDER THAN GUESS
INCREASINGLY OPAQUE: WEIGHTING FUNCTIONS PEAK HIGHER FOR INCREASINGLY OPAQUE BANDS
I←KC1
I←I+1
I≤KC2
156∅
L
JJ=KC2+1-I
J=JJ
REVERSE ORDER BECAUSE CHAN 5 IS MORE TRANSPARENT
JJ=6
J=5
F
JJ=5
J=6
F
DELTT=DELT(J)
IF THIS CONDITION IS TRUE, THE BAND IS NOT CLOUD CONTAMINATED
DELTT ≥-∅.25
155∅
IF 5 PASSES SKIP OUT, BUT CHECK 4 EVEN IF 6 PASSES


KBUG ≠ ∅
CALL SDEST ('REMOVING CLOUDY CHANNEL ', J)
J≠6
158∅
K
F
DELETE BAND "J" FROM RETRIEVAL
KUSE(J)=∅
THIS RET. WILL HAVE ITYP=23, IFLG≠∅
SET RETRIEVAL TYPE TO OVERCAST AT THIS POINT
END OF DO LOOP 156∅
L
STATEMENT 158∅
K
KUSE(4) = ∅
22∅∅
B
IF BAND 4 (WEIGHTING FUNCTION PEAK AT ≈ 45∅ MB) DELETED DUE TO CLOUD CONTAMINATION, ABORT RETRIEVAL
KBUG = ∅
166∅
CALL ENKODE('(T11,"T =",*15F7.1/)', LBUF, T(26))
CALL ENKODE('(T11,"W =",*15F7.3/)', LBUF, W(26))
CALL ENKODE('(T11,"DTDB",*6F8.2/)', LBUF, DELT)
IF RECALCULATION ON SECOND PASS SHOWS LARGE RESIDUAL ASSUME THAT SOMETHING HAS GONE WRONG
KC←KC1
KC←KC+1
KC≤KC2
1661
M
IUSE(KC) = ∅
1661
M
KUSE(KC) = ∅
1661
M
FORCE=∅ FOR "B" (DEFAULT) OR "N" (N-STAR) RETS., =1∅ FOR "F" (FORCED) OR "C" (CLEAR) RETRIEVALS
FORCE > ∅.
1661
M
IF ON 2ND OR SUBSEQUENT ITERATIVE PASS, AND THIS CONDITION IS MET, ABORT RETRIEVAL
IPASS≠∅ AND ABS (DELT(KC)) >2.5
22∅∅
B
M
IPASS ≤ 3
1662
F

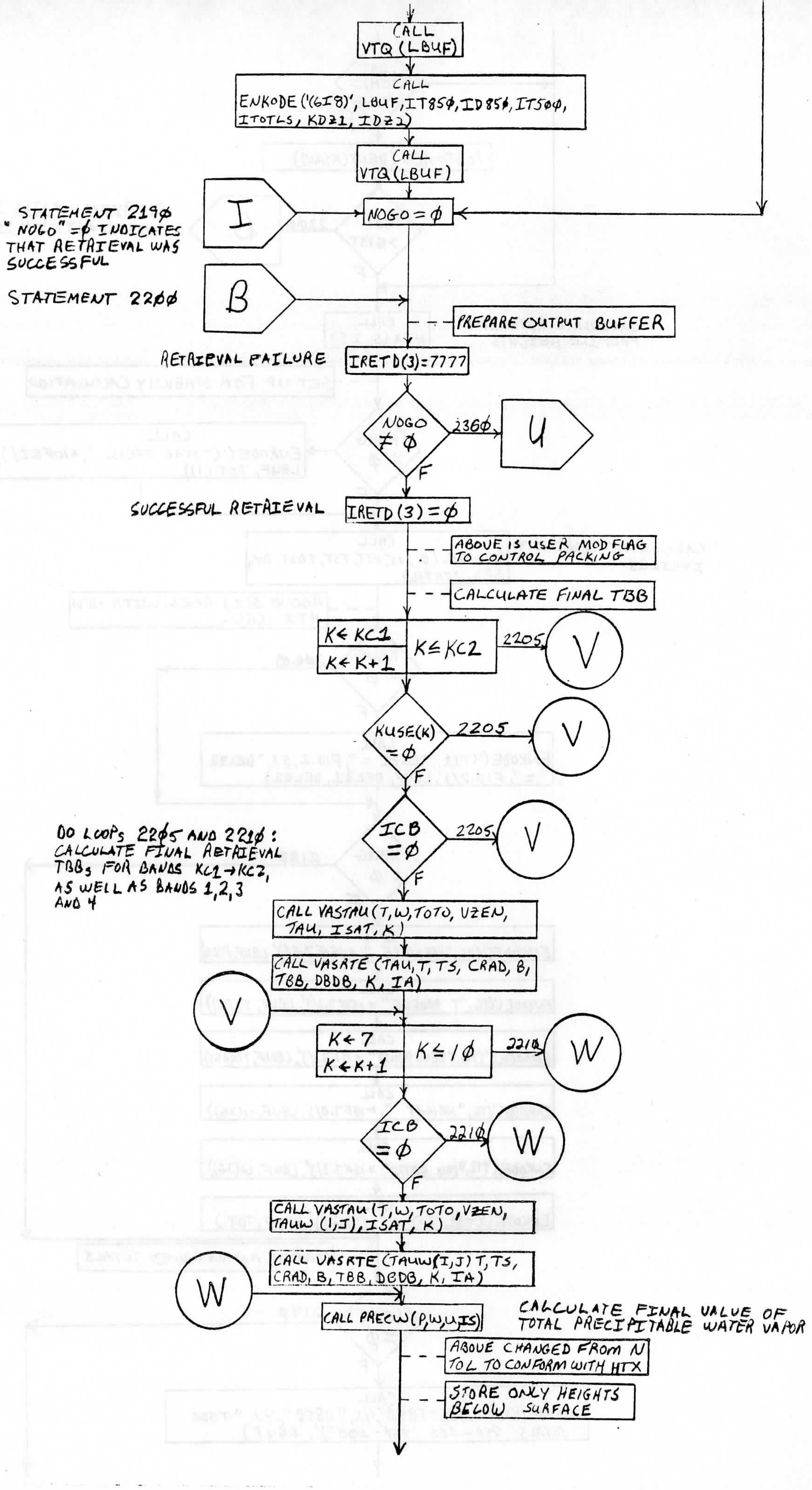


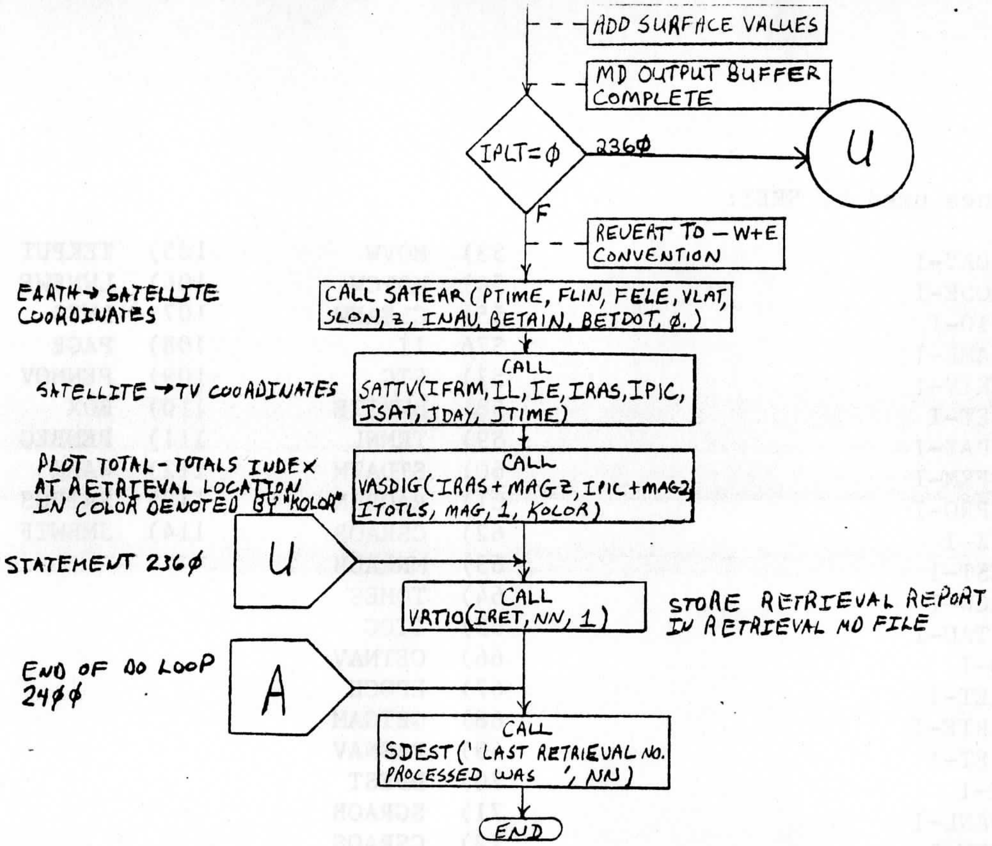
IPASS = ϕ
1670
F
CHECK CONVERGENCE OF ALL CHANNELS WITHIN NOISE
I ← KC1
I ← I+1
I ≤ KC2
1665
O
DTABS = ABS(DELT(I))
TEST CONV. OF INDIVIDUAL BANDS WITHIN EXPECTED ERROR OF TBB OBS FOR THAT BAND
DTABS > EX(I)
1668
GO TO 1951 IF TEST PASSES
F
O
1951
P
Q
IFLG ≠ ϕ
1940
N
F
IOK = -1
TEMPERATURE ENHANCEMENT CALL
IENH ≠ 1
VTRET(KC1, KC2, DELT, KUSE, WG, IS, IPASS, EX, IOK)
F
IOK < ϕ
1882
R
F
1730
IOK > ϕ
1940
N
F
ABOVE JUMPS OUT WHEN ENHANCEMENT OF T. COMPLETED SUCCESSFULLY; OTHERWISE, CONTINUE A SECOND ITERATIVE PASS WITH NEW TAU (UNLESS CONVERGED)
CORRECT GUESS TEMP. PROFILE WITH OBSERVED VAS RADIANCES
APPLY CORRECTION TO TEMP PROFILE
CHECK FOR GUESS T(P) BLENDING
ADJUST LOW LEVEL T(P) USING SFC. AIR TEMP
ADJUSTMENTS NOT DONE IF FIRST LEVEL BELOW SFC. (IS) IS ≤ 850 MB
IF SURF AT OR ABOVE 850 EXIT, OTHERWISE BLEND LOWER LEVELS WITH SURFACE AIR TEMP RETAINING SHAPE OF PROFILE
DO LOOP 1860: CALCULATE RETRIEVAL MIXING RATIO, SATURATED MIXING RATIO, AND DEWPOINT
1885
IPASS = ϕ OR IPASS > 3
F
R
IPASS = IPASS + 1
IOK ≥ ϕ
1390
S
F
IPASS ≤ 3
1668
Q
F
ABOVE AVOIDS REPEATING RADIATIVE TRANSFER WHEN ENHANCEMENT FAILED; FOR FAILED ENHANCEMENT, APPLY NORMAL ITERATIVE CORRECTION
IOK = ϕ
ITLIM = ITLIM + 3
RAISE ITERATION LIMIT WHEN WE GO TO SECOND PASS OF ITERATIVE
1670
CHECK CONVERGENCE
IOK = ϕ
SUM = ϕ.
SUM = SQRT(SUM / D2)
ITER = ITER + 1
TEST CONVERGENCE OF RETRIEVAL ON THE WHOLE
SUM < ϕ.025
1940
F


KBUG ≠ Ø
CALL SDEST(' *** END ITER =', ITER)
F
ITER ≥ ITLIM
1940
F
1390
S
STATEMENT 1940
N
IENH ≠ Ø
1949
ROUTE FOR NO MOISTURE ENH.
F
***** BEGIN MOISTURE ENHANCEMENT
REDEFINE CLOUD LEVEL
DO LOOP 1942: "LS" = MINIMUM OF THE LEVEL BELOW THE FIRST LEVEL COLDER THAN THE SFC. SKIN TEMP. AND THE NUMBER 40 (=NL)
K←1
K≤4
K←K+1
1944
T
CALL VASTAU(T, W, TOTO, VZEN, TAUW(1,K), ISAT, KC)
CALCULATE TRANSMITTANCE FOR BAND "KC"
T
CALL VWRET(TS, TAUW, U, NL, NLS, TSP, USP)
MOISTURE PROFILE ENHANCEMENT
CALCULATE TOTAL PRECIPITABLE WATER VAPOR
CALL PRECW(P, W, U, NL)
IPASS ≠ Ø
F
IPASS = Ø
IPASS = 1
F
STATEMENT 1951 (END OF RETRIEVAL PROCESS)
P
RETURN TO RECALCULATE TRANSMITTANCE AND ATTEMPT ENHANCEMENT
1390
S
FORCE ≠ Ø.
1960
FORCE=Ø FOR DEFAULT OR N-STAR TYPE RETS.
FORCE IS ≠ Ø FOR FORCED OR CLEAR-TYPE RETRIEVALS
F
GTST = 2.5
ITCH = 6
GTST = 2.Ø
F
"GTST" IS COMPARED TO THE MOST RECENT VALUE OF THE DIFFERENCE BETWEEN THE OBSERVED AND CALCULATED TBB FOR BAND "KSAV" ("KSAV" DETERMINED BACK IN DO LOOP 1920)
TS > 305.
GTST = 5.Ø
F
1960
KUSE (ITCH) = Ø
F



VDAT (ITCH) = VMISG
1960
F
TEST=ABS (DELT(KSAV))
TEST >GTST
2200
B
RETRIEVAL FAILS IF THIS CONDITION IS TRUE
F
CALCULATE RETRIEVAL PROFILE HEIGHTS
CALL HTX(S, IS)
SET UP FOR STABILITY CALCULATION
KBUG ≠ Ø
CALL ENKODE('("STAB. PRESS. ",*10F8.2/)', LBUF, PST(1))
F
CALCULATE STABILITY INDICES
CALL SNDANL (Ø., 15, PST, TST, TDST, DIR, SPD, STABIL)
ABOVE S(I) GOES WITH NEW HTX CALL
KBUG = Ø
2Ø6Ø
F
CALL ENKODE('(T11, "DELZ1 =", F1Ø.2, 5X, "DELZ2 =", F1Ø.2/)', LBUF, DELZ1, DELZ2)
KBUG = Ø
218Ø
F
CALL ENKODE('(T6, "PRESSURE ", *15F7.Ø/)', LBUF, P(26))
CALL ENKODE('(T6, "T PROFILE", *15F7.1/)', LBUF, T(26))
CALL ENKODE('(T6, "DEW POINT", *15F7.1/)', LBUF, TD(26))
CALL ENKODE('(T6, "HEIGHT ", *15F7.0/)', LBUF, U(26))
CALL ENKODE('(T6, "MIX RATIO", *15F7.3/)', LBUF, W(26))
CALL ENKODE('(T11, "TOT = ", F7.2/)', LBUF, TOT)
COMPUTE AN AVERAGED TOTALS
LCRT = Ø
219Ø
F
CALL ENKODE('(132X, T5, "T85Ø", 4X, "D85Ø", 4X, "T5ØØ TOTALS 850-5ØØ 85Ø-2ØØ")', LBUF)


CALL VTQ (LBUF)
CALL ENKODE ('(6I8)', LBUF, IT85Ø, ID85Ø, IT50Ø, ITOTLS, KDZ1, IDZ2)
CALL VTQ (LBUF)
STATEMENT 219Ø "NOGO" = Ø INDICATES THAT RETRIEVAL WAS SUCCESSFUL
I
NOGO = Ø
STATEMENT 220Ø
B
PREPARE OUTPUT BUFFER
RETRIEVAL FAILURE
IRETD(3)=7777
NOGO ≠ Ø
236Ø
U
F
SUCCESSFUL RETRIEVAL
IRETD (3) = Ø
ABOVE IS USER MOD FLAG TO CONTROL PACKING
CALCULATE FINAL TBB
K ← KC1
K ← K+1
K ≤ KC2
2205
V
KUSE(K) = Ø
2205
V
F
DO LOOPS 22Ø5 AND 221Ø: CALCULATE FINAL RETRIEVAL TBBs FOR BANDS KC1→KC2, AS WELL AS BANDS 1,2,3 AND 4
ICB = Ø
2205
V
F
CALL VASTAU (T, W, TOTO, UZEN, TAU, ISAT, K)
CALL VASRTE (TAU, T, TS, CRAD, B, TBB, DBDB, K, IA)
V
K ← 7
K ← K+1
K ≤ 1Ø
221Ø
W
ICB = Ø
221Ø
W
F
CALL VASTAU (T, W, TOTO, VZEN, TAUW (I,J), ISAT, K)
CALL VASRTE (TAUW(I,J) T, TS, CRAD, B, TBB, DBDB, K, IA)
W
CALL PRECW (P, W, U, IS)
CALCULATE FINAL VALUE OF TOTAL PRECIPITABLE WATER VAPOR
ABOVE CHANGED FROM N TO L TO CONFORM WITH HTX
STORE ONLY HEIGHTS BELOW SURFACE



ADD SURFACE VALUES
MD OUTPUT BUFFER COMPLETE
IPLT=Ø
236Ø
U
F
REVERT TO — W+E CONVENTION
EARTH → SATELLITE COORDINATES
CALL SATEAR (PTIME, FLIN, FELE, VLAT, SLON, Z, INAV, BETAIN, BETDOT, Ø.)
SATELLITE → TV COORDINATES
CALL SATTV(IFRM, IL, IE, IRAS, IPIC, JSAT, JDAY, JTIME)
PLOT TOTAL-TOTALS INDEX AT RETRIEVAL LOCATION IN COLOR DENOTED BY "KOLOR"
CALL VASDIG(IRAS+MAGZ, IPIC+MAGZ, ITOTLS, MAG, 1, KOLOR)
STATEMENT 236Ø
U
CALL VRTIO(IRET, NN, 1)
STORE RETRIEVAL REPORT IN RETRIEVAL MD FILE
END OF DO LOOP 24ØØ
A
CALL SDEST('LAST RETRIEVAL NO. PROCESSED WAS ', NN)
END

Subroutines used by SRET:

1) CALDAY-I
2) ENKODE-I
3) VRTIO-I
4) MDNAME-I
5) PLNKIV-I
6) TQSET-I
7) GETDAY-I
8) GETFRM-I
9) GESPRO-I
10) WMIX-I
11) SDEST-I
12) PRECW-I
13) VASTAU-I
14) VTQ-I
15) VWRET-I
16) VASRTE-I
17) VTRET-I
18) HTX-I
19) SNDANL-I
20) SATTV-I
21) VASDIG-I
22) TVSAT-I
23) NVINIT-I
24) SATEAR-I
25) ENCODX
26) CONTNT
27) LTQ
28) ZECONV
29) MOVB
30) CLEANA
31) TQ
32) PRLINX
33) PRCLOS
34) PROPEN
35) PRPRPR
36) LOCK
37) PRWR
38) PRRD
39) UNLOCK
40) PRCL
41) POST
42) BLKA
43) ABORT
44) DOPEN
45) ENCODE
46) DREAD
47) DCLOSE
48) LWCLOS
49) MOVWC
50) LWMOP
51) LWPO
52) EDEST
53) MOVW
54) MOVCW
55) CLEANW
576) II
57) STC
58) DWRITE
59) TRMNL
60) STDATM
61) RAOBIN
62) CSRAOB
63) MNRAOB
64) TQMES
65) ITOC
66) GETNAV
67) EPOCH
68) GETGAM
69) VASNAV
70) DDEST
71) SGRAOB
72) CSRAOS
73) CSRAOM
74) CSRAOZ
75) CSRAOP
76) CSRAOR
77) CSRAOI
78) INTPTW
79) EXTEMP
80) CLMGES
81) VASGES
82) SURGES
83) GETSFV
84) IGNAME
85) PROFIX
86) PRETAV
87) PREATV
88) ULMR
89) CO2TAV
90) H2OTAV
91) CONTAV
92) O3TAV
93) GAMTAV
94) INITPL
95) QGDASH
96) DSHOFF
97) PLOT
98) DSHON
99) ENDPLT
100) ENPT
101) PENADD
102) PACK
103) SENOUT
104) ATOE
105) TEKPUT
106) LWNEWF
107) LWSO
108) PAGE
109) PENMOV
110) BOX
111) PENBEG
112) WALK
113) SATPOS
114) JMBWTF

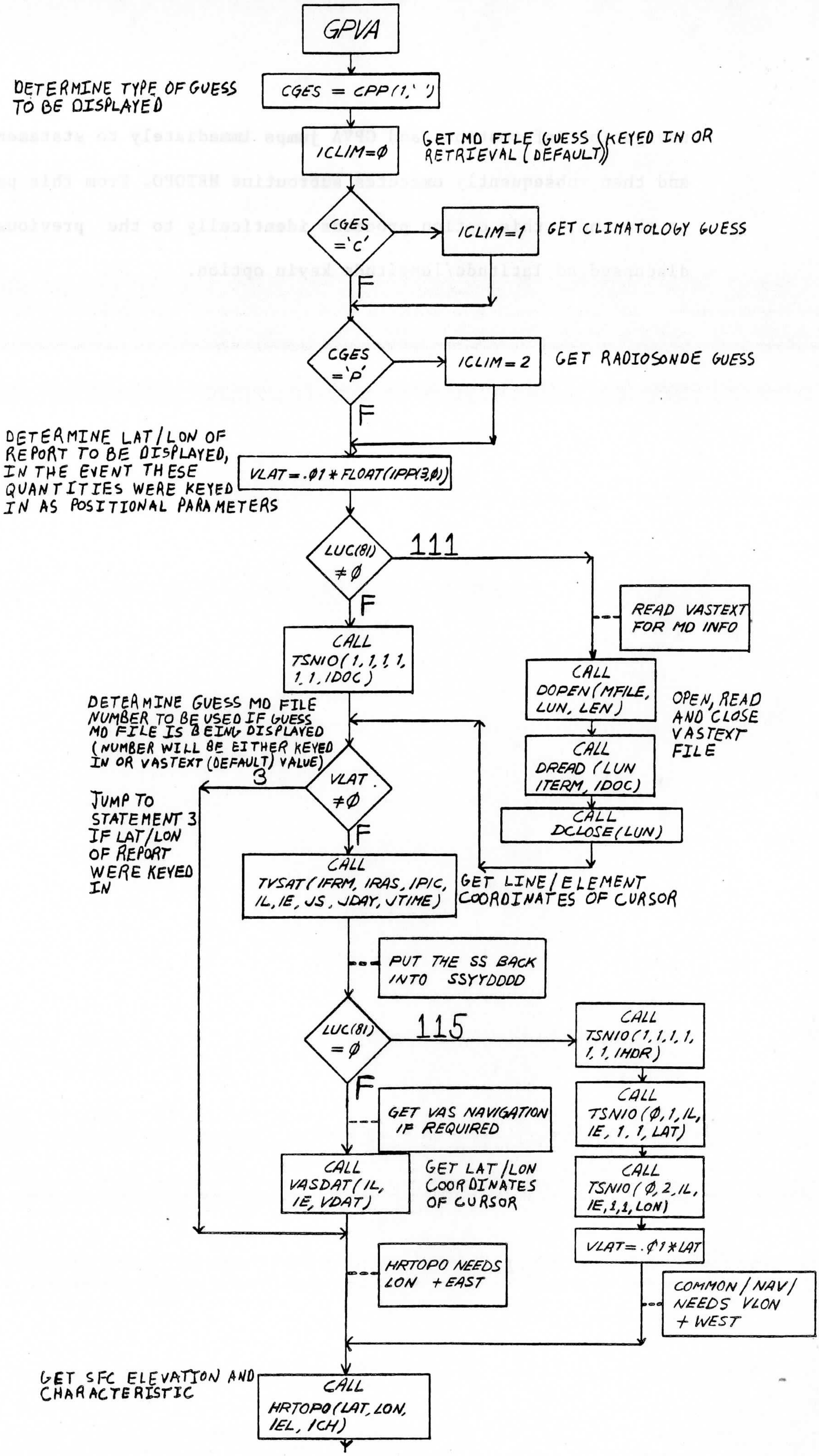
GPVA

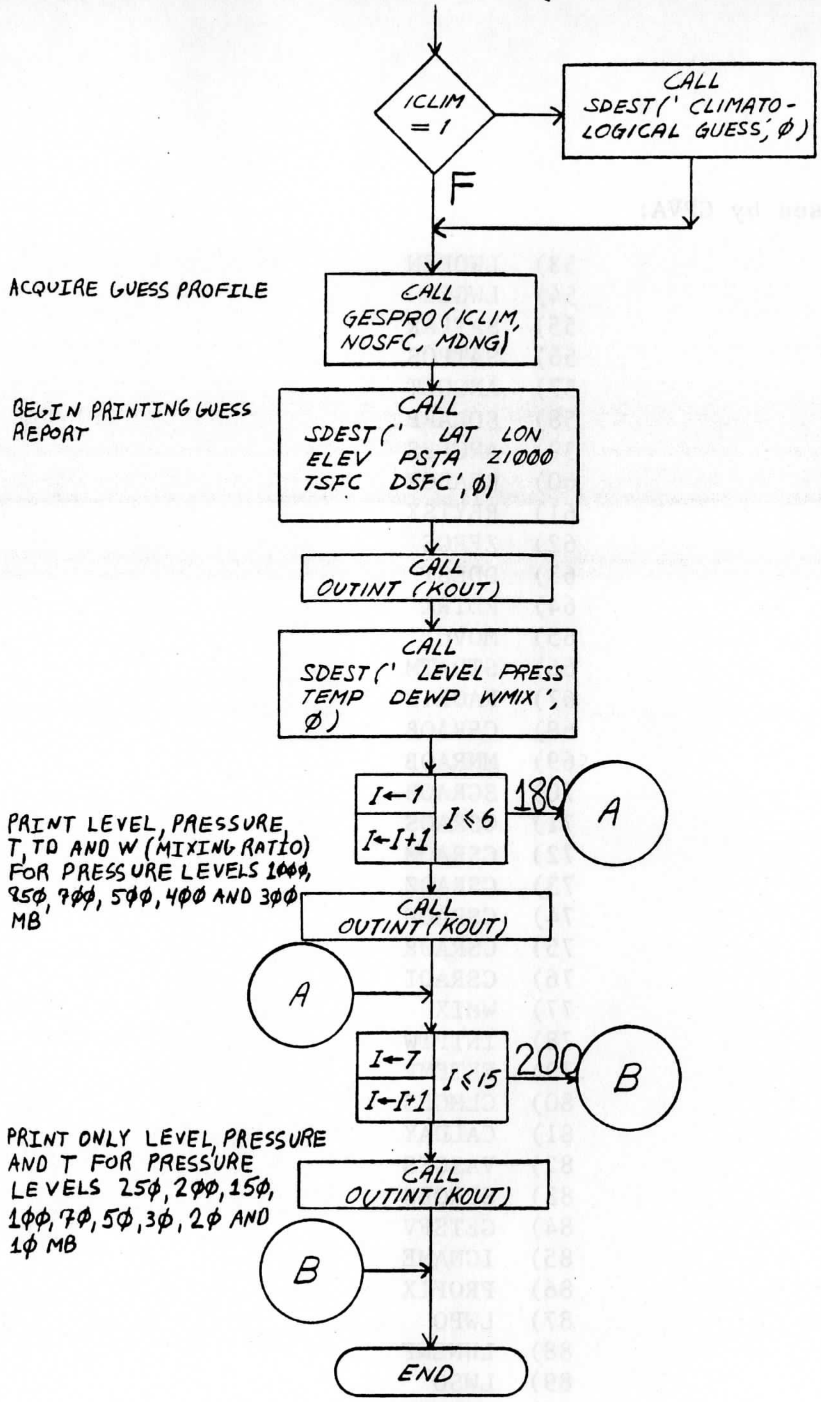
GPVA is used to display VAS guess profiles of temperature, dewpoint and mixing ratio at either the cursor location (default), or a keyed-in latitude/longitude position. Three different types of guess profiles can be displayed: the guess profiles in either a keyed in MD file or the MD file defined under heading MDNG in the VASTEXT file (default), climatology guess profiles, or radiosonde profiles. Note that if the guess in the guess MD file (MDNG) is to be displayed, the user must run program GSVA before executing GPVA.

First, the type of guess to be displayed (CGES), as well as the surface analysis and debug options, are determined. In general, a user will probably want to use surface analyses when generating retrievals, so he/she should operate GPVA in the surface analysis mode to get an idea of what this data looks like at a particular location. This can be accomplished simply by keying in a 0 for the second positional parameter, if any positional parameters beyond the second are also going to be keyed in, or by keying in nothing, if only the first positional parameter is going to be used. In addition, variable ICLIM is set to 0,1 or 2, depending on the guess profile chosen (0 for keyed in or VASTEXT file-defined MD file, 1 for climatology, 2 for radiosonde profile) . This variable is vital later when accessing the actual guess information, because it tells subroutine GESPRO which guess data is to be read. Then, after variables VLAT and VLON (latitude/longitude) have been determined (= 0 if positional parameters in third and fourth positions have not been keyed in), and assuming the sounder area has been set with program VPVA (LUC(81) .NE. 0), the VASTEXT file is opened, read and closed. One of the items read from the VASTEXT file is the guess MD file number MDNG (IDOC(38))(see reference to MDNG in above paragraph). At this point, the rest of the discussion of GPVA, beyond the code where the MD file number is set, will be broken down into two sections. In the first section, it will be assumed that the latitude and longitude for the guess have not been keyed in, while the second section will discuss the case when these quantities have been keyed in.

Assuming there are no latitude/longitude keyins (VLAT,VLON =0), the cursor line and element positions (IL,IE) are determined via subroutine TVSAT, and a call to subroutine VASDAT (with VDAT(1)=-1.) returns navigation data only, including the latitude and longitude positions of the cursor. Next, subroutine HRTOPO returns the surface elevation and characteristic (land = 1, ocean = 0), using a global 10 minute resolution topography. Finally, subroutine GESPRO gathers the rest of the guess data, taking into account the guess option desired. The data returned by GESPRO includes station (surface) pressure, 1000 mb height, and surface temperature and dewpoint. In addition, GESPRO accesses 15 temperature values from 1000-10 mb, 6 dewpoint values from 1000-300 mb, and 6 mixing ratio values, also from 1000-300 mb. After GESPRO has finished, subroutines SDEST and OUTINT print the results on the CRT.

On the other hand, if the latitude and longitude values have been keyed in, there is no need to call subroutine VASDAT for navigation information, and GPVA jumps immediately to statement 3 and then subsequently executes subroutine HRTOPO. From this point to the end, this option proceeds identically to the previously-discussed no latitude/longitude keyin option.


GPVA
DETERMINE TYPE OF GUESS TO BE DISPLAYED
CGES = CPP(1,' ')
ICLIM=Ø
GET MD FILE GUESS (KEYED IN OR RETRIEVAL (DEFAULT))
CGES ='C'
ICLIM=1
GET CLIMATOLOGY GUESS
F
CGES ='P'
ICLIM = 2
GET RADIOSONDE GUESS
F
DETERMINE LAT/LON OF REPORT TO BE DISPLAYED, IN THE EVENT THESE QUANTITIES WERE KEYED IN AS POSITIONAL PARAMETERS
VLAT = .Ø1 * FLOAT(IPP(3,Ø))
LUC(81) ≠ Ø
111
F
READ VASTEXT FOR MD INFO
CALL TSNIO(1,1,1,1, 1,1,IDOC)
CALL DOPEN(MFILE, LUN, LEN)
OPEN, READ AND CLOSE VASTEXT FILE
CALL DREAD (LUN ITERM, IDOC)
CALL DCLOSE(LUN)
DETERMINE GUESS MD FILE NUMBER TO BE USED IF GUESS MD FILE IS BEING DISPLAYED (NUMBER WILL BE EITHER KEYED IN OR VASTEXT (DEFAULT) VALUE)
3
VLAT ≠Ø
JUMP TO STATEMENT 3 IF LAT/LON OF REPORT WERE KEYED IN
F
CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, JS, JDAY, JTIME)
GET LINE/ELEMENT COORDINATES OF CURSOR
PUT THE SS BACK INTO SSYYDDDD
LUC(81) = Ø
115
CALL TSNIO(1,1,1,1, 1,1,IHDR)
F
GET VAS NAVIGATION IF REQUIRED
CALL TSNIO(Ø,1,IL, IE,1,1,LAT)
CALL VASDAT(IL, IE, VDAT)
GET LAT/LON COORDINATES OF CURSOR
CALL TSNIO(Ø,2,IL, IE,1,1,LON)
VLAT=.Ø1*LAT
HRTOPO NEEDS LON +EAST
COMMON/NAV/ NEEDS VLON +WEST
GET SFC ELEVATION AND CHARACTERISTIC
CALL HRTOPO(LAT, LON, IEL, ICH)


ICLIM = 1
CALL SDEST(' CLIMATOLOGICAL GUESS', Ø)
F
ACQUIRE GUESS PROFILE
CALL GESPRO(ICLIM, NOSFC, MDNG)
BEGIN PRINTING GUESS REPORT
CALL SDEST(' LAT LON ELEV PSTA Z1000 TSFC DSFC', Ø)
CALL OUTINT (KOUT)
CALL SDEST(' LEVEL PRESS TEMP DEWP WMIX', Ø)
I←1
I≤6
I←I+1
180
A
PRINT LEVEL, PRESSURE, T, TD AND W (MIXING RATIO) FOR PRESSURE LEVELS 1ØØØ, 85Ø, 7ØØ, 5ØØ, 4ØØ AND 3ØØ MB
CALL OUTINT(KOUT)
A
I←7
I≤15
I←I+1
200
B
PRINT ONLY LEVEL, PRESSURE AND T FOR PRESSURE LEVELS 25Ø, 2ØØ, 15Ø, 1ØØ, 7Ø, 5Ø, 3Ø, 2Ø AND 1Ø MB
CALL OUTINT(KOUT)
B
END

Subroutines used by GPVA:

1) TSNIO-I
2) DOPEN-I
3) DREAD-I
4) DCLOSE-I
5) TVSAT-I
6) VASDAT-I
7) HRTOPO-I
8) GESPRO-I
9) SDEST-I
10) OUTINT-I
11) II
12) STC
13) ENCODE
14) ENCODX
15) ABORT
16) CONTNT
17) LTQ
18) ZECONV
19) MOVB
20) CLEANA
21) TQ
22) PRLINX
23) PRCLOS
24) PROPEN
25) PRPRPR
26) LOCK
27) PRWR
28) PRRD
29) UNLOCK
30) PRCL
31) POST
32) BLKA
33) LWCLOS
34) MOVWC
35) LWMOP
36) ITOC
37) GETFRM
38) TRMNL
39) TQMES
40) MOVCW
41) NVINIT
42) GETNAV
43) EPOCH
44) GETGAM
45) VASNAV
46) DDEST
47) CLEANW
48) EDEST
49) MOVW
50) READD
51) PLNKIV
52) OPNA
53) LWOPEN
54) LWGET
55) SATEAR
56) SATPOS
57) ANGGET
58) SOLARP
59) ANGLES
60) READOF
61) RBYTSX
62) ZEROS
63) DDEST
64) RDTRK
65) MOVC
66) STDATM
67) RAOBIN
68) CSRAOB
69) MNRAOB
70) SGRAOB
71) CSRAOS
72) CSRAOM
73) CSRAOZ
74) CSRAOP
75) CSRAOR
76) CSRAOI
77) WMIX
78) INTPTW
79) EXTEMP
80) CLMGES
81) CALDAY
82) VASGES
83) SURGES
84) GETSFV
85) IGNAME
86) PROFIX
87) LWPO
88) LWNEWF
89) LWSO
90) JMBWTF
91) VRTOPO

ESVA

This program, which works only for retrieval surface MD files (schema RSVC), edits surface reports at the cursor location, and operates mainly in either of two modes. First, it can delete any combination of 1000 mb height (Z100), mean sea level temperature (TSL) or surface dewpoint depression (DD) values from the retrieval surface MD file. Otherwise, keyword data values for any or all three of these parameters can be substituted in. In addition, different combinations of the two modes (deletion, substitution) can also be keyed in. Besides these two main modes, there is also an option to add a report (where none existed previously) to the retrieval surface MD file. As part of this option, it is possible to calculate and store the TSL value corresponding to a keyed-in TSFC (surface temperature) value.

The first major step in ESVA is the calculation of the cursor line and element positions (IL,IE) via subroutine TVSAT. Next, the VASTEXT file is read, the navigation is initialized, and the MD file and row numbers are set to IDOC (36) and (37), respectively, which correspond to the retrieval surface MD file and row numbers listed in the VASTEXT file. This MD file is then opened for read/write, and the row header corresponding to row MDR is read via function MDGET (IOK=MDGET(MDNO, MDR, 0, IOUT)). Then, after a test has been performed to assure that data exists in this row of the retrieval surface MD file, the keys for the file are read in via function MDKEYS. This allows ESVA (in DO LOOP 11) to (among other things) locate the positions within the array IOUT which contain the values for a given MD file report of Z100, TSL and DD. These are the positions (words) that will later be updated with new values, or have their old values deleted. Following DO LOOP 11, the values keyed in for Z100, TSL and(or) DD are read separately using function IKWP, and the value of variable ITEM is determined. If a particular parameter is to be deleted from a given report, it should be keyed in with a "1" (TSL=1, for example). On the other hand, if some new value is to be substituted into the report in place of the old value, it should be keyed in *100 for TSL, TSFC and DD, but only as the value itself for Z100. Note that if "ESVA" alone is keyed in, the program will respond with "NO ACTION INDICATED AT THIS LOCATION" and terminate.

After frame information is accessed (GETFRM) and the plot package is started by subroutine INITPL, the cursor dimensions in latitude and longitude are determined. First, the NW corner of the cursor in satellite coordinates (line, element--TVSAT) and then earth coordinates (latitude, longitude--SATEAR) is calculated, after which ESVA repeats the process for the SE corner of the cursor.

The next step involves the retrieval surface MD file itself. A large DO LOOP (DO LOOP 70) reads MREC (= IOUT(3) in row header, which was read approximately 100 code lines previously) MD file entries, determines which ones should be edited, and then proceeds to do the actual data editing itself. A given report will be edited ONLY if it falls within the bounds of the cursor box. (Note that the cursor dimensions, if large enough, may encompass more than one surface report.) For a given report which is contained by the cursor, the TV coordinates (raster, pixel) are obtained via subroutines SATEAR and SATTV. Following this, the correct words of the array IOUT are updated with the deletions or new values of Z100, TSL and(or) DD, with the array subsequently being put back into the retrieval surface MD file via function MDPUT (IOK=MDPUT(MDNO, MDR, M, IOUT)). In addition, variable ITEM is plotted on the video screen (via subroutine PLTDIG) at the report location within the cursor, after which the program returns to examine the next surface MD file report to see if it is also encompassed by the cursor box.

After DO LOOP 70 is exited, and assuming at least one report was located within the cursor box (NDAT .NE. 0), control transfers to statement 100, where an option to pack the retrieval surface MD file will be carried out if keyword parameter PACK was keyed in equal to 1. Normally, this keyword will not be used, and ESVA terminates. This concludes the discussion of how ESVA can be used for editing existing data within a cursor box. The following paragraph describes how ESVA operates when data is to be edited into an area where no prior data existed.

If ESVA is to be used in this mode, the cursor should be placed in the desired data void before the program is executed. Up to DO LOOP 70, ESVA functions exactly the same way in this "data addition" mode as it did when reports within the cursor were to be edited. However, DO LOOP 70 will not do anything this time, since all the existing retrieval surface MD file reports will be outside the cursor boundaries. Hence, variable NDAT will equal 0 when DO LOOP 70 is exited, and control will transfer to statement 85. At this point, if a value or values of Z100, TSL and(or) DD was(were) keyed in and are to be added to the retrieval surface MD file, the latitude and longitude of the cursor (ALAT,ALON) are determined by subroutine SATEAR. Then, if TSFC was keyed in, a corresponding mean sea level temperature value is determined. In either case (TSFC keyed or not keyed in), ESVA eventually reaches statement 300, and the new report is placed in the retrieval surface MD file one column beyond the last previous report (column IOUT(3) + 1). In addition, the updated row header is written into the MD file via function MDPUT, and variable ITEM is plotted on the video screen via subroutine PLTDIG near the new report location. Finally, the PACK option is again encountered, and ESVA terminates.

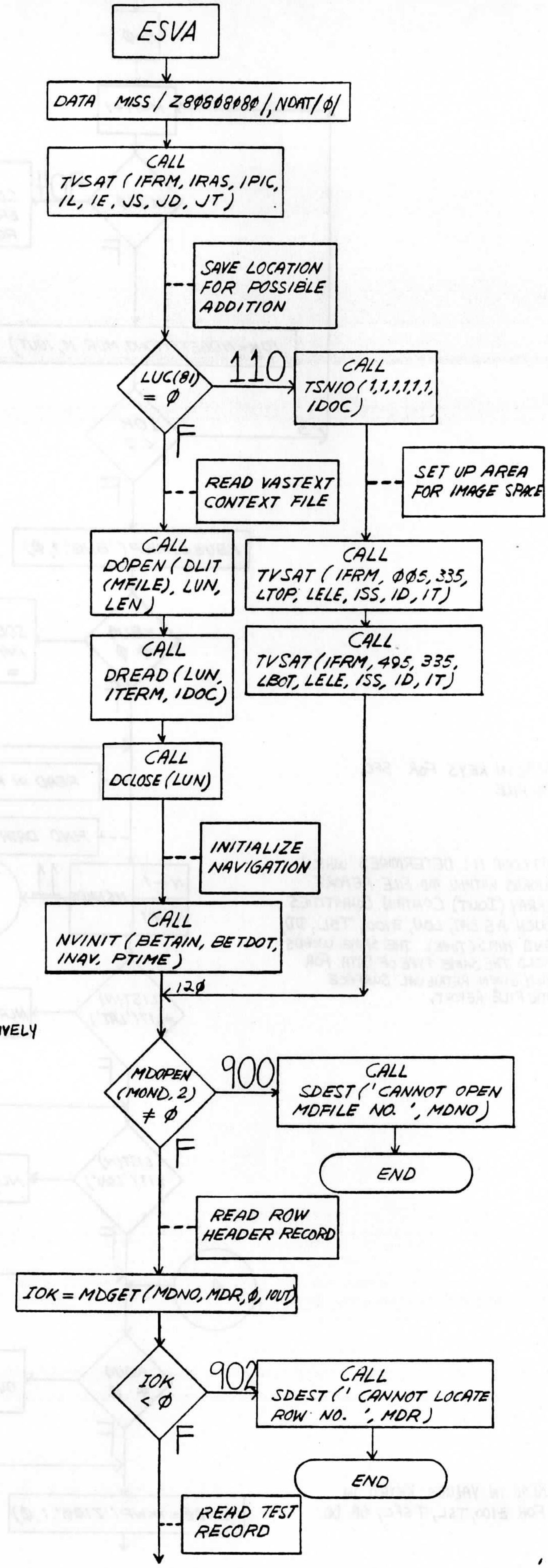
SAVE LINE/ELEMENT COORDINATES OF CURSOR CENTER FOR CASE WHEN DATA IS TO BE ADDED WHERE NONE EXISTED PREVIOUSLY

OPEN AND READ VASTEXT FILE

SET MD FILE AND ROW #'s TO VASTEXT "MONS" AND "MORS," RESPECTIVELY

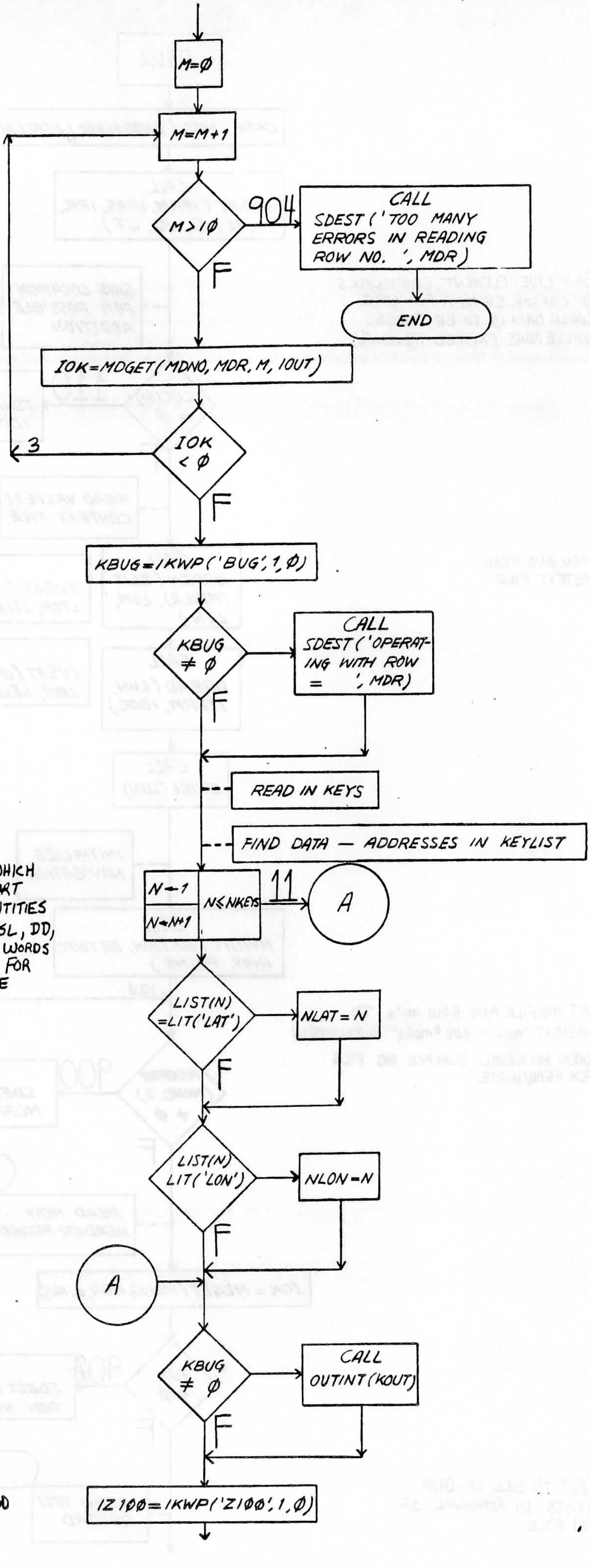
OPEN RETRIEVAL SURFACE MD FILE FOR READ/WRITE

TEST TO SEE IF DATA EXISTS IN RETRIEVAL SFC. MD FILE

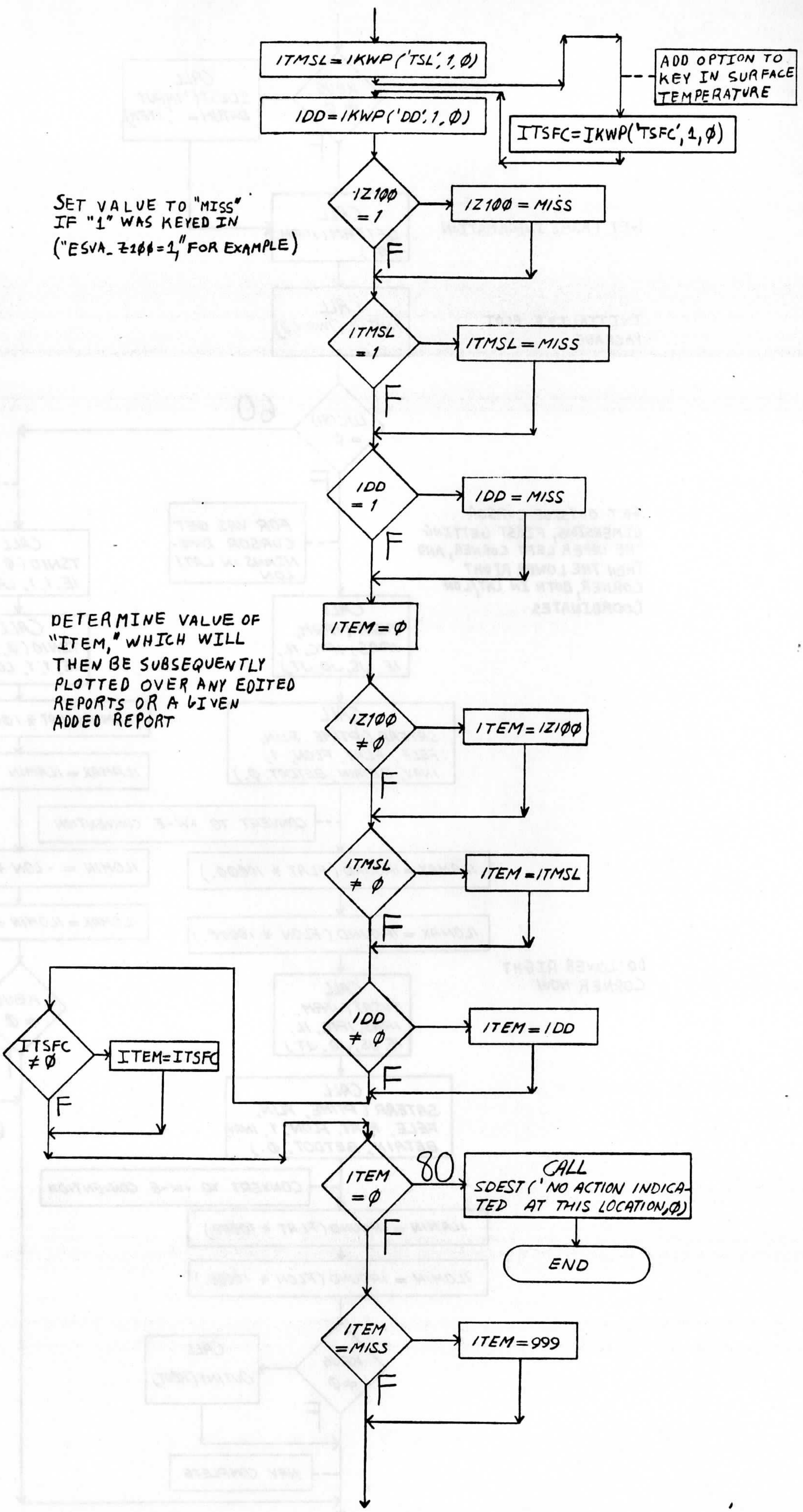


READ IN KEYS FOR SFC.
MD FILE

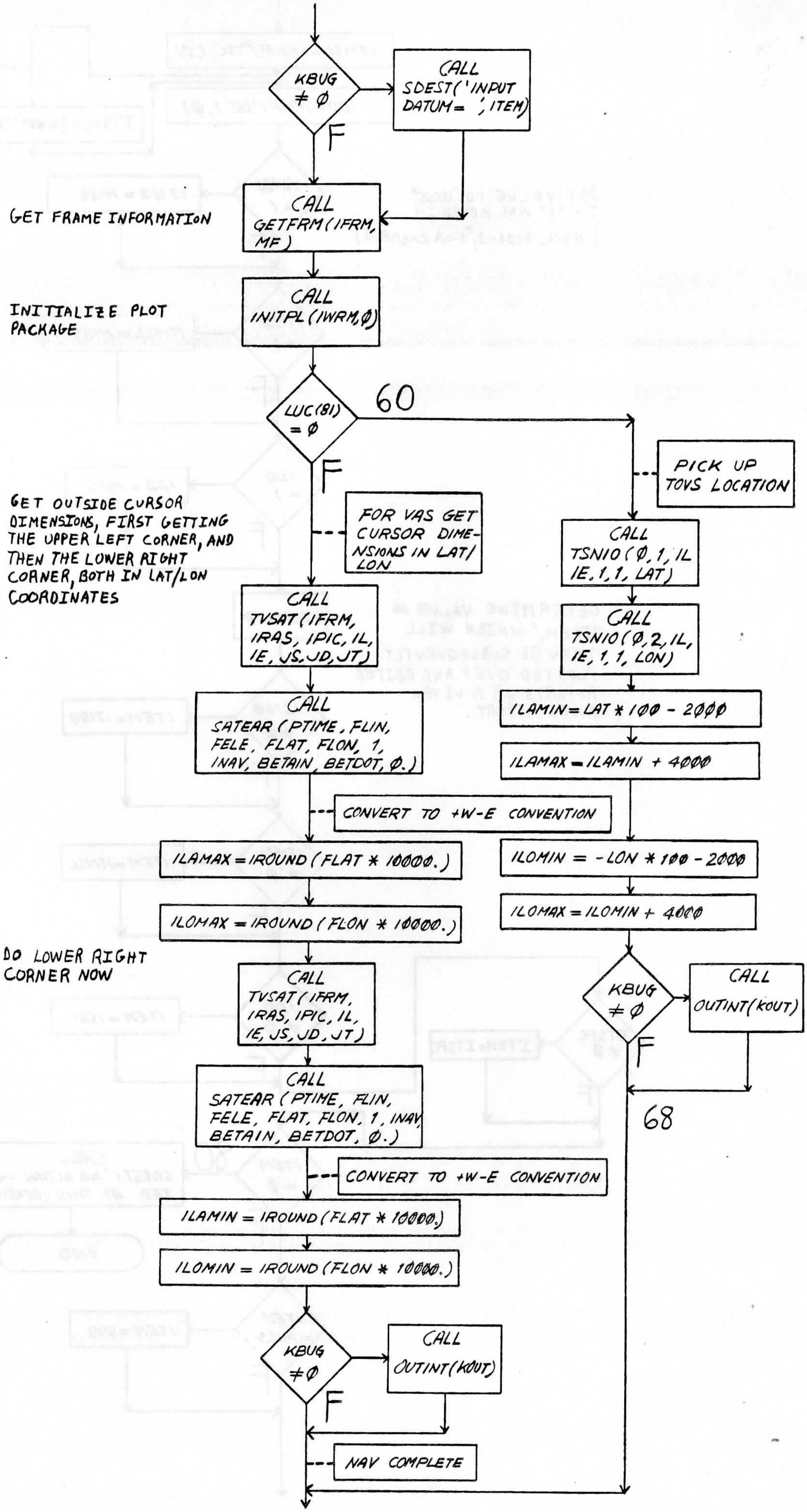
DO LOOP 11: DETERMINES WHICH WORDS WITHIN MD FILE REPORT ARRAY (IOUT) CONTAIN QUANTITIES SUCH AS LAT, LON, Z100, TSL, DD, AND HMS(TIME). THE SAME WORDS HOLD THE SAME TYPE OF DATA FOR ANY GIVEN RETRIEVAL SURFACE MD FILE REPORT.

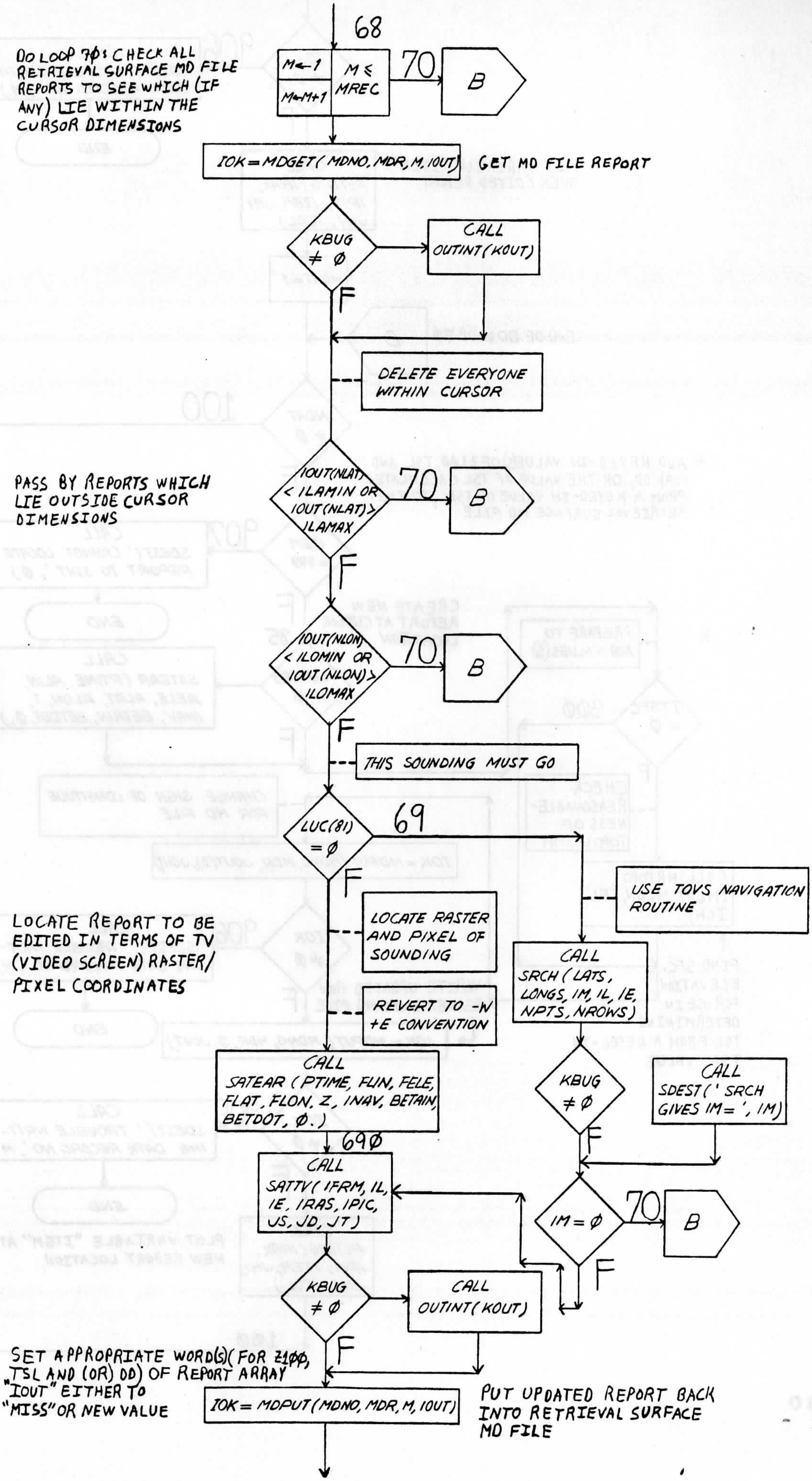


READ IN VALUES KEYED IN
FOR Z100, TSL, TSFC, OR DD

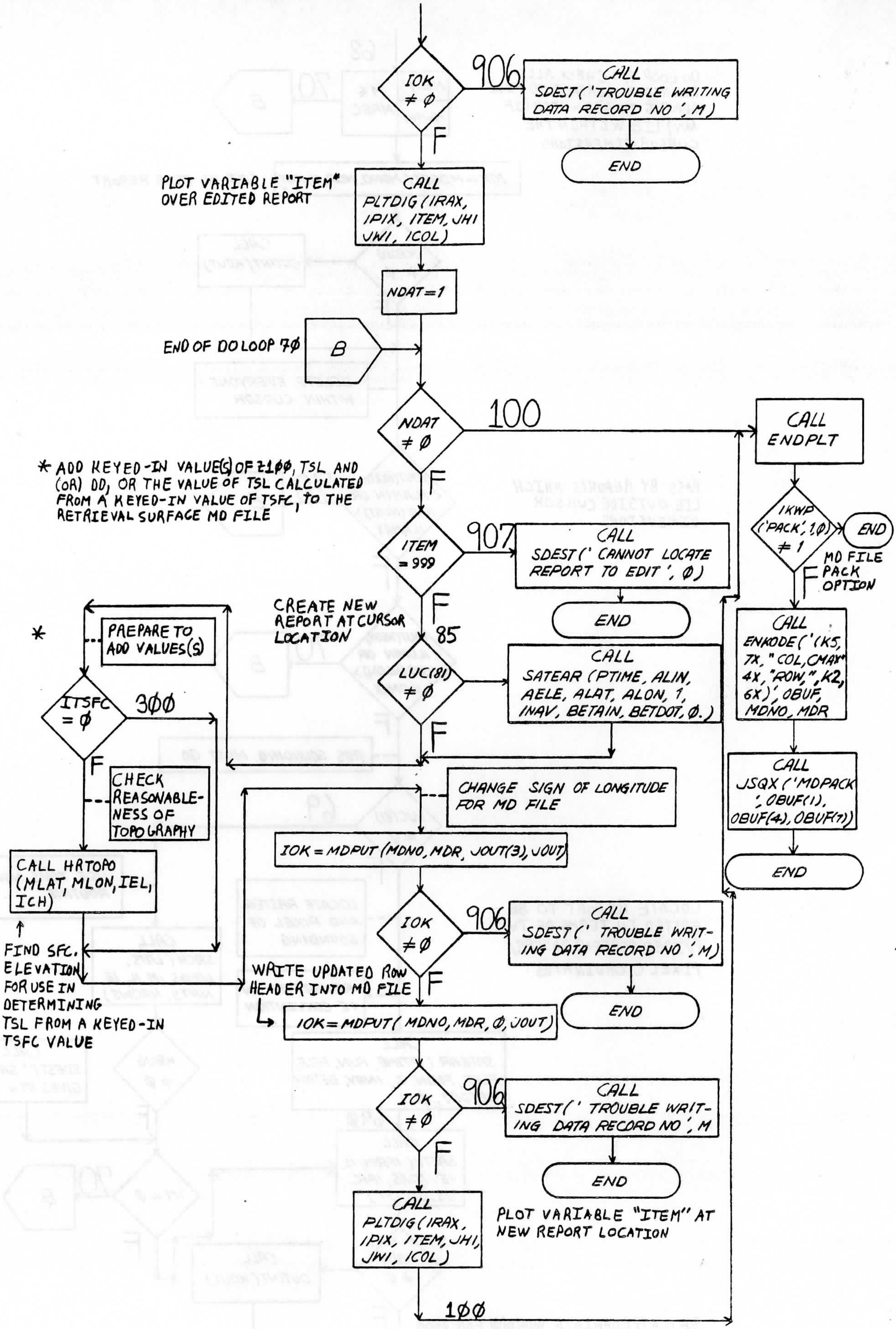


ITMSL=IKWP('TSL',1,Ø)
ADD OPTION TO KEY IN SURFACE TEMPERATURE
IDD=IKWP('DD',1,Ø)
ITSFC=IKWP('TSFC',1,Ø)
SET VALUE TO "MISS" IF "1" WAS KEYED IN ("ESVA_Z1ØØ=1," FOR EXAMPLE)
IZ1ØØ =1
IZ1ØØ = MISS
F
ITMSL =1
ITMSL = MISS
F
IDD =1
IDD = MISS
F
DETERMINE VALUE OF "ITEM," WHICH WILL THEN BE SUBSEQUENTLY PLOTTED OVER ANY EDITED REPORTS OR A GIVEN ADDED REPORT
ITEM=Ø
IZ1ØØ ≠Ø
ITEM=IZ1ØØ
F
ITMSL ≠Ø
ITEM=ITMSL
F
ITSFC ≠Ø
ITEM=ITSFC
F
IDD ≠Ø
ITEM=IDD
F
ITEM =Ø
80
CALL SDEST('NO ACTION INDICATED AT THIS LOCATION,Ø)
F
END
ITEM =MISS
ITEM=999
F


KBUG ≠ Ø
CALL SDEST('INPUT DATUM= ', ITEM)
F
GET FRAME INFORMATION
CALL GETFRM(IFRM, MF)
INITTIALIZE PLOT PACKAGE
CALL INITPL(IWRM,Ø)
LUC(81) = Ø
60
F
PICK UP TOVS LOCATION
GET OUTSIDE CURSOR DIMENSIONS, FIRST GETTING THE UPPER LEFT CORNER, AND THEN THE LOWER RIGHT CORNER, BOTH IN LAT/LON COORDINATES
FOR VAS GET CURSOR DIMENSIONS IN LAT/LON
CALL TSNIO(Ø,1,IL, IE,1,1, LAT)
CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)
CALL TSNIO(Ø,2,IL, IE,1,1, LON)
CALL SATEAR(PTIME, FLIN, FELE, FLAT, FLON, 1, INAV, BETAIN, BETDOT, Ø.)
ILAMIN = LAT * 1ØØ - 2ØØØ
ILAMAX = ILAMIN + 4ØØØ
CONVERT TO +W-E CONVENTION
ILAMAX = IROUND(FLAT * 1ØØØØ.)
ILOMIN = -LON * 1ØØ - 2ØØØ
ILOMAX = IROUND(FLON * 1ØØØØ.)
ILOMAX = ILOMIN + 4ØØØ
DO LOWER RIGHT CORNER NOW
CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
CALL SATEAR(PTIME, FLIN, FELE, FLAT, FLON, 1, INAV, BETAIN, BETDOT, Ø.)
68
CONVERT TO +W-E CONVENTION
ILAMIN = IROUND(FLAT * 1ØØØØ.)
ILOMIN = IROUND(FLON * 1ØØØØ.)
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
NAV COMPLETE



68
DO LOOP 7Ø: CHECK ALL RETRIEVAL SURFACE MD FILE REPORTS TO SEE WHICH (IF ANY) LIE WITHIN THE CURSOR DIMENSIONS
M←1
M ≤ MREC
M←M+1
70
B
IOK = MDGET(MDNO, MDR, M, IOUT)
GET MD FILE REPORT
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
DELETE EVERYONE WITHIN CURSOR
PASS BY REPORTS WHICH LIE OUTSIDE CURSOR DIMENSIONS
IOUT(NLAT) < ILAMIN OR IOUT(NLAT) > ILAMAX
70
B
F
IOUT(NLON) < ILOMIN OR IOUT(NLON) > ILOMAX
70
B
F
THIS SOUNDING MUST GO
LUC(81) = Ø
69
F
USE TOVS NAVIGATION ROUTINE
CALL SRCH (LATS, LONGS, IM, IL, IE, NPTS, NROWS)
LOCATE REPORT TO BE EDITED IN TERMS OF TV (VIDEO SCREEN) RASTER/ PIXEL COORDINATES
LOCATE RASTER AND PIXEL OF SOUNDING
REVERT TO -W +E CONVENTION
CALL SATEAR (PTIME, FLIN, FELE, FLAT, FLON, Z, INAV, BETAIN, BETDOT, Ø.)
KBUG ≠ Ø
CALL SDEST(' SRCH GIVES IM= ', IM)
F
69Ø
CALL SATTV(IFRM, IL, IE, IRAS, IPIC, US, JD, JT)
IM = Ø
70
B
F
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
SET APPROPRIATE WORD(S) (FOR Z1ØØ, TSL AND (OR) DD) OF REPORT ARRAY "IOUT" EITHER TO "MISS" OR NEW VALUE
IOK = MDPUT(MDNO, MDR, M, IOUT)
PUT UPDATED REPORT BACK INTO RETRIEVAL SURFACE MD FILE



IOK ≠ Ø
906
CALL SDEST('TROUBLE WRITING DATA RECORD NO', M)
END
F
PLOT VARIABLE "ITEM" OVER EDITED REPORT
CALL PLTDIG(IRAX, IPIX, ITEM, JHI, JWI, ICOL)
NDAT=1
END OF DO LOOP 7Ø
B
NDAT ≠ Ø
100
CALL ENDPLT
F
* ADD KEYED-IN VALUE(S) OF Z1ØØ, TSL AND (OR) DD, OR THE VALUE OF TSL CALCULATED FROM A KEYED-IN VALUE OF TSFC, TO THE RETRIEVAL SURFACE MD FILE
ITEM = 999
907
CALL SDEST(' CANNOT LOCATE REPORT TO EDIT', Ø)
END
IKWP ('PACK', 1Ø) ≠ 1
END
MD FILE PACK OPTION
F
CREATE NEW REPORT AT CURSOR LOCATION
F
85
*
PREPARE TO ADD VALUES(S)
LUC(81) ≠ Ø
CALL SATEAR(PTIME, ALIN, AELE, ALAT, ALON, 1, INAV, BETAIN, BETDOT, Ø.)
CALL ENKODE('(K5, 7X, "COL, CMAX" 4X, "ROW", K2, 6X)', OBUF, MDNO, MDR)
ITSFC = Ø
3ØØ
F
F
CHECK REASONABLENESS OF TOPOGRAPHY
CHANGE SIGN OF LONGITUDE FOR MD FILE
CALL JSQX('MDPACK', OBUF(1), OBUF(4), OBUF(7))
IOK = MDPUT(MDNO, MDR, JOUT(3), JOUT)
CALL HRTOPO (MLAT, MLON, IEL, ICH)
END
FIND SFC. ELEVATION FOR USE IN DETERMINING TSL FROM A KEYED-IN TSFC VALUE
IOK ≠ Ø
906
CALL SDEST(' TROUBLE WRITING DATA RECORD NO', M)
WRITE UPDATED ROW HEADER INTO MD FILE
F
END
IOK= MDPUT(MDNO, MDR, Ø, JOUT)
IOK ≠ Ø
906
CALL SDEST(' TROUBLE WRITING DATA RECORD NO', M
F
END
CALL PLTDIG(IRAX, IPIX, ITEM, JHI, JWI, ICOL)
PLOT VARIABLE "ITEM" AT NEW REPORT LOCATION
1ØØ

Subroutines used by ESVA:

1) TVSAT-I
2) DOPEN-I
3) DREAD-I
4) DCLOSE-I
5) NVINIT-I
6) TSNIO-I
7) SDEST-I
8) OUTINT-I
9) GETFRM-I
10) INITPL-I
11) SATEAR-I
12) SRCH-I
13) SATTV-I
14) PLTDIG-I
15) ENDPLT-I
16) ENKODE-I
17) JSQX-I
18) HRTOPO-I
19) TRMNL
20) TQMES
21) BLKA
22) STC
23) ITOC
24) MOVB
25) LTQ
26) CLEANA
27) TQ
28) PRLINX
29) PRCLOS
30) PROPEN
31) PRPRPR
32) LOCK
33) PRWR
34) PRRD
35) UNLOCK
36) PRCL
37) POST
38) ABORT
39) ENCODE
40) ENCODX
41) CONTNT
42) ZECONV
43) LWCLOS
44) MOVWC
45) LWMOP
46) GETNAV
47) EPOCH
48) GETGAM
49) VASNAV
50) DDEST
51) MOVCW
52) CLEANW
53) EDEST
54) MOVW
55) II
56) DWRITE
57) SATPOS
58) QGDASH
59) DSHOFF
60) PLOT
61) DSHON
62) PUC
63) WD
64) ENPT
65) PENADD
66) PACK
67) SENOUT
68) ATOE
69) TEKPUT
70) ISQX
71) TOKANL
72) SQSLED
73) LWPO
74) LWNEWF
75) LWSO
76) PAGE
77) PENMOV
78) BOX
79) PENBEG
80) WALK
81) JMBWTF

VTPW

VTPW is used to retrieve values of total precipitable water vapor (TPWV) for individual fields of view (FOVs) within a retrieval area, and can also be keyed in to generate an image of the results. Note that the resolution of these retrievals will almost always be much greater than the usual temperature/moisture retrievals discussed in SRAD/SRET or VTPZ, because those retrievals were made using brightness temperature (TBB) measurements averaged over anywhere from 1 to 121 FOV retrieval boxes. In other words, VTPW will have from 1 to 121 individual retrievals PER RETRIEVAL BOX. Note also near code line 30 which VASTEXT file pointers must be set before the execution of the program. Most of these pointers would be set by the time a typical retrieval area was processed; however, if VTPW is used BEFORE a given retrieval area has been processed, any pointers not set must be set using program SPVA. In addition, the pointers MDNR, MDRR and NRET (retrieval MD file and row numbers, and number of retrievals performed so far, respectively) must be set to the proper values with SPVA before VTPW is run, irrespective of whether a retrieval area has been processed previously or not. The reason for this is that the retrieval MD file used to store the results of VTPW must be created using a different schema (VTWV) than is used for typical temperature/moisture retrieval MD files (VRET). In other words, these three pointers will all necessarily have different values than their predecessors from the previous retrieval area, while the rest of the VASTEXT file pointers may or may not be the same. Finally, it will be assumed that the area of retrievals to be processed will have its NW and SE corners determined by the cursor position and a keyin via keyword parameter END, respectively. (Note that keyword parameter BEG must be keyed in as BEG 0 0 for this option to work properly. This will become more clear as the discussion proceeds. Also, the cursor should be positioned near the NW corner of the image before VTPW is executed.)

Initially, the number of the digital area to be used for storing the image of the TPWV results is determined via keyword parameter ARA (NARA=IKWP('ARA',1,0)). ARA should normally be included in the program keyin. However, if it is not keyed in (NARA=0), no image of TPWV will be created or stored in the digital area. Next, subroutine VRTIO reads both the VASTEXT file and the retrieval MD file row header, after which variables MDNG (upper air guess MD file), MDNR and MDRR are set to VASTEXT file values (through common block DOC and hence the array IDOC). Then, after variable LRO (the total number of retrievals as recorded in the VASTEXT file) is determined, the type of guess to be used for the retrievals (climatology or grid (default)), as well as the surface analysis option, are stored in variables CGES and NOSFC, respectively. Next, the size (in FOVs) of one side of a given retrieval box is determined (11 by default if no value is keyed in with keyword parameter BOX)) and stored in variable NBXS. If the sounder area data is large detector in nature, keyword parameter BOX should be keyed in equal to 6. This will allow VTPW to set the box size to 5 (the typical size of large detector retrieval boxes) and variable IFILL to 1 (as opposed to 0 for small detector data). For small detector data, since the usual size of small detector retrieval boxes IS 11*11 FOVs, keyword parameter BOX usually will not be keyed in.

Following the box size determination, the line and element retrieval box spacings in FOVs (INCRL, INCRE), and the end and initial line and element coordinates defining a retrieval area (LLINE, LELEM, ILINE, IELEM) are set. Note that these coordinates, if not keyed in via keyword parameters END and BEG, will take on VASTEXT file values by default. At this point, several other variables, such as the debug option (LBUG) and plot option (IPLT) are established, after which the cursor location in TV coordinates (raster-INRAS, pixel-INPIC) are determined from User Common via function LUC.

After the row header for retrieval MD file MDNR has been written (put) into the file via function MDPUT (IOK=MDPUT(MDNR,MDRR,0,IRET)), a jump is taken to statement 140 (remember, keyword parameter BEG was keyed in as BEG 0 0), where subroutine TVSAT determines the retrieval area NW corner satellite coordinates from the cursor postion. Then, since LLINE*LELEM .NE. 0 (remember, keyword para. END supposedly was keyed in as END line# element#), VTPW will jump to statement 200.

As an aside, the preceeding code between and including statements 120 and 200 will now be discussed in terms of some other modes of operation of VTPW. First, if both the initial and ending line and element coordinates have been keyed in via keyword parameters BEG and END, and the processing is being done on a video terminal (LUC(16)=1), a call to subroutine GETFRM will return image frame information, and VTPW will jump to statements 160 and then 200 to process a retrieval area. On the other hand, if both the initial and final line and element coordinates have been keyed in, but the processing is being done on a non-video terminal (LUC(16)=0), the call to GETFRM will not occur before the area is processed. (This makes sense, because it is impossible to have image frame information for a non-video terminal.) Finally, a single retrieval box at the cursor location will be processed if keyword parameters BEG and END are both entered with two zeros.

Returning now to the previous discussion, VTPW calls subroutine VASDAT, which returns navigation data only, because VDAT(1) is set to -1 before VASDAT is called. Important information returned by VASDAT includes the line and element resolution of the sounder area image (ILRES,IERES), which are then used, in turn, to calculate the line and element spacing of each retrieval box (INCIL,INCIE). At this point, it would be a good idea to view the layout of a typical retrieval area by glancing at Diagram 3 immediately following this discussion.

Next, assuming the area number (NARA) is .NE. 0, the EXACT line and element coordinates of the NW and SE corners of the image (as opposed to ILINE/IELEM, LLINE/LELEM, which are the satellite coordinates for the CENTERS of the NW and SE corner retrieval boxes within the image) are calculated. Following this, the dimensions of the retrieval area, in terms of lines and elements, are calculated and placed in the variables MLIN and MELE, respectively. Then, a digital area and its attendant directory are created via subroutines ARASIZ and ENAREA, after which the digital area itself is opened via subroutine OPNA. This area will be used to store the TPWV values retrieved by VTPW as an image.

After variables KLINES and KELEMS (number of FOVs from top to bottom and side to side, respectively, in a retrieval box) are set, VTPW enters DO LOOP 820, which comprises almost all of the remaining program. Nested within DO LOOP 820 are DO LOOPs 740, 640 and 620. Together, these four loops create the image of TPWV from the vapor totals retrieved for each FOV in the retrieval area. VAS data is accessed retrieval line by retrieval line within the retrieval area, retrieval box by retrieval box within each retrieval line, and from the NW to the SE corner within each retrieval box, moving successively FOV by FOV to the right within each scan line of a given box. (Again, see Diagram 3 at the end of this discussion.)

The VAS data itself is accessed by a call to subroutine VASDAT about 10 lines inside of DO LOOP 620 (CALL VASDAT(ILINES,IELES,VDAT)). After the call to VASDAT, several tests are performed to see if a retrieval should be attempted. Note that a retrieval will NOT be attempted if: latitude is .GE. 90 degrees, satellite zenith angle is .GT. 60 degrees, land elevation cannot be returned due to a problem with the land elevation file, or TBB data for that particular FOV is missing. DO LOOP 320 then checks bands 1 through 12, setting a given band to VMISG if its TBB value is unreasonable. If there is reasonable data, and variable IMF=0, variable MSAM is incremented by 1. MSAM will be incremented only once per FOV, since IMF is set to 0 after the initial incrementation. In addition, assuming that neither the band 7 or 8 TBB data for the given FOV is missing, the spin budget for each band is saved in the array NSPIN.

Upon exiting DO LOOP 320, if MSAM has not been incremented (i.e., each band from 1 through 12 either has bad data or is missing), VTPW will move to statement 600, where the element of the array IARRAY (used for generating the image of TPWV) corresponding to the FOV in question is filled with the default TPWV image value of 255. After this, VTPW moves to the next FOV to attempt a new retrieval. For the first FOV to be processed in a given retrieval box (which will be the FOV in the center of the box), variable IGFLG=1, and subroutine GESPRO accesses the guess information, depending on the type of guess chosen (IGES). Within GESPRO, in addition to the accessing of the guess profiles of temperature and mixing ratio, several surface parameters are determined, including surface pressure, temperature, and dewpoint. It should be noted that only ONE set of guess profiles is accessed for each retrieval box, since variable IGFLG is set to 0 after the guess profiles are accessed successfully for the first time.

Assuming the guess for a given retrieval box has been accessed successfully (TGES(1).GT.0) the first time and the surface parameters have been determined, DO LOOP 360 calculates the first pressure level below the surface for the retrieval box and stores the result in the variables IS and NLS. Following this, DO LOOP 440 finalizes the guess temperature and moisture profiles. The resulting guess temperature profile will be identical to the original guess above 300 mb, but will consist of the surface value (TSTA) blended with the original guess from 300 mb down to level NLS, and values equal to the surface value itself from the first level below NLS down to 1000 mb. In addition, the resulting guess moisture profile will be equivalent to .02 g/kg above 100 mb, but will equal (.2)*(saturated value) from 100-250 mb, a blend between the surface value (WSTA) and the original guess from 300 mb down to level NLS, and the surface value itself again from the first level below NLS down to 1000 mb.

After the guess profile (if LBUG .NE. 0) has been printed on the line printer, and other information concerning the FOV location and zenith angles has been displayed (depending on the value(s) of variables LCRT and LBUG) on either the CRT or line printer, subroutine VASTPW calculates the TPWV amount for the FOV in question, returning the result via output parameter URET. The maximum possible value of URET allowed by VASTPW is 12 cm.

Assuming that VASTPW is successful, variable NDONE is incremented by 1. Note that only every "nth" retrieval will be outputted to the retrieval MD file (i.e., when variable MDOUT=1). After statement 560, assuming the plot option is in effect (IPLT.NE.0), subroutine SATTV calculates the FOV raster/pixel coordinates, after which subroutine VASDIG plots the TPWV value on the video screen at that location. Then, assuming the report is to be written into the retrieval MD file (MDRR.GE.0, MDOUT.EQ.1), several elements of the output array IRET are filled with retrieval data, after which the entire array is put into column LRO of the retrieval MD file via function MDPUT. Following this, the retrieval MD file row header is updated with the new number of completed TPWV retrievals via a second MDPUT. Finally, DO LOOP 620 concludes by updating several arrays with TBB, TPWV and fail (IFAIL) flag data. Note in particular the data stored in the array IARRAY. This data will be used later to generate the image of TPWV. (As an aside, an image value of 0 is pure black, while a value of 255 is pure white.)

Following DO LOOP 640 (entire retrieval box completed at this point), and assuming the VAS data is small detector (IFILL.EQ.0), VTPW enters DO LOOP 700. This loop, together with DO LOOP 680, is designed to fill in blank scan lines within a given retrieval box column by column with an estimated TPWV image value (remember the relative widths of scan lines and retrieval lines). The estimated value is based on TPWV information from bordering scan lines, as well as band 8 TBB data (which exists for ALL of the FOVs in the retrieval area, regardless of whether the data is small or large detector).

Finally, DO LOOPs 780 and 760 prepare the TPWV image results stored previously in the array IARRAY for input into digital area NARA. DO LOOP 760 prepares the array IARET for input into subroutine PACK. After DO LOOP 760 has been completed, IARET will contain TPWV image information for all the FOVs in one scan line of the retrieval area. Then, subroutines PACK and WRITA together write the scan line of image data into digital area NARA. After DO LOOP 780 has been completed, a message saying FINISHED LINE is displayed on the CRT, and VTPW moves to the next retrieval line within the retrieval area (DO LOOP 820 increments).

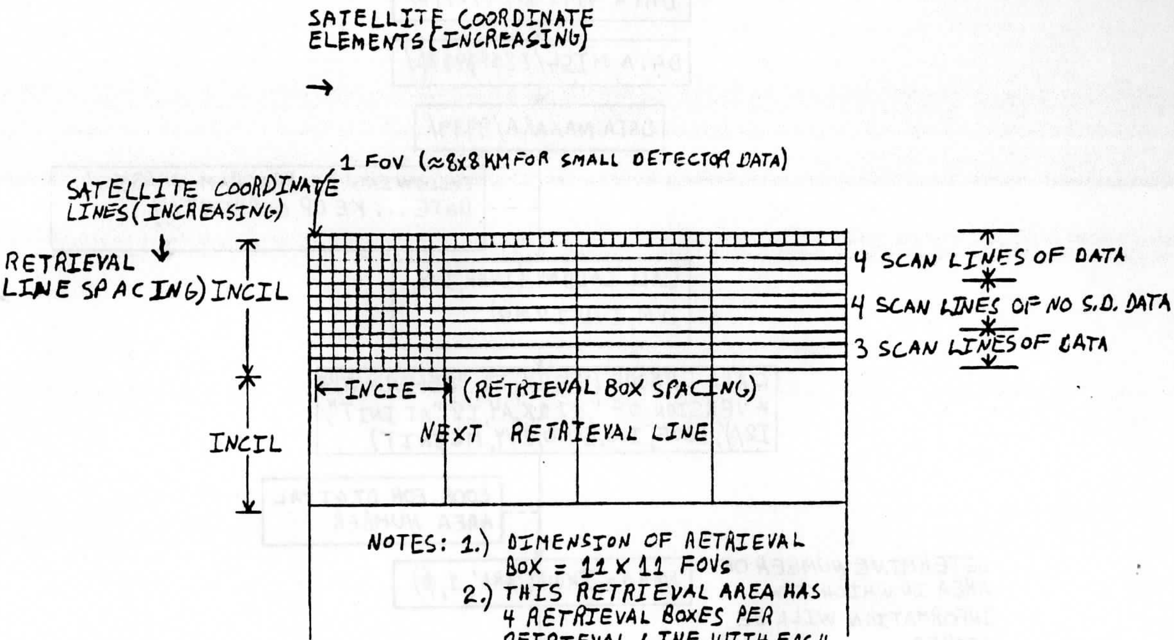
After all the retrieval lines in the retrieval area have been processed, the VASTEXT file is updated with the total number of retrievals generated (stored in variable LRO). This is accomplished by the call to subroutine VRTIO just before statement 840.

Finally, the output digital area (NARA) is closed, and a summary of the failures encountered during the processing of the retrieval area (the array NFAIL) is printed on the CRT. At this point, after VTPW has finished, system program DF can be used to load digital area NARA into a given image frame (i.e., it can be used to display the previously-generated image of TPWV on the video screen).

DIAGRAM 3

TYPICAL RETRIEVAL AREAS

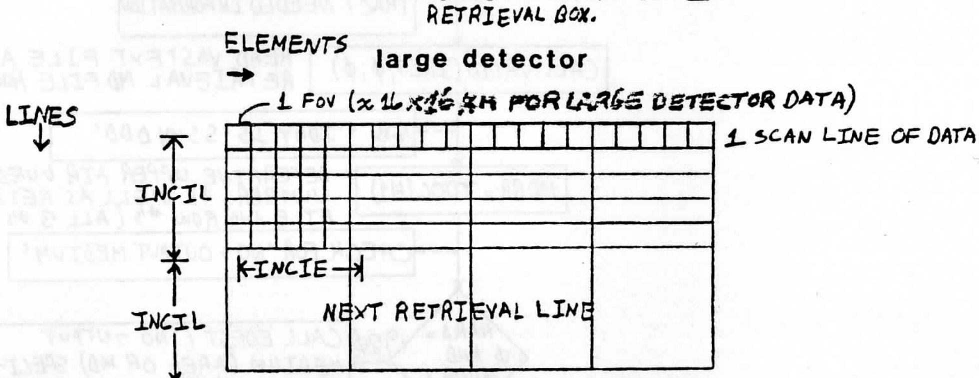
small detector



NOTES: 1.) DIMENSION OF RETRIEVAL BOX = 11 x 11 FOVs

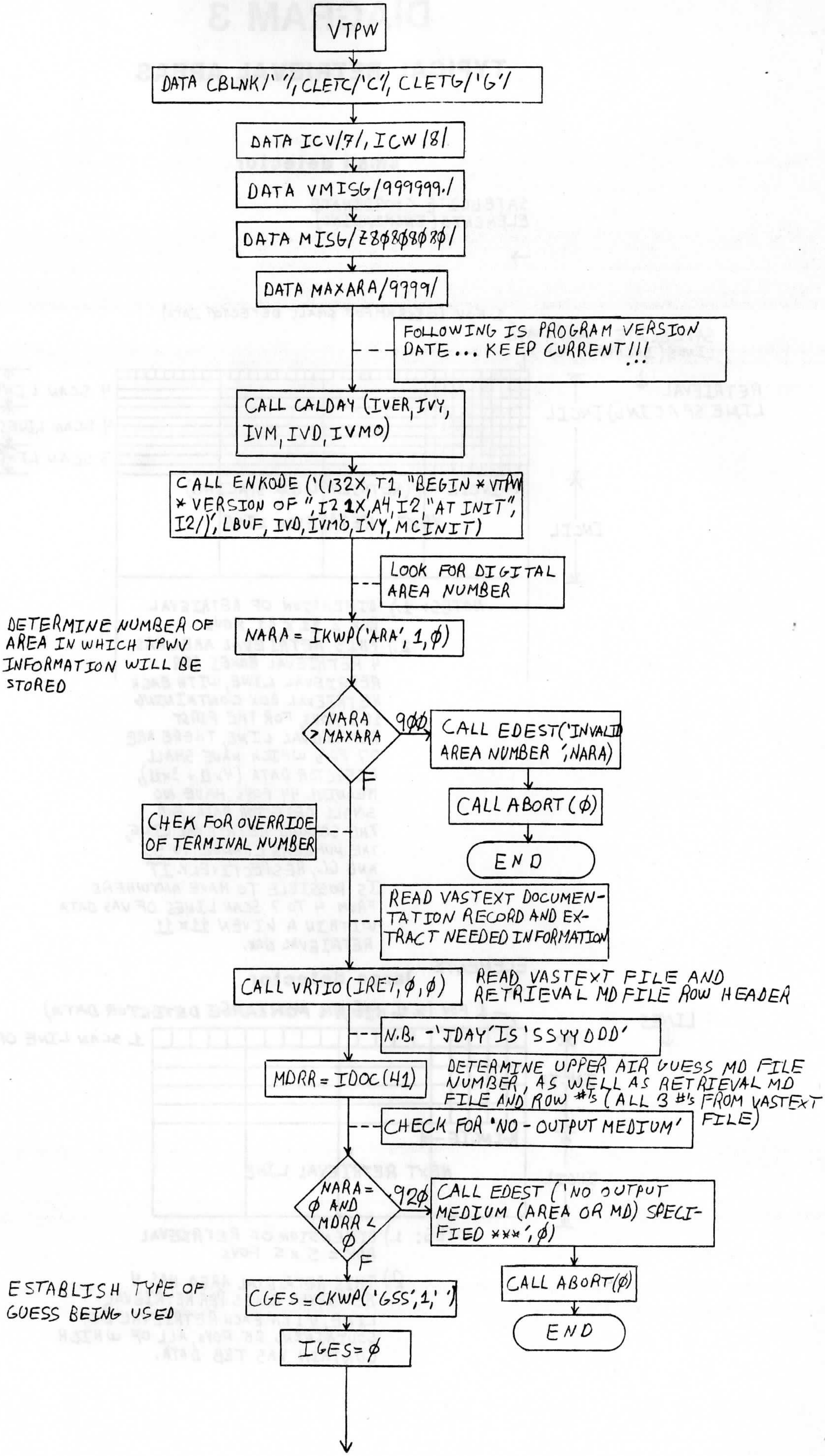
2.) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 121 FOVs, FOR THE FIRST RETRIEVAL LINE, THERE ARE 77 FOVs WHICH HAVE SMALL DETECTOR DATA (4x11 + 3x11), MEANING 44 FOVs HAVE NO SMALL DETECTOR DATA. FOR THE SECOND RETRIEVAL LINE, THE NUMBERS CHANGE TO 55 AND 66, RESPECTIVELY. IT IS POSSIBLE TO HAVE ANYWHERE FROM 4 TO 7 SCAN LINES OF VAS DATA WITHIN A GIVEN 11 x 11 RETRIEVAL BOX.

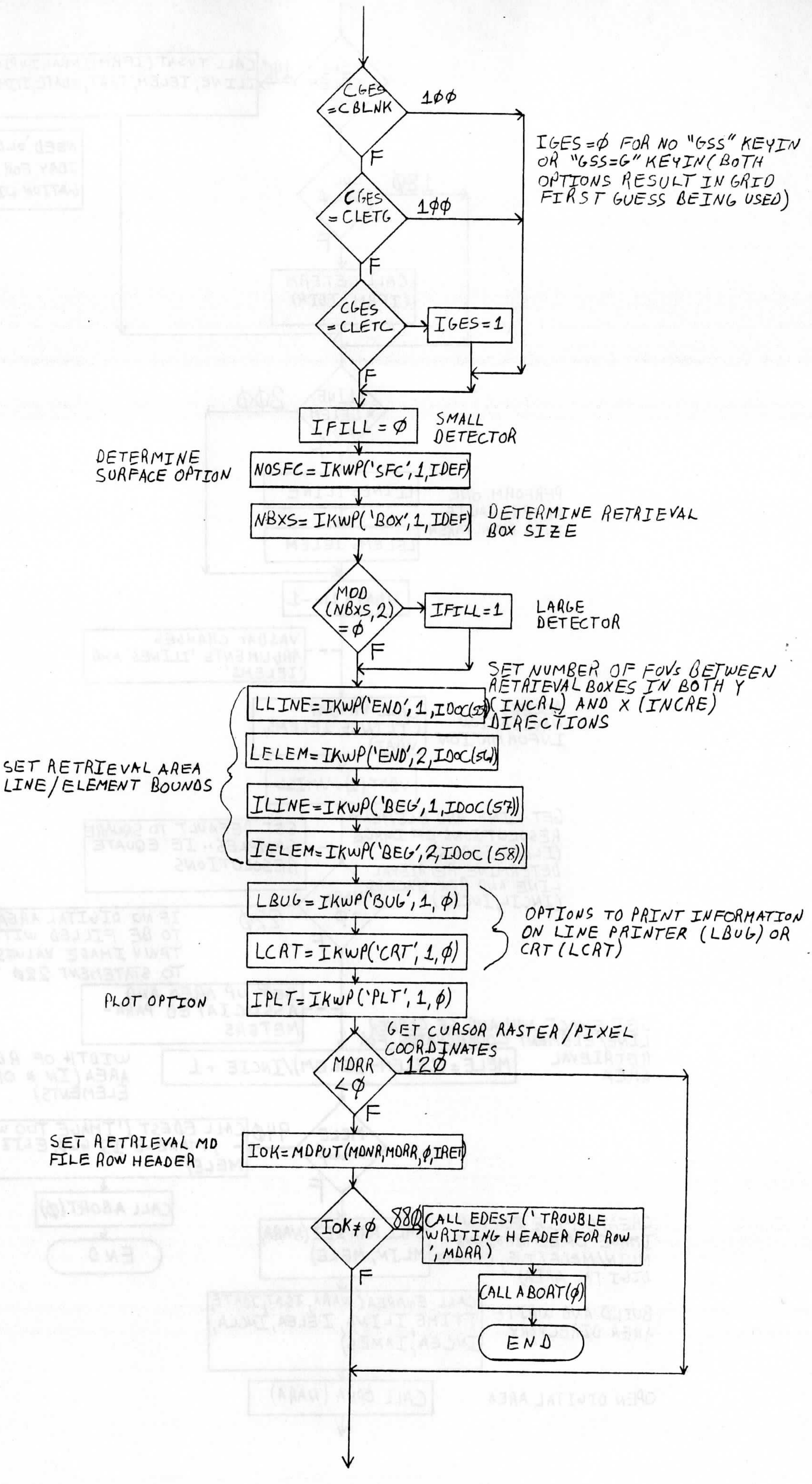
large detector



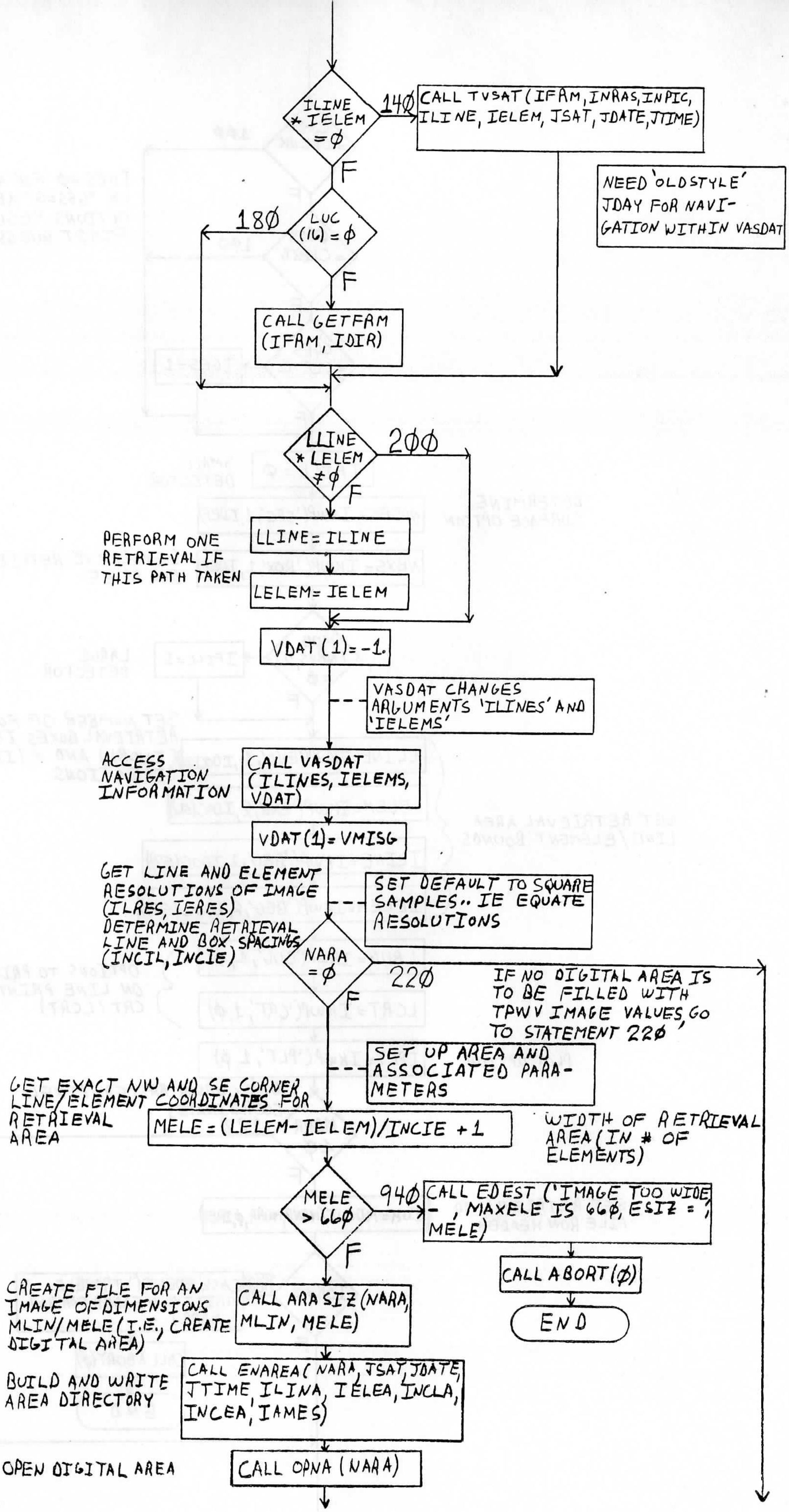
NOTES: 1.) DIMENSION OF RETRIEVAL BOX = 5 x 5 FOVs

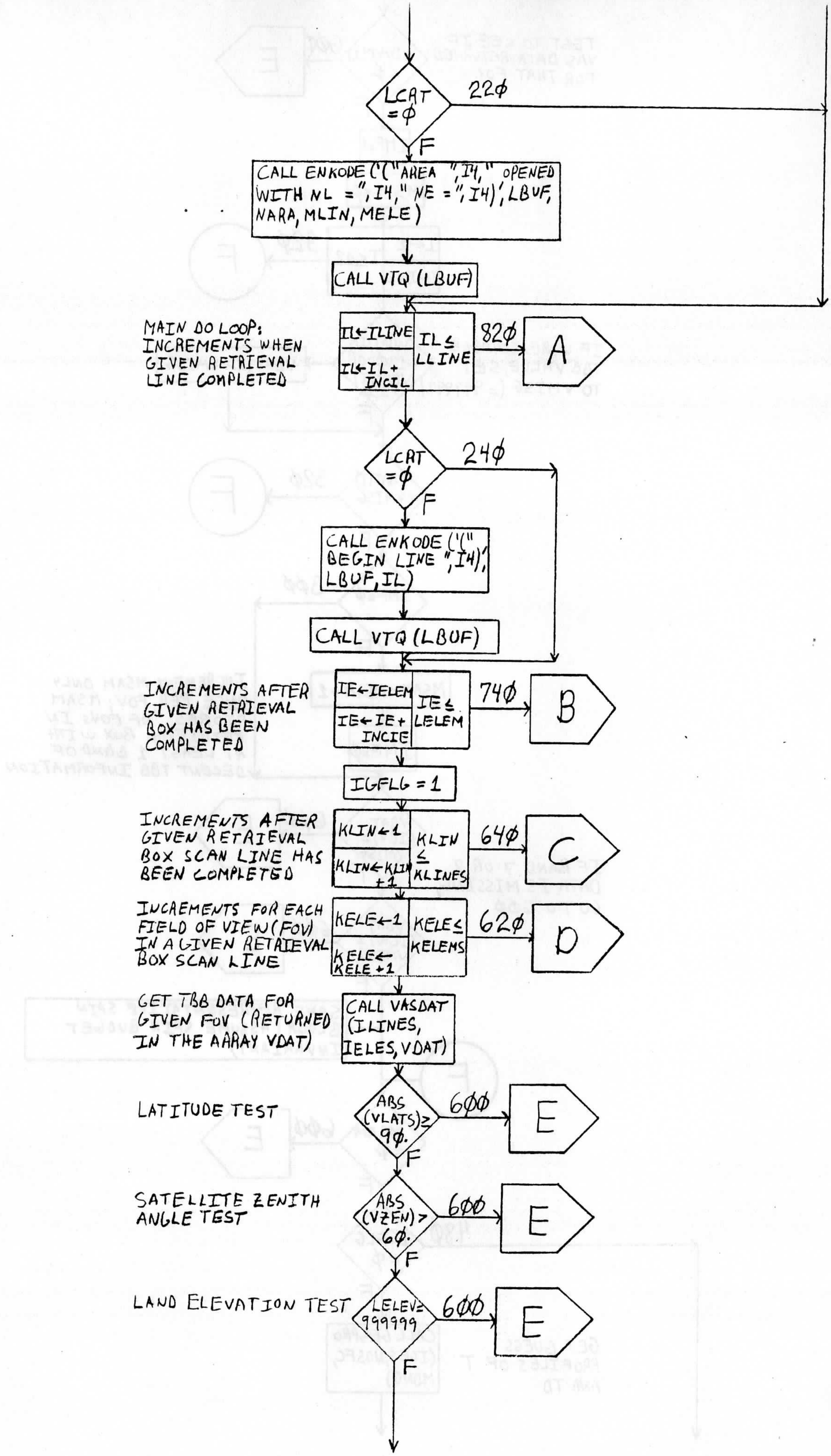
2.) THIS RETRIEVAL AREA HAS 4 RETRIEVAL BOXES PER RETRIEVAL LINE, WITH EACH RETRIEVAL BOX CONTAINING 25 FOVs, ALL OF WHICH CONTAIN VAS TBB DATA.


VTPW
DATA CBLNK/' '/, CLETC/'C'/, CLETG/'G'/
DATA ICV/7/, ICW/8/
DATA VMISG/999999./
DATA MISG/Z8Ø8Ø8Ø8Ø/
DATA MAXARA/9999/
FOLLOWING IS PROGRAM VERSION DATE... KEEP CURRENT!!!
CALL CALDAY (IVER, IVY, IVM, IVD, IVMO)
CALL ENKODE ('(132X, T1, "BEGIN * VTPW * VERSION OF ", I2 1X, A4, I2, "AT INIT", I2/)', LBUF, IVD, IVMO, IVY, MCINIT)
LOOK FOR DIGITAL AREA NUMBER
DETERMINE NUMBER OF AREA IN WHICH TPWV INFORMATION WILL BE STORED
NARA = IKWP('ARA', 1, Ø)
NARA > MAXARA
9ØØ
CALL EDEST('INVALID AREA NUMBER ', NARA)
CALL ABORT (Ø)
END
F
CHEK FOR OVERRIDE OF TERMINAL NUMBER
READ VASTEXT DOCUMENTATION RECORD AND EXTRACT NEEDED INFORMATION
CALL VRTIO (IRET, Ø, Ø)
READ VASTEXT FILE AND RETRIEVAL MD FILE ROW HEADER
N.B. - 'JDAY' IS 'SSYYDDD'
MDRR = IDOC(41)
DETERMINE UPPER AIR GUESS MD FILE NUMBER, AS WELL AS RETRIEVAL MD FILE AND ROW #'s (ALL 3 #'s FROM VASTEXT FILE)
CHECK FOR 'NO - OUTPUT MEDIUM'
NARA = Ø AND MDRR < Ø
92Ø
CALL EDEST ('NO OUTPUT MEDIUM (AREA OR MD) SPECIFIED ***', Ø)
CALL ABORT(Ø)
END
F
ESTABLISH TYPE OF GUESS BEING USED
CGES = CKWP('GSS', 1, ' ')
IGES = Ø

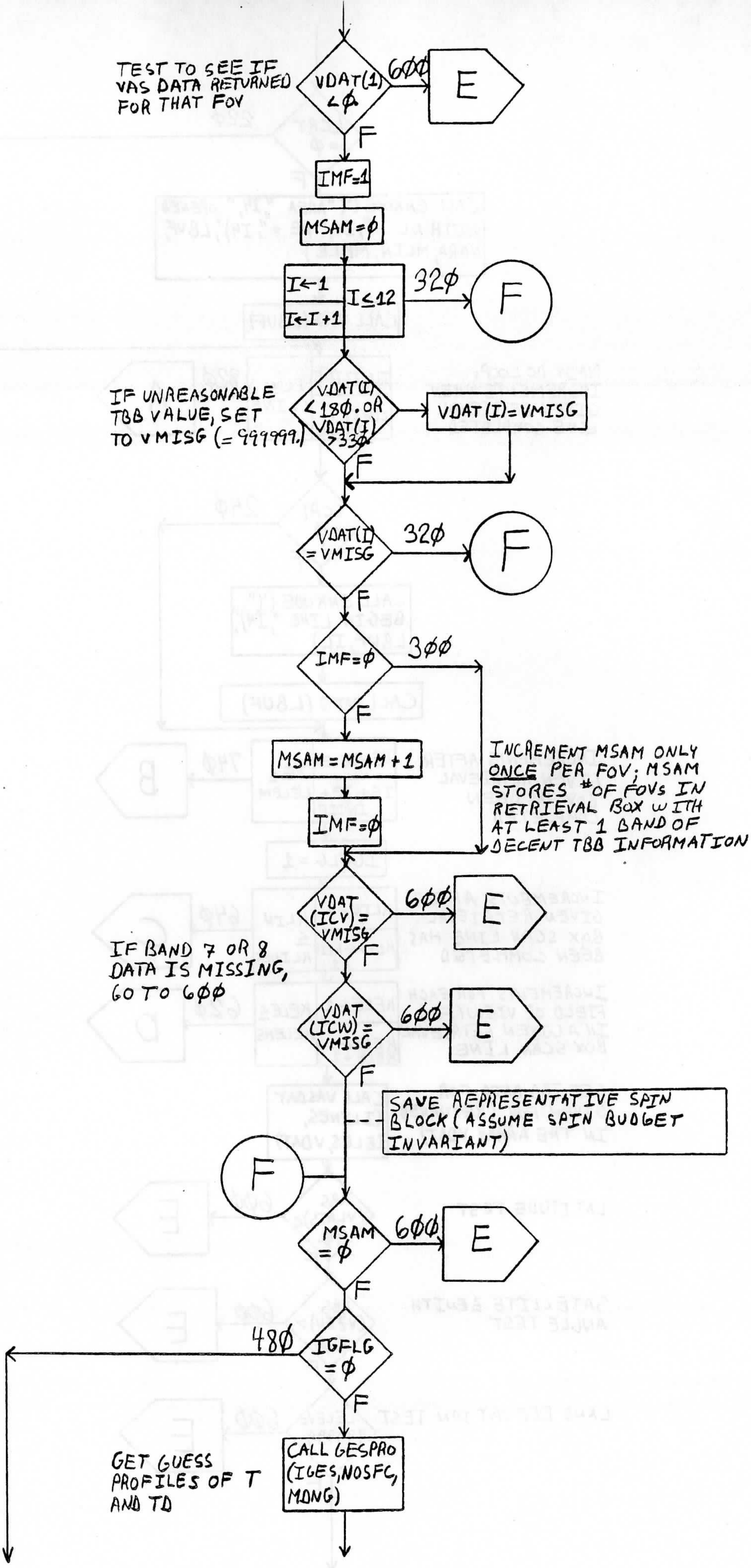


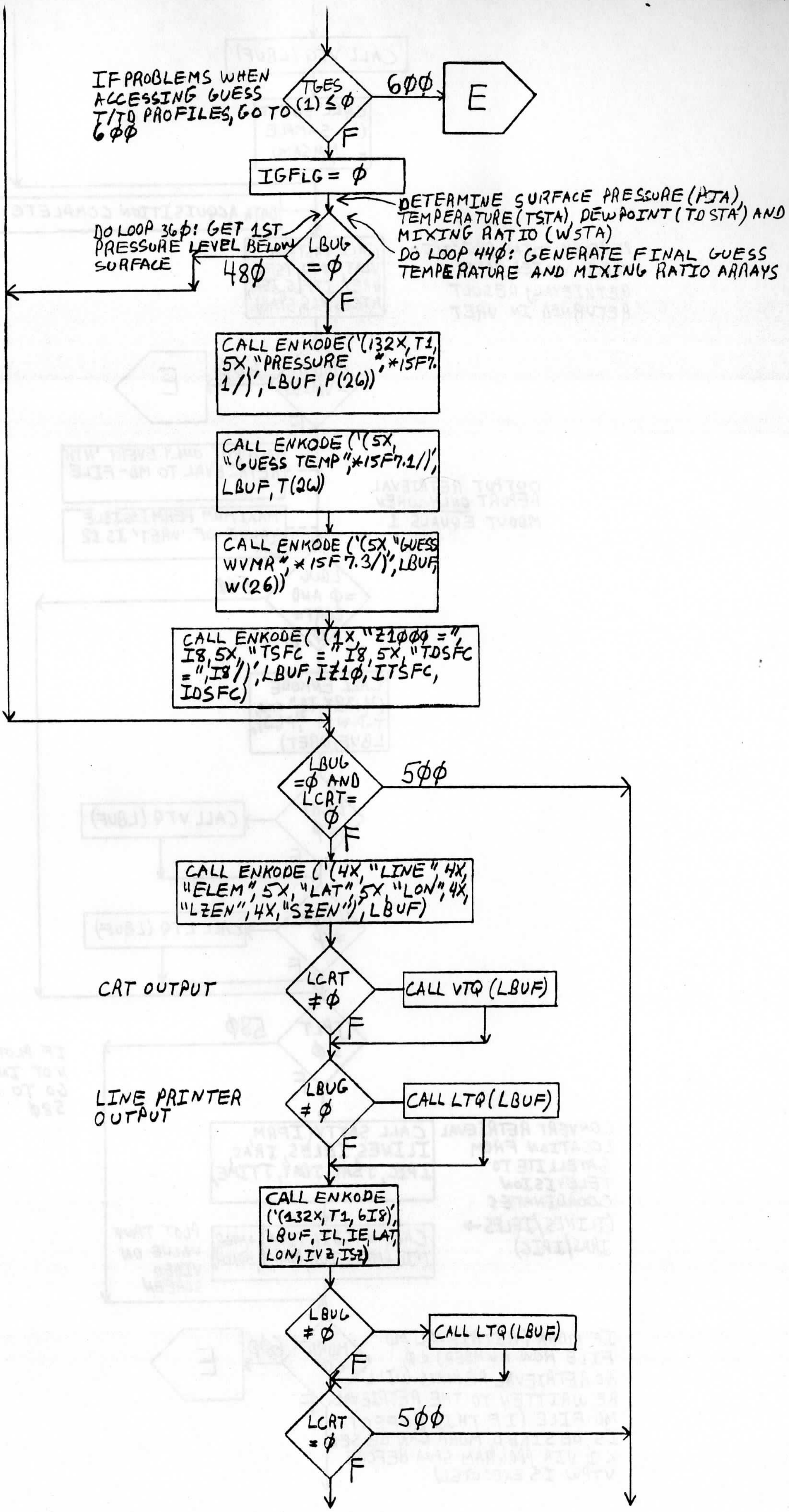
CGES =CBLNK
1ØØ
F
IGES=Ø FOR NO "GSS" KEYIN OR "GSS=G" KEYIN (BOTH OPTIONS RESULT IN GRID FIRST GUESS BEING USED)
CGES =CLETG
1ØØ
F
CGES =CLETC
IGES=1
F
IFILL=Ø
SMALL DETECTOR
DETERMINE SURFACE OPTION
NOSFC=IKWP('SFC',1,IDEF)
NBXS=IKWP('BOX',1,IDEF)
DETERMINE RETRIEVAL BOX SIZE
MOD (NBXS,2) =Ø
IFILL=1
LARGE DETECTOR
F
SET NUMBER OF FOVs BETWEEN RETRIEVAL BOXES IN BOTH Y (INCRL) AND X (INCRE) DIRECTIONS
LLINE=IKWP('END',1,IDOC(53))
LELEM=IKWP('END',2,IDOC(54))
SET RETRIEVAL AREA LINE/ELEMENT BOUNDS
ILINE=IKWP('BEG',1,IDOC(57))
IELEM=IKWP('BEG',2,IDOC(58))
LBUG=IKWP('BUG',1,Ø)
OPTIONS TO PRINT INFORMATION ON LINE PRINTER (LBUG) OR CRT (LCRT)
LCRT=IKWP('CRT',1,Ø)
PLOT OPTION
IPLT=IKWP('PLT',1,Ø)
GET CURSOR RASTER/PIXEL COORDINATES
MDRR <Ø
12Ø
F
SET RETRIEVAL MD FILE ROW HEADER
IOK=MDPUT(MDNR,MDRR,Ø,IRET)
IOK≠Ø
88Ø
CALL EDEST('TROUBLE WRITING HEADER FOR ROW ',MDRR)
F
CALL ABORT(Ø)
END

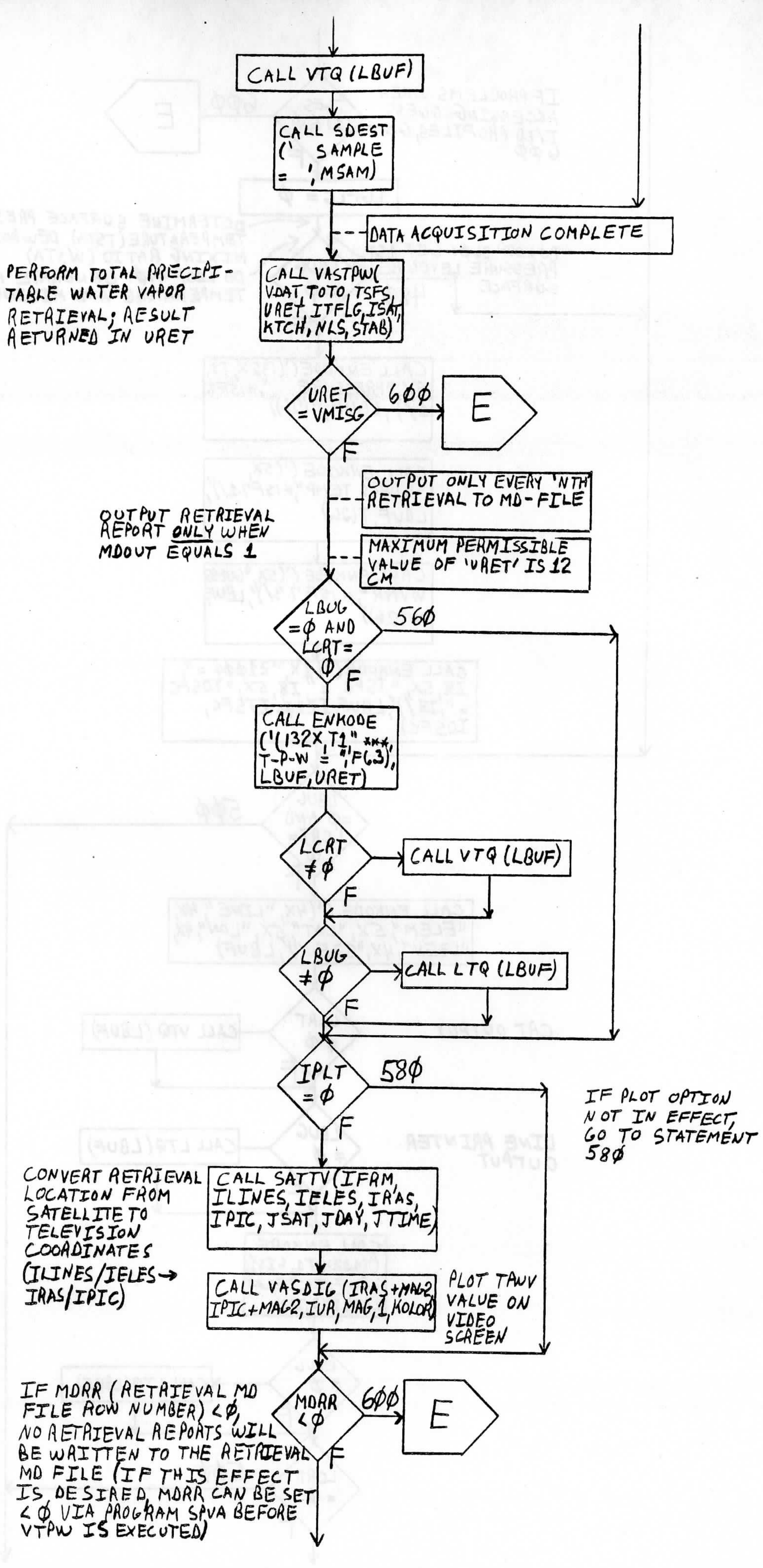

ILINE * IELEM = Ø
140
CALL TVSAT (IFRM, INRAS, INPIC, ILINE, IELEM, JSAT, JDATE, JTIME)
NEED 'OLDSTYLE' JDAY FOR NAVIGATION WITHIN VASDAT
F
180
LUC (16) = Ø
F
CALL GETFRM (IFRM, IDIR)
LLINE * LELEM ≠ Ø
200
F
PERFORM ONE RETRIEVAL IF THIS PATH TAKEN
LLINE = ILINE
LELEM = IELEM
VDAT(1) = -1.
VASDAT CHANGES ARGUMENTS 'ILINES' AND 'IELEMS'
ACCESS NAVIGATION INFORMATION
CALL VASDAT (ILINES, IELEMS, VDAT)
VDAT(1) = VMISG
GET LINE AND ELEMENT RESOLUTIONS OF IMAGE (ILRES, IERES)
DETERMINE RETRIEVAL LINE AND BOX SPACINGS (INCIL, INCIE)
SET DEFAULT TO SQUARE SAMPLES.. IE EQUATE RESOLUTIONS
NARA = Ø
220
F
IF NO DIGITAL AREA IS TO BE FILLED WITH TPWV IMAGE VALUES, GO TO STATEMENT 220
SET UP AREA AND ASSOCIATED PARAMETERS
GET EXACT NW AND SE CORNER LINE/ELEMENT COORDINATES FOR RETRIEVAL AREA
MELE = (LELEM-IELEM)/INCIE + 1
WIDTH OF RETRIEVAL AREA (IN # OF ELEMENTS)
MELE > 66Ø
940
CALL EDEST ('IMAGE TOO WIDE - , MAXELE IS 66Ø, ESIZ = ', MELE)
CALL ABORT(Ø)
END
F
CREATE FILE FOR AN IMAGE OF DIMENSIONS MLIN/MELE (I.E., CREATE DIGITAL AREA)
CALL ARASIZ (NARA, MLIN, MELE)
BUILD AND WRITE AREA DIRECTORY
CALL ENAREA(NARA, JSAT, JDATE, JTIME, ILINA, IELEA, INCLA, INCEA, IAMES)
OPEN DIGITAL AREA
CALL OPNA (NARA)

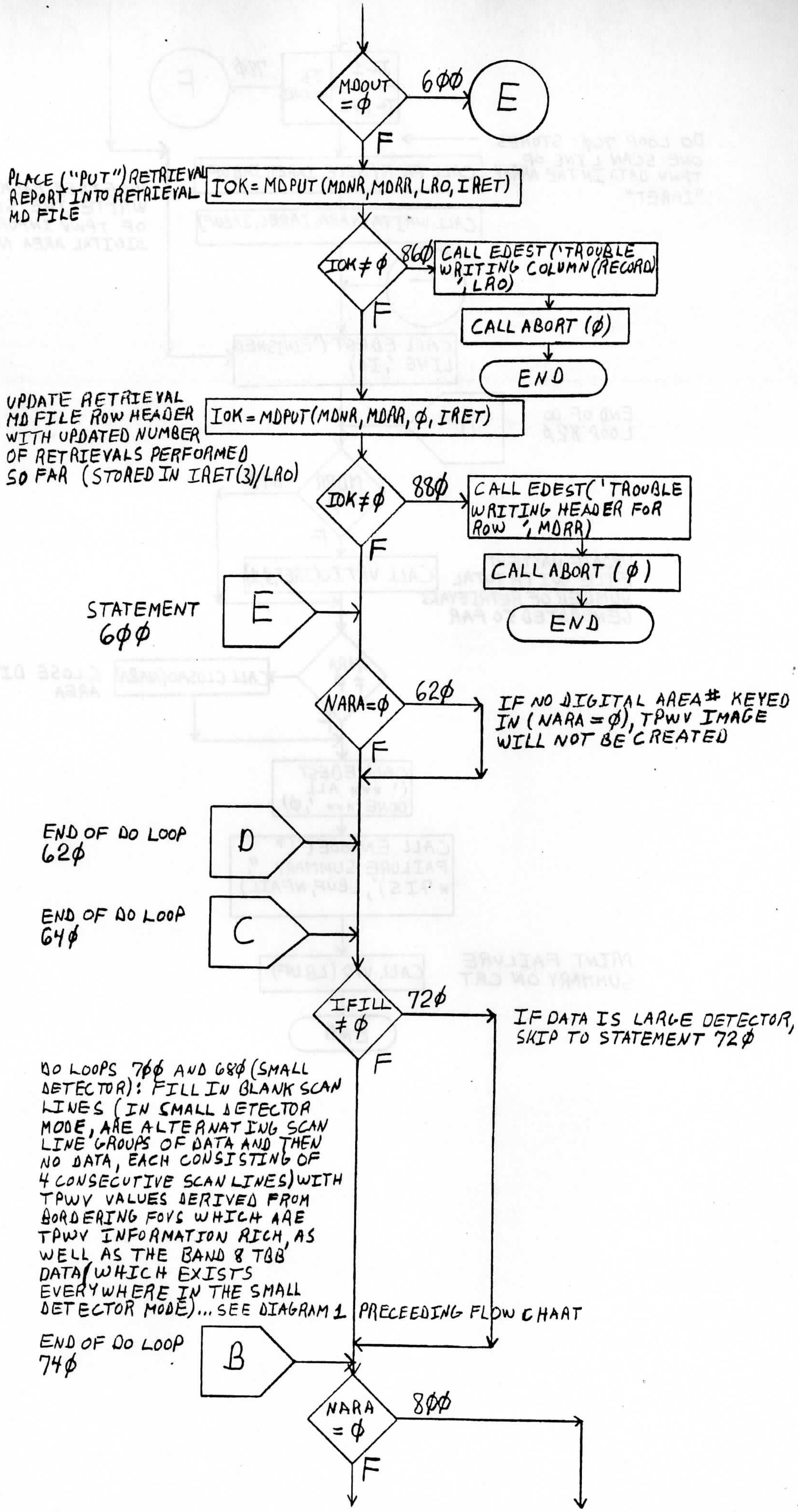


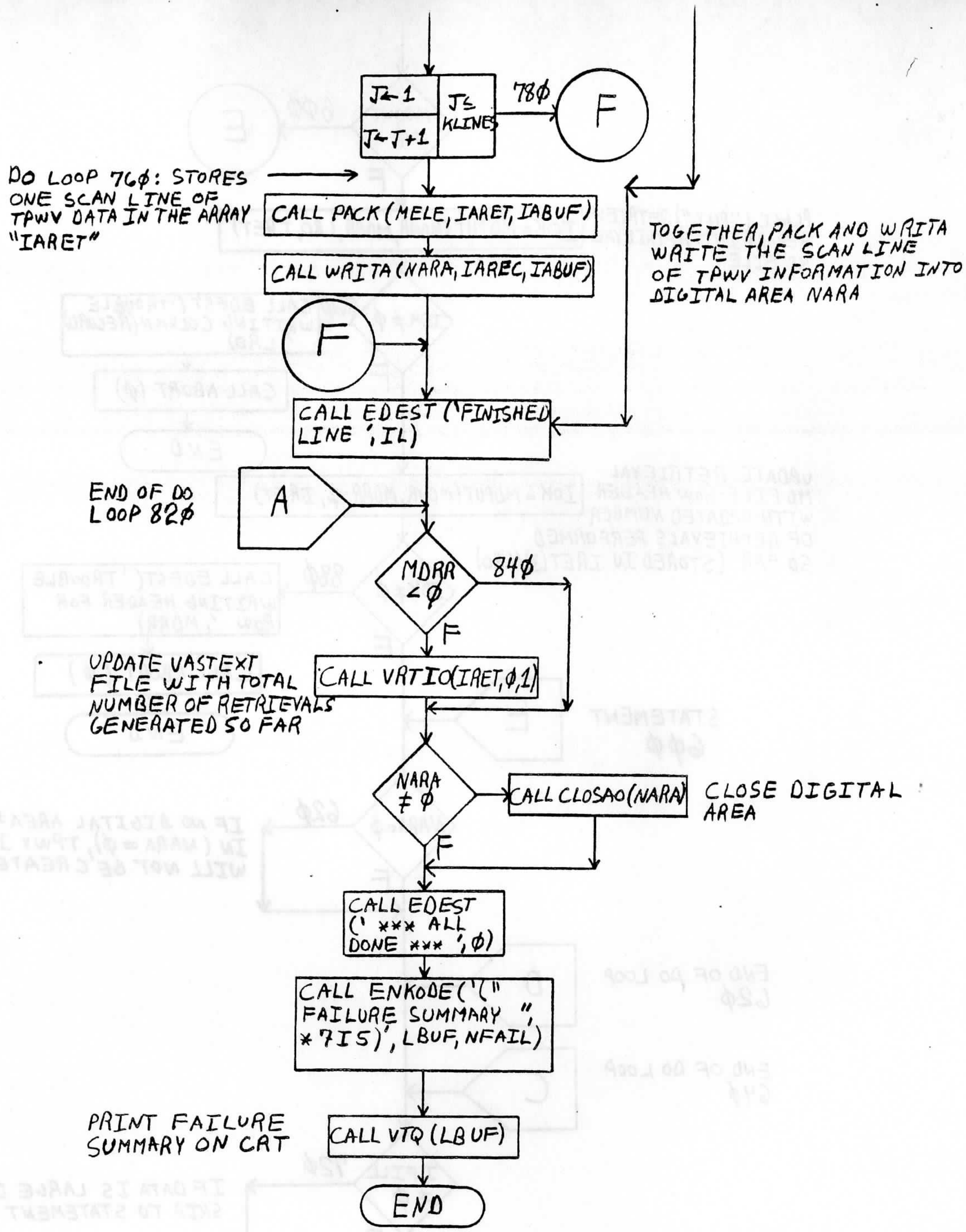
LCAT = ϕ
22ϕ
F
CALL ENKODE ('("AREA ",I4," OPENED WITH NL = ",I4," NE = ",I4)', LBUF, NARA, MLIN, MELE)
CALL VTQ (LBUF)
MAIN DO LOOP: INCREMENTS WHEN GIVEN RETRIEVAL LINE COMPLETED
IL←ILINE
IL←IL+INCIL
IL ≤ LLINE
82ϕ
A
LCAT = ϕ
24ϕ
F
CALL ENKODE ('(" BEGIN LINE ",I4)', LBUF, IL)
CALL VTQ (LBUF)
INCREMENTS AFTER GIVEN RETRIEVAL BOX HAS BEEN COMPLETED
IE←IELEM
IE←IE+INCIE
IE ≤ LELEM
74ϕ
B
IGFLG = 1
INCREMENTS AFTER GIVEN RETRIEVAL BOX SCAN LINE HAS BEEN COMPLETED
KLIN←1
KLIN←KLIN+1
KLIN ≤ KLINES
64ϕ
C
INCREMENTS FOR EACH FIELD OF VIEW (FOV) IN A GIVEN RETRIEVAL BOX SCAN LINE
KELE←1
KELE←KELE+1
KELE ≤ KELEMS
62ϕ
D
GET TBB DATA FOR GIVEN FOV (RETURNED IN THE AARAY VDAT)
CALL VASDAT (ILINES, IELES, VDAT)
LATITUDE TEST
ABS (VLATS) ≥ 9ϕ.
6ϕϕ
E
F
SATELLITE ZENITH ANGLE TEST
ABS (VZEN) > 6ϕ.
6ϕϕ
E
F
LAND ELEVATION TEST
LELEV= 999999
6ϕϕ
E
F


TEST TO SEE IF VAS DATA RETURNED FOR THAT FOV
VDAT(1) < ∅
6∅∅
E
F
IMF=1
MSAM=∅
I←1
I≤12
I←I+1
32∅
F
IF UNREASONABLE TBB VALUE, SET TO VMISG (=999999.)
VDAT(I) <18∅. OR VDAT(I) >33∅
VDAT(I)=VMISG
F
VDAT(I) = VMISG
32∅
F
F
IMF=∅
3∅∅
F
MSAM = MSAM + 1
IMF=∅
INCREMENT MSAM ONLY ONCE PER FOV; MSAM STORES # OF FOVs IN RETRIEVAL BOX WITH AT LEAST 1 BAND OF DECENT TBB INFORMATION
VDAT (ICV) = VMISG
6∅∅
E
F
IF BAND 7 OR 8 DATA IS MISSING, GO TO 6∅∅
VDAT (ICW) = VMISG
6∅∅
E
F
SAVE REPRESENTATIVE SPIN BLOCK (ASSUME SPIN BUDGET INVARIANT)
F
MSAM = ∅
6∅∅
E
F
48∅
IGFLG = ∅
F
GET GUESS PROFILES OF T AND TD
CALL GESPRO (IGES, NOSFC, MONG)


IF PROBLEMS WHEN ACCESSING GUESS T/TD PROFILES, GO TO 6Ø0
TGES (1) ≤ Ø
6ØØ
E
F
IGFLG = Ø
DETERMINE SURFACE PRESSURE (PSTA), TEMPERATURE (TSTA), DEWPOINT (TDSTA) AND MIXING RATIO (WSTA)
DO LOOP 36Ø: GET 1ST PRESSURE LEVEL BELOW SURFACE
DO LOOP 44Ø: GENERATE FINAL GUESS TEMPERATURE AND MIXING RATIO ARRAYS
LBUG = Ø
48Ø
F
CALL ENKODE ('(132X, T1, 5X, "PRESSURE ", *15F7.1/)', LBUF, P(26))
CALL ENKODE ('(5X, "GUESS TEMP", *15F7.1/)', LBUF, T(26))
CALL ENKODE ('(5X, "GUESS WVMR", *15F7.3/)', LBUF, W(26))
CALL ENKODE ('(1X, "Z1000 =", I8, 5X, "TSFC =" I8, 5X, "TDSFC =", I8/)', LBUF, IZ1Ø, ITSFC, IDSFC)
LBUG = Ø AND LCRT = Ø
5ØØ
F
CALL ENKODE ('(4X, "LINE", 4X, "ELEM", 5X, "LAT", 5X, "LON", 4X, "LZEN", 4X, "SZEN")', LBUF)
CRT OUTPUT
LCRT ≠ Ø
CALL VTQ (LBUF)
F
LINE PRINTER OUTPUT
LBUG ≠ Ø
CALL LTQ (LBUF)
F
CALL ENKODE ('(132X, T1, 6I8)', LBUF, IL, IE, LAT, LON, IVZ, ISZ)
LBUG ≠ Ø
CALL LTQ (LBUF)
F
LCRT = Ø
5ØØ
F


CALL VTQ (LBUF)
CALL SDEST (' SAMPLE = ', MSAM)
DATA ACQUISITION COMPLETE
PERFORM TOTAL PRECIPITABLE WATER VAPOR RETRIEVAL; RESULT RETURNED IN URET
CALL VASTPW(VDAT, TOTO, TSFC, URET, ITFLG, ISAT, KTCH, NLS, STAB)
URET = VMISG
6Ø
E
F
OUTPUT ONLY EVERY 'NTH' RETRIEVAL TO MD-FILE
OUTPUT RETRIEVAL REPORT ONLY WHEN MDOUT EQUALS 1
MAXIMUM PERMISSIBLE VALUE OF 'URET' IS 12 CM
LBUG =Ø AND LCRT=Ø
56Ø
F
CALL ENKODE ('(132X,T1," ***, T-P-W = ",F6.3)', LBUF, URET)
LCRT ≠Ø
CALL VTQ (LBUF)
F
LBUG ≠Ø
CALL LTQ (LBUF)
F
IPLT =Ø
58Ø
IF PLOT OPTION NOT IN EFFECT, GO TO STATEMENT 58Ø
F
CONVERT RETRIEVAL LOCATION FROM SATELLITE TO TELEVISION COORDINATES (ILINES/IELES → IRAS/IPIC)
CALL SATTV(IFRM, ILINES, IELES, IRAS, IPIC, JSAT, JDAY, JTIME)
CALL VASDIG (IRAS+MAG2, IPIC+MAG2, IUR, MAG, 1, KOLOR)
PLOT TAUV VALUE ON VIDEO SCREEN
IF MDRR (RETRIEVAL MD FILE ROW NUMBER) <Ø, NO RETRIEVAL REPORTS WILL BE WRITTEN TO THE RETRIEVAL MD FILE (IF THIS EFFECT IS DESIRED, MDRR CAN BE SET <Ø VIA PROGRAM SPVA BEFORE VTPW IS EXECUTED)
MDRR <Ø
6ØØ
E
F


MDOUT =ϕ
600
E
F
PLACE ("PUT") RETRIEVAL REPORT INTO RETRIEVAL MD FILE
IOK= MDPUT (MDNR, MDRR, LRO, IRET)
IOK≠ϕ
860
CALL EDEST ('TROUBLE WRITING COLUMN (RECORD)', LRO)
F
CALL ABORT (ϕ)
END
UPDATE RETRIEVAL MD FILE ROW HEADER WITH UPDATED NUMBER OF RETRIEVALS PERFORMED SO FAR (STORED IN IRET(3)/LRO)
IOK= MDPUT (MDNR, MDRR, ϕ, IRET)
IOK≠ϕ
880
CALL EDEST ('TROUBLE WRITING HEADER FOR ROW ', MDRR)
F
CALL ABORT (ϕ)
END
STATEMENT 6ϕϕ
E
NARA=ϕ
62ϕ
IF NO DIGITAL AREA # KEYED IN (NARA = ϕ), TPWV IMAGE WILL NOT BE CREATED
F
END OF DO LOOP 62ϕ
D
END OF DO LOOP 64ϕ
C
IFILL ≠ϕ
72ϕ
IF DATA IS LARGE DETECTOR, SKIP TO STATEMENT 72ϕ
F
DO LOOPS 76ϕ AND 68ϕ (SMALL DETECTOR): FILL IN BLANK SCAN LINES (IN SMALL DETECTOR MODE, ARE ALTERNATING SCAN LINE GROUPS OF DATA AND THEN NO DATA, EACH CONSISTING OF 4 CONSECUTIVE SCAN LINES) WITH TPWV VALUES DERIVED FROM BORDERING FOVS WHICH ARE TPWV INFORMATION RICH, AS WELL AS THE BAND 8 TBB DATA (WHICH EXISTS EVERYWHERE IN THE SMALL DETECTOR MODE)... SEE DIAGRAM 1 PRECEEDING FLOW CHART
END OF DO LOOP 74ϕ
B
NARA =ϕ
8ϕϕ
F


J ← 1
J ← J+1
J ≤ KLINES
78∅
F
DO LOOP 76∅: STORES ONE SCAN LINE OF TPWV DATA IN THE ARRAY "IARET"
CALL PACK (MELE, IARET, IABUF)
CALL WRITA (NARA, IAREC, IABUF)
TOGETHER, PACK AND WRITA WRITE THE SCAN LINE OF TPWV INFORMATION INTO DIGITAL AREA NARA
F
CALL EDEST ('FINISHED LINE', IL)
END OF DO LOOP 82∅
A
MDRR ≤ ∅
84∅
F
UPDATE VASTEXT FILE WITH TOTAL NUMBER OF RETRIEVALS GENERATED SO FAR
CALL VRTIO(IRET, ∅, 1)
NARA ≠ ∅
CALL CLOSAO(NARA)
CLOSE DIGITAL AREA
F
CALL EDEST (' *** ALL DONE *** ', ∅)
CALL ENKODE ('(" FAILURE SUMMARY ", * 7I5)', LBUF, NFAIL)
PRINT FAILURE SUMMARY ON CRT
CALL VTQ (LBUF)
END

Subroutines used by VTPW:

1) CALDAY-I
2) ENKODE-I
3) VRTIO-I
4) GETFRM-I
5) ARASIZ-I
6) OPNA-I
7) VTQ-I
8) GESPRO-I
9) SDEST-I
10) VASDAT-I
11) ENAREA-I
12) LTQ-I
13) VASTPW-I
14) SATTV-I
15) VASDIG-I
16) PACK-I
17) CLOSAO-I
18) EDEST-I
19) TVSAT-I
20) WRITA-I
21) ABORT-I
22) ENCODX
23) CONTNT
24) LTQ
25) ZECONV
26) MOVB
27) CLEANA
28) TQ
29) PRLINX
30) PRCLOS
31) LOCK
32) PRWR
33) PRRD
34) UNLOCK
35) PRCL
36) POST
37) PROPEN
38) BLKA
39) PRPRPR
40) DOPEN
41) DREAD
42) MOVCW
43) II
44) STC
45) DWRITE
46) ENCODE
47) DCLOSE
48) LWCLOS
49) MOVWC
50) LWMOP
51) LWPO
52) TRMNL
53) LWNEWF
54) DDEST
55) LWSO
56) CLEANW
57) MOVW
58) LWOPEN
59) LWGET
60) READDL
61) TQMES
62) ITOC
63) GETDAY
64) GETTIM
65) WRITDU
66) READD
67) RAOBIN
68) CSRAOB
69) CSRAOR
70) SGRAOB
71) CSRAOS
72) CSRAOM
73) CSRAOZ
74) CSRAOP
75) CSRAOI
76) MNRAOB
77) WMIX
78) INTPTW
79) EXTEMP
80) STDATM
81) CLMGES
82) VASGES
83) OUTINT
84) SURGES
85) GETSFV
86) IGNAME
87) PROFIX
88) SATEAR
89) SATPOS
90) ANGGET
91) SOLARP
92) ANGLES
93) VRTOPO
94) NVINIT
95) GETNAV
96) EPOCH
97) GETGAM
98) VASNAV
99) PLNKIV
100) OPNA
101) READOF
102) RBYTSX
103) ZEROS
104) RDTRK
105) MOVC
106) READDL
107) VASTAU
108) PRETAV
109) PREATV
110) ULMR
111) CO2TAV
112) H2OTAV
113) CONTAV
114) O3TAV
115) GAMTAV
116) PRECW
117) INITPL
118) PLTDIG
119) QGDASH
120) DSHOFF
121) PLOT
122) PAGE
123) WALK
124) PENMOV
125) PENBEG
126) PENADD
127) ENPT
128) SENOUT
129) ATOE
130) TEKPUT
131) BOX
132) DSHON
133) ENDPLT
134) WRTRK
135) CLOSA
136) EMES
137) JMBWTF

Acknowledgements

I wish to thank Dr. W. P. Menzel of the NOAA/NESDIS Advanced Satellite Products Project for his guidance and assistance throughout this project. In addition, Dr. C. M. Hayden of the NOAA/NESDIS Development Laboratory was indispensible in helping me understand the software itself, as were Dr. W. L. Smith and Messrs. H. M. Woolf and F. W. Nagle, of the NOAA/NESDIS Development Laboratory. Mr. G. S. Wade of the NOAA/NESDIS Development Laboratory helped extensively with Chapter 1 in addition to answering questions from time to time. Mr. P. Raphael helped with the flowcharting, and Messrs. A. J. Schreiner and T. H. Achtor also provided timely assistance.

Many other people helped with various other aspects of the manual, including Dr. J. F. Le Marshall, Messrs. G. M. Callan, H. B. Howell, R. C. Dengel, M. Barrett, D. A. Santek, W. L. Smith, Jr., and J. M. Benson, as well as Mrs. G. A. Dengel and Mses. D. J. Laitsch and A. M. Weickmann. Finally, I also thank Mesdames G. M. Turluck, A. Crowell and L. Beckett, Mses. J. M. Waite, A. L. North, S. Spellman, J. Slavin and S. Adler, and Messrs. A. W. Wendricks, J. Lam, J. Maynard and J. Sitzman for their help in preparing this manuscript.

Appendix I

General McIDAS Terminology

Appendix I contains descriptions of many of the McIDAS terms used in the manual, and consists of two sections. Section I consists of brief descriptions of many terms the user is likely to encounter while processing VAS retrievals. If the user has access to a McIDAS User's Manual, he/she may wish to augment Section I with the Manual's glossary. Section II, on the other hand, deals with file structure, and details the structure of files used during the retrieval process, including the VASTEXT file, User Common, MD files, grid files/grids, sounder areas, and digital areas.

I.) McIDAS Terminology

1.) alphanumeric display monitor (CRT): the display device used to present McIDAS programs, program outputs, etc., as opposed to images and (or) graphics (see also COLOR-VIDEO DISPLAY MONITOR).

2.) area: a section of disk which contains digital information; is used to store the representation of an image, in which the data is arranged by location in a 2-dimensional array; there are several types of areas, including

A.) VISSR (Visible Infrared Spin-Scan Radiometer) area: stores either a visible image or an 11-micron IR image

B.) VAS area (sounder area): stores VAS radiance information in anywhere from 1-12 bands

(see also SOUNDER AREA, VISSR AREA, or DNGSWT and DNSOUND from Section II).

3.) background: refers to McIDAS jobs that are submitted from editor mode which are usually heavily involved with numerical computation and development; as a result, background programs usually take longer to run than foreground jobs (see also FOREGROUND).

4.) band: a very narrow range of wavelengths to which the VAS instrument is sensitive (for example, the 11 micron band refers to the measurement of radiation in a very narrow band centered at 11 microns).

5.) color-video display monitor (TV): the display device used to view VAS images, graphics, etc., as opposed to program text and (or) output (see also ALPHANUMERIC DISPLAY MONITOR (CRT)).

6.) column: refers to a column within a given type of MD file (surface, retrieval, etc.) within which a certain number of reports exist, and is one of the coordinates used to locate reports within MD files (other coordinate is the row) (see also ROW, as well as DNMDDOC in Section II).

7.) column header: describes location (e.g., station number, latitude/longitude, elevation, etc.) for all reports at a particular geographic location (all reports in given column are from same geographic location), as well as number of reports in that column (see also DNMDDOC in Section II).

8.) coordinate system: a particular system of planar coordinates (x,y) for keeping track of location of surface reports, retrievals, etc.; there are 3 systems in use for the processing of VAS retrievals: satellite (line, element), TV (television) (raster, pixel) and earth (latitude, longitude); vertical coordinate is pressure in all 3 cases.

9.) cursor: a visual indicator which can be moved about the color-video display monitor either manually via the joysticks or automatically via a McIDAS keyin.

10.) digital area: refers to same entity as an area (see AREA).

11.) earth coordinates: (x,y) coordinates of given location measured relative to the earth, known obviously as latitude (y) and longitude (x).

12.) editor mode: one of 2 McIDAS system modes (other is McIDAS mode), is the mode in which programs can be modified using different edit commands, as well as executed.

13.) element coordinate: one of 2 satellite coordinates, gives location in the east-west (x) direction relative to the satellite image.

14.) foreground: refers to McIDAS jobs that are submitted from the McIDAS mode and are usually of an interactive nature; foreground jobs generally run quicker than background jobs (see also BACKGROUND).

15.) graphics: various depictions of maps, derived fields, parameter values, etc. that can be overlayed over different images (see also GRAPHICS FRAME).

16.) graphics frame: a color-video display monitor (TV) section of memory which is capable of holding the graphics for 1 image frame (note that a color-video display monitor has half as many graphics frames as image frames).

17.) grid: see DNGRID and DNGRIDS in Section II.

18.) grid file: see DNGRID and DNGRIDS in section II.

19.) image: the radiance "picture" sensed by the VAS for a given band (are therefore band 1 images, band 2 images, etc.) (see also DNGSWT).

20.) image frame: a color-video display monitor (TV) section of memory which is capable of holding an image of a given VAS band (note that color-video display monitors have a variable number of image frames, depending on the amount of memory accorded to the monitor in question).

21.) image loading : the act of placing a given VAS band's image in a certain color-video display monitor (TV) image frame via McIDAS program "DF" (see also IMAGE FRAME).

22.) initiator: refers to a part of system software which allows jobs to execute, a connection between a job and the system, are 10 initiators on the McIDAS system.

23.) joysticks: two small stick-like controls which are used to manually move the cursor; one control is for short, detailed movements, while the other is for movements which are longer, but less detailed.

24.) keylist: the list of keys for a schema which corresponds to a particular type of MD file (see also KEYS, SCHEMA).

25.) keys: alphanumeric labels which are exclusive to a given type of MD file (retrieval, upper air, surface, etc.) which represent various quantities stored within a given report from that type of MD file; for example, one might have a surface MD file, with the keys for this MD file's reports consisting of TSFC (surface temperature), TDSFC (surface dewpoint), Z100 (1000 mb height), etc.; then, a program which needed to access surface temperature from different surface reports would run through the keylist corresponding to the schema for that particular type of MD file, stopping when the key for surface temperature had been determined (see also DNMDDOC in Section II).

26.) keyword parameter: a small "word" (the keyword) or sequence of letters used to input data to a given McIDAS program, such as MD file number, grid file number, MD file row number, etc.; some examples would be: SPVA MDNR=4185, ESVA TSL=29100; keyword parameters are usually thought of as being used to input optional data to McIDAS programs (i.e., data secondary in importance to that inputted via positional parameters).

27.) Large Word (LW) array file: refers to a file that is stored on disk (MD files and gridfiles are 2 examples of LW array files).

28.) line coordinate: one of 2 satellite coordinates, gives location in the north-south (y) direction relative to the satellite image.

29.) macro: a type of program which consists mainly of execution of other programs, is usually used to consolidate the execution of a series of related programs.

30.) McIDAS mode: one of 2 McIDAS modes (other is editor mode), is the mode in which programs can only be executed, not modified or edited.

31.) MD file: see DNMDDOC in Section II.

32.) navigation: refers to images; relating satellite coordinates (line/element) in images to earth coordinates (latitude/longitude).

33.) non-video terminal: McIDAS terminals which consist solely of a keyboard input device coupled with an alphanumeric display monitor (see also VIDEO TERMINAL).

34.) page: a section of a Large Word (LW) array file which consists of 1024 4-byte words.

35.) pixel: the color-video display monitor (TV) coordinate in the x-direction.

36.) pointer: refers to a word in User Common which identifies the current sounder area, gridfile, or MD file for the user's terminal (see also USER COMMON).

37.) positional parameter: a given variable is referred to as such when it is included in the program keyin independent of any keyword (as opposed to keyword parameters, where the value of a given variable is entered in the program keyin in the company of a small keyword or sequence of letters); in McIDAS programs where both positional and keyword parameters are involved, positional parameters are usually thought of as being used to input data vital to the programs, while positional parameters are usually used to input optional data; some examples would be VPVA 4181 and PLVA Z 500.

38.) raster: the color-video display monitor (TV) coordinate in the y-direction.

39.) retrieval: the final temperature, moisture or total precipitable water vapor profile (value) derived from a first guess modified using the VAS radiance measurements (see also SOUNDING).

40.) retrieval area: a subset of a given sounder area within which retrievals are to be generated (can be smaller than or equal to sounder area in size) (see also SOUNDER AREA).

41.) retrieval line: an east-west line of retrieval boxes within a given retrieval area (see also RETRIEVAL BOX, RETRIEVAL AREA).

42.) row: refers to a row within a given type of MD file (surface, retrieval, etc.) within which a certain number of reports exist, and is one of the two coordinates used to locate reports within MD files (other coordinate is the column) (see also COLUMN, as well as DNMDDOC in Section II).

43.) row header: describes a date and time (or sensor number, date and time) which is valid for all reports in a given row of an MD file (note that all reports in a given row are valid for same time), as well as the number of reports in that row (see also DNMDDOC in Section II).

44.) satellite coordinates: (x,y) coordinates of given location measured relative to the VAS image, known as lines (y) and elements (x).

45.) scaling: relates to manner in which different variable quantities are stored in grid/MD files; for instance, whether given variable is stored as 1/100, 1/10, 100, etc. times its true value.

46.) scan line: an east-west line of individual VAS fields of view within a given sounder or retrieval area (note that a retrieval line will almost always have a latitudinal width of several scan lines) (see also RETRIEVAL LINE).

47.) schema: the way in which reports from a given type of MD file are structured, with each quantity within the report being represented by a corresponding key, such as TSFC for surface temperature, TDSFC for surface dewpoint, etc. (see also KEYS, KEYLIST; see also DNMDDOC in Section II).

48.) sounder area/file: an area used to store VAS radiance information in anywhere from 1-12 VAS bands; the user can think of a sounder area as a multi-spectral VISSR area, with each sounding location (field of view) being represented by 1-12 VAS measurements (see also DNGSWT and DNSOUND in Section II).

49.) sounding: the VAS radiance (brightness temperature) measurements for a given field of view, which are then used to generate the temperature/moisture retrievals (see also RETRIEVAL).

50.) spin budget: refers to the number of spins the geostationary satellite completes while the VAS instrument is sensing in a given band in order that a sufficiently strong signal-to-noise ratio is achieved.

51.) terminal: a McIDAS user's station which can be used to communicate with the system via a keyboard input; there are both video and non-video terminals.

52.) track: one circular band of storage area on a disk.

53.) TV: see COLOR-VIDEO DISPLAY MONITOR.

54.) TV coordinates: (x,y) coordinates of given location measured relative to the color-video display monitor, known as rasters (y) and pixels (x).

55.) User Common: terminal-dependent directory of pointers, terminal/cursor control information, etc. (see also DNUC in Section II).

56.) VAS pointer: the word of User Common which refers to the sounder area of VAS radiances the user wishes to process (see also POINTER).

57.) VASTEXT file: a file used to store many quantities vital to the user during his/her McIDAS activites (retrieval processing, in particular), including retrieval MD file number, surface MD file number, sounder area number, etc.; the contents of this file can be displayed at any time on the CRT via program LOVA (see also DNVASTXT in Section II).

58.) video terminal: a McIDAS user's station with a keyboard input to the system, an alphanumeric display monitor (CRT), and a color-video display monitor (TV) (see also NON-VIDEO TERMINAL).

59.) VISSR area: a type of area which stores either a visible image or an 11-micron IR image (see also AREA).

60.) VRET: the schema used when creating the retrieval MD file which stores retrievals of temperature/moisture (see also SCHEMA).

61.) water vapor and totals imagery: refers to individual images of total precipitable water vapor or total-totals index generated by various VAS retrieval programs (total precipitable water vapor--VTPZ or VTPW; total-totals index--VTPZ) (see also IMAGE).

62.) weighting function: a mathematical expression which evaluates quantitatively the contribution to radiation of a given wavelength and satellite zenith angle at any atmospheric pressure level, is unique for each wavelength, pressure level and satellite zenith angle.

II.) File Structure

List of Documents in Section II: DNGRID

DNGRIDS

DNGSWT

DNMDDOC

DNSOUND

DNUC

DNVASTXT

GRIDS ARE THE MCIDAS DATA STRUCTURES WHICH REPRESENT GEOGRAPHICALLY DISTRIBUTED DATA OF LOW RESOLUTION OR VOLUME (AS COMPARED TO SOUNDINGS OR AREAS). THEY ARE N X M MATRICES, LIMITED TO A TOTAL SIZE OF N X M = 10240 (ABOUT 4 TIMES GREATER THAN ON THE HARRIS MCIDAS SYSTEM). EACH NUMBER IN THE GRID IS AN OBSERVATION OR A CALCULATED VALUE OF A METEROLOGICAL QUANTITY, WHOSE GEOGRAPHICAL COORDINATES ARE INFERRED FROM THE MATRIX COORDINATES AND INFORMATION ABOUT THE PROJECTION THAT IS KEPT IN THE HEADER ASSOCIATED WITH THE GRID.

GRIDS ARE GROUPED INTO "GRID FILES", UP TO 159 GRIDS PER FILE. THERE IS NO NECESSARY RELATIONSHIP AMONG THE GRIDS IN A GRID FILE; THE FILES JUST PROVIDE A CONVENIENT WAY OF HANDLING LARGE NUMBERS OF GRIDS. MCIDAS PROVIDES FOR UP TO 9999 GRID FILES.

GRID FILE FORMAT

GRID FILES, LIKE MOST OTHER MCIDAS FILES, ARE DESIGNED TO BE ACCESSED BY THE LW- DISK I/O SUBROUTINES. THAT IS, THEY ARE THOUGHT OF AS A LARGE, CONTINUOUS ARRAY OF WORDS, WITH THE FIRST WORD NUMBERED 0. THE WORD NUMBERS OF THE VARIOUS COMPONENTS ARE AS FOLLOWS:

```
WORD
----
0-9    FILE ID. 0-7 ARE 32 CHARACTERS OF (EBCDIC) LABEL INFORMATION.
       8 IS THE PROJECT NUMBER OF THE USER WHO CREATED THIS FILE.
       9 IS THE FILE CREATION DATE (YYDDD).
----------
64-127    HEADER (64 WORDS) FOR GRID #1 (FORMAT BELOW)
128-191   HEADER FOR GRID #2
  ...
  ...
10176-10239   HEADER FOR GRID #159
----------
10240-20479    GRID #1
20480-30719    GRID #2
  ...
  ...
----------
```

THE FIRST WORD OF THE GRID HEADER FOR GRID N IS WORD # 64 X N. THE FIRST WORD OF GRID N (THE MATRIX ITSELF) IS WORD # 10240 X N. AN EMPTY GRID SLOT IS REPRESENTED BY A NEGATIVE VALUE IN THE FIRST WORD OF THE HEADER, WHICH OTHERWISE GIVES THE TOTAL SIZE OF THE GRID.

GRID FILE DOCUMENTATION
8/82

GRID FILES ON IBM/MCIDAS ARE LARGELY A CARRYOVER FROM THE SAME FILE STRUCTURE ON HARRIS/MCIDAS. A GRID STORES OBSERVATIONS MADE AT STATIONS ON A GEOGRAPHIC LATTICE. THE LATTICE SHOULD BE THOUGHT OF AS A CHECKERBOARD SUPERIMPOSED ON A PROJECTION OF THE EARTH WHICH IS EITHER (IN THE CURRENT IMPLEMENTATION) 1--PSUEDO-MERCATOR (I.E., LAT/LON GRID) OR 2--CONFORMAL. THE RECORDED OBSERVATIONS (HENCEFORTH CALLED THE "GRIDDED VARIABLE") ARE ASSUMED MADE AT SOME (POSSIBLY VARIABLE) HEIGHT WHICH IS DESCRIBED BY THE "LEVEL" PARAMETER IN THE GRID HEADER.

GRIDS ARE TYPICALLY MADE BY SOME PROCESS OF INTERPOLATION FROM AN MD ("METEOROLOGICAL DATA") FILE CONTAINING THE RAW OBSERVATIONS.

A GRID HAS 2 COMPONENTS: A 64-WORD HEADER WHICH DESCRIBES THE GRID PROJECTION, THE GRIDDED VARIABLE, LEVEL PARAMETER, THE TIME OF THE DATA. AND SO FORTH, AND THE ACTUAL GRID, WHICH IS JUST AN M X N-WORD MATRIX IN WHICH M X N CANNOT EXCEED 10240.

A GRID FILE CAN CONTAIN UP TO 159 GRIDS, NUMBERED 1...159. THERE IS ACTUALLY SPACE FOR A GRID "0", BUT THIS SPACE IS TAKEN UP BY THE COLLECTION OF 64-WORD GRID HEADERS. THESE HEADERS ARE ALSO NUMBERED 0...159 (HEADER 0 CONTAINS AN ARBITRARY 8-WORD ID FOR THE WHOLE FILE).

THE FORMULAE FOR LOCATING GRID HEADERS AND MATRICES IN LW- TERMS (THAT IS, TREATING THE FILE AS A GIANT ARRAY WHOSE FIRST WORD IS NUMBERED 0) ARE AS FOLLOWS:

HEADER N BEGINS AT WORD # 64 X N.
MATRIX N BEGINS AT WORD # 10240 X N.

THE VALUE 10240 WAS CHOSEN TO BE BIG ENOUGH FOR PRACTICAL APPLICATIONS (IT ADMITS GRIDS OF 100 X 100, 4 TIMES LARGER THAN THE MAXIMUM ON THE HARRIS). AND ALSO TO BE A MULTIPLE OF THE 1024-WORD DISK PAGE SIZE ON IBM/MCIDAS. GRID MATRICES CAN BE SMALLER THAN 10240; THE ACTUAL SIZE IS RECORDED IN THE GRID HEADER. BUT BECAUSE THE MCIDAS DISK I/O PACKAGE ALLOCATES PAGES ONLY AS NEEDED, SMALL GRIDS RESULT IN THE CONSUMPTION OF ONLY A SMALL AMOUNT OF DISK SPACE, EVEN THOUGH 10240 "VIRTUAL WORDS" ARE RESERVED FOR EACH GRID MATRIX.

GRID TYPES:

CURRENTLY 2 TYPES OF GRIDS ARE DEFINED (THE TYPE REFERS TO THE GEOGRAPHICAL PROJECTION): PSEUDO-MERCATOR (LAT/LON) GRIDS, AND CONFORMAL GRIDS. THE CONSTANTS DESCRIBING THESE 2 PROJECTIONS ARE GIVEN IN WORDS 33-38 OF THE GRID HEADER. LAT/LON GRIDS ARE 2-DIM MATRICES WITH 1-BASED (FORTRAN-TYPE) SUBSCRIPTS, THEN THE NORTHWEST CORNER (UPPER LEFT-HAND CORNER) CORRESPONDS TO THE (1,1) COORDINATE; LATITUDE DECREASES AS THE FIRST COORDINATE INCREASES; LONGITUDE DECREASES AS THE SECOND COORDINATE INCREASES; AND THE MATRIX IS STORED INTERNALLY BY COLUMN . YOU SHOULD FOLLOW THESE CONVENTIONS WHEN CREATING GRIDS IF THEY ARE TO BE PROPERLY HANDLED BY THE GRID UTILITIES.

KEYINS

IMAGE TAPES PRODUCED BY UW/SSEC
VERSION 3.0, 4/84

THIS DOCUMENT DESCRIBES THE TAPE FORMAT DEVISED BY UW/SSEC FOR DISTRIBUTION OF SATELLITE IMAGERY TO EXTERNAL USERS. THIS FORMAT IS DESIGNED TO BE READILY TRANSPORTABLE TO A WIDE CLASS OF MACHINES, AND PARTICULARLY EASILY TO MACHINES WITH 32-BIT WORDS.

THE FORMAT IS CHARACTERIZED BY FIXED-LENGTH TAPE BLOCKS WHICH ARE SELF-IDENTIFYING; THAT IS, EACH BLOCK CONTAINS WORDS SPECIFYING THE TYPE OF THAT BLOCK AND THE POSITION OF THE BLOCK'S DATA WITHIN THE WHOLE DATA SET. FIXED-LENGTH BLOCKS MAKE IT EASIER TO COMBINE DATA SETS OF VARYING TYPES ON A SINGLE TAPE, WITHIN THE LIMITS OF IBM JOB-CONTROL LANGUAGE; AND THE SELF-IDENTIFYING NATURE OF THE BLOCKS FACILITATES RECOVERY OF DATA AFTER A BAD PATCH ON THE TAPE, WHICH IS IMPORTANT WHEN THE TAPE FORMAT IS VIEWED AS AN ARCHIVE MEDIUM.

SATELLITE IMAGES AT SSEC

SATELLITE IMAGES PROCESSED AT SSEC OCCUR IN SEVERAL DIFFERENT FORMS, DEPENDING ON THE DATA SOURCE. TWO OF THE MOST IMPORTANT SOURCES ARE THE GOES VISSR AND GOES VAS INSTRUMENTS. THE IMAGES FROM THESE INSTRUMENTS ARE REFERRED TO AND DOCUMENTED IN DETAIL BELOW. REGARDLESS OF THE SOURCE, AN IMAGE CAN BE CONCEIVED OF AS A SEQUENCE OF IMAGE 'LINES' ARRANGED ONE BELOW THE OTHER AND NUMBERED FROM TOP TO BOTTOM, THE TOP LINE BEING NUMBERED ONE. EACH LINE CONSISTS OF A SEQUENCE OF IMAGE 'ELEMENTS' ARRANGED ACROSS THE LINE AND NUMBERED LEFT TO RIGHT, THE LEFT-MOST ELEMENT BEING NUMBERED ONE. AN ELEMENT MAY BE A SINGLE UNSIGNED 8-BIT QUANTITY, OR A SEQUENCE OF SIGNED 2'S COMPLEMENT 16-BIT QUANTITIES, DEPENDING ON THE IMAGE FORMAT. (IN VAS IMAGES, THE 16-BIT QUANTITIES REPRESENT RAW OBSERVATIONS IN DIFFERENT SPECTRAL BANDS AT THE SAME SPATIAL LOCATION.) IN ANY CASE, THE LINE/ELEMENT NUMBERING SCHEME ASSIGNS A PAIR OF COORDINATES TO EACH ELEMENT, CALLED THE 'IMAGE COORDINATES' OF THE ELEMENT. THIS COORDINATE SYSTEM IS DEFINED BY THE SATELLITE/CAMERA COMBINATION AND IS INDEPENDENT OF HOW THE DATA MAY BE STORED ON DISK OR TAPE.

AS AN EXAMPLE, GOES VISSR IMAGES IN THE VISIBLE-LIGHT BAND CONTAIN 14568 LINES, EACH WITH 15288 ELEMENTS.

WHEN A SATELLITE IMAGE IS TO BE TRANSFERRED TO TAPE BY THE MCIDAS SYSTEM, IT IS FIRST PREPARED IN A SPECIAL DISK STORAGE REGION CALLED A 'DIGITAL AREA', OR WHEN NO CONFUSION OF TERMS WOULD RESULT, JUST AN 'AREA'.

ASSOCIATED WITH THE SYSTEM OF DIGITAL AREAS IS AN ADDITIONAL DATA SET CALLED THE 'AREA DIRECTORY'. THE AREA DIRECTORY ENTRY FOR A GIVEN AREA CONTAINS AUXILIARY INFORMATION DESCRIBING IMPORTANT CHARACTERISTICS AND PARAMETERS OF THE AREA DATA, INCLUDING THE IMAGE FORMAT. THIS AREA DIRECTORY ENTRY FOR

AN AREA IS ALSO INCLUDED ON THE TAPE AND IS DOCUMENTED IN DETAIL BELOW.

THE DATA STORED IN A DIGITAL AREA IS ARRANGED IN A TWO-DIMENSIONAL ARRAY, LIKE THE IMAGE WHICH IT REPRESENTS. EACH ELEMENT IN AN AREA HAS A LINE NUMBER, STARTING FROM ONE AT THE TOP, AND A ELEMENT NUMBER, STARTING FROM ONE AT THE LEFT. THE LINE/ELEMENT NUMBER PAIR DEFINES A COORDINATE SYSTEM FOR THE ELEMENTS IN AN AREA, CALLED 'AREA COORDINATES'.

IF EVERY ELEMENT IN EVERY LINE OF A SATELLITE IMAGE WERE ALWAYS STORED IN A DIGITAL AREA, THERE WOULD BE NO POINT IN DISTINGUISHING BETWEEN IMAGE AND AREA COORDINATES. THIS IS NOT USUALLY THE CASE HOWEVER, SINCE IMAGES ARE ORDINARILY TOO LARGE TO PROCESS EFFICIENTLY IN TOTO. (THE VISIBLE-BAND IMAGE MENTIONED ABOVE WOULD REQUIRE OVER 200 MEGABYTES.) WHAT IS STORED IN AN AREA IS A RECTANGULAR SUBSET OF THE IMAGE, PRODUCED FROM THE ORIGINAL BY SAMPLING (IN THE VERTICAL DIRECTION) AND AVERAGING (IN THE HORIZONTAL DIRECTION. IN THE CASE OF VAS IMAGES, IT IS ALSO POSSIBLE TO ONLY INCLUDE IN THE DISK AREA A SUBSET OF THE AVAILABLE SPECTRAL BANDS, SO THAT EACH ELEMENT CONTAINS FEWER VALUES THAN ARE REPRESENTED IN THE ORIGINAL IMAGE.

THE IMAGE COORDINATES FOR AN ELEMENT ARE OBTAINED FROM ITS AREA COORDINATES BY THE FOLLOWING FORMULAS:

IMAGE LINE = UPPERLEFTLINE + (AREA LINE - 1) LINERES
IMAGE ELEMENT = UPPERLEFTELE + (AREA ELEMENT - 1) ELERES .

WHERE

UPPERLEFTLINE := IMAGE LINE # OF AREA ELEMENT (1,1)
UPPERLEFTELE := IMAGE ELEMENT # OF AREA ELEMENT (1,1)
LINERES := SPACING IN IMAGE LINES BETWEEN CONSECUTIVE AREA LINES
ELERES := SPACING IN IMAGE ELEMENTS BETWEEN CONSECUTIVE AREA ELEMENTS

THESE FOUR PARAMETERS MAY BE OBTAINED FROM THE AREA DIRECTORY ENTRY. WHEN LINERES = ELERES = 1, THE AREA IS SAID TO BE AT 'RESOLUTION 1', OR 'FULL VISIBLE RESOLUTION' (SINCE THE GOES MODE A/MODE AA VISIBLE CHANNEL IS CAPABLE OF A MAXIMUM RESOLUTION OF 1.) WHEN (AS ANOTHER EXAMPLE) LINERES = ELERES = 4, ONLY EVERY FOURTH LINE AND ELEMENT ARE REPRESENTED IN THE AREA, AND THE AREA IS SAID TO BE AT 'RESOLUTION 4', OR HAVE A 'BLOW-DOWN FACTOR' OF 4 FROM RESOLUTION 1.

INTERNAL ORGANIZATION OF A DIGITAL AREA * (AREA)
(COMMON FEATURES)

THE DATA RECORDED ON A TAPE FOR A SATELLITE IMAGE DISTRIBUTED BY SSEC IS AN EXACT COPY OF A DIGITAL AREA IN THE MCIDAS SYSTEM, SO IT IS NECESSARY TO KNOW THE INTERNAL STRUCTURE OF AREAS. EVERY AREA IS VIEWED AS A CONTINUOUS STREAM OF BYTES NUMBERED FROM 0. (EACH DATA BLOCK ON THE TAPE CONTAINS AN INDICATOR SHOWING WHERE THE BLOCK IS POSITIONED IN THIS CONTINUOUS STREAM.) WITHIN THE STREAM OF BYTES, THE AREA DATA IS PRESENTED LINE-BY-LINE, WITH THE FIRST LINE FIRST. EACH LINE IS FURTHER DIVIDED INTO

TWO PARTS, THE 'LINE PREFIX' AND THE ACTUAL LINE DATA (IMAGE ELEMENTS.) THE LINE PREFIX CONTAINS FURTHER DOCUMENTATION ABOUT THE IMAGE AND THE GIVEN LINE, AND CAN BE ESSENTIALLY IGNORED WHEN THAT INFORMATION IS NOT OF INTEREST.

THE SIZE OF THE PREFIX AND THE INFORMATION WITHIN IT DEPEND HEAVILY ON THE AREA TYPE, WHICH IN TURN IS DETERMINED BY THE DATA SOURCE. THE AREA TYPE IS GIVEN BY WORD 2 OF THE AREA DIRECTORY ENTRY (Q.V.). REGARDLESS OF AREA TYPE, EVERY LINE WITHIN THE AREA HAS THE SAME LENGTH PREFIX. THIS LENGTH, IN BYTES, IS GIVEN BY WORD 15 OF THE AREA DIRECTORY ENTRY. IT IS ALWAYS A MULTIPLE OF 4.

THE LINE PREFIX MAY OPTIONALLY BEGIN WITH A 'VALIDITY CODE'. ITS PRESENCE IS SIGNIFIED BY A NON-ZERO VALUE IN WORD 36 OF THE AREA DIRECTORY ENTRY. WHEN PRESENT, THE VALIDITY CODE OCCUPIES THE FIRST 4 BYTES OF THE LINE PREFIX. THESE FIRST 4 BYTES MUST MATCH EXACTLY WORD 36 OF THE AREA DIRECTORY ENTRY, OR ELSE THE LINE IS INVALID AS DATA AND MUST BE IGNORED.

THE LINE PREFIX IS FOLLOWED BY THE IMAGE ELEMENTS IN THAT LINE. AN ELEMENT MAY BE AS SHORT AS 1 BYTE (SINGLE-BAND IMAGES, WHICH HAVE 8-BIT PIXELS) OR AS LONG AS 24 BYTES (12 BANDS OF 16-BIT PIXELS.) THE NUMBER OF BYTES IN AN ELEMENT IS GIVEN BY WORD 11 OF THE AREA DIRECTORY ENTRY.

EACH LINE IN AN AREA HAS THE SAME TOTAL LENGTH, AND THIS LENGTH (IN BYTES) IS ALWAYS A MULTIPLE OF 4.

ADDITIONAL STRUCTURE OF GOES VISSR AREAS

THE GOES VISSR INSTRUMENT PRODUCES IMAGES IN TWO SPECTRAL BANDS, DUBBED 'VISIBLE' AND 'INFRARED' (IR). A PARTICULAR AREA CONTAINS ONLY ONE OF THESE TWO BANDS. WHICH BAND IT IS MAY BE DETERMINED FROM THE SATELLITE CODE (SSS) IN WORD 3 OF THE AREA DIRECTORY ENTRY. GENERALLY, EVEN-NUMBERED SSS REPRESENT VISIBLE IMAGES, AND ODD-NUMBERED SSS REPRESENT IR IMAGES.

EVERY ELEMENT IN A VISSR AREA CONTAINS JUST ONE 8-BIT PIXEL, REPRESENTING RAW DATA FROM THE INSTRUMENT. IF THE AREA CONTAINS IR DATA, THE OBSERVED TEMPERATURE MAY BE INFERRED FROM THE PIXEL VALUE, ACCORDING TO THE FOLLOWING FORMULAE:

$$T = 418 - B \quad (B>176 \text{ OR } B=176)$$

$$T = 330 - (B / 2) \quad (B<176 \text{ OR } B=176)$$

WHERE:

T IS BRIGHTNESS TEMPERATURE (DEGREES K)

B IS PIXEL VALUE (0...255)

NOTE THAT FOR IR DATA, THE HIGHEST PIXEL VALUES CORRESPOND TO THE COLDEST TEMPERATURES ("SPACE IS WHITE").

THE LINE PREFIX IN A VISSR AREA MAY BE ABSENT ENTIRELY, OR MAY BE JUST 4 BYTES FOR A VALIDITY CODE. FOR IR AREAS ONLY, THE LINE PREFIX MAY INCLUDE 128 BYTES FOR THE IR DOCUMENTATION TRANSMITTED BY THE SATELLITE ALONG WITH THE DATA. BUT NOTE THAT MOST OF THE USEFUL INFORMATION IN THE IR DOCUMENTATION HAS BEEN PROCESSED BY SSEC INTO MORE USEFUL FORM IN NAVIGATION

RECORDS, WHICH ARE INCLUDED ON TAPE ALONG WITH THE DATA.

THE HIGHEST RESOLUTION (I.E. LOWEST VALUES OF LINERES AND ELERES IN THE AREA DIRECTORY ENTRY) POSSIBLE FOR A VISIBLE AREA IS 1; FOR AN IR AREA IT IS ONLY 4 (LONGER WAVELENGTHS HAVE INHERENTLY LESS RESOLUTION).

ADDITIONAL STRUCTURE OF VAS AREAS (SOUNDER AREAS) *

THE VAS INSTRUMENT PRODUCES OBSERVATIONS IN UP TO 12 DIFFERENT IR SPECTRAL BANDS AT A GIVEN SPATIAL LOCATION, PLUS A VISIBLE BAND. ALL OR SOME OF THE IR BANDS MAY BE REPRESENTED TOGETHER IN A SINGLE AREA OF VAS TYPE; BUT THE VISIBLE BAND IS REPRESENTED SEPARATELY IN AN AREA OF VISSR TYPE. THUS, AS WITH GOES VISSR IMAGES, IT MAY REQUIRE A PAIR OF AREAS TO OBTAIN THE TOTAL INFORMATION TRANSMITTED BY THE SATELLITE DURING A GIVEN TIME PERIOD.

THE CHARACTERISTICS OF THE VAS BANDS ARE GIVEN BELOW:

BAND NUM.	SPECTRAL CENTER UM	1/CM	FILTERS WIDTH 1/CM	PURPOSE FOR SOUNDING	MAIN ABSORBING GAS	OTHER SIGNIFICANT EFFECTS
1	14.73	678.7	10.	TEMP	CO2	O3
2	14.48	690.6	16.	TEMP	CO2	O3
3	14.25	701.6	16.	TEMP	CO2	O3
4	14.01	713.6	20.	TEMP	CO2	O3
5	13.33	750.	20.	TEMP	CO2,H2O	O3
6	4.525	2210.	45.	TEMP+CLOUD	N2O	SUN
7	12.66	790.	20.	MOISTURE	H2O	CO2+DUST
8	11.17	895.	140.	SURFACE	H2O	CO2+DUST
9	7.261	1377.2	40.	MOISTURE	H2O	CO2
10	6.725	1487.	150.	MOISTURE	H2O	--
11	4.444	2250.	40.	TEMP+CLOUD	N2O,CO2	SUN
12	3.945	2535.	140.	SURFACE	H2O	SUN+DUST

MD- (METEROLOGICAL DATA) FILES DOCUMENTATION

INTRODUCTION

CONCEPTUALLY, MD FILES LOOK LIKE A LARGE 2-DIMENSIONAL MATRIX OF DATA RECORDS. THESE RECORDS ARE ADDRESSED BY 2-D COORDINATES: E.G. (3,4). (THE UPPER-LEFT-MOST RECORD IS (1,1).) ATTACHED TO EACH ROW IS A LABEL-- AN ADDITIONAL RECORD OF SEVERAL WORDS-- KNOWN AS THE ROW HEADER. SIMILARLY, ATTACHED TO EACH COLUMN IS A COLUMN HEADER. LOGICALLY SPEAKING, THE WHOLE FILE THUS LOOKS LIKE A LARGE TABLE, WITH MULTIPLE ITEMS OF INFORMATION IN EACH BOX, AND LABELS ATTACHED TO EACH ROW AND COLUMN. THE ACTUAL USEFUL INFORMATION ASSOCIATED WITH EACH BOX CONSISTS OF THE INFORMATION IN THE BOX, PLUS THE ASSOCIATED ROW AND COLUMN LABELS. USING THE MD JARGON:

=====> RECORD= ROW HEADER + COLUMN HEADER + DATA PORTION

IN MATRIX COORDINATES, THE ROW HEADERS ARE THOUGHT TO BE IN COLUMN 0, AND THE COLUMN HEADERS IN ROW 0. THUS:

=====> RECORD = (M,0) + (0,N) + (M,N)

THE RECORD THUS DEFINED IS SIMPLY A GROUP OF WORDS (32-BIT ENTITIES), IN THE ORDER JUST GIVEN. THE OTHER KEY CONCEPT OF MD FILES IS THAT THE WORDS IN A RECORD ARE GIVEN NAMES, CALLED "KEYS" THESE NAMES ARE RECORDED IN ADVANCE, IN A LIST CALLED A FILE "SCHEMA", AND THE ENTIRE SCHEMA IS GIVEN A NAME, WHICH IN A SENSE DEFINES THE FILE TYPE. FOR EXAMPLE, THE SCHEMA DEFINING MD FILES TO CONTAIN DATA FROM RADIOSONDE OBSERVATIONS IS CALLED "RAOB", AND CONSISTS OF KEYS SUCH AS "P" (PRESSURE) "T" (TEMPERATURE) "TD" (DEWPOINT), AND SO FORTH. (ALL KEY NAMES ARE 1-4 LETTERS.) ONCE DEFINED, THIS SCHEMA DESCRIBES THE DATA IN ALL MD FILES OF THIS TYPE.

A USEFUL CONVENTION, WHICH SHOULD BE FOLLOWED WHENEVER THERE IS NO COMPELLING REASON NOT TO, IS THAT SCHEMAS ARE DESIGNED SO THAT ALL DATA FOR A PARTICULAR TIME IS GROUPED IN A SINGLE ROW, AND THE ROW HEADER DESCRIBES THE TIME (I.E., DATE AND TIME, PERHAPS SENSOR TYPE,DATE, AND TIME); ALL DATA OBSERVED AT A PARTICULAR GEOGRAPHIC LOCATION IS GROUPED IN A SINGLE COLUMN, AND THE COLUMN HEADER DESCRIBES THE LOCATION (E.G., STATION #, ITS LATITUDE AND LONGITUDE, ELEVATION, ETC.). THEN THE TIME AND PLACE OF AN OBSERVATION CAN DETERMINE A UNIQUE ROW,COLUMN COORDINATE AT WHICH TO PLUNK DOWN THE DATA COMPRISING THE OBSERVATION. THIS KIND OF FILE ORGANIZATION KEEPS THE DATA FORMAT VERY CLOSE TO THE ACTUAL STRUCTURE OF THE DATA, AND TO INTUITION.

SCHEMAS ARE PREPARED USING THE EDITOR; THEIR FORMAT WILL BE SPECIFIED ELSEWHERE. ONCE PREPARED, THE SCHEMA MUST BE "REGISTERED" WITH THE MD SYSTEM (USING A PROGRAM CALLED SCHEMA) BEFORE THEY CAN BE USED TO CREATE NEW MD FILES.

MD FILE LAYOUT. (ALL WORD #'S ARE 0-BASED)

THE MD FILES ARE IMPLEMENTED AS LW ("LARGE WORD ARRAY") LEVEL FILES, USING THE LW- PREFIX SUBROUTINES. IN OTHER WORDS, THE MATRIX-OF-RECORDS STRUCTURE THAT DISTINGUISHES THE MD LEVEL IS SUPERIMPOSED ON A STRUCTURE THAT PRESENTS A FILE AS A SIMPLE CONTIGUOUS ARRAY OF WORDS. THE LOCATION OF THE VARIOUS MD FILE COMPONENTS IN THE CONTIGUOUS-WORD SPACE ARE GIVEN BELOW (ALL WORD #'S ARE 0-BASED)

FIRST 4096 WORDS:

FIRST 64 WORDS ARE THE FILE HEADER (LABEL)

NEXT 400 WORDS ARE THE USER RECORD (MD COORDINATES 0,0). THIS BLOCK OF WORDS IS NOT DESCRIBED BY ANY PART OF THE SCHEMA AND IS AVAILABLE FOR STORAGE OF ARBITRARY INFORMATION.

NEXT 400 WORDS CONTAIN THE NAMES OF THE FILE KEYS

NEXT 400 WORDS CONTAIN THE SCALE FACTORS OF THE KEYS

NEXT 400 WORDS CONTAIN THE UNITS OF THE KEYS

ALL WORDS BETWEEN HERE AND WORD# 4095 ARE RESERVED FOR LATER EXPANSION OF THE SCHEMA, SO DONT USE THIS AREA FOR ANYTHING

BEGINNING AT WORD# 4096:

FIRST NR NRKEYS WORDS CONTAIN ALL THE ROW HEADERS

NEXT NC NCKEYS WORDS CONTAIN ALL THE COLUMN HEADERS

NEXT NR NC NDKEYS WORDS CONTAIN THE DATA PORTION: THE MAIN MATRIX. THE DATA RECORDS ARE STORED ROW-BY-ROW, SO ROW-BY-ROW ACCESS WILL PROVE MOST EFFICIENT.

THE NEXT WORD AFTER HERE IS MARKED BY FENDOF IN MDCOM. ALL STORAGE BEYOND THIS POINT IS AVAILABLE FOR USER PURPOSES.

SOUNDING FILES. (SOUNDER AREAS)

11/29/82

THIS DOCUMENT DESCRIBES THE STRUCTURE OF SOUNDING FILES ON IBM MC-IDAS SYSTEMS.

SOUNDING FILES ON THE IBM ARE ONE OF THE SUBTYPES OF THE AREA SYSTEM. THAT IS THEY HAVE A LINE ORIENTED STRUCTURE AND ARE ACCESSED WITH SAME ROUTINES (E.G. OPNA, READA) AS AREAS. THE DIFFERENCES ARE:

1) A POINT IN IMAGE SPACE HAS MORE THAN ONE ASSOCIATED VALUE
2) VALUES ARE SIGNED 2 BYTE INTEGERS RATHER THAN UNSIGNED BYTES
3) THERE IS A LINE HEADER FOR EACH LINE

THE ORGANIZATION OF THE FILES IS SIMILAR TO THAT OF SOUNDING FILES ON THE HARRIS WITH THE EXCEPTION THAT THE LINE HEADER IS ORGANIZED DIFFERENTLY. THE DIFFERENCE IS THAT ALL THE CHANNEL SPECIFIC INFORMATION FOR A CHANNEL IS CONTIGUOUS. THERE ARE FOUR DATA ITEMS FOR EACH CHANNEL WHICH ARE, IN ORDER, CHANNEL NUMBER, NUMBER OF SPINS, DELTA F, AND Y SUB Z. EACH OF THESE CAN BE CONSIDERED A SIGNED 2 BYTE QUANTITY.

```
               USER COMMON ("UC") LAYOUT
             LATEST VERSION: 2/10/83
             LATEST VERSION: 5/06/83 (TERMINAL STATE CHANGES)
             LATEST VERSION: 12/02/83 (ADD TERMINAL NAME)
             LATEST VERSION: 1/12/84 (DATA TABLET GROUP)

          USER COMMON IS ACCESSED BY SUBROUTINES 'LUC' (LOOK AT ) AND
     'PUC' (PUT INTO)

          UC VARIABLES RESIDE IN THE SYSTEM COMMUNICATIONS REGION (SYSCOM);
     THEY ARE NOT LOST WHEN YOUR INITIATOR ABORTS, BUT ONLY WHEN MCIDAS
     ABORTS. UC VARIABLES WITH NEGATIVE SUBSCRIPTS PERTAIN TO THE INITI-
     ATOR AND RESIDE IN THE INITIATOR REGION OF SYSCOM (THIS INCLUDES
     UC(0)); UC VARIABLES WITH POSITIVE SUBSCRIPTS PERTAIN TO YOUR TERMINAL
     AND RESIDE IN THE (SEPARATE) TERMINAL REGION OF SYSCOM. (ALL THIS IS
     TRANSPARENT TO ANY PROGRAM WHICH CALLS 'LUC' OR 'PUC'.)
          PRECEDING THE UC VARIABLES IN SYSCOM IS THE GLOBAL SYSCOM AREA,
     WHICH CONTAINS VARIABLES ACCESIBLE TO ALL INITIATORS AND AT ALL
     TERMINALS. THESE VARIABLES ARE INSPECTED BY FUNCTION ND(IPOS)
     AND WRITTEN BY SUBROUTINE WD(IPOS,IVAL). ND(0) RETURNS THE FIRST
     WORD OF THIS AREA.


     LAYOUT OF THE GLOBAL SYSCOM REGION


     (INDICES ARE THOSE USED TO ACCESS SYSCOM VIA SUBPROGRAMS ND/WD)

     1-2)   SYSTEM VERSION.
     3)     YYDDD WHEN 'WAKEUP' WAS EXECUTED
     4)     HHMMSS WHEN 'WAKEUP' WAS EXECUTED
     5)     LARGEST FOREGROUND TERMINAL NUMBER DEFINED IN SYSTEM
               (1/84: CURRENT NUMBER IS 71)
     6)     MAXIMUM PERMITTED AREA NUMBER (BETWEEN 1 AND 9999)
     7-299) RESERVED



     LAYOUT OF UC


         THE FOLLOWING WORDS GIVE THE NAMES OF NAVIGATION CORELOADS WHICH
           HAVE BEEN DYNAMICALLY LOADED INTO CORE. THESE ARE REFERENCED
           BY SUBROUTINE NVLOPE, WHICH RELEASES THE CORE USED BY THESE
           CORELOADS AT THE CONCLUSION OF EACH SQX'ED MODULE.

     -94,-95)  RESERVED FOR FUTURE NAVIGATION CORELOADS
     -93)   SUFFIX OF NAVIGATION-3 CORELOAD (NAME='NV3' + SUFFIX)
               (E.G. 'GOES')
     -92)   SUFFIX OF NAVIGATION-2 CORELOAD (NAME='NV2' + SUFFIX)
     -91)   SUFFIX OF NAVIGATION-1 CORELOAD (NAME='NV1' + SUFFIX)
```

```
---DATA TABLET GROUP. THESE WORDS CONTAIN THE INFORMATION
RETURNED FROM THE TERMINAL ON THE TABLET STATE, AS WELL AS
THE TABLET BOX NUMBER, NAME AND POSITION AS DETERMINED BY
THE SCANNER.

-79)  HIGHEST Y COORDINATE OF BOX (TOP)
-78)  LOWEST Y COORDINATE OF BOX (BOTTOM)
-77)  HIGHEST X COORDINATE OF BOX (RIGHT)
-76)  LOWEST X COORDINATE OF BOX (LEFT)
-75)  BOX STRINGNAME (A4 EBCDIC)
-74)  BOX NUMBER (BOX 1 IS AT UPPER LEFT)
-73)  PEN MODE (2 OR 3=PEN DOWN, 1=PROXIMATE, 0=PEN UP)
-72)  PEN POSITION-- Y COORDINATE: (0,0) IS BOTTOM-LEFT
-71)  PEN POSITION-- X COORDINATE


MOTION VECTOR (WINDCO) UCWORDS USED (SEE ALSO UC 98,100):

-62)  JOYSTICK CONTROL VALUE SAVE
-61)  MODE CONTROL FOR JOYSTICK 2
-60)  MODE CONTROL FOR JOYSTICK 1
-58)  ENDING COLUMN NUMBER IN CURRENT MD 'WIND' FILE
-57)  ENDING ROW NUMBER IN CURRENT MD WIND FILE
-56)  STARTING ROW NUMBER IN CURRENT MD WIND FILE
-55)  CURRENT MD WIND FILE USED FOR CORE OUTPUT
-54)  CURRENT MD WIND FILE USED IMV, SELECTOR AND
      CORE INPUT
-53)  ELEMENT SIZE OF TARGET CURSOR (TV COORD)
-52)  LINE SIZE OF TARGET CURSOR (TV COORD)
-51)  SPACE BAR TOGGLE
-50)  ERROR STATUS WORD


-44)  2ND CURSOR STATE CONTROL WORD
         FORMAT:BITS 0-2  NOT USED
         BITS 3-5 ARE CURSOR COLOR (DT COMMAND)
         BITS 6-8 ARE CURSOR TYPE (DT COMMAND)
-43)  2ND CURSOR ELEMENT POSITION
-42)  2ND CURSOR LINE POSITION
-41)  2ND CURSOR ELEMENT SIZE
-40)  2ND CURSOR LINE SIZE

-34)  CURRENT GRAPHICS VIRTUAL FRAME
-33)  CURRENT DMES DEVICE:1(CRT),2(LOCAL PRTR),3(SYS PRTR),0(BLACK HOLE)
-32)  CURRENT EMES DEVICE:1(CRT),2(LOCAL PRTR),3(SYS PRTR),0(BLACK HOLE)
-31)  CURRENT LTQ DEVICE: 1(CRT),2(LOCAL PRTR),3(SYS PRTR),0(BLACK HOLE)
-30)  AUTO-CONTEXT-TABLE-SEARCH: WHEN 1, PARAMETER FETCHING SUBRS. (E.G.
         CKWP) RESORT TO SYSTEM STRING TABLE FOR MISSING KEYWORD PARAMS
-29)  (RESERVED FOR FUTURE USE BY SYSTEM SUBROUTINE SQX)
-28)  NAME (CHARS 5-8) OF PROGRAM CURRENTLY RUNNING
-27)  NAME (FIRST 4 CHARS.) OF PROGRAM CURRENTLY RUNNING
```

```
   -26)  PROGRAM CURRENTLY RUNNING IS MACRO (=1), NON-MACRO (=0)
   -25)  PROGRAM CURRENTLY RUNNING IS FOREGROUND (=1), BACKGROUND (=0)
             BACKGROUND W/ FORTRAN MAIN, CAN DO FORTRAN I/O (=-1)
             CONSOLE STARTED TASK I.E. TAPE JOB (=2)
   -24)  CURRENT SQX LEVEL (SCANNER IS 0, NEXT PGM IS 1, ETC.)
   -23)  NUMBER OF THIS INITIATOR
   -22)  KEYIN WAS STARTED BY SCHEDULER (=1); NOT BY SCHEDULER (=0)
   -18...-21)  RESERVED FOR LATER USE
   -17)  USER INITIALS UNDER WHICH THIS KEYIN RUNS (MAY BE DIFFERENT
             FROM LOGGED-ON INITIALS IN UC(2) IF KEYIN IS SCHEDULED)
   -16)  PROJECT # UNDER WHICH THIS KEYIN RUNS (MAY BE DIFFERENT
             FROM LOGGED-ON PROJECT # IN UC(1))

   -1...-13 CONTAIN A SNAPSHOT OF THE TERMINAL STATE TAKEN
         JUST BEFORE THE KEYIN BEGINS. THIS IS IN A FORMAT CLOSE TO THE
         TERMINAL PROTOCOL.

   -13)  CURSOR STATE CONTROL WORD
             FORMAT:BITS 0-2=0 MEANS CURSOR FROZEN (SET BY PC COMMAND)
             BITS 0-2=1 MEANS CURSOR POS IS JOYSTICK-CONTROLLED (P KEY)
             BITS 0-2=2 MEANS CURSOR SIZE & POS ARE JOYSTICK-CNTRLD (Z KEY)
             BITS 3-5 ARE CURSOR COLOR (DT COMMAND)
             BITS 6-8 ARE CURSOR TYPE (DT COMMAND)
             FORMAT 7/83: BITS 0-2 CONTROL JOYSTICK ASSIGNMENTS:
             0--CURSOR IS HOST-CONTROLLED
             1--JOYSTICK 1 IS POS VERNIER & J2 IS COARSE POS
             2--J1 IS CURSOR SIZE & J2 IS DISCONNECTED
             3--J1 IS CURSOR SIZE & J2 IS COARSE POSITION
             4--J1 IS VELOCITY & J2 IS DISCONNECTED
             5--J1 IS VELOCITY & J2 IS COARSE POSITION
             6--DUAL CURSOR MODE; HOST CONTROLLED
             7--DUAL CURSOR MODE; J1 IS VERNIER POSITION OF 2ND CURSOR,
                WHILE J2 IS COARSE POSITION OF BOTH CURSORS
   -12)  CURSOR ELEMENT POSITION
   -11)  CURSOR LINE POSITION
   -10)  CURSOR ELEMENT SIZE
   -9)   CURSOR LINE SIZE
   -8)   GRAPHICS STATE CONTROL WORD
             FORMAT:BIT 0=1 MEANS GRAPHICS CONNECTED TO LOOP CONTROL (J KEY)
             BIT 1=1 MEANS GRAPHICS FRAME IS LOOPING (L KEY)
             BIT 2=0 MEANS GRAPHICS FRAME IS BLANKED (W KEY)
   -7)   GRAPHICS LOWER BOUND
   -6)   GRAPHICS UPPER BOUND
   -5)   CURRENT GRAPHICS FRAME
   -4)   FRAME STATE CONTROL WORD
             FORMAT: BIT 0=1 MEANS FRAME CONNECTED TO LOOP CONTROL (Y KEY)
             BIT 1=1 MEANS FRAME IS LOOPING (L KEY)
             BIT 2=0 MEANS FRAME IS BLANKED (K KEY)
   -3)   IMAGE FRAMES LOWER BOUND
   -2)   IMAGE FRAMES UPPER BOUND
   -1)   CURRENT IMAGE FRAME
```

```
0)      USER'S TERMINAL NUMBER
1)      PROJECT # UNDER WHICH CURRENT USER IS LOGGED ON (MAY BE
            DIFFERENT FROM UC(-16))
2)      USER'S INITIALS
4)      CURRENT NAV FILE #
5)      CURRENT MD FILE #
6)      CURRENT GRID FILE #
9)      SECOND PRINTER NUMBER FOR THIS USER (SYS PRTR=0)
10)     FIRST PRINTER NUMBER (LOCAL PRINTER FOR THIS USER; SYS PRTR=0)
11)     # OF LINES ON TV SCREEN
12)     # OF ELEMENTS ON TV SCREEN
13)     # OF IMAGE FRAMES
14)     # OF GRAPHICS FRAMES
15)     TERMINAL IS REMOTE (=1), LOCAL (=0)
16)     TERMINAL IS VIDEO (=1) NONVIDEO (=0)
17)     TERMINAL HAS RECEIVED MODS OF 7/83 (=0); HAS NOT (=1)
            (THIS AFFECTS THE OPERATION OF THE F KEY PROGRAM)
18)     SVCT FLAG ON THE TERMINAL WANTING 604 OVER COMM
19)     MCIDAS DISK FILE I/O TRACE: 1=DO TRACE, ANYTHING ELSE=DO NOT
20)     SET TO 1 BY "G" KEY
21)     SET TO 1 BY "Q" KEY
22)     IMPORT POINTER FOR OTHERS...JMB
23-24)    TERMINAL IDENTIFIER (NAME OR LOCATION)-- 8 CHARACTERS
25)     NAME OF CRT COMPONENT OF TERMINAL--4 CHARACTERS
            ('TLVD'=TELEVIDEO '3278'=IBM 3278 'CRT '=OTHERS)
26)     DATA TABLET-- HORIZONTAL EXTENT (MAX X COORDINATE + 1)
27)     DATA TABLET-- VERTICAL EXTENT (MAX Y COORDINATE + 1)
28)     NON ZERO ==> TERMINAL DOWN
29      NON-ZERO ==> MAY QUIT AREAS EVEN IF NOT THE OWNING PROJECT
            (OPERATOR'S TERMINAL ONLY)
38)     0=DRAW GRAPHICS ON WRRRM   , 1=DO NOT DRAW GRAPHICS ON WRRRM
39)     0=DO NOT WRITE VIRTUAL GRAPHIC , NON ZERO=VIRTUAL FRAME TO WRITE
40)     3-D GRAPHICS MENU POINTER
41)     PLANETARY PROGRAMS NAVIGATION POINTER
42)     FLTPAK RIGHT/LEFT FRAME INVERSION FLAG FOR TWO SCREEN STEREO
43)     3-D GRAPHICS AND PLANETARY PROGRAMS STATUS FILE ID
44)     RESERVED FOR 3-D GRAPHICS AND PLANETARY PROGRAMS
45)     3-D GRAPHICS MESSAGE WAITING FLAG
46)     DEFAULT GRAPHICS LINE WIDTH
47)     GRAPHICS DASH PATTERN (1)
48)     GRAPHICS DASH PATTERN (2)
49)     GRAPHICS DASH PATTERN (3)

        WORDS 50 ... 79 ARE RESERVED FOR KEEPING THE HOST'S COPY OF THE
        TERMINAL STATE
        THIS IS IN A FORMAT EASY TO MANIPULATE AND RELATED TO THE
        FUNCTIONS OF THE TERMINAL AS AN ABSTRACT DEVICE.
        SUBROUTINE STATEU COMPOSES THIS STATE INTO A VALID TERMINAL
        ORDER (PROTOCOL RESEMBLES SNAPSHOT IN UC -1...-13)

50)     LOOP CONTROL: 1 (SYSTEM IS LOOPING) 0 (NOT LOOPING)
        LOOP CONTROL SYSTEM CONSISTS OF LB COMMAND & A,B,J,Y,O & L KEYS
```

```
          IMAGES & GRAPHICS ARE INDEPENDENTLY CONNECTED TO & DISCONNECTED
          FROM THE LOOP SYSTEM VIA UC 54 AND 59.
    51)   CURRENT IMAGE FRAME
    52)   IMAGE FRAME LOOP-- UPPER BOUND
    53)   IMAGE FRAME LOOP-- LOWER BOUND
    54)   1 (IMAGE FRAMES CONNECTED TO LOOP CONTROL) 0 (NOT CONNECTED)
    55)   1 (IMAGE FRAMES VISIBLE) 0 (IMAGE FRAMES BLANKED)
    56)   CURRENT GRAPHICS FRAME
    57)   GRAPHICS FRAME LOOP-- UPPER BOUND
    58)   GRAPHICS FRAME LOOP-- LOWER BOUND
    59)   1 (GRAPHICS FRAMES CONNECTED TO LOOP CONTROL) 0 (NOT CONNECTED)
    60)   1 (GRAPHICS FRAMES VISIBLE) 0 (BLANKED)
    61)   CURSOR SIZE (VERTICAL)
    62)   CURSOR SIZE (HORIZONTAL)
    63)   CURSOR POSITION (LINE #)
    64)   CURSOR POSITION (ELEMENT #)
    65)   CURSOR TYPE 1 (BOX) 2 (CROSSHAIR) 3 (BOX&CROSSHAIR) 4 (SOLID BOX)
                                  5 (STARS WARS)
    66)   CURSOR COLOR (0...7)
    67)   JOYSTICK 1: 0 (DISCONNECTED) 1 (CONTROLS CURSOR POSITION)
             2 (VERNIER CURSOR CONTROL) 3 (CONTROLS CURSOR SIZE)
             4 (VELOCITY CURSOR)
    68)   JOYSTICK 2 (SAME VALUES AS JOYSTICK 1)
          CURRENTLY THE ONLY LEGAL STATES FOR JOYSTICK1/JOYSTICK2 ARE:
             0/0 , 2/1 , 3/1 , 3/0 , 4/1 , 4/0   IN SINGLE CURSOR MODE
             0/0 , 2/1          IN DUAL CURSOR MODE
    69)   2ND CURSOR SIZE(VERTICAL)
    70)   2ND CURSOR SIZE(HORIZONTAL)
    71)   2ND CURSOR POS(LINE #)
    72)   2ND CURSOR POS(ELEMENT #)
    73)   2ND CURSOR TYPE (AS FOR PRIMARY CURSOR...WORD 65)
    74)   2ND CURSOR COLOR(AS FOR PRIMARY CURSOR...WORD 66)
    75)   CURSOR MODE SWITCH   0=SINGLE CURSOR MODE   1=DUAL CURSOR MODE

    76...79) (RESERVED FOR TERMINAL STATE EXPANSION)


    80)   CURRENT TOVS ORBIT/SOUNDER/RTVL FILE #
    81)   CURRENT VAS AREA/SOUNDER FILE #

          DATA TABLET RELATED SETUP OPTIONS
    91)   TABLET- TV SPACE MODE
    92)   TABLET- INACTIVE AREA (BORDER) AROUND OUTSIDE
    93)   TABLET- CURSOR FOLLOWING STATE
    94)   TABLET- LOWER RIGHT CORNER OF TV SPACE (LINE POSITION)
    95)   TABLET- LOWER RIGHT CORNER OF TV SPACE (ELEMENT POSITION)
    96)   TABLET- WHEN TO START SIGNIFICANT EVENT
    97)   TABLET- WHAT TYPE OF EVENT TO START

    98)   CURRENT MD CORE OUTPUT FILE FOR MOTION VECTORS(WINDCO)
    99)   NORTLE TOGGLE
    100)   CURRENT MD SELECTOR FILE FOR MOTION VECTORS(WINDCO)
```

```
101-120)  COPY OF RAW TEXT OF CURRENT OR LAST KEYIN ENTERED FROM KBD
121)    SINGLE LETTER KEYIN (ZERO (0) OTHERWISE)




        CALLING SEQUENCES

      I=LUC(INDEX)
C
C------PEEK/POKE USER COMMON
C
C
C------LUC "LOOKS AT" USER COMMON (PEEK)
C------INDEX IS WHERE (IN USER COM.) TO DO TRANSFER
C------FN VAL IS  WORD EXTRACTED FROM USER COMMON
C
C
      I=PUC(FROM,INDEX)
      OR: CALL PUC(FROM,INDEX)     (NO FUNCTION VALUE)
C
C------PUC POKES WORD INTO USER COMMON.
C------INDEX AS IN LUC
C------FROM IS FULLWORD DATA VALUE TO TRANSFER TO USER COM.
C------FN VAL IS (ALSO) WORD TRANSFERRED
C
C
C
```

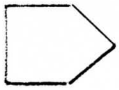
```
            VASTEXT ... CONTEXT FILE FOR VAS SOUNDING PROGRAMS

                OWNER ... C.M.HAYDEN, NOAA/NESDIS, ROOM 219
                VERSION OF 21 APRIL 1983

WORD                CONTENT

 1                  SSYYDDD (SAT-YR-DAYOFYEAR)
 2                  HHMMSS (TIME AT BEGINNING OF SOUNDER FILE)
 3                  UPPER LEFT LINE NUMBER IN SATELLITE COORDINATES
 4                  UPPER LEFT ELEM NUMBER IN SATELLITE COORDINATES
 5                  UPPER LEFT HAND Z COORDINATE
 6                  NUMBER OF LINES IN IMAGE
 7                  NUMBER OF ELEMENTS IN IMAGE
 8                  NUMBER OF BYTES IN IMAGE
 9                  LINE RESOLUTION
10                  ELEMENT RESOLUTION
11                  Z RESOLUTION
12-24               CHANNELS(BANDS) AVAILABLE, 1 = YES
25                  LLNW (LAT/LON AT NORTHWEST CORNER OF IMAGE)
26                  LLSE (LAT/LON AT SOUTHEAST CORNER OF IMAGE)
27                  DDHHMM OF SURFACE-DATA USED
28                  STATUS WORD
29                  SURFACE GRID FILE NUMBER
30-33               GRID NUMBERS FOR SURFACE ANALYSES (Z,T,TD,P)
34                  GUESS PROFILE IDENTIFICATION
35                  SOUNDER FILE NUMBER
36                  SURFACE EDIT FILE MD NUMBER
37                  SURFACE EDIT FILE ROW NUMBER
38                  GUESS FILE MD NUMBER
39                  GUESS FILE MD ROW NUMBER
40                  RETRIEVAL FILE MD NUMBER
41                  RETRIEVAL FILE MD ROW NUMBER
42                  GUESS GRID FILE NUMBER
45                  VISIBLE AREA NO. (CORRESPONDS TO SOUNDER FILE)

50                  RETRIEVAL TYPE
51                  FIRST GUESS
52                  RETRIEVAL SPACING
53                  FOV BOX SIZE
54                  MOSFC OPTION
55                  LAST LINE
56                  LAST ELEMENT
57                  BEGIN LINE
58                  BEGIN ELEMENT
59                  TERMINAL NUMBER
60                  PLOT OPTION
61-72               BIAS VALUES FOR CHAN RADIANCES

100                 NUMBER OF RETRIEVALS
```

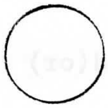
APPENDIX II

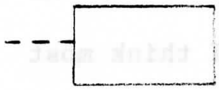
Level I Flowchart Constructs

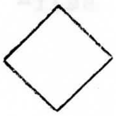
The following brief appendix contains examples and(or) descriptions of all the major FORTRAN constructs used in the VAS program Level I Flowcharts discussed in this manual. I think most people with FORTRAN experience will find this appendix self-explanatory.

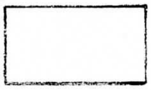
LEVEL 1 FLOWCHART CONSTRUCTS

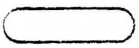
1.  : Offpage connector or beginning and end of DO-loops.

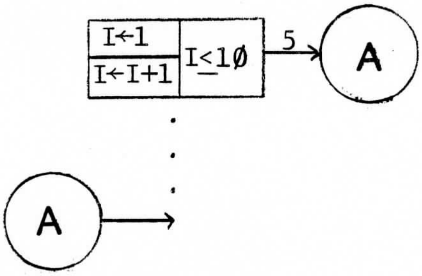
2.  : O-page connector or beginning and end of DO-loops.

3.  : Comment statement.

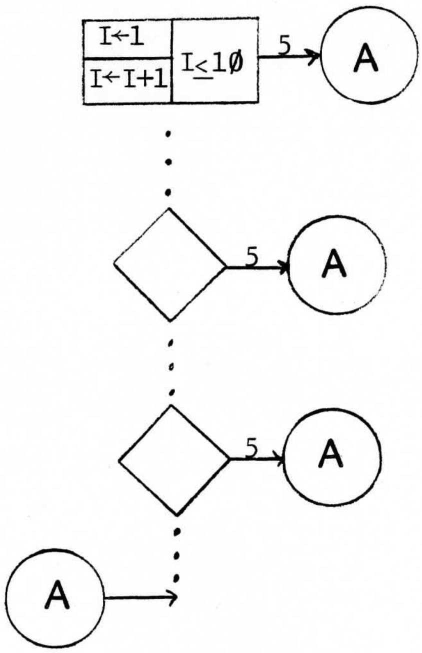
4.  : IF conditional.

5.  : CALL statement or arithmetic statement.

6.  : END statement.

7. 

: DO-loop construct; loops back to beginning when A is reached in flow. If A is used, it means the loop extends past the end of the page. Note that loops do not always extend vertically; they can extend horizontally as well. **Note: If a DO-loop extends beyond the end of the page, and there are jumps to the end of the DO-loop on the subsequent page where the loop terminates, then these jumps will be denoted by circles rather than the semi-box notation.**

8. 

: When leaving DO-loop A, control passes beyond last A. This also holds for DO-loops with off-page connectors, in which case control would pass beyond the last A.

9. Note that F (false) paths are NOT ALWAYS at the bottom of IF conditionals. (

). Occasionally, the false path is to either the right or left of the conditional.

10. Numbers outside IF constructs, DO-loops, etc., indicate transfer of control to that particular statement of the program.

APPENDIX III

Subroutine List and Descriptions

The following appendix contains, in alphabetical order, information concerning the vast majority of the subroutines called by VAS retrieval programs. Included are all the Level I Subroutines, as well as further subservient subroutines. In particular, each subroutine included in the Chapter 2 Modular Flowcharts is listed and described here.

As part of each subroutine's description, the prefix or program/subroutine location within the McIDAS source library of each routine is listed. If the reader is referred to a subroutine/program location, it means the subroutine in question is located in its entirety somewhere within the designated subroutine/program library member. For example, the code for subroutine VTRET exists after the code for program SRET in library member VLSRET.

In addition, a brief discussion of what each subroutine does is also presented, along with descriptions of input arguments and(or) output parameters and lists thereof whenever possible. Finally, the reader may occasionally be referred to the source code itself for further information.

COMPLETE ALPHABETICAL SUBROUTINE LIST

II	(MR)
ABORT	(SR)
ANGGET	(VR)
ANGLES	(MR)
ANGSS	(in VLBNVA)
ARASIZ	(MR)
ATOE	(MR)
BARB	(VR)
BLKA	(SR)
BOX	(in MRPLTPBK)
CALDAY	(VR)
CLEANA	(SR)
CLEANW	(SR)
CLMGES	(TR)
CLOSA	(in MROPNA)
CLOSAO	(in MRWRITA)
CO2TAV	(VR)
CONTAV	(VR)
CONTNT	(VR)
CSRAOB	(MR)
CSRAOI	(in MRCSRAOB)
CSRAOM	(in MRCSRAOB)
CSRAOP	(in MRCSRAOB)
CSRAOS	(in MRCSRAOB)
CSRAOZ	(in MRCSRAOB)
DCLOSE	(in VRDAIO)
DDEST	(MR)
DIRADJ	(VR)
DOPEN	(in VRDAIO)
DREAD	(in VRDAIO)
DSHOFF	(in MRPLTPBK)
DSHON	(in MRPLTPBK)
DWRITE	(in VRDAIO)
EDEST	(MR)
EMES	(MR)
ENAREA	(MR)
ENCODE	(VR)
ENCODX	(TR)
ENDPLT	(in MRPLTPBK)
ENKODE	(in VRENCODE)
ENPT	(in MRPLTPBK)
EPOCH	(in MRGETNAV)
EXTEMP	(TR)
FBARN	(MR)
FILCLD	(in VLSRAD)
FILCLR	(in VLSRAD)
FILL	(TR)
FILTER	(in VLVTPZ)
FLOC	(in VLXRVA)
GAMTAV	(VR)
GESPRO	(TR)
GETDAY	(MR)
GETFRM	(MR)
GETGAM	(MR)
GETNAV	(MR)
GETSFV	(VR)
GETTIM	(in MRGETDAY)
GRADWI	(in VLGWVA)
H2OTAV	(VR)
HEAPFY	(in MRFBARN)
HRTOPO	(TR)
HTV	(VR)
HTX	(in VLSRET)
IGNAME	(in MRIGMAKE)
INITPL	(in MRPLTPBK)
INTER	(in MRFBARN)
INTPTW	(TR)
IQ	(VR)
ISQX	(MR)
ITOC	(SR)
JMBWTF	(foreground version in MRJMBFUN)
JSQX	(VR)
LINFIL	(in TRFILL)
LOCK	(SR)
LTQ	(MR)
LWCLOS	(in MRLWGETX)
LWGET	(in MRLWGETX)
LWMOP	(in MRLWSUBS)
LWNEWF	(in MRLWSUBS)
LWOPEN	(in MRLWGETX)
LWPO	(in MRLWSUBS)
LWSO	(one version in MRS2LWSI)
MDCLOS	(in MRMDMAKE)
MDNAME	(in MRMDMAKE)
MNRAOB	(MR)
MOVB	(SR)
MOVC	(SR)
MOVCW	(SR)
MOVW	(SR)
MOVWC	(in SRMOVCW)
NSTAR	(in VLSRAD)
NVINIT	(VR)
O3TAV	(VR)
OPNA	(MR)
OUTINT	(MR)
PACK	(in SRCRACK)
PAGE	(in MRPLTPBK)
PENADD	(in MRPLTPBK)
PENBEG	(in MRPLTPBK)
PENMOV	(in MRPLTPBK)
PLNKIV	(VR)
PLOT	(in MRPLTPBK)
PLTDIG	(MR)

POST	(SR)
PRCL	(in SRPRRD)
PRCLOS	(in SRPROPEN)
PREATV	(VR)
PRECW	(TR)
PRETAV	(VR)
PRLINX	(SR)
PROFIX	(TR)
PROPEN	(SR)
PRPRPR	(in SRPROPEN)
PRRD	(SR)
PRWR	(in SRPRRD)
PUC	(in MRLUC)
PUTCHR	(VR)
QGDASH	(in MRPLTPBK)
QVTWR	(in VLVTPZ)
RAOBIN	(VR)
RBYTSX	(MR)
RDTRK	(SR)
READD	(MR)
READDL	(in MRREADD)
READOF	(MR)
RETIO	(TR)
RORDER	(TR)
SATEAR	(MR)
SATPOS	(MR)
SATTV	(MR)
SDEST	(MR)
SENOUT	(foreground version in MRJMBFUN)
SGRAOB	(MR)
SNDANL	(MR)
SOLARP	(MR)
SOLVEX	(NM)
SQSLED	(MR)
SRCH	(TR)
SRGSS	(in VLSRVA)
STC	(in SRIC)
STDATM	(TR)
SURGES	(TR)
SYMVRT	(AR)
TEKPUT	(in MRPLTPBK)
TOKANL	(MR)
TQ	(foreground version in MRJMBFUN)
TQMES	(MR)
TQSET	(in MRLTQ)
TRMNL	(MR)
TSNIO	(TR)
TVSAT	(MR)
ULMR	(TR)
UNLOCK	(in SRLOCK)
UNLOCK	(in SRLOCK)
VALUE	(in VLSRVA)
VASDAT	(VR)
VASDIG	(VR)
VASGES	(VR)
VASNAV	(in MRGETGAM)
VASRTE	(VR)
VASTAU	(VR)
VASTPW	(VR)
VRTIO	(VR)
VRTOPO	(in VRVASDAT)
VTQ	(VR)
VTRET	(in VLSRET)
VWRET	(VR)
WALK	(in MRPLTPBK)
WD	(foreground version in MRJMBFUN)
WMIX	(VR)
WRBOX	(in VLXRVA)
WRITA	(MR)
WRITDU	(in MRWRITD)
WRMAR	(TR)
WRTRK	(in SRRDTRK)
ZECONV	(VR)
ZEROS	(SR)
ZWIND	(in VLGWVA)

SUBROUTINE DESCRIPTION LIST

1) II (SIZE, VAL, STR, POS): Convert integer to string (fixed length field); 'SIZE' is length of field in bytes, 'VAL' is the integer to convert, 'STR' is the output array, 'POS' is the offset (0-based) at which to start the output. If the integer (plus sign, if necessary) doesn't fit into 'STR', an asterisk is placed in the high order position.

2) ABORT: Aborts (quits) a program or subroutine (plots 'INIT RESTARTED' on screen).

3) ANGGET: Gets angles (such as local zenith angle (angle between perpendicular to earth at observation point and satellite) and solar zenith angle (same as local zenith angle except sun instead of satellite).

4) ANGLES (JDAY, JTIME, XLAT, XLON, GHA, DEC, SATANG, SUNANG, RELANG): Computes zenith angles to sun and satellite and relative azimuth angle. Input parameters: 'JDAY' = picture day, 'JTIME' = picture start time, 'XLAT' = latitude of point, 'XLON' = longitude of point, 'GHA' = Greenwich hour angle of sun, 'DEC' = declination of sun. Output parameters: 'SATANG' = zenith angle of satellite, 'SUNANG' = zenith angle of sun, 'RELANG' = relative angle.

5) ANGSS (IFLD, ITAG, NCOLS, NROWS, TLAT, WLON, DING, LEV, ICHR, MDNG, NSFC): Functions similarly to SRGSS, except for upper air data rather than surface data, gets guess information at gridpoint locations in gridfile for upper air analysis for whatever parameter is being analyzed (gets data from guess MD file after guess grids have been reformatted into it via program GSVA).

6) ARASIZ (AREA, LINSIZ, ELESIZ): Creates a file for an image 'LINSIZ' x 'ELESIZ.' Input parameters: 'AREA' = area number, 'LINSIZ' = number of lines requested, 'ELESIZ' = number of elements requested.

7) ATOE (NBYTES, ARRAY, LEFPOS): Convert characters from ASCII to EBCDIC. 'LEFPOS' is 0-based offset within 'ARRAY', 'NBYTES' characters are converted in place. This is one of the character conversion routines.

8) BARB (DIREC, SPEED, XPG, YPG, KOL, SIZE): Simply plots wind barbs; 'DIREC' is direction relative to north (or top of TV screen) from which wind blows, 'SPEED' is in any arbitrary units, 'SIZE' indicates the size of the barb to be determined from experience. Actually, 'SIZE' is the desired length of a 10-knot barb. A negative 'SIZE' denotes southern hemisphere, to place wind barbs on opposite side of arrow.

9) BLKA (N, ARRAY): Set 'N' words of 'ARRAY' to EBCDIC blanks (blanks N words of an array).

10) BOX (DROW, DCOL): Is internal subroutine, does wide line graphics, sets up box.

11) CALDAY (YYDDD, KYEAR, NMON, KDAY, AMON): Converts 'YYDDD' to normal calendar day (year, month, day), month is in three-letter form.

12) CLEANA (NBYTES, IARRAY): Change all unprintable characters to blank; 'IARRAY' is array, 'NBYTES' long (clean an array by transforming bad or invalid characters to blanks.

13) CLEANW (NWORDS, IARRAY): Change unprintable characters to blanks; 'IARRAY' is array, 'NWORDS' words ('NWORDS'*4 bytes) long, 'NWORDS' must be less than or equal to 64. Cleans unprintable characters out of 'NWORDS' words, is like subroutine CLEANA, except number of words (not bytes) to be cleaned is specified.

14) CLMGES: Gives the climatological guess for satellite temperature retrievals, output consists of 40 levels (1000-.1 mb) of temperature data, 20 levels (1000-115 mb) of w (mixing ratio) data at 15°, 30°, 45°, 60° and 75°N, from 30°-75°N, have January and July data sets of T and w, 15°N is same data for both January and July, 1000 mb height is accessed within the subroutine proper and outputted through common block SURF, also returns 15-level temperature profile and 6-level dewpoint profile through common block GUESS.

15) CLOSA (IAREA): Closes input area, removes area from lists.

16) CLOSAO (IAREA): Closes an output area (digital segment on disk). One might think of an area as "external memory" (unlike core, which is "internal memory"), also writes onto disc any data in buffer.

17) CO2TAV (PATH): Calculates transmissivities for CO2 channels (is of course channel dependent), output consists of transmittance through each of 40 pressure levels for a given channel.

18) CONTAV (WMIX, PATH): Computes transmittances for water vapor/trace gas continuum.

19) CONTNT (ADDR, LEN, RETURN): Accesses a word or array from memory when its byte length and absolute memory address are known.

20) CSRAOB (IOPT, MD1, MD2, IDAY, ITIME, IDNO, NLEVS, ISTDAT, LEV, PRES, TEMP, TD, DIR, SPD, Z, ISTAT): Input raob data from file(s), construct complete sounding for given day/time/station from mandatory and significant levels (get mandatory + significant levels from raob-type files), levels are returned in descending order by pressure, missing data is marked -99999.

Inputs: 'IOPT' describes desired options. It is Ø or any sum of:

1: Recompute heights for all levels with p/T/TD data; if this option is not taken, no interpolation of any kind is performed--the raw data only is returned to the caller.

2: Interpolate pressures for the significant level winds.

4: Interpolate missing temperatures and dewpoints.

'MD1' is MD file number of mandatory data file.
'MD2' is MD file number of significant data file. If less than or equal

to Ø, no significant levels are included. 'IDAY' is day of desired data (YYDDD), 'ITIME' is time of desired data (HHØØØØ), e.g., Ø or 12ØØØØ, 'IDNO' is desired station number.

Outputs: 'NLEVS' is total number of levels returned (maximum of 1ØØ), 'ISTDAT' is four word array to contain latitude, longitude, elevation and state code of desired station. (Following arrays should be dimensioned 1ØØ. In the real arrays, missing data values are returned as -99999.) 'LEV' is integer array containing the level (for mandatories, "SFC," the integer pressure (mb), "TRO1," "TRO2" or "MAXW," for significants, either "SIGT" or "SIGW"), 'PRES' is real array containing pressure (mb), 'TEMP is real array containing temperature (deg K), 'TD' is real array containing dewpoint (deg K), 'DIR' is real array containing wind direction, 'SPD' is real array containing wind speed, 'Z' is real array containing height of level, 'ISTAT' is status code (Ø=OK, but some data may be missing, -1=data missing, no sounding returned, -2=no 'MD1' file).

Note: 'CSRAOB' employs several auxilliary subroutines whose names begin with 'CSRAO-.' Avoid naming other subroutines 'CSRAO-.'

21) CSRAOI (NLEV, PRES, TEMP, TD, DIR, SPD, Z): Routine to interpolate missing values of temperature, dewpoint, wind direction, wind speed. Selected by IOPT in CSRAOB.

22) CSRAOM (LEV, PRES, TEMP, TD, DIR, SPD, Z, NLEVS, JPR): Merge significant levels with mandatories and drop extraneous levels (prefer mandatory to significant data when both present for a given level). 'NLEVS' is number of levels in the various arrays. It is revised (downward) on output. 'JPR' is pointer array giving on input the observations by descending pressure and on output is revised to drop unneeded levels.

23) CSRAOP (NLEVS, JPR, Z, SCALHT, PRES): interpolate pressures for intermediate levels with heights.

24) CSRAOR (NLEVS, JPT, LEV, PRES, TEMP, TD, DIR, SPD, Z, SCALHT): Routine which reorders data arrays and deletes any levels with either missing pressure, temperature or height.

25) CSRAOS (NLEVS, X, JPR): Sort observations on descending 'X,' result is permutation of pointer array 'JPR,' which should be initialized with pointers before entry. Sort into descending-'X' order using pointer array 'JPR.' This is an internal subroutine of CSRAOB.

26) CSRAOZ (NLEVS, JPR, PRES, TEMP, TD, Z, SCALHT): Recompute heights ('Z' array) using p, T, TD. 'NLEVS' is total number of levels. 'JPR' is a pointer (subscript) array which orders the levels by descending pressure. 'PRES' is the array of pressures. 'TEMP' is the array of temperature. 'TD' is the array of dewpoints. Input/output: the first valid height is selected as a baseline and all subsequent heights are re-computed (whenever p/T/TD are all present) for hydrostatic consistency. Output: 'SCALHT' is an array of scale heights for the layer just below the current level (will be used later in interpolating pressures).

27) DCLOSE (LUN): Closes an LW (large word array) file.

28) DDEST (CTEXT, IVAL): Display message + integer/A4 on CRT (debug message destination). Output debug message on CRT/printer.

29) DIRADJ (FLAT, FLONG, SATLON, DIREC, ADJDIR): Makes slight adjustment to any wind to adjust it for parallax when plotting on a grid superimposed over a hemispheric image; in other words, winds will be plotted from their true directions anywhere on the grid.

30) DOPEN (DFILE, LUN, LREC, *): Opens LW (large word array) file. When file is opened, computer is told how many words are to be read; from then on any reads or writes (subroutines DREAD or DWRITE) will access or write out that number of words onto or from THAT FILE (does not apply to other DOPENs and files). Different LW files can have different amounts of words read by different settings in DOPEN.

31) DREAD (LUN, NREC, JBUF, *): Reads from an LW (large word array) file.

32) DSHOFF: Dash off (--).

33) DSHON: Dash on (--).

34) DWRITE (LUN, NREC, JBUF, *): Writes onto an LW (large word array) file.

35) EDEST (CTEXT, IVAL): Display message + integer/A4 on CRT (error message class).

36) EMES (TEXT, VAL): Output error message + integer/A4; 'TEXT' is up to 20 word array of message ending with "$" or "$$." If "$," 'VAL' is treated as integer, if "$$," 'VAL' is treated as integer or A4.

37) ENAREA (AREA, SS, DAY, TIME, LINE, ELEM, INCL, INCE, MEMO): Builds and writes area directory, writes area parameters into area directory. Input parameters: 'AREA' = area number, 'SS' = satellite identification number, 'DAY' = day in YYDDD format, 'TIME' = time in HHMMSS format, 'LINE' = satellite line coordinate of upper lefthand corner, 'ELEM' = satellite element coordinate of upper lefthand corner, 'INCL' = line resolution, 'INCE' = element resolution, 'MEMO' = 8-word integer array to hold a memo on area.

38) ENCODE (FMT, OUTBUF, ARG1, ARG2, ARG3, . . .): The encode package for the IBM 4341. It may have as many as 21 or as few as 0 arguments. ENCODE is designed for use with FORTRAN-66, while subroutine ENKODE is to be used with FORTRAN-77. Both ENCODE and ENKODE work in a fashion similar to PRINT statements, with the arguments being the data printed. See DNENCODE for further details.

39) ENCODX (NARGS, LFMT, LBUF, LIST): Special routine which is called only by the ALC routines "ENCODE" and "ENKODE", does the actual encoding of the variables.

40) ENDPLT: Subroutine called at end of a plot, is an interface subroutine to force binding off of buffer. Makes sure all of buffer is plotted on video screen (only applies to video screen), no matter how little information is present.

41) ENKODE (FMT, OUTBUF, ARG1, ARG2, ARG3, . . .): Used in conjunction with FORTRAN-77, encodes up to 21 variables and places then in suitable buffer which can then be displayed by a specific display instruction such as LTQ (buffer can also be displayed by suitable construction of the ENKODE statement itself). Variables are encoded into character strings. It is like a formatted "WRITE" statement. The variables can be displayed on screen (CRT) or printer. Arguments: 'FMT' - either an array name or a literal string containing a FORTRAN-type format, describing the manner in which the variables are to be encoded. If it is an array name, it must be type CHARACTER using ENKODE, 'OUTBUF' - an array into which the variables are to be encoded (the array must be sufficiently long to contain the encoded variables, and it must not be type CHARACTER), 'ARGN' the internal variables to be encoded, 21 or fewer, possibly none. See DNENCODE for further details.

42) ENPT: An interface subroutine to force binding off of buffer. Makes sure all of buffer is plotted on video screen (only applies to video screen) no matter how little information is present (is called by ENDPLT).

43) EPOCH (IETIMY, IETIMH, SEMIMA, OECCEN, XMEANA): Finds time of perigee from Keplerian epoch.

44) EXTEMP (TT,LAT): Extrapolates temperature from top of NMC guess (either 50 or 10 mb, depending on guess being used) to 0.1 mb using regression equations based on NMC climatology, EXTEMP used only with NMC guess (either LFM or hemispheric) because climatology guess has 40 levels in it already.

45) FBARN (XLATN, NR, NC, XGA, XGD, XGE, XDA, XROW, XCOL, NSTA, XINC, IFREQ, IGUESS): A fast approximation to Barnes interpolation. This routine implements a fast algorithm for a Barnes interpolation of randomly-located observations into a uniform latitude-longitude grid of values. This is done for one set of observations. The computing time of this algorithm is proportional to NR*NC+NSTA*LOG(NSTA), compared to NR*NC*NSTA for INTPO parameters. 'XLATN' is input north latitude of grid top in degrees, 'NR' is input number of rows in grid, 'NC' is input number of columns in grid, 'XGA' is output array (NR,NC) interpolated from 'XDA,' 'XGD' is work array (NR, NC) for weight sum denominator for first pass, 'XGE' is work array (NR, NC) for weight sum denominator for second pass, 'XDA' is input array (NSTA) containing first observed parameter, 'XROW' is input array (NSTA) containing latitudes of observations, 'XCOL' is input array (NSTA) containing longitudes of observations, 'XROW' and 'XCOL' are measured as grid row, column. Thus, the geographical point at the upper left corner of the grid has 'XROW'=1 and 'XCOL'=1. 'NSTA' is input number of observations (stations), 'XINC' is input latitude between rows in degrees, 'IFREQ' is input lowpass filter bandwidth parameter, 'IGUESS' is non-zero if guess grid input through 'XGA' is proportional to spatial frequency. Good values for 'IFREQ' depend on station density per grid square. Too low a value may cause a floating point abort. Values from 30-60 are common on McIDAS. These parameters are identical to the parameters for the INTPO routine in the McIDAS SVCLIB library, except for the addition of the work arrays 'XGD' and 'XGE.' In particular, if 'IFREQ' is the same in this routine and in the routine INTPO, then they should give the same interpolations, up to the approximation. Note that

the values in 'XDA, 'XROW' and 'XCOL' are changed by this routine, final value is Ø (OK), -1 (bad parameters).

46) FILCLD (NSAM, NRAS, RADS, AVG, STD, BDAT, RMAX, TCLRMN, FAC, NS): Called by subroutine NSTAR, sets up file of clear radiances, but in somewhat different way than FILCLR, is registered (TBB data for given band for given FOV accepted only if all bands (1-10 and 12) have reasonable TBB data; hence, average TBB for any band is always calculated from same FOVs).

47) FILCLR (NSAM, NRAS, RADS, AVG, STD, BOAT, KM, NS): Also called by NSTAR, sets up file of clear radiances, is not registered (TBB data for given band for given FOV accepted even if some other bands for that FOV have unreasonable data; hence, average TBB for different bands may be computed from different FOVs). Both FILCLD and FILCLR filter out noise as well.

48) FILL (IGRID, ITAG, IBUF, IFG, NROW, NCOL): Fills missing elements of grid with appropriate values by linear extrapolation.

49) FILTER (VDAT, NSPIN, ERM, MSAM, TSFS, LDETR): Filters brightness temperatures for "one-step" physical matrix inversion retrieval (done in VLVTPZ). Gets retrieval box-averaged brightness temperatures for each band, average surface elevation for retrieval box, expected error of brightness temperature observations, etc.

50) FLOC: Gets latitude and longitude from line and element (for either TOVS or VAS).

51) GAMTAV (TAU, KCHAN): Adjusts transmittances for up to 40 levels of a given sounding by raising τ at each level to an empirical correction factor "G" (for example, $\tau(I) = \tau(I)^G$).

52) GESPRO (ICLIM, NOSFC, MDNO): Get 40 level profile from VASGSS or climatology, add surface. Gets guess profile of temperature and mixing ratio for all 40 retrieval levels (uses guess MD file and statistics).

53) GETDAY: Get system date, in terms of integer values, from computer's own clock (format of YYDDD).

54) GETFRM (FRAME, ENTRY): Reads entry from frame directory "FRAMED," takes input parameter 'FRAME' (frame number) and returns output parameter 'ENTRY' (64-word array containing directory 'ENTRY'). The directory contains such things as frame number, magnification, etc.

55) GETGAM (DAY, TIME, GAMMA, GAMDOT): Gets 'GAMMA,' 'GAMDOT' for 'DAY,' 'TIME' from NAV file. Searches NAV files in order: current file, file 2, file 1. 'GAMMA,' 'GAMDOT' give E-W shift misalignment adjustments--one adjustment per picture.

56) GETNAV (IDAY, IEXIST): Fills GAMCOM, BETCOM with values from NAV (nagivation) file, gets navigation information (parameters) from navigation files and puts it in navigation common block. Gets navigation (latitude and longitude) for VISSR images (imaging mode). It is satellite dependent, not wavelength dependent.

57) GETSFV: Gets surface data from gridfile for VAS or TOVS retrievals (1000 mb height, sea level temperature and dewpoint depression).

58) GETTIM: Get system time, returns integer value in HHMMSS format (current time of day) from computers' clock.

59) GRADWI (JHITE, II, JJ, FCORIO, U, V, JHIT2, DELT, JHIT3): Get gradient wind from analyzed field in height, wind direction returned is clockwise from grid north.

60) H2OTAV (PATH): Calculates transmittances for H2O channels.

61) HEAPFY (XDA, XROW, XCOL, LL, K): Sorting routine. Sorts data according to latitude and longitude.

62) HRTOPO (MLAT, MLON, IELEV, ICHAR): Takes input latitude and longitude and returns output as shown below. Gives High Resolution Topography (global-10 minute resolution). Input: latitude (0-89, +N, -S, *100), longitude (0-180, +E, -W, *100). Output: elevation (meters), surface characteristic (0=ocean, 1=land).

63) HTV (S, LSTA): Calculate heights from 40 level T, W profiles, beginning at station pressure/elevation. File lower mandatory levels by direct linear (ln(p)) interpolation between surface and IZ1Ø.

64) HTX (S, LSTA): Calculates heights from 40-level temperature, moisture profiles beginning at station pressure/elevation. Fills lower mandatory levels by direct linear (ln (p)) interpolation between surface and IZ10 (1000 mb height).

65) IECONV (ITEM, JRMP, LBUF): For use with ENCODE, called by ENCODX, is part of encoding package.

66) IGNAME (GFNO, FILNAM): Construct gridfile name from number. Name is of form "GRIDNNNN."

67) INITPL (IGD, NWD): Entry in PLTPAK to initialize plot routines. 'IGD' = output frame number, or negative of area to use, 'NWD' = graphics width. Starts plot package.

68) INTER (XDA, XROW, XCOL, LL, K): Is an entry point within subroutine HEAPFY.

69) INTPTW (PI, TI, WI, PF, TF, WF, NI, NF): Interpolates temperature and mixing ratio in pressure.

70) IQ (MSG): Allows user to conform to FORTRAN-77 keyin standards (MAINØ) without having to make significant changes in his/her existing main routine. Brings the values contained within positional parameters into the main program via an array (the argument of the subroutine call).

71) ISQX (CPGM, NVAL, CTOK): Load and start (sequential) execution of McIDAS load module. Control does not return to caller until execution complete. 'CPGM' is load module name (blanks on the right if necessary). In calling program, 'CPGM' is character *8 or more, 'NVAL' is the number of tokens in 'CTOK,' 'CTOK' is an array of character *12 tokens (the first should be the program name and the remaining are parameters). Unless

1-letter keyin (or "CNTRL"), ISQX does load of link. Load/link is mode via "linker." Transfer vector is passed along with 'NTOK,' 'CTOK.' Final value is 0 (OK; execution completed), or -1 (couldn't do it).

72) ITOC (SIZE, VAL, STR, POS): Converts an integer to a character string (variable length of output field). 'SIZE' is maximum size of output field, in bytes, 'VAL' is integer to be converted (+ or - OK), 'STR' is distination string (A4), 'POS' is offset (0-based) within string where to begin output. NOTE: the output takes only as many characters as are required for all the digits in the string, plus the - if negative. Zero is output as a single "0." If the number doesn't fit, the high-order character is set to "*." In summation, ITOC converts binary to EBCDIC-variable width.

73) JMBWTF: Called within subroutine LWSO, does the actual writing of the block to disk.

74) JSQX (CNAME, CARG1, CARG2, CARG3, . . .): Allows a program to perform a keyin or keyins, rather than the operator or user having to do them. For instance, a graphic could be plotted by using JSQX rather than manually punching in keyins. The arguments to the function/subroutine are the keyins themselves. In other words, JSQX allows one foreground program to invoke another. The routine is called by either a FORTRAN-66 or 77 program containing the arguments to be passed to Barrett's routine 'ISQX.' As many as 20 CARG*'s may be passed to JSQX. It is unimportant that the arguments be typed character or otherwise in the calling routine, because JSQX can determine which they are. The argument 'CNAME' must contain at least eight characters expressing the keyin name to be invoked. The remaining arguments contain 12 EBCDIC characters, expressing the keyed parameters. If the calling program is written in FORTRAN-77, the arguments need not be 8 or 12 bytes in length. For example,

```
call JSQX ('CNAME,' 'ARG1,' 'ARG2')
```

is permissible. But in FORTRAN-66, we would need

```
call JSQX ('CNAME___,' 'ARG1________,' 'ARG2________').
```

The functional value returned by the routine is zero (JSQX was successful) or non-zero (JSQX was not successful). The routine may also be called as a subroutine. For example,

```
call JSQX (CNAME, CARG1, CARG2, CARG3, . . .)
```

though in this case there is no indication of success or failure.

75) LINFIL (LINE, N, IPASS): Do linear interpolation on values along a single line.

76) LOCK: Locks a resource with an arbitrary 8-character name, makes sure no other users can modify particular file being worked on until the work is done; file is henceforth unlocked by using subroutine UNLOCK.

77) LTQ (LINE): Display a line of 80 characters; 'LINE' is array containing message to display and 'LINE' is displayed on device 0 (black hole), 1 (CRT), 2 (printer) or 3 (system printer) according to setting made by calling subroutine TQSET. If you never call TQSET, output goes to 1 (CRT). Function value is total number of calls made to LTQ so far.

78) LWCLOS: Close file, remove it from open tables. No harm is done if the file is not already open. Call LWCLOS (0) causes all files open for this initiator to be closed. It is not strictly necessary to call LWCLOS (except when writing with locks) as LWCLOS (0) is called when your program terminates or aborts.

79) LWGET (IFILE, IWORD, NWORDS, IAREA): Service routine for LW (large word array) level disk input/output. Reads 'NWORDS' words from an open McIDAS file, beginning at virtual word 'IWORD,' into 'IAREA.' 'IFILE' is the file name (8 bytes). If value returned (I=LWGET(.....)) = 0(OK), if -1 (request goes beyond the possible file extents). Reading from pages that were never written is OK, the data in the missing pages is returned as HEX 80808080. If the file is not already open for this initiator, it is opened (with an LWOPEN 'IACCES' = 1). Technical note: the read request is fulfilled through a number of buffers (8 is standard) which are core-resident in the scanner. As a result, consecutive requests for data in a given page, or in a page recently accessed, may not result in actual I-O, but rather in a core-to-core transfer.

80) LWMOP (CFNAME): Flush LW buffers. Buffers with write-back flags (IBFMOD value > 0) are dumped. Then all buffers which match 'CFNAME' are set to "EMPTY." All other buffers remain full of data. Exception--if 'CFNAME' = ' ', all buffers are set to empty. This routine is called automatically at end of each McIDAS program, so it doesn't have to be called explicitly in the program code. This routine replaces old subroutine LWCLOS (which still works, however).

81) LWNEWF (CFNAME, IPAGE, ISECTR): Make entry into filename tables for new extent. 'CFNAME' is file name, 'IPAGE' is file extent (page number of file's page table) and must be negative, 'ISECTR' is sector location of file's page table, which must already be in proper format and nonempty.

82) LWOPEN (IFILE, IACCES): Causes a file to be opened. Opens an existing McIDAS file for read access ('IACCES'=1) or read/write access ('IACCES'=2). It is permissible to write on a file when it has been opened with 'IACCES'=1, but it is not safe (loss of data integrity) unless your program is the only program in the system that is allowed to write on the file. If value (I=LWOPEN(.....)) returned = 0 (OK), if -1 (no such file). If the open table for your initiator is full (current capacity: 16 files), LWOPEN aborts: U030. Finally, the file opening is recorded in SYSCOM so that the file may be closed automatically when the program terminates.

83) LWPO (ICB): Write a file page out to disk. 'ICB' is the number of a buffer containing the file page.

84) LWSO (ISECTR, IBUF): Writes block to disk (page), called by LWI and LWO, is part of McIDAS operating system.

85) MDCLOS (MDNO): Close MD file(s). 'MDNO' is MD number of file to be closed, or 0 to close all.

86) MDNAME (MDNA, FILNAM): Make name of MD file.

87) MNRAOB (MD, IDAY, ITIME, IDNO, ISTDAT, LEVDAT, ISTAT): Gets mandatory level raob data. 'MD' is MD file number, 'IDAY' (YYDDD) and 'ITIME' (HH0000) define the day/time, 'IDNO' is the station number, 'ISTDAT' is returned as a 4-word array: LAT (degrees *10000), LON (degrees *10000), Z0(surface elevation in meters), STATE ID (A4 format). If any of these are not in the schema, Z80808080 is returned. The level data is returned in array 'LEVDAT' (dimensioned 7*30); that is, maximum of 16 levels returned. Actual number will be in 'ISTAT'. Level (A4 format) is same as p or "SFC" or "TRO." P is in mb*10, T (K*100), TD (K*100), DIR (wind direction in degrees), SPD (wind speed in meters per second), and Z (in meters). 'ISTAT' is number of levels (OK), -1 (data not present), or -2 (invalid MD file). MNRAOB will extract data from any MD file whose schema has the following properties: (1) keys day,' 'time' are present and in the row header; (2) key 'ID' identifying the station is present in the column header or in the data portion, (3) the data portion contains from 1 to 16 repeat groups, each with 7 fields in the order: LEV, P, T, TD, DIR, SPD, Z. Note that the schema need not contain column headers.

88) MOVB (NBYTES, FROM, TO, TOPOS): Moves N bytes from word boundary to anywhere. 'TOPOS' is offset within 'TO' array (0-based), if 'NBYTES' is less than or equal to 0, nothing happens.

89) MOVC (NBYTES, FROM, FRMPOS, TO, TOPOS): Move N characters with source/destination offsets. 'FRMPOS' is offset within 'FROM' array (0-based), 'TOPOS' is offset within 'TO' array (0-based). If 'NBYTES'/NWORDS is less than or equal to 0, nothing happens. MOVC essentially moves N bytes from anywhere to anywhere.

90) MOVCW (CARR, N, IARR): Gets literal data into non-character-type variables, move character variable to word array; 'CARR,' 'N' represents character variable of length 'N,' 'IARR' is destination array.

91) MOVW (NWORDS, FROM, TO): Moves N words from one array to another. If 'NWORDS' is less than or equal to 0, nothing happens.

92) MOVWC (IARR, CARR, N): Move word data to character variable. 'IARR' is source array, 'CARR,' 'N' represents character variable of length 'N.'

93) NSTAR (RADS, VCLR, IFAIL): Infers clear column radiances from partially-clouded fields of view.

94) NVINIT (BETAIN, BETDOT, INAV, PTIME): Refers to opening up of navigation (NAV) file.

95) O3TAV (TOTO, PATH): Computes transmittances for channels affected by ozone absorption (especially channel 2 (approximately 690 cm^{-1})).

96) OPNA (IAREA): Collect a list of the track groups assigned to the area and place them in the common block.

97) OUTINT (KOUT): Outputs integer values (maximum of 9) in an array onto the video screen (CRT) or printer. It is somewhat tedious, since only one array at a time can be printed, whereas subroutines ENKODE/ENCODE can print several variables (including arrays) at once. In addition, OUTINT works only for integers, while ENKODE/ENCODE can print characters and reals, in addition to integers.

98) PACK (N, IS, ID): Moves 'N' bytes from 'IS' at one per word to 'ID' at four per word.

99) PAGE (LINB, NELB, LINE, NELE, N): Basic graphics package (WRRRMS only, WRRRM refers to old graphics program). Modified to compute distances to edge of CRT, defines plotting area.

100) PENADD (IVAL): WRRRM ("Write Random Read Raster Memory") interface subroutine to add 3-bit values.

101) PENBEG: New WRRRM interface subroutine to begin a new packet, deals with terminal protocol when new line is to be added.

102) PENMOV (ILINED, IPIXLD): WRRRM ("Write Random Read Raster Memory") interface subroutine to move the pen a "delta."

103) PLNKIV (ISAT): Brings in coefficients used for any Planck function calculations from disk.

104) PLOT (LIN, NELEM, IPN): Is called to plot a line. It does any scaling requested, checks that the points will go on the screen and adjusts the line if not. In addition, PLOT handles the generation of wide and dashed lines. Note that the last point of the line is not displayed under the assumption that it will be the first point displayed on the next call. 'LIN' and 'NELEM' represent the logical end of a line.

105) PLTDIG (LINE, LEL, NVAL, LHGT, NFW, LEVEL): Plots a single digit.

106) POST: Part of PRINT package, "wakes up" printer handlers. Is valid only in McIDAS.

107) PRCL: Is an entry within PRRD on the line immediately below the PRWR entry. Closes a transfer file.

108) PRCLOS (I): End a print spool string and connect it to que for printer 'I.'

109) PREATV (TEMP, WMIX): Takes atmospheric information, gets quantities required for transmittance model (also computes derived functions of temperature and mixing ratio).

110) PRECW (P, W, U, NP): Calculates precipitable water for a given atmospheric layer or layers using pressure and mixing ratio. Calculates 'U' between two given p-levels.

111) PRETAV (ISAT, KCHAN): More preparation for transmittance calculations, initializes pressure functions and reads in coefficients.

112) PRIOUT (JDEV): To regulate printed output, as follows:

Ø. Suppress all LTQ-generated output (black hole).

1. To terminal monitor

2. Local printer if there is one, otherwise to second floor, provided you are at a second floor terminal.

3. System printer (sixth floor).

113) PRLINX (N, LINE): Spooled printing subroutine. Prints a line on printer number 'N' (≧ 0); 'N'=0 implies system printer. Interleaving lines to different printers is possible, but each time the printer number is changed, the last batch of lines is tied off and spooled out. Hence, the printer number should be changed only infrequently. To cause a skip to top of page, call PRLINX with LINE (1)=-1. Calling PRLINX with 'N' < 0 just closes any open printer.

114) PROFIX (PR, TR, DR, WR, P, T, W, IL, IM, IS, IT, LAT): Takes first guess profile from NMC grid and transforms it to a form needed for internal levels (temperature at 15 levels becomes temperature at 40 levels; dewpoint at 6 levels becomes mixing ratio at 15 levels, etc.).

115) PROPEN: Start a print spool string.

116) PRPRPR (K): Add a line of print to spool.

117) PRRD: Printer spool read/write, reads record from transfer file, is not called by user.

118) PRWR: Is also an entry within PRRD, writes record on transfer file, is not called by user.

119) PUC (FROM, INDEX): "Put user common," pokes word into user common. Subroutines LUC/PUC map user-common indices into indices in core-resident SYSCOM. The constants required for this mapping are computed when the system (the keyin scanner) calls "INILUC." System must call "INILUC" before LUC or PUC can be called. 'INDEX' is where (in user common) to do transfer, 'FROM' is fullword data value to transfer to user common, final value is word transferred.

120) PUTCHR (NBYTE, ARRAY, ITEM): Stores simple byte in an array (the opposite of LCHAR).

121) QGDASH (JDASH): Return current status of dash mode.

122) QVTWR (TBO, ERO, TOTO, TSKIN, TSTA, WSTA, LSTA, NSAT, IFAIL): Does "one-step" physical matrix inversion retrieval described in Smith and Woolf, 1984.

123) RAOBIN (P, T, W, IOK): Ingest radiosonde data, gets radiosonde report (mandatory + significant level data).

124) RBYTSX (IAREA, NREC, IOFF, NUMBYT, IARRAY): Read a line from an area, returning a specified subset of the line. 'IAREA' = area number, 'NREC' = line ioff (0-based), 'IOFF' = offset to first byte desired. 0 returns the first byte of the data. Negative offsets are used to access the prefix. A large negative offset will reference the first byte of the prefix. 'NUMBYT' = number of bytes desired; will not return any past the line end. 'IARRAY' = (output) array to hold returned data.

125) RDTRK (INDEX, TRACK, ARRAY): Area access method BDAM read/write, reads a track off an area file. 'INDEX' is a number between 0 and 7 specifying which DCB is used, 'TRACK' is a number between 0 and 16383 specifying track number, 'ARRAY' is the I 0 buffer address.

126) READD (ANUM, ENTRY): Reads directory entry from area/sounding directory "DATDIR." Input parameter--'ANUM' = area number. Output parameter: 'ENTRY' = 64 word directory entry (integer array).

127) READDL (ANUM, ENTRY): Locks and reads directory entry from area/sounding directory "DATDIR." Input parameter: 'ANUM' = area number. Output parameter: 'ENTRY' = 64 word directory entry (integer array).

128) READOF (IAREA, LINOFF, IOFF, NUMBYT, IARRAY): Read a line from an area, returning a specified subset of the line. 'IAREA' = area number, 'LINOFF' = line offset (0-based), 'IOFF' = offset to first byte desired. 0 returns the first byte of the data. Negative offsets are used to access the prefix. A large negative offset will reference the first byte of the prefix. 'NUMBYT' = number of bytes desired. Will not return any past the line end. 'IARRAY' = (output) array to hold returned data.

129) RETIO (IOUT, NSEC, IOPT): TOVS equivalent to VRTIO, except doesn't access any context file or MD file. Only accesses TOVS retrieval files.

130) RORDER (B, A, NI, M, IDIR): Orders the data in the array 'B' from smallest to largest ('IDIR'=1), or largest to smallest ('IDIR'=2). The reordered data is returned in the array 'A.' Upon completion of the ordering, the array 'NI' contains the original positional subscripts of 'B' in the new order. This index array can then be used to order other data associated with the data in 'B.' 'M' is the number of elements to be ordered.

131) SATEAR (PICTIM, XLIN, XELE, XLAT, XLON, ITYPE, INAV, BETAIN, BETDOT, ATFRAC): Most general satellite-earth coordinate transformation program, computes satellite, earth coordinates, earth edges, sub points. T(0) is defined to be Greenwich hour 0 of day of navigation. Latitude ranges from +90 north to -90 south. Longitude ranges from +180 east to -180 west. Input parameters: 'PICTIM' = picture start time (hours from T(0)), 'XLIN' = satellite coordinate (line), 'XELE' = satellite coordinate (element), 'XLAT' = earth coordinate (degrees latitude), 'XLON' = earth coordinate (degrees longitude), 'ITYPE' = 1 for satellite coordinate to earth coordinate transformation, 2 for earth coordinate to satellite coordinate transformation, etc. (see MRSATEAR for more information). 'INAV' = +1 - for use of updated

navigation parameters (sets limit to 5), 0 for use of previous navigation parameters, -1 for use of updated navigation parameters (sets limit to 2), 'BETAIN' = beta angle at T(0) (elements), 'BETDOT' = rate of change of beta (elements per hour), 'ATFRAC' = cloud height coefficient (ranges from 0 to 1).

132) SATPOS (INORB, NTIME, X, Y, Z): Generates satellite position vector from earth center. Input parameters (all integers): 'INORB' = initialization flag (should = 0 on first call to SATPOS, 1 on all subsequent calls), 'NTIME' = time (hours, minutes, seconds) in HHMMSS format. Output parameters (all floating point): 'X,' 'Y,' 'Z' = coordinates of position vector.

133) SATTV (FRAM, ILIN, IELE, ITVLIN, ITVELE, SS, JDAY, TIME): Transforms from satellite to TV coordinates. Input parameters: 'FRAM' = frame number, 'ILIN' = line satellite coordinate, 'IELE' = element satellite coordinate. Output parameters: 'ITVLIN' = line TV coordinate, 'ITVELE' = element TV coordinate, 'SS' = satellite identification number, 'JDAY' = YYDDD of frame, 'TIME' = HHMMSS of frame.

134) SDEST (CTEXT, IVAL): Display 'CTEXT' + integer/A4 on CRT (standard message class), displays message at standard message destination (similar to subroutine TQMES).

135) SENOUT: Changes terminal state, sends arbitrary messages to terminal.

136) SGRAOB (MD, IDAY, ITIME, IDNO, NSIGT, NSIGW, ISND): Returns (1 station per call) significant level raob reports. Input parameters: 'MD' is MD file number, 'IDAY' (YYDDD) is day, 'ITIME' (HH0000) is nominal time, usually 0 or 120000, 'IDNO' is station number. Output parameters: 'NSIGT' is number of returned significant temperatures (maximum is 100). If less than 0, MD file is bad or not schema "RSIG," 'NSIGW' is number of returned significant winds (maximum is 100), 'ISND' is array (7*100) to contain significant temperatures and winds. Each column contains level (either "SIGT" or "SIGW"), p is in mb *10, T(K*10), TD (K*10), DIR (degrees), SPD (meters per second), Z (meters). Note that for "SIGT," DIR, SPD and Z will be missing (=Z80808080), and for "SIGW," P, T and W will be missing. Function value is 0 (OK), or -1 (bad MD file specifications). Note: This subroutine is intended only for files with schema RSIG and assumes the positions of variables within the schema. If schema RSIG is changed, data values for LOCDAY and LOCREP may have to change.

137) SNDANL (HOUR, NLEV, P, T, TD, DIR, SPD, STABIL): Subroutine to compute stability indices for given sounding. Such as total-totals, lifted, K and SWEAT. 'HOUR'=HH.00 of raob time (used for parcel characteristics), 'NLEV' = number of input levels in data arrays, 'P' = pressure in mb, 'T' = temperature in degrees Kelvin, 'TD' = dewpoint temperature (K), 'DIR' = wind direction, in degrees, 'SPD' = wind speed, in meters per second, 'STABIL' = array containing outputted stability indices.

138) SOLARP (JDAY, JTIME, GHA, DEC, XLAT, XLON): Computes Greenwich hour angle and declination of sun. Input parameters: 'JDAY' = satellite year/day, 'JTIME' = hour/minute/second. Output parameters: 'GHA' Greenwich hour angle, 'DEC' = declination, 'XLAT' = latitude of sun position, 'XLON' = longitude of sun position.

139) SOLVEX (GAM, X, XIT, XTX, XIV, NF, NE): Calculates $(\Phi^T \Phi)^{-1}\Phi^T$. See Smith and Woolf, 1984, equation (6) therein.

140) SQSLED (ICODE): Set line (LTQ) device. Error (EMES) device, debug (DMES) device into UC (user common). 'ICODE' is 1-3 letters (A1, A2 or A3 format) which specify the output devices for the 3 message categories. The first letter directs the "line output" category, the second the "error message" category and the third the "debug message" category. The letters can be: C . . . CRT, P . . . local printer, S . . . system printer, or N . . . nowhere (output disappears). The values designated are recorded in UC words 31-33.

141) SRCH (LATS, LONGS, IM, ILIN, IELE, NPTS, NROWS): A subroutine which uses trigonometry to locate latitude/longitude coordinates in terms of TIROS-N line and element number. Basic framework is determined from top and bottom line numbers fed by 'NROWS' and LTOP. 'LONGS' is +west, -east.

142) SRGSS (IFLD, IDL, NCOLS, NROWS, TLAT, WLON, DINC, IP, MDNG): Gets guess information from guess MD file to use in creating surface analysis grids (for VAS or TOVS). Gets guess information at gridpoint locations.

143) STC (VAL, STR, J): Store a character at offset 'J' in string, store character in rightmost byte of 'VAL' at offset 'J' in 'STR' ('J' is 0-based), final value is 'VAL.'

144) STDATM (P, T, W): Returns 40 level (0.1-1000 mb) standard atmosphere for P, T, and W (mixing ratio). 1962 version for U.S.A.-derived from radiosondes, rocketsondes, etc.

145) SURGES (NOSFC): Get values from gridpoint surface analysis for TOVS or VAS, according to latitude and longitude (output is 1000 mb height, surface temperature, surface dewpoint depression).

146) SYMVRT (A, S, KN, N): Inverts the matrix $\Phi^T\Phi$. See Smith and Woolf, 1984, equation (6) therein.

147) TEKPUT (L, P): In graphics package, outputs to Techtronics (4010) compatible displays.

148) TOKANL (CPARR, NTOK, NKW, NARR, IDEVAL): Analyze token sequence produced by MCTOKN. 'CPARR' contains the token sequence, 'NTOK' is total number of tokens. Outputs: 'NKW' is the number of keywords found, which equal the number of items stored in 'NARR,' includes one item for the positional tokens and one for the field (if any). 'NARR' contains one entry for each keyword which gives the number of tokens belonging to that keyword (includes the keyword name), 'IDEVAL' (integer) contains the first four characters of the value assigned to keyword DEV= . (If DEV= not present, 'IDEVAL' contains blanks.)

149) TQ (BUFFER): Involved with writing message to terminal, is part of operating system, sends a text message to CRT.

150) TQMES (TEXT, VAL): Output 'TEXT' + integer/A4 on CRT. 'TEXT' is up to 20 word array of MSG, ending with "$" or "$$" if "$," 'VAL' treated as integer, if "$$" 'VAL' treated as integer or A4.

151) TQSET (DVC): Redirects destination of standard class error messages. Set/examine display device. To set LTQ device, 'DVC' should be 0 (black hole, no output), 1 (CRT), 2 (local printer), 3 (system printer). To see what is current LTQ device, 'DVC' should be -1; current device is initially set to value in UC (-31); function value is current LTQ device.

152) TRMNL (IS): Returns the terminal number.

153) TSNIO (IOP, IFL, IR, IC, NR, NC, IA): TIROS-N data input/output (operates on orbit/image file).

154) TVSAT (FRAM, ITVLIN, ITVELE, ILIN, IELE, SS, JDAY, TIME): Transforms from TV to satellite coordinates. Output parameters give line and element satellite coordinates (Note: satellite coordinates are often called simply "line" and "element") corresponding to input TV coordinates, as well as satellite identification number, date of frame image in YYDDD, plus time of frame image in HHMMSS.

155) ULMR (W): Upper level mixing ratio. Extrapolates moisture from 300 mb up to 70 mb, thereby giving a smooth vertical decrease in moisture rather than a sudden discontinuity (w decreases ultimately to .001 g/Kg for the top 20 levels of the standard atmosphere, so after guess is complete moisture will be .001 g/Kg from 70 to 0.1 mb).

156) UNLOCK: Releases file for subsequent use by other programs.

157) VALUE (LAN, LAS, LOW, LOE, INC, IDL, FSCL, VAL): Given latitude and longitude, returns a value interpolated from a grid.

158) VASDAT (IL, IE, VDAT): Routine to access VAS sounder file according to image line and element obtained from subroutine TVSAT, if VDAT(1)=-1, returns only navigation data, if JDAY=0, navigation skipped. 'IL,' 'IE' are cursor line and element positions, 'VDAT' is array used to return brightness temperatures at cursor location for different VAS bands. Note: IL and IE are changed in this routine.

159) VASDIG (IRAS, IPIC, ITEM, MAG, IWD, KOLOR): Simply calls subroutines INITPL, PLTDIG and ENDPLT, interfaces to plotting routines, causes a quantity to be plotted on video screen.

160) VASGES (MDNO, FLAT, FLON): Enters and extracts guess vectors (arrays of T or TD for 21 or 40 levels) of temperature, dewpoint from the VASGSS file (created by program GSVA). Guess values are moved through common blocks /GUESS/ and /SURF/.

161) VASNAV (FNUM, SYDREC, ISYD, IHMS): Partially sets up navigation common block, is VAS navigation package (sounding mode).

162) VASRTE (TAU, TEMP, TSFC, RAD, DBDT, TBB, DBDTBB, KCHAN, LGND): Calculates radiance being emitted toward the VAS via the radiative transfer equation (RTE). Calculates total radiance (surface + atmospheric), integrating downward from 0.1 mb to the surface, also gives profiles of $(\partial B/\partial T)(\gamma, T^n)$ where T^n = atmospheric temperature (this is one term used in VAS weighting function), as well as $(\partial B/\partial T_B)(\gamma, T_B{}^n)$, where $T_B{}^n$ = brightness temperature for a given channel calculated from the RTE and the Planck function. Note: VAS weighting function:

$$W(p) = \frac{\partial\tau}{\partial \ln p}(\gamma, p, \theta) \cdot \frac{\partial B}{\partial T}(\gamma, T^n)\Big|_p \Big/ \frac{\partial B}{\partial T_B}(\gamma, T_B{}^n)\Big|_T$$

163) VASTAU (TEMP, WMIX, TOTO, ZENANG, TAU, ISAT, KCHAN): Subroutine which calculates atmospheric transmittances for VAS channels through each of 40 pressure levels for a given channel. Transmittance will vary depending upon which channel is being used, as well as atmospheric transmittances at that channel's wavelength for CO2, water vapor/trace gas continuum, H2O vapor and ozone.

164) VASTPW (TBB, EBB, TOTO, TSFS, URET, ITMAX, ISAT, KCT, NP): Retrieves total precipitable water from VAS radiances.

165) VRTIO (IREC, NREC, IOPT): VAS retrieval input/output. Data from MD file and to MD file. Accesses VASTEXT (context file) which gives context in which program is running.

166) VRTOPO (MLAT, MLON, IELEV, ICHAR, NIN, IBEG): Functions like HRTOPO, except is optimized for VAS usage. Input parameters: 'MLAT' (Ø-89, +N, -S, *1ØØ), 'MLON' (Ø-18Ø, +E, -W, *1ØØ). Output parameters: 'IELEV' (elevation in meters), 'ICHAR' (surface characteristic, Ø=ocean, 1=land).

167) VTQ (TQBUF): Same as TQ on the Harris, fills array 'TQBUF'(20) with 20 values of 4H. Works the same as LTQ, except buffer is cleared as well (is not cleared in LTQ). Displays contents of subroutine ENKODE buffer.

168) VTRET (KC1, KC2, DELB, KUSE, WG, IS, M, ER, IOK): VAS temperature retrieval enhancement routine.

169) VWRET (TS, TAUW, U, NL, NLS, TSP, USP): VAS water vapor retrieval enhancement routine.

170) WALK (DROW, DCOL): Is internal subroutine related to wide line graphics, does actual plotting.

171) WD: Retrieves words from user common (UC), is part of the scanner (operating system), alters SYSCOM (systems communications region).

172) WMIX (P, T, DD, W, NL): Calculates mixing ratios for 'NL' atmospheric levels using pressure 'P', temperature 'T', dewpoint depression 'DD' and saturation vapor pressure (ES) at the dewpoint. Note: saturation vapor pressure at the dewpoint = vapor pressure at the dewpoint.

173) WRBOX (IR, IP, IZR, IZP, IC): Writes (plots) a box in color 'IC.'

174) WRITA (IAREA, NREC, IADDR): Writes into an area.

175) WRITDU (ANUM, ENTRY): Writes and unlocks area/sounding directory. Input parameters: 'ANUM' = area number, 'ENTRY' = 64-word directory entry to be written (integer array).

176) WRMAR (IR, IP, IZ, IC, IN): Program to operate "worm" (WRRRM-"Write Random Read Raster Memory"--graphics package) and indicate on graphics frame where retrieval has been attempted or deleted. 'IL' is line, 'IE' is element, 'IZ' is size of indicator (all in TV units). 'IN' is 1 for square, 0 for "X." 'IC' is color: 1-red, 2-green, 3-yellow.

177) WRTRK: Writes onto a track, is an entry within SRRDTRK.

178) ZECONV (ITEM, NBYTES, JRMP, OUTBUF): Used by subroutine ENCODE, is for conversion from integer to sexadecimal.

179) ZEROS: Sends a bunch of zeroes to FORTRAN, makes all elements of an array equal to zero.

180) ZWIND (JHITE, I, J, JLAT, SPEED, DIREC, JHIT2, DELT, JHIT3): Calculates a given type of wind (geostrophic, isallobaric, ageostrophic) according to the input arguments, also indirectly calculates gradient wind by calling subroutine GRADWI.

APPENDIX IV

Function List and Descriptions

Appendix IV contains descriptions of functions called within VAS retrieval software. However, the only functions described here are functions which are called in the main programs themselves, not the functions called within subroutines subservient to the main programs. In this sense, Appendix IV cannot be considered as comprehensive as Appendix III.

As in the third appendix, the McIDAS source library location for each function is listed, as well as the argument/parameter list and descriptions of the argument(s)/parameter(s) for each function whenever possible. Finally, the purpose of each function is also detailed.

COMPLETE ALPHABETICAL FUNCTION LIST

1)	AZMUTH	(VR)
2)	CHOP	(SR)
3)	CKWP	(MR)
4)	CLIT	(SR)
5)	CPP	(MR)
6)	DEWPT	(TR)
7)	DLIT	(ML)
8)	DKWP	(MR)
9)	DPP	(MR)
10)	ICURG	(in VLXRVA)
11)	IGGET	(in MRIGMAKE)
12)	IGOPEN	(in MRIGMAKE)
13)	IGPUT	(in MRIGMAKE)
14)	IKWP	(MR)
15)	ILALO	(MR)
16)	IPP	(MR)
17)	IROUND	(MR)
18)	ISATNV	(VR)
19)	IVASCL	(VR)
20)	LANSEA	(TR)
21)	LIT	(SR)
22)	LOGAND	(in VRLOGOR)
23)	LUC	(MR)
24)	LWI	(in MRLWSUBS)
25)	MDGET	(in MRMDMAKE)
26)	MDINFO	(MR)
27)	MDKEYS	(MR)
28)	MDOPEN	(in MRMDMAKE)
29)	MDPUT	(in MRMDMAKE)
30)	RADENC	(VR)
31)	VBRITE	(in subroutine PLNKIV)
32)	VBDTAU	(VR)
33)	VDBDTB	(in subroutine PLNKIV)
34)	VPLANC	(in subroutine PLNKIV)
35)	VSKINT	(VR)
36)	WSAT	(VR)

FUNCTION DESCRIPTION LIST

1) AZMUTH (ALAT, ALON, BLAT, BLON, DIST): To get azimuth from point A to point B with distance between returned as a by-product.

2) CHOP (X, YMIN, YMAX): Limit size of 'X' to values between and including 'YMIN' and 'YMAX.'

3) CKWP (CKW, I, CDEFLT): Get a program keyword parameter in character string form. 'CKW' is character and is "POS" or the name of a keyword, 'I' specifies which argument from the argument sequence to take (1-based). For example, "POS," 1 specifies the first positional parameter, 'I'=Ø gives the program name for positional parameters ('CKW'=' '), or "="//CKW for keyword parameters (if present). 'CDEFLT' is character * 12 to use as missing value. Function value is character string representing the desired parameter or 'CDEFLT' if parameter is missing.

4) CLIT (I): Type transformer: integer/real to character * 4.

5) CPP (I, CDEFLT): Get positional parameter (character * 12). Function value is 'I'th positional parameter or 'CDEFLT' if parameter is missing.

6) DEWPT (P, T, W): Calculates dewpoint using given P, T and mixing ratio.

7) DLIT (C): Returns own argument (literal). Result is real * 8 bitwise identical to 'C,' 'C' is character * 8.

8) DKWP (CKW, I, DDEFLT): Get keyword parameter (real*8), returns real*8 value.

9) DPP (I, DDEFLT): Get a positional parameter (real * 8).

10) ICURG (IL, IE, IRINC, IPINC): Get response from user, cull out TV line and element as well as cursor dimensions. Function value of 1 if user says 'END,' else Ø. <u>Output</u> <u>parameters</u>: 'IL,' 'IE' are cursor line/element positions.

11) IGGET (GFNO, GNO, MAXWDS, GRID, NR, NC, TABLE): Get a grid from a gridfile. 'GFNO' is gridfile number, 'GNO' is grid number within gridfile (1 . . . 999), 'MAXWDS' is maximum size of grid allowed to read, 'GRID' is array to contain grid, 'NR' is returned as number of rows in grid, 'NC' is returned as number of columns in grid, 'TABLE' is 64-word array to receive grid header. Final value is Ø (OK), -1 (no such grid or too big), or -2 (no such file).

12) IGOPEN (GFNO, FILNAM): Open gridfile, return file reference number. 'GFNO' is grid file number, 'FILNAM' is returned as file name of the specified gridfile (8-characters). Final value is Ø (OK), or -1 (can't, e.g., no such file or not a gridfile).

13) IGPUT (GFNO, IGNO, GRID, NR, NC, TABLE, GNO): Put a grid into a gridfile. 'GFNO' is gridfile number, 'IGNO:" if + or Ø, grid is written

in next empty slot after 'IGNO,' if negative, grid is written in ABS ('IGNO'), overwriting any grid that is there. 'GRID' is the grid array, 'NR' is number of rows in grid, 'NC' is number of columns in grid. If 'NR' or 'NC' is less than or equal to Ø, grid 'IGNO' is deleted from file. 'TABLE' is 64-word grid header. Caller must set it up, except that IGPUT stuffs in the 'NR' and 'NC' fields, 'GNO' is returned as actual grid number stored, final value is Ø (OK), -1 (no room) or -2 (no such grid file).

14) IKWP (CKW, I, IDEFLT): Get keyword parameter (KP) (integer). The keyword (first argument) is searched for in program keyin. If found, the first, second, third, etc., integer value denoted by the second argument is given to the variable. If the KP is not defined, then the variable is given the third argument's integer value by default. For example, the statement I=IKWP('IAD,'2,Ø) causes a search for the keyword parameter 'IAD.' If it is found, the second value of the keyword parameter is assigned to variable I. Otherwise, if 'IAD' is not found, I is assigned the value Ø by default.

15) ILALO (X): Changes a floating point latitude-longitude to a packed integer (sign DDD‸MM‸SS). Input parameter: 'X'=floating point latitude or longitude.

16) IPP (I, IDEFLT): Get positional parameter (integer). The first argument tells computer which value of the positional parameter to look for and assign to the variable. The second argument is the default value assigned to the variable if no such value of the positional parameter has been keyed in (or if the positional parameter has not been keyed in at all).

17) IROUND (X): Rounds a floating point value. Input parameters: 'X'-floating point value.

18) ISATNV (JSAT): Obtain VAS satellite number from 'JSAT' (obtained via TVSAT).

19) IVASCL (TBO, FAC): Predicts VAS brightness temperatures at 2210 cm^{-1} based on brightness temperatures at 715, 750 and 790 cm^{-1}; compare with observed and return the following: Ø if ABS (PRE-OBS) is within 'FAC' standard error, 1 if ABS (PRE-OBS) is not within 'FAC' standard error, 2 if check cannot be done because of missing data.

20) LANSEA (ILAT, ILON, LEVEL): Distinguishes between land or ocean, using a low resolution topography (1 degree resolution in both latitude and longitude, as opposed to HRTOPO, which uses 10 minute resolution). Is used when low accuracy distinction is needed. If result shows land, then elevation is also returned. One enters with latitude/longitude (positive E, negative W) in degrees * 1ØØ. Final value=2 (ocean) or 1 (land).

21) LIT (C): Returns integer bitwise identical to character * 4. This is type transformer for character * 4. 'C' is character * 4 datum (character * 8 for DLIT). Result of (LIT, ALIT, DLIT) is (integer, real * 4, real * 8) bitwise identical to 'C.'

22) LOGAND (A, B): Logical 'and' of two 4-byte arguments.

23) LUC (INDEX): "Look at user common," subroutines LUC/PUC map user-common indices into indices in core-resident SYSCOM. The constants required for this mapping are computed when the system (the keyin scanner) calls "INILUC." System must call "INILUC" before LUC or PUC can be called. 'INDEX' is where (in user common) to do transfer. If greater than Ø, terminal-related area is addressed, if less than Ø, initiator-related area is addressed. Final value is word extracted from user common.

24) LWI (CFNAME, IBEGWD, NWDS, IARR): This is the service routine for LW (large word)--level disk input-output, entry points LWI (input), LWO (output). 'CFNAME' is file name (12 bytes), 'IBEGWD' is first virtual word in transfer (Ø based), 'NWDS' is number of words to transfer (if 'NWDS' less than or equal to Ø, nothing happens), 'IARR' is an array containing record to read/write. Final value for LWI: if Ø (last data transferred was from an actual (allocated) file page), or -1 (last data transferred was from a null page, and was equal to Z80808080. Note: the function value from LWO should be ignored (always Ø)).

25) MDGET (MDNO, M, N, ARRAY): Get record from MD file. 'MDNO' is the MD file number, 'M,' 'N' are the row and column numbers of the desired record. If 'M'=Ø, a column header is designated. If 'N'=Ø, a row header is designated. 'ARRAY' must be long enough to contain the desired record--all of it, not just the portion being read. This is because row headers, column headers and data records are each read into different regions within 'ARRAY,' so that they will match up properly with their respective keys. If final value=Ø (OK), if=-1 (can't, i.e., no such record).

26) MDINFO (MDNO, MDHD): Check existence of MD file, return header. 'MDNO' is MD file number, 'MDHD' is 64-word array to receive MD file header. Function value is Ø (OK) or -1 (can't, file doesn't exist). Note: last seven words of 'MDHD' are initialized by MDINFO for buffer management. See MDI (MRMDI).

27) MDKEYS (MDNO, NLIST, LIST, SCALES, UNITS, LOCS): Returns information corresponding to keys in schema. Returns meteorological data (MD) values/units/location of keys in MD file. 'MDNO' is the MD file number, 'NLIST' is -1 to retrieve all keys in schema or else the number of keys in the list, 'LIST' is an array of keys, 1 per word (unless 'NLIST'=-1), 'SCALES' is an array of words to receive scale factors, 'UNITS' is an array of words to receive key units (A4), 'LOCS' is an array of words to receive subscripts within a complete (row header + column header + data record) MD file record where the fields specified by the keys are to be found. (Each such field is exactly one word long.) Final value is number of keys for which information is returned. If 'NLIST' is greater than Ø. Thus final value is less than or equal to 'NLIST;' if not equal, it indicates that one or more keys were not found in schema. If 'NLIST'=-1, final value is total number of keys in schema (less than or equal to 4ØØ); final value is -1 if error (i.e., file does not exist). When 'NLIST' is -1, arrays should be at least 4ØØ words in order to contain maximum possible schema. Since the MD file structure permits multiple fields with the same name, MDKEYS is careful to return the values for the first key in the schema for the first occurrence of that key in 'LIST,' the values for the second key in the schema for the second occurrence, and so on.

28) MDOPEN (MDNO, ACCESS): Open MD file. 'MDNO' is MD file number (1-9999), 'ACCESS' is 1 (open for read) or 2 (open for read/write). Final value is Ø (OK or already open) or -1 (can't open).

29) MDPUT (MDNO, M, N, ARRAY): Put record to MD file. 'MDNO' is MD file number 'M,' 'N' are row, column number of desired record. If M=Ø, a row header must be designated. If n=0, a column header must be designated. 'ARRAY' must be long enough to contain the desired record, all of it, not just the portion being read. This is done because row headers, column headers and data records are each read into different regions within 'ARRAY' so that they will match up properly with their respective keys. If final value=Ø (OK), if it =-1 (can't, i.e., no such record).

30) RADENC (VALUE, CHANL, DELTAF, YSUBZ): Converts radiance in given channel from counts to radiance in units of mWm^{-2} str^{-1} cm^{-1}.

31) VBRITE (R, K): Calculates brightness temperature based on given radiance and wavenumber.

32) VBDTAU (TAU, B, BS, KCHAN, NL): Calculates radiance using radiative transfer equation (RTE--integrated from top to bottom of atmosphere at retrieval location, i.e., integrated from top down through last full layer above ground). Final radiance includes both surface and atmospheric contributions as well as an empirical correction to the radiance based on the channel being used. Radiance returned is at least .001 mWm^{-2} str^{-1} cm^{-1}.

33) VDBDTB (TBB, K): Computes derivative of Planck function (radiance) with respect to brightness temperature.

34) VPLANC (T, K): Calculates Planck radiance from given temperature and channel.

35) VSKINT (TBB, JCOF, JUSE, JACT): Obtain surface skin temperature from VAS brightness temperatures. 'JCOF'=Ø for empirical coefficients, 'JUSE'=Ø to use day (2-channel) or night (3-channel) equation according to solar zenith angle, 'JUSE'=1 to use day equation, 2 to use night equation, 'JACT'=0 if nothing done, =1 if day equation actually used, =2 if night equation actually used.

36) WSAT (P, T): Calculates saturation mixing ratio for a given P and T.

Appendix V
Miscellaneous Program Information

This appendix will be of interest to users who wish to know about VAS retrieval programs in terms of core memory usage, code length, number of physical input-outputs (I-O's) during each execution, and(or) how many times a given program is called during the processing of a typical retrieval area.

The information for all the programs is arranged in a tabular format. The four columns headed 1-4 correspond to the above four pieces of information for each program. Column 1 contains core memory usage (in bytes * 1024, or "K" bytes). Column 2 holds program code length, including both total number of lines and number of comment statements. Column 3 lists the number of physical I-O's during a given program run, and column 4 indicates approximately how many times a given program would be executed during the processing of a typical retrieval area. It must be kept in mind that numbers 3 and 4 are only approximate figures, and can vary depending both on the particular data set being processed and the time at which the processing is being done.

Before the presentation of the table, a few comments on physical I-O's is in order. Physical I-O's are those I-O's which cause a "wait state" on the McIDAS system; that is, they cause system activity to be suspended until the input has been completed. For a given program, the following constitute physical I-O's:

1.) 1 I-O for program keyin

2.) 1 I-O for every page read off an LW (Large Word array) file

3.) 1 I-O for every track read off an area

4.) 1 I-O for every line which goes to a display device (TV or CRT line)

(Note: to see definitions of page, Large Word (LW) array file and track, turn to Appendix I, Section I.

PROGRAM	**1**	**2**	**3**	**4**
VPVA	26	38=total 32=code	8	1
IDVA	164	348=total 290=code	54 in auto mode:47	1
LOVA	33	51=total 46=code	15	optional, usually .LE. 5
SPVA	51	60=total 49=code	from 12 to 24, depending on # of keyword parameters used	1
GSVA	350	297=total 245=code	82	1
CSVA	61	228=total 187=code	230, depends on # of reports written into the sfc. MD file	1
SRVA	186	365=total 292=code	26	3, more if sfc. analyses must be redone
XRVA	117	269=total 230=code	72	depends on # of sfc. and (or) retrieval reports deleted
VTPZ	694	562=total 486=code	434 in auto mode:442 (when auto mode was done over a retrieval area approximately one-half the size of the "442" area above, I-O's only decreased to 411)	1

PLVA	121	324=total 277=code	73, when plotting approximately 200 sfc. reports; 118, when plotting approximately 195 retrieval 500 mb height reports	usually no more than 6-8
BNVA	279	542=total 436=code	65	optional, depending upon how many grids of retrieval parameters are desired (9,for example, if gradient winds at standard levels (850-100 mb) are computed
SRAD	571	484=total 415=code	1998 in auto mode:2001 (for retrieval area of approximately one-half the size of the "1998" area:1071)	1 (not used at present;VTPZ replaces SRAD)
SRET	257	799=total 646=code	1772 (for same half-area as above:997)	1(not used at present;VTPZ replaces SRET)

Note: physical I-O's for both SRAD and SRET depend heavily on how many retrievals are to be processed in the retrieval area.

GWVA	159	288=total 234=code	325	9, for gradient winds at standard levels (850-300 mb)
UGVA	51	136=total 98=code	1806	1
EXVA	110	124=total 113=code	25,for 1 report, 46 for 5 reports; that is, about 5 more I-O's for each additional report	lesser-used program;used few times if at all during processing of typical area of retrievals
GPVA	548	100=total 85=code	61	lesser-used program;used

				few times if at all during processing of typical area of retrievals
VDVA	147	96=total 86=code	21	lesser-used program;used few times if at all during processing of typical area of retrievals
VTPW	661	428=total 376=code	not determined	obsolete at present; the function of VTPW is accomplished by VTPZ
ESVA	113	267=total 233=code	44	not used at present;if used in future, will probably only be used at most a few times during the processing of a typical retrieval area

APPENDIX VI

VAS Retrieval Software

The final appendix consists of copies of all the programs discussed in this manual, in the same order as listed in the Table of Contents. This appendix should be used in conjunction with a given Level I Flowchart and program description to achieve the best possible understanding of the program in question.

```
//VPVA7000 JOB  CLASS=A,MSGLEVEL=(0,0)
//  VLVPVA   SSG  12/23/83: ENTERED USER MANUAL CARD
//  EXEC  MCPRG,MOD=VPVA,LANGLVL=66
//FOPT.SYSIN  DD
@PROCESS SC(TQMES,EMES,DMES,MCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTOF,OPCON,SGX,SGW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAIN0
C ?   FILE POINTER FOR VAS PROCESSING
C ?      KEYIN: VPVA NN
C ?           NN=SOUNDER AREA NO. (TO SET)
C ?           NN=0 TO INTERROGATE
C ?           NN=-1 TO TURN OFF
C    SSEC/MCIDAS USERS MANUAL - CHAP12
      DIMENSION MIN(32),KOUT(10),IBUF(112),IDIR(64)
      COMMON /ARENT/IDIR
      CALL IQ(MIN(2))
      IF (MIN(2).EQ.LIT('HEL ').OR.MIN(2).EQ.LIT('HELP'))GO TO 900
      IFILE=MIN(2)
      NFILE=IFILE
      IF(IFILE.GE.0) GO TO 10
      NFILE=0
      GO TO 20
   10 IF (IFILE.EQ.0)GO TO 200
      IF (IFILE.GT.9999)GO TO 800
   20 CONTINUE
      CALL PUC(NFILE,81)
  100 CALL TQMES(' POINTER SET TO SOUNDER FILE$',NFILE)
      IF(NFILE.EQ.0) RETURN
      CALL READD(NFILE,IDIR)
      IF (IDIR(1).EQ.-1)GO TO 120
      KOUT(1)=3
      KOUT(2)=IDIR(3)
      KOUT(3)=IDIR(4)
      KOUT(4)=IDIR(5)
      CALL OUTINT(KOUT)
      RETURN
  120 CALL TQMES(' POINTER NOT SET TO VALID SOUNDER FILE$$',0)
      RETURN
  200 NFILE=LUC(81)
      GO TO 100
  800 CALL TQMES(' ILLEGAL POINTER REQUEST ...$$',0)
      RETURN
  900 CALL TQMES('  (SNDR FILE NO.)  -1 SETS POINTER TO 0$$',0)
      RETURN
      END
/
```

```
//IDVASECC JCE CLASS=E,MSGLEVEL=(0,C)
//   VLIDVA    JPN   03/29/84: MEMBER UPDATED
//   VLIDVA    GSW   02/28/84; NOTE "IBOX" DEFAULTS.
//   VLIDVA    JPN   03/02/84; "APPROVED"
// EXEC MCPRG,MOD=IDVA
//FORT.SYSIN DD *
@PROCESS SC(TG,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTOR,CPCOM,SGX,SGW,DOPEN,DREAD,DWRITE)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAIN0
C ?   TO INITIALIZE VASTEXT DOCUMENTATION SECTOR (IF VAS PNTR SET)
C ?   OR TOVORB DOCUMENTATION
C ?   KEYIN: IDVA <MDNR> <MDRR> <LALONW> <LALOSE>
C ?     MDNR IS MD FILE FOR RETRIEVALS
C ?     MDRR IS ROW NO. (1-16)
C ?        IF MDRR IS SPECIFIED,ROW HEADER IS INITIALIZED.
C ?     LALONW AND LALOSE ARE LAT,LON BOUNDARIES FOR PROCESSING
C ?        IF LALONW IS NOT SPECIFIED THE CURRENT TV IMAGE IS USED
C ?
C ?            KEYWORD:  AUTO = NON ZERO FORCES NON-VIDEO PATH
C ?                      IBOX = RETRVL BOX SIZE (DEF: 5*5 FOR RES 16,
C ?                                                   11*11 FOR RES 8)
      DIMENSION VAS(13),MFR(64),KCUT(10)
      DIMENSION LAT(56),LON(56)
      DIMENSION IRET(300)
      DIMENSION LBUF(33)
      DATA LBUF/33 Z40404040/
      DATA NCOLS/56/
      DATA IRET/300 Z80808080/
      DATA LUN/20/,LEN/100/
      DATA WLON/360./,ELON/0./,TOPE/360./,BOTW/0./
      DATA LETOP/0/,IEBCT/999999/
      DATA VMISG/999999./
      COMMON /NAV/FLAT,FLON,ZENLOC,SZEN,IL,IE,IRAS,IPIC,IHMS,JT,JD
      COMMON /DOC/IDOC(112)
      COMMON /ARENT/IDIR(64)
C     ADD COMMON BLOCKS FOR VASDAT
      COMMON/AUTO/IBOX,IDUM(5)
      COMMON/LAST/LASLIN,LASELE,LELE,ICHAR
      MDNR=IPP(1,0)
      MDRR=IPP(2,0)
      LALONW=IPP(3,0)
      LALOSE=IPP(4,0)
      KBUG=IKWP('BUG',1,0)
      IF (LUC(81).EQ.0)GO TO 10
C     FOLLOWING FOR VAS
      ITERM=LUC(-20)
      CALL DOPEN('VASTEXT ',LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      GO TO 20
   10 CALL TSNIC(1,1,1,1,1,1,IDOC)
   20 CONTINUE
```

```
      IF (LALONW.EQ.0)GO TO 100
      IF (LALOSE.EQ.0)GO TO 920
C     FOR TOVS FILL DIRECTORY AND EXIT
      IDOC(25)=LALONW
      IDOC(26)=LALOSE
      IF (LUC(81).EQ.0)GO TO 280
      LAN=LALONW/1000
      FLAT=LAN
      LOW=MOD(LALONW,1000)
      WLON=LOW
      IF(LOW.GT.180)LOW=LOW-360
      FLON=-LOW
C      USE NAV TO GET LINE,ELE
      NSND=LUC(81)
      CALL READD(NSND,IDIR)
      JD=IDIR(4)+IDIR(3)*100000
      JT=IDIR(5)
      CALL NVINIT(BETAIN,BETDOT,INAV,PTIME)
C      LOOK AT ALL CORNERS TO GET MAX AND MIN ELEMENT
      CALL SATEAR(PTIME,FLIN,FELE,FLAT,FLON,2,INAV,BETAIN,BETDOT,0.)
      IL=FLIN+0.5
      ITE=FELE+0.5
      LAS=LALOSE/1000
      FLAT=LAS
      CALL SATEAR(PTIME,FLIN,FELE,FLAT,FLON,2,INAV,BETAIN,BETDOT,0.)
      IBE=FELE+0.5
      IE=MIN(ITE,IBE)
      LOE=MOD(LALOSE,1000)
      ELON=LOE
      IF (LOE.GT.180)LOE=LOE-360
      FLON=-LOE
      CALL SATEAR(PTIME,FLIN,FELE,FLAT,FLON,2,INAV,BETAIN,BETDOT,0.)
      LL=FLIN+0.5
      IBE=FELE+0.5
      FLAT=LAN
      CALL SATEAR(PTIME,FLIN,FELE,FLAT,FLON,2,INAV,BETAIN,BETDOT,0.)
      ITE=FELE+0.5
      LE=MAX(ITE,IBE)
      ILR=IDIR(12)
      IER=IDIR(13)
      GO TO 115
  100 IF (LUC(81).EQ.0)GO TO 280
C     TRANSFER FOR TOVS
      NFR=LUC(-1)
C     PICK UP IMAGE INFO
      CALL GETFRM(NFR,MFR)
C     LOAD TIME PARAMETERS IN COMMON /NAV/
      JD=MFR(1)*100000+MFR(2)
      JT=MFR(2)
      IL=MFR(5)
      IE=MFR(6)
C     IL AND IE GIVE AREA LINE AND ELEMENT OF UPPER LEFT DATA
```

```
C       DISPLAYED ON TV FRAME (NOT NECESSARILY RAS/PIX 1,1)
        IY=IL
        IX=IE
        MAG=MFR(10)
        ILR=MFR(11)
        IER=MFR(12)
C       LAST ARE LINE AND ELEMENT RESOLUTION
C       PICK UP SOUNDER AREA NO
        NSND=MFR(17)
        IDOC(3)=IL
        IDOC(4)=IE
C       CALCULATE AREA COORDINATES OF LOWER RIGHT RAS/PIX ON TV SCREEN
        LR=LUC(11)
        LP=LUC(12)
        LL=IY+((LR-MFR(8))*ILR)/MAG
        LE=IX+((LP-MFR(9))*IER)/MAG
    115 IF (KBUG.NE.0)CALL ENKODE('("IL,IE,ILR,IER",4I8/)',LBUF,
       -IL,IE,ILR,IER)
C       ENSURE INITIAL ELEMENT IS OK
C       AND GET LAT AND LON OF POSITION
        IEB=(IE+LE)/2
        IF (KBUG.NE.0)CALL ENKODE('("LL,LE,IEB",3I8/)',LBUF,
       -LL,LE,IEB)
C       USE VASDAT TO PICK UP BOUNDS OF DATA ON TV SCREEN OR AS FORCED
C       BY LAT AND LON ENTRIES
C       PASS FLAGS TO VASDAT THRU COMMON
        IBOX=0
        LASELE=-1
        LASLIN=-1
        ISEC=0
C       USE ISEC TO INSIST ON 4 CONSECUTIVE LINES OF DATA.  THIS
C       IS TO POSITION SPACING OF RETRIEVALS IN SMALL DETECTOR MODE
    120 VAS(1)=0.
C       VAS(1) MUST BE NON NEG TO RETURN DATA VECTOR
C        CHECK TOP LINE
        CALL VASDAT(IL,IEB,VAS)
        LAN=FLAT+0.5
        IF (VAS(1).GT.0.AND.VAS(7).NE.VMISG)GO TO 125
C        VAS(7) IS SMALL DETECTOR...MAYBE
        ISEC=0
    122 IL=IL+ILR
        IF (IL.GT.LL)GO TO 300
        GO TO 120
    125 ISEC=ISEC+1
        IF (ISEC.NE.1)GO TO 126
        ILS=IL
        LAN=FLAT+0.5
    126 IF(ISEC.GE.4)GO TO 130
        GO TO 122
    130 VAS(1)=0.
        IL=ILS
        IETOP=IE
```

```
      LASLIN=1
  133 VAS(1)=0.
      CALL VASDAT(IL,IETOP,VAS)
      IF (FLON.LT.0.)FLON=FLON+360.
      IF (VAS(1).GE.0..AND.FLON.LE.WLON)GO TO 135
C     LONG TEST IS TO CORRECT PARALLAX
      IETOP=IETOP+IER
      IF (IETOP.GT.LE)GO TO 300
      GO TO 133
  135 TOPW=FLON
C     CHECK LAT/LON FORCE
      IF (LALONW.EQ.0)GO TO 138
C     MOVE ACROSS TO GET EASTERNMOST ELEMENT
      LETOP=IETOP
  136 LETOP=LETOP+IER
      VAS(1)=0.
      CALL VASDAT(IL,LETOP,VAS)
      IF (FLON.LT.0)FLON=FLON+360.
      IF (VAS(1).GT.0.AND.FLON.GT.ELON)GO TO 136
      TOPE=FLON
C     PICK UP INITIAL TIME FROM SOUNDER FILE
  138 IDOC(10)=IHMS
      CALL SDEST('...UPPER LEFT DONE',0)
C     GO TO LOWER RIGHT CORNER (OF TV) AND BACK UP TO DARA
      LASLIN=-1
  140 CONTINUE
      VAS(1)=0.
      CALL VASDAT(LL,IEB,VAS)
      IF(VAS(1).GE.0.) GO TO 150
      LL=LL-ILR
      IF(LL.LE.IL) GO TO 400
      GO TO 140
  150 CONTINUE
      LAS=FLAT-0.5
      LOE=FLON-0.5
      LEBOT=LE
      IF (LALONW.EQ.0)GO TO 156
C     MARCH ALONG LINE TO AGAIN GET ELEMENT NOS. WITH LAT/LON BOUNDS
      IEBOT=IE
      LASLIN=1
  153 VAS(1)=0.
      CALL VASDAT(LL,IEBOT,VAS)
      IF (FLON.LT.0)FLON=FLON+360.
      IF(VAS(1).GE.0..AND.FLON.LE.WLON)GO TO 155
      IEBOT=IEBOT+IER
      IF(IEBOT.GT.LE) GO TO 400
      GO TO 153
  155 BOTW=FLON
  156 VAS(1)=0.
      CALL VASDAT(LL,LEBOT,VAS)
      IF (FLON.LT.0.)FLON=FLON+360.
      IF (VAS(1).GE.0.AND.FLON.GE.ELON)GO TO 157
```

```
      LEBOT=LEBOT-IER
      IF (LEBOT.LT.IE)GO TO 400
      GO TO 156
  157 BOTE=FLON
      IE=MIN(IETOP,IEBOT)
      LE=MAX(LETOP,LEBOT)
      LOW=AMAX1(TOPW,BOTW)+0.1
      LOE=AMIN1(TOPE,BOTE)-0.1
  158 IF (LOE.LT.0.)LOE=360.+LOE
      IF (LOW.LT.0.)LOW=360.+LOW
      IF (KBUG.NE.0)CALL ENKODE('(1X,"TOPW,TOPE,BOTW,BOTE ",4F7.2/)'
      LBUF,TOPW,TOPE,BOTW,BOTE)
      CALL SDEST('...LOWER RIGHT DONE',0)
      IDOC(11)=IHMS
      DO 160 L=1,13
      IDOC(11+L)=0
      IF (VAS(L).NE.99999.)IDOC(11+L)=1
  160 CONTINUE
      LALONW=IABS(LAN)*1000+LOW
      IF (LAN.LT.0)LALONW=-LALONW
      LALOSE=IABS(LAS)*1000+LOE
      IDOC(25)=LALONW
      IF (LAS.LT.0)LALOSE=-LALOSE
      IDOC(26)=LALOSE
      CALL SDEST('     LINUL    ELEUL    LATUL    LONUL    LINLR    ELELR
      TLR   LONLR',0)
      KOUT(1)=8
      KOUT(2)=IL
      KOUT(3)=IE
      KOUT(4)=LAN
      KOUT(5)=LOW
      KOUT(6)=LL
      KOUT(7)=LE
      KOUT(8)=LAS
      KOUT(9)=LOE
      CALL OUTINT(KOUT)
C     FILL VASRET DOCUMENTATION FROM SOUNDER DIRECTORY
      IDOC(1)=IDIR(3)-100000+IDIR(4)
      DO 170 I=2,11
  170 IDOC(I)=IDIR(I+3)
      IF (MDNR.EQ.0)GO TO 260
      IF (MDOPEN(MDNR,2).LT.0)GO TO 900
C     INITIALIZE RETRIEVAL MD ROW
      IF (MDRR.EQ.0)GO TO 250
      IRET(1)=MOD(IDOC(1),100000)
      IHR=IDOC(2)/10000
      IRET(2)=IDOC(2)
      IRET(3)=0
      ICK=MDPUT(MDNR,MDRR,0,IRET)
      IF (ICK.LT.0)GO TO 910
  250 CONTINUE
      IDOC(40)=MDNR
```

```
      IDCC(41)=MCRR
  260 CONTINUE
C     SET STATUS FLAG TC READY
      IDCC(28)=1
C     PUT IN SNDR FILE NC
      NSND=LUC(81)
      IDOC(35)=NSND
C     SET LINE AND ELEMENTS FOR SRAD
      IDCC(50)=0
      IDCC(51)=0
C     SET SPACING DEFAULTS FCR SRAD
      IDCC(52)=1C 16/ILR+0.5
C     FCLLOWING IS FOR BCX SIZE (FER RETRIEVAL..5 5 FOR RES 16)
      IDCC(53)=5*16/ILR+0.5
C     LOOK FOR FORCED OVERRIDE
      IDOC(53)=IKWF('IBCX',1,IDCC(53))
C     INDENT BY HALF THE BCX SIZE
      IOFF=IDCC(53)/2
      IF (MCD(IOFF,2).NE.1)IOFF=IOFF+1
      IOFF=IOFF ILR
      IDOC(54)=0
      IDOC(55)=LL-IOFF
      IDOC(56)=LE-IOFF
      IDOC(57)=IL+IOFF
      IDOC(58)=IE+IOFF
      IDOC(59)=ITERM
      IDOC(60)=0
      CALL DWRITE(LUN,ITERM,IDOC)
      CALL DCLOSE(LUN)
      CALL SDEST('  DOCUMENTATION COMPLETE FOR TERMINAL ',ITERM)
      RETURN
  280 CONTINUE
C
C     BEGIN SEARCH FOR MAX AND MIN LAT AND LONG REQUIRED BY ORBIT
C     WATCH FOR MISSING LINES AT BEGINNING AND END OF ORBIT
      NROW=IDOC(4)
      DO 2 N=5,NROW
      K=N
      CALL TSNIC(0,1,K,1,1,NCOLS,LAT)
      IF (LAT(1).LE.9000.AND.LAT(NCOLS).LE.9000)GO TO 3
    2 CONTINUE
    3 CALL TSNIC(0,2,K,1,1,NCOLS,LON)
      NODE=1
      IF (LAT(NCOLS).LT.LAT(1))NODE=2
C     NODE=2 IS DESCENDING
      MAXLAT=LAT(NCOLS)+300
      MAXLON=LON(1)-500
      IF (NODE.EQ.1)GO TO 4
      MAXLAT=LAT(1)+300
      MINLON=LON(NCOLS)+500
    4 NR=NROW-5
      DO 5 N=NR,1,-1
```

```
      K=N
      CALL TSNIO(0,1,K,1,1,NCOLS,LAT)
      IF (LAT(1).LE.9000.AND.LAT(NCOLS).LE.9000)GO TO 44
    5 CONTINUE
   44 CALL TSNIO(0,2,K,1,1,NCOLS,LON)
      IF (NODE.EQ.2)GO TO 6
      MINLAT=LAT(1)-300
      MINLON=LON(NCOLS)+500
      GO TO 7
    6 MAXLON=LON(1)-500
      MINLAT=LAT(NCOLS)-300
C     CHANGE LOCALLY TO LONG CONV POS W NEG E
    7 MAXLON=-MAXLON
      MINLON=-MINLON
C     CHANGE TO 0-360 W FOR STORAGE
      IF (MAXLON.LT.0)MAXLON=MAXLON+36000
      IF (MINLON.LT.0)MINLON=MINLON+36000
      MAXLAT=MAXLAT/100
      MINLAT=MINLAT/100
      LALONW=IABS(MAXLAT)*1000+MAXLON/100
      IF (MAXLAT.LT.0)LALONW=-LALONW
      LALOSE=IABS(MINLAT)*1000+MINLON/100
      IF (MINLAT.LT.0)LALOSE=-LALOSE
      IDOC(25)=LALONW
      IDOC(26)=LALOSE
      IDOC(40)=MDNR
      IDOC(41)=MDRR
C     SET STATUS FLAG TO READY
      IDOC(28)=1
      CALL TSNIO(3,1,1,1,1,1,IDOC)
C     LATER WE WILL RETURN TO 200 HERE
      CALL SDEST(' DUN..',0)
      RETURN
  300 CALL SDEST('...PROBLEM LOCATING NW CORNER ... ENTER LINE + ELEM V
     * KEYIN',0)
      CALL ABORT
  400 CALL SDEST('...PROBLEM LOCATING SE CORNER ... ENTER LINE + ELEM V
     * KEYIN',0)
      CALL ABORT
  900 CALL SDEST(' CANNOT OPEN RETRIEVAL MD FILE NO ',MDNR)
      GO TO 260
  910 CALL SDEST(' CANNOT WRITE MD ROW HEADER FOR ROW ',MDRR)
      GO TO 260
  920 CALL SDEST(' LALOSE HAS NOT BEEN SPECIFIED',0)
      RETURN
      END
```

```
//LOVA7000 JOB CLASS=A,MSGLEVEL=(0,0)
//  VLLOVA   SEG  12/23/83; ENTERED USER MANUAL CARD
// EXEC MCPRG,MOD=LOVA
//FORT.SYSIN DD *
&PROCESS SC(TGMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
&PROCESS SC(ISFILE,WTOR,CPCOM,SQX,SQW)
&PROCESS SC(MOVB,MOVC,MOVW,CLEANW,DOPEN,DREAD,DWRITE,DCLOSE)
      SUBROUTINE MAINO
C ?   TO QUERY VASTEXT DOCUMENTATION SECTOR (IF VAS POINTER SET)
C ?   OR TOVCRB TO WHICH TOVS POINTER IS SET
C ?      KEYIN: LOVA      -NO PARAMETERS-
C   SSEC/MCIDAS USERS MANUAL - CHAP12
      DIMENSION VAS(13),IBUF(112),MFR(64),KOUT(10)
      DIMENSION TGBUF(20)
      COMMON /DOC/IDOC(112)
      CHARACTER 80 TITLE,TIT2,TIT3,TIT4
      REAL*8 MFILE
      DATA TITLE/'    YYDDD   BEGIN   Y-RES   X-RES   LLNW   LLSE  STAT
     NSAT  SNDAREA          '/
      DATA TIT2 /'    MDNS    MDRS    MDNG    MDRG    MDNR    MDRR
                              '/
      DATA TIT3 /'    NGFG    NGFS   ZGRID   TGRID   DGRID
                              '/
      DATA TIT4/'   TYP   GSS   SPC   SIZ   SFC  ENDL  ENDE  BEGL  BEG
       TER   PLT              '/
      DATA LUN/20/,MFILE/'VASTEXT '/,LEN/100/
      COMMON /NAV/FLAT,FLON,ZENLOC,SZEN,IL,IE,IRAS,IPIC,IHMS,JT,JD
      COMMON /ARENT/IDIR(64)
      IF (LUC(81).EQ.0)GO TO 1
C     FOLLOWING FOR VAS
      ITERM=LUC(-20)
      CALL DOPEN(MFILE,LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      NRET=IDOC(100)
      NSAT=IDOC(1)/100000
      GO TO 2
    1 CALL TSNIO(1,1,1,1,1,1,IDOC)
      CALL RETIO(IBUF,0,0)
      NRET=IBUF(112)
      NSAT=IDOC(5)
      IDOC(9)=IDOC(4)
      IDOC(10)=56
    2 CALL SDEST(TITLE,0)
      IYD=MOD(IDOC(1),100000)
      ITIM=IDOC(2)
      NSND=IDOC(35)
      CALL ENKODE(' (4I8,2I7,I6,I7,I9/)',TGBUF,IYD,ITIM,
     1 IDOC(9),IDOC(10),IDOC(25),IDOC(26),IDOC(28),NSAT,NSND)
      CALL SDEST(TIT2,0)
      CALL ENKODE(' (6I8/)',TGBUF,IDOC(36))
      CALL SDEST(TIT3,0)
      CALL ENKODE(' (I8,*4I8/)',TGBUF,IDOC(42),IDOC(29))
```

```
      CALL SDEST(' NO. RETRIEVALS= ',NRET)
      IF (LLC(81).EQ.0)RETURN
      CALL SDEST(' CURRENT RETRIEVAL OPTIONS..',0)
      CALL SDEST(TIT4,0)
      CALL ENKODE(' (11I6/)',TGBUF,IDOC(50))
      RETURN
      END
```

```
//SPVAR680 JOB CLASS=A,MSGLEVEL=(0,0)
//   VLSPVA    AJS   01/06/84: MEMBER UPDATED
//   VLSPVA    SSG   12/23/83: ENTERED USER MANUAL CARD
// EXEC MCPRG,MOD=SPVA
//FORT.SYSIN DD
@PROCESS SC(TGMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(DOPEN,DREAD,DWRITE,ISFILE,WTOR,OPCOM,SQX,SQW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW)
          SUBROUTINE MAIN0
C ?       PROGRAM TO SET VASTEXT MD POINTERS AND SFC GRID INFO
C ?       ALSO NRET WITHIN MD FILE
C ?          KEYIN: SPVA  -KEYWORDS-
C ?       NRET MDNS MDRS MDNG MDRG MDNR MDRR NGFG NGFS ZGRID TGRID DGRID
C ?       LLNW LLSE
C      SSEC/MCIDAS USERS MANUAL - CHAP12
          COMMON /DOC/IDOC(112)
          COMMON /TERMNL/ITERM
          DIMENSION IRET(300),CPAR(10)
          CHARACTER*12 CPAR
          DATA LUN/20/,LEN/100/
          DATA CPAR/'MDNS','MDRS','MDNG','MDRG','MDNR','MDRR',
         * 'NGFS','ZGRID','TGRID','DGRID'/
          KBUG=IKWP('DBG',1,0)
          IF (LUC(81).EQ.0)GO TO 1
C         FOLLOWING FOR VAS APPLICATIONS
          ITERM=LUC(-20)
          CALL DOPEN('VASTEXT ',LUN,LEN)
          CALL DREAD(LUN,ITERM,IDOC)
          GO TO 2
        1 CALL TSNIC(1,1,1,1,1,1,IDOC)
        2 CONTINUE
C         CHECK LAT LON COVERAGE
          LLNW=IKWP('LLNW',1,0)
          LLSE=IKWP('LLSE',1,0)
          IF (LLNW.NE.0)IDOC(25)=LLNW
          IF (LLSE.NE.0)IDOC(26)=LLSE
C         CHECK MD NO.
          DO 10 M=1,6
          MESS=IKWP(CPAR(M),1,0)
          IF (MESS.EQ.0)GO TO 10
          IDOC(35+M)=MESS
       10 CONTINUE
C         CHECK GRID DOC
          DO 20 M=7,10
          MESS=IKWP(CPAR(M),1,0)
          IF (MESS.EQ.0)GO TO 20
          IDOC(22+M)=MESS
       20 CONTINUE
          MESS=IKWP('NGFG',1,0)
          IF (MESS.NE.0)IDOC(42)=MESS
          IF (LUC(81).EQ.0)GO TO 25
          CALL DWRITE(LUN,ITERM,IDOC)
```

```
          CALL DCLOSE(LUN)
          GO TO 28
       25 CALL TSMIC(3,1,1,1,1,1,IDOC)
C         CHECK ON RETRIEVAL NO. EDIT
       28 NRET=IKWF('NRET',1,-1)
          IF (NRET.EQ.-1)GO TO 30
          IF (LUC(81).EQ.0)GO TO 35
          CALL VRTIO(IRET,0,0)
          IRET(3)=NRET
          IDOC(100)=NRET
          CALL VPTIO(IRET,0,1)
          GO TO 30
       35 IDOC(112)=NRET
          CALL RETIO(IDOC,0,1)
       30 IF(KBUG.GE.0)CALL SDEST(' DUN...',0)
          RETURN
          END
```

```
//GSVA5660 JOB  CLASS=A,MSGLEVEL=(0,0)
//  EXEC  MCPRG,MOD=GSVA
//* VLGSVA   CMH  04/24/84: MEMBER UPDATED
//  VLGSVA   SBG  12/23/83; ENTERED USER MANUAL CARD
//FORT.SYSIN  DD  *
@PROCESS SC(TGMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTOR,DPCOM,SGX,SGW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW,DOPEN,DREAD,DWRITE,DCLOSE)
      SUBROUTINE MAIN0
C ?  PROGRAM TO PREPARE MD FILE OF GUESS PROFILES (CMH)
C ?  KEYIN: GSVA NGFG NGR MDNO  LLNW LLSE INC NGRB DELT
C ?      POSITIONAL PARAMETERS:
C ?      NGFG - GRID FILE OF GUESS GRIDS
C ?      NGR - FIRST GRID OF LATER (LATEST) SET OF GUESS GRIDS
C ?      MDNO - MDNO OF OUTPUT ARRAY
C ?      LLNW - NW LAT LON FOR MD FILE (0-360W)
C ?      LLSE - SE LAT LON FOR MD FILE
C ?      INC - LAT/LON INCREMENT IN DEGREES 10
C ?      NGRB - FIRST GRID OF EARLIER SET FOR TIME INTERP
C ?      DELT - FRACTIONAL TIME INTERP (EARLY+DELT*(LATE-EARLY))
C   SSEC/MCIDAS USER MANUAL - CHAP12
      COMMON /DOC/IDOC(112)
      DIMENSION LIST(26),KOUT(10),ISCALE(26),IUNIT(26),LOCS(26)
C     ABOVE TO RECEIVE MCKEY INFO
      DIMENSION IGRID(3200,23),MES(64),JREC(26),IREC(26),FISC(22)
      DIMENSION ICHAR(22),KCHAR(22),JCHAR(22),IPRESS(22),IGOT(22)
      DIMENSION LCHAR(22)
      INTEGER IGHD(64),IGHDT(64),IGID(8),ROW(3),COL
C-----DESCRIPTION OF 64-WORD GRID HEADERS
C
C-----GIVE TOTAL SIZE (WORDS), # ROWS, # COLS. (IGSIZE=IGNR*IGNC)
      EQUIVALENCE (IGSIZE,IGHD(1)),(IGNR,IGHD(2)),(IGNC,IGHD(3))
C-----YYDDD, HHMMSS AND VALID-TIME (IF APPLICABLE) FOR GRID
      EQUIVALENCE (IGDAY,IGHD(4)),(IGTIME,IGHD(5)),(IGTIMV,IGHD(6))
C-----DESCRIPTION OF GRIDDED VARIABLE (IN MD-FILE TERMS):
C-----     NAME, SCALE, AND UNITS
      EQUIVALENCE (IGVNAM,IGHD(7)),(IGVSCA,IGHD(8)),(IGVUNI,IGHD(9))
C-----DESCRIPTION OF VERTICAL LEVEL: VALUE, SCALE, AND UNITS
      EQUIVALENCE (IGLEVL,IGHD(10)),(IGLSCA,IGHD(11)),(IGLUNI,IGHD(12)
C-----GRIDDED-VARIABLE TYPE: =1 (TIME DIF) 2(TIME AVG) 4 (LEVEL DIF)
C-----     8 (LEVEL AVG) OR ANY SUM OF THE FOREGOING
      EQUIVALENCE (IGVTYP,IGHD(13))
C-----FOLLOWING USED IF PARAMETER IS A VERTICAL (LEVEL) DIF OR AVG
C-----     (SAME SCALE AS IGLEVL)
      EQUIVALENCE (IGLDIF,IGHD(14))
C-----FOLLOWING USED IF PARAMETER IS A TIME DIF OR AVG (HHMMSS)
      EQUIVALENCE (IGTDIF,IGHD(15))
C-----GRID ORIGIN, TYPE (I.E. TYPE OF PROJECTION)
      EQUIVALENCE (IGORG,IGHD(33)),(IGTYPE,IGHD(34))
C-----SUBSEQUENT COORDS (IGLAMX,IGLOMX,IGLAMN,IGLOMN,IGINCR,
C-----     IGPOLR,IGPOLC,IGSP60,IGCLON) ALL HAVE 4 IMPLIED DEC. PLACES.
C-----     LAT GOES FROM -900000 TO 900000, LON GOES FROM -1800000
```

```
C-----   TO 1800000 (WEST IS +)
C-----TYPE 1 GRIDS ARE PSEUDO-MERCATOR
      EQUIVALENCE (IGLAMX,IGHD(35)),(IGLOMX,IGHD(36)),(IGLAMN,IGHD(37)
     ,(IGLOMN,IGHD(38)),(IGINCR,IGHD(39))
C-----TYPE 2 GRIDS ARE POLAR-STEREOGRAPHIC
C-----GIVE ROW # OF NORTH POLE, COL # OF N.P., COL SPACING AT 60 DEG N
C-----   (DEG), LONGITUDE PARALLEL TO COLUMNS (DEG)
      EQUIVALENCE (IGPOLR,IGHD(35)),(IGPOLC,IGHD(36)),(IGSP60,IGHD(37)
     ,(IGCLON,IGHD(38))
C-----INITIALS OF USER AND PROJECT # UNDER WHICH GRID CREATED
      EQUIVALENCE (IGUSER,IGHD(41)),(IGPROJ,IGHD(42))
C-----CHARACTER ID SUPPLIED BY PROGRAM (ARBITRARY)
      EQUIVALENCE (IGID,IGHD(43))
      EQUIVALENCE (ROW(1),IREC(1))
      EQUIVALENCE (COL,IREC(4))
      DATA IPRESS/1000,850,700,500,400,300,250,200,150,100,70,50,30,20
     10,1000,850,700,500,400,300,1000/
C     FOLLOWING 2 SPECIAL CASES FOR CALLAN'S ANMRC AND ECMWF FIASCOS
      DATA LCHAR/15*' TMP',6*' DPT',' HGT'/
      DATA ICHAR/15*'TEMP',6*'TDPT','HGHT'/
      DATA KCHAR/15*'   T',6*'  TD','   H'/
      DATA JCHAR/15*'T   ',6*'TD  ','Z   '/
      DATA IGOT/22*0/
      DATA MISG/Z80808080/
      DATA LUN/20/,LEN/100/
C
C
C
      NGFG=IPP(1,0)
      IF (LUC(81).EQ.0)GO TO 110
      ITERM=LUC(-20)
      CALL DOPEN('VASTEXT ',LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      GO TO 120
  110 CALL TSNIO(1,1,1,1,1,1,IDOC)
  120 CONTINUE
      DO 1 N=1,22
      DO 1 I=1,3200
    1 IGRID(I,N)=MISG
      NGB=IPP(2,1)
      MDNG=IPP(3,0)
      IF (MDOPEN(MDNG,2).LT.0)GO TO 960
      LLNW=IPP(4,0)
      IF (LLNW.EQ.0)LLNW=IDOC(25)
      LLSE=IPP(5,0)
      IF (LLSE.EQ.0)LLSE=IDOC(26)
      INC=IPP(6,0)
      IF (INC.EQ.0)INC=20
      KBUG=IKWP('BUG',1,0)
C     READ IN ENTIRE SET OF GUESS GRIDS
C     GET KEYS TO ESTABLISH SCALING
      IOK=MDKEYS(MDNG,-1,LIST,ISCALE,IUNIT,LOCS)
```

```
      IF (ICK.LT.0)GO TO 920
C     IMPLICIT LOOP OVER TWO TIME PERIODS
      NBEF=0
  100 NGE=NGB+70
      L=0
      DO 10 N=NGB,NGE
      ICK=IGGET(NGFG,N,3200,IGRID(1,23),NR,NC,IGHDT)
C     FOLLOWING NECESSARY BECAUSE OF GRIDS WITH HOLES
      ITVNAM=IGHDT(7)
      ITLEVL=IGHDT(10)
      IF (IOK.EQ.0)GO TO 2
      IF(IOK.LT.-1)GO TO 930
      GO TO 10
    2 CONTINUE
C      ORDER THE GRIDS FOR MDFILE
      NPT=NR NC
      DO 3 I=1,22
      JB=I
      IF (LCHAR(I).EQ.ITVNAM)GO TO 4
      IF (ICHAR(I).EQ.ITVNAM)GO TO 4
      IF (KCHAR(I).EQ.ITVNAM)GO TO 4
      IF (JCHAR(I).EQ.ITVNAM)GO TO 4
    3 CONTINUE
      GO TO 10
    4 DO 5 J=JB,22
      IF (IGOT(J).EQ.1)GO TO 5
      L=J
      IF (IPRESS(J).EQ.ITLEVL)GO TO 6
      IF (MOD(J,15).EQ.1.AND.ITLEVL.EQ.900) GO TO 6
    5 CONTINUE
      GO TO 10
    6 DO 8 J=1,NPT
      IGRID(J,L)=IGRID(J,23)
    8 CONTINUE
      DO 80 I=1,64
   80 IGHD(I)=IGHDT(I)
      NRSAV=NR
      NCSAV=NC
      KOUT(1)=7
      KOUT(2)=IGDAY
      KOUT(3)=IGTIME
      KOUT(4)=IGLEVL
      KOUT(5)=L
      KOUT(6)=NRSAV
      KOUT(7)=NCSAV
      KOUT(8)=IGRID(100,23)
      IF(KBUG.NE.0)CALL OUTINT(KOUT)
C     SET UP FACTOR BETWEEN GRID SCALING AND MD SCALING
      ICON=ISCALE(L+4)-IGVSCA
      SCL=10.**ICON
      FISC(L)=SCL
      IGOT(L)=1
```

```
      NGOT=0
      DO 11 J=1,22
   11 NGOT=NGOT+IGOT(J)
      IF(NGOT.EQ.22) GO TO 12
   10 CONTINUE
      IF (L.EQ.0)GO TO 550
   12 CONTINUE
      CALL SDEST(' PREPARING GUESS FOR DAY ',IGDAY)
      CALL SDEST(' HOUR =  ',IGTIME)
      CALL SDEST(' VALID TIME = ',IGTIMV)
C     GET DATE INFORMATION FROM GRID AND MOVE TO ROW HEADER
      ROW(1)=IGDAY
      ROW(2)=IGTIME
      FGINC=FLOAT(IGINCR)
      LAN=LLNW/1000
      LAS=LLSE/1000
      NLA=(LAN-LAS)*10/INC+1
      LOW=IABS(MOD(LLNW,1000))
      LOE=IABS(MOD(LLSE,1000))
      IF (LOE.GT.LOW)LOE=LOE-360
      IF (LOW.GT.180)LOW=LOW-360
      IF (LOE.GT.180)LOE=LOE-360
      NLO=(LOW-LOE)*10/INC+1
      NLAT=LAN 10000
      JINC=INC 1000
      IF (KBUG.EQ.0)GO TO 45
      CALL SDEST('   IGLAMX  IGLAMN  IGLOMX  IGLOMN  IGINCR ',0)
      KOUT(1)=5
      KOUT(2)=IGLAMX
      KOUT(3)=IGLAMN
      KOUT(4)=IGLOMX
      KOUT(5)=IGLOMN
      KOUT(6)=IGINCR
      CALL OUTINT(KOUT)
      KOUT(2)=LAN
      KOUT(3)=LAS
      KOUT(4)=LOW
      KOUT(5)=LOE
      KOUT(1)=4
      CALL OUTINT(KOUT)
      CALL SDEST(' NLA= ',NLA)
      CALL SDEST(' NLO= ',NLO)
   45 CONTINUE
      IF (IGLOMX.LT.0.AND.IGLOMN.GT.0)IGLOMX=3600000+IGLOMX
      NN=0
      DO 50 N=1,NLA
      IF (NLAT.GT.IGLAMX)GO TO 50
      IF (NLAT.LT.IGLAMN)GO TO 50
      FJ=FLOAT(IGLAMX-NLAT)
      SJ=FJ/FGINC+1.0
      J=SJ
      DJ=SJ-J
```

```
      JP=J+1
      IF (JP.GT.NRSAV)JP=J
      ROW(3)=NLAT
C     WRITE ROW HEADER
      NN=NN+1
      ICK=MDPUT(MDNG,NN,0,IREC)
      IF (IOK.LT.0)GO TO 970
      LON=LOW 10000
      MM=0
      DO 40 M=1,NLO
      IF (LON.GT.IGLOMX)GO TO 40
      IF (LON.LT.IGLOMN)GO TO 40
      FI=FLOAT(IGLOMX-LON)
      SI=FI/FGINC+1.0
      I=SI
      DI=SI-I
      IP=I+1
      IF (IP.GT.NCSAV)IP=I
      L=(I-1)*NRSAV+J
      L2=(IP-1)*NRSAV+J
      L3=(I-1)*NRSAV+JP
      L4=(IP-1)*NRSAV+JP
      MM=MM+1
      IF (NN.NE.1)GO TO 7
      COL=LON
      IF (COL.GT.1800000)COL=COL-3600000
C     WRITE COL HEADER
      ICK=MDPUT(MDNG,0,MM,IREC)
      IF (IOK.LT.0)GO TO 980
C      WRITE DARA
    7 DO 9 K=1,22
      IVAL=IGRID(L,K)
      IF (IVAL.EQ.MISG)GO TO 9
      TL=IGRID(L,K)
      TR=IGRID(L2,K)
      BL=IGRID(L3,K)
      BR=IGRID(L4,K)
      TVAL=TL+DI*(TR-TL)
      BVAL=BL+DI*(BR-BL)
      VAL=TVAL+DJ*(BVAL-TVAL)
      VAL=VAL FISC(K)
      IVAL=VAL
    9 IREC(4+K)=IVAL
C     CHECK ON MISSING 20ME T (HARRIS)
      IF (IREC(18).EQ.MISG)IREC(18)=(IREC(17)+IREC(19))/2
      DO 35 L=1,21
      K=L+4
      IF (IREC(K).EQ.MISG)GO TO 35
      IF (IREC(K).LT.15000)IREC(K)=IREC(K)+27312
   35 CONTINUE
      IF (NBEF.EQ.0)GO TO 30
      ICK=MDGET(MDNG,NN,MM,JREC)
```

```
      DO 28 L=1,22
      K=L+4
      FINC=DT FLOAT(JREC(K)-IREC(K))
      IREC(K)=IREC(K)+FINC
   28 CONTINUE
   30 CONTINUE
C     CHECK FOR CENTIGRADE (HARRIS GRIDS)
      ICK=MDPUT(MDNG,NN,MM,IREC)
      IF (ICK.LT.0)GO TO 990
   40 LON=LON-JINC
   50 NLAT=NLAT-JINC
      IF (NBEF.NE.0)GO TO 65
      NGB=IPP(7,0)
      NBEF=1
      DT=DPP(8,0.)
      DO 60 K=1,22
   60 IGOT(K)=0
      IF (NGB.NE.0)GO TO 100
   65 CALL MDCLOS(MDNG)
      IDOC(33)=IGDAY
      IDOC(34)=IGTIME 100+IGTIMV
      IDOC(38)=MDNG
      IDOC(39)=MDRG
      IDOC(42)=NGFG
      IF (LUC(81).EQ.0)GO TO 70
      CALL DWRITE(LUN,ITERM,IDOC)
      CALL DCLOSE(LUN)
      GO TO 75
   70 CALL TSNIC(3,1,1,1,1,1,IDOC)
   75 CALL SDEST(' DUN       ',0)
      RETURN
  920 CALL SDEST(' UNABLE TO OBTAIN MD KEYS      ',0)
      RETURN
  930 CALL SDEST(' UNABLE TO OPEN GRID FILE NO. ',NGFG)
      RETURN
  950 CALL SDEST(' CANNOT FIND ONE SINGLE LOUSY GRID...NOT ONE',0)
      RETURN
  960 CALL SDEST(' UNABLE TO OPEN MD FILE NO. ',MDNG)
      RETURN
  970 CALL SDEST(' CANNOT WRITE ROW HEADER ',N)
      RETURN
  980 CALL SDEST(' CANNOT WRITE COL HEADER ',MM)
      RETURN
  990 CALL SDEST(' CANNOT WRITE DARA VECTOR ',L)
      RETURN
      END
```

```
//CSVA5660 JOB  CLASS=A,MSGLEVEL=(0,0)
//  EXEC  MCPRG,MOD=CSVA
//  VLCSVA   RUF  02/29/84: MEMBER UPDATED
//  VLCSVA   SEG  12/23/83; ENTERED USER MANUAL CARD
//FORT.SYSIN  DD  *
@PROCESS SC(TGMES,EMES,DMES,NCCD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,MDKEYS,MDGET,MDPUT,WTOR,OPCOM,SGX,SQW)
@PROCESS SC(MOVB,DOPEN,DREAD,DWRITE,MOVC,MOVW,CLEANW)
      SUBROUTINE MAINC
C ?   LOAD SURFACE MD INTO SPECIAL EDIT MD FILE (MDCUT)
C ?   KEYIN: CSVA MDCUT
C ?      KEYWORDS:
C ?          LAT =  MIN AND MAX LATITUDE EXTENT (WHOLE DEGREES)
C ?          LON =  MIN AND MAX LONGITUDE EXTENT (POS W NEG E)
C ?          TIME = HOUR OF SURFACE DATA
C ?          MDF =  MDFILE OF INPUT DATA
C ?          DAY =  DAY OF SURFACE DATA (JULIAN DAY)
C ?          AUTO = FORCE ACCEPTANCE OF FIRST VALID ROW
C ?          SHIP = NON ZERO GOES TO CURRENT SHIP DATA
C ?      KEYWORDS DEFAULT TO VASTEXT PARAMETERS AND CURRENT MD FILE
C   SSEC/MCIDAS USERS MANUAL - CHAP12
      DIMENSION IBUF(100),IDAT(2),IDOC(112),KBUF(14)
      DIMENSION KOUT(10),MDHD(64),ISCL(13),IUNITS(13),LOCS(13)
      CHARACTER*52 LST
      REAL*8 MFILE
      DATA LST/'TYPEDAY TIMEID  LAT LON HMS ZS  T   TD  DIR SPD PSL '/
      DATA MFILE/'VASTEXT '/,MSG/Z80808080/,LUN/20/,LEN/100/,ITRY/0/
      DATA NOUT/0/
      IPRNT=IKWP('BUG',1,0)
      KBUG=IPRNT
C     IF (IPRNT.EQ.0)IPRNT=1
C     CALL TQSET(IPRNT)
C      CHECK ON AVAILABILITY OF OUTPUT FILE
      MDO=IPP(1,0)
      IF (MDOPEN(MDO,2).LT.0)GO TO 960
C     PICK UP DOCUMENTATION RECORD
      IF (LUC(81).EQ.0)GO TO 1
C     FOLLOWING FOR VAS APPLICATIONS
      ITERM=LUC(-20)
      CALL DOPEN(MFILE,LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      GO TO 2
C      FOLLOWING FOR TOVS APPLICATIONS
    1 CALL TSNIO(1,1,1,1,1,1,IDOC)
    2 CONTINUE
C     CHECK STATUS WORD
      IF (IDOC(28).NE.1)GO TO 800
C     FILL IN MD OUTPUT NO
      IDOC(36)=MDO
C     CHECK FOR TIME FORCE
      IHR=IKWP('TIME',1,0)
C     PICK UP TIME FROM DOC
```

```
      ITM=IDOC(2)
      IF(IHR.EQ.0)IHR=ITM/10000
      IDOC(1)=IKWP('DAY',1,IDOC(1))
      IDAY=MOD(IDOC(1),1000)
      IDHR=IDAY 100+IHR
   10 CONTINUE
      NFIL=IKWP('MDF',1,0)
      IF (NFIL.GT.0)GO TO 11
C     ASSUME FILE IS IN CURRENT SECTION
      NFIL=MOD(IDAY,10)
      IF (NFIL.EQ.0)NFIL=NFIL+10
      IF (IKWP('SHIF',1,0).NE.0)NFIL=NFIL+30
   11 IF (KBUG.NE.0)CALL SDEST(' OPENING HOUR NO.',IHR)
      IF(MDINFO(NFIL,MDHD).NE.0)GO TO 970
      IOPN=MDOPEN(NFIL,1)
      IF (IOPN.LT.0)GO TO 970
C     PICK UP KEYS
      IKEY=MDKEYS(NFIL,13,LST,ISCL,IUNITS,LOCS)
      IF (KBUG.EQ.0)GO TO 12
      DO 85 J=1,8
   85 KOUT(J+1)=LOCS(J)
      KOUT(1)=8
      CALL OUTINT(KOUT)
      DO 86 J=1,8
   86 KOUT(J+1)=ISCL(J)
      KOUT(1)=8
      CALL OUTINT(KOUT)
      DO 83 J=1,4
   83 KOUT(J+1)=LOCS(J+8)
      KOUT(1)=4
      CALL OUTINT(KOUT)
      DO 88 J=1,4
   88 KOUT(J+1)=ISCL(J+8)
      KOUT(1)=4
      CALL OUTINT(KOUT)
   12 CONTINUE
C     SET UP DEFAULT LAT/LON LIMITS
      LLNW=IDOC(25)
      LLSE=IDOC(26)
      LAN=LLNW/1000
      LAN=IKWP('LAT',2,LAN)
      MAXLAT=LAN 100
      LAS=LLSE/1000
      LAS=IKWP('LAT',1,LAS)
      MINLAT=LAS 100
      LOW=MOD(LLNW,1000)
      LOW=IABS(LOW)
      IF (LOW.GT.180)LOW=LOW-360
      LOW=IKWP('LON',2,LOW)
      MAXLON=LOW 100
      LOE=MOD(LLSE,1000)
      LOE=IABS(LOE)
```

```
          IF (LOE.GT.180)LOE=LOE-360
          LOE=IKWP('LON',1,LOE)
          MINLON=LOE 100
          CALL SDEST('    MAXLT    MINLT    MAXLN    MINLN',0)
          KOUT(1)=4
          KOUT(2)=MAXLAT
          KOUT(3)=MINLAT
          KOUT(4)=MAXLON
          KOUT(5)=MINLON
          CALL OUTINT(KOUT)
C
C         SET UP INDICES
          JTYP=LOCS(1)
          JDAY=LOCS(2)
          JTIM=LOCS(3)
          JLAT=LOCS(5)
          JLON=LOCS(6)
          JHMS=LOCS(7)
          JELV=LOCS(8)
          JTEM=LOCS(9)
          JDEW=LOCS(10)
          JDIR=LOCS(11)
          JSPD=LOCS(12)
          JPRE=LOCS(13)
          DO 32 I=1,72
C          SEARCH ROW HEADER FOR TIME AND DAY...UNLESS WE
C         ARE FORCING FIRST AVAILABLE ROW AS IN AUTO PROCESSING
          IOK=MDGET(NFIL,I,0,IBUF)
          IF (IOK.NE.0)GO TO 32
          ISV=I
          ITIME=IBUF(JTIM)/10000
          IF (MOD(IBUF(JDAY),1000).NE.IDAY)GO TO 32
          IF (IKWP('ALTO',1,0).NE.0)GO TO 35
       31 IF (IHR.EQ.ITIME)GO TO 35
       32 CONTINUE
          GO TO 900
       35 KBUF(1)=IBUF(JDAY)
          KBUF(2)=IBUF(JTIM)
C         DO NOT PERMIT A ROW VALUE OF ZERO
          IF (ITIME.EQ.0)ITIME=24
C         WRITE OUTPUT ROW HEADER
          IF(KBUG.NE.0)CALL SDEST(' WRITING ROW HEADER FOR ROW ',ITIME)
          IF (KBUG.NE.0)CALL SDEST (' THIS IS MD FILE NO ',MDO)
          IOK=MDPUT(MDO,ITIME,0,KBUF)
          IF (IOK.LT.0)GO TO 940
          NREP=0
          NOUT=0
       40 NREP=NREP+1
C         PICK UP COLUMN HEADERS
          IF (MDHD(8).NE.0)IOK=MDGET(NFIL,0,NREP,IBUF)
          IF (IOK.NE.0)GO TO 50
C         IF (IBUF(JTYP).NE.0)GO TO 40
```

```
      ICK= MDGET(NFIL,ISV,NREP,IBUF)
      IF (IOK.NE.0)GO TO 40
      IDAT(1)=IBUF(JLAT)/100
      IDAT(2)=IBUF(JLON)/100
      IF (IDAT(1).LT.MINLAT.OR.IDAT(1).GT.MAXLAT)GO TO 40
      IF (IDAT(2).LT.MINLON.OR.IDAT(2).GT.MAXLON)GO TO 40
C     SET USER MOD FLAG KBUF(4)
      KBUF(4)=0
      DO 45 K=5,13
      I=LOCS(K)
      IF(I .LT. 0)GO TO 45
      KBUF(K)=IBUF(I)
   45 CONTINUE
C     CHANGE TO TSL,DD,AND ADD Z10
      IF (IBUF(JTEM).EQ.MSG)GO TO 40
      IF (IBUF(JDEW).NE.MSG)KBUF(10)=IBUF(JTEM)-IBUF(JDEW)
C     CHECK REASONABLENESS OF TOPOGRAPHY
      MLAT=IDAT(1)
      MLON=-IDAT(2)
      CALL HRTOPO(MLAT,MLON,IEL,ICH)
      Z=0
C     GIVE SHIP FILE SPECIAL TREATMENT
      IF (IEL.EQ.0.AND.JELV.EQ.-1)GO TO 46
      IF (IABS(IEL-KBUF(8)).GT.1000)GO TO 40
      Z=KBUF(8)
   46 TSL=.01*FLOAT(KBUF(9))+Z*0.0065
      KBUF(9)=100.*TSL
C     FIND Z1000
      IF (IBUF(JPRE).EQ.MSG)GO TO 48
      PSL=0.1*FLOAT(IBUF(JPRE))
      PST=PSL*(1.-.00002256944*Z)**5.256
      ALG=ALOG(1000./PST)
      Z10=Z-29.2898*TSL*ALG/(1.+.09519*ALG)
      KBUF(14)=Z10
   48 CONTINUE
C     OUTPUT RECORD
      NOUT=NOUT+1
      KOUT(2)=IDAT(1)
      KOUT(3)=IDAT(2)
      KOUT(4)=KBUF(14)
      KOUT(5)=KBUF(9)
      KOUT(6)=KBUF(10)
      KOUT(7)=NREP
      KOUT(8)=NOUT
      KOUT(1)=7
      IF (KBUG.NE.0)CALL OUTINT(KOUT)
      IOK=MDPUT(MDC,ITIME,NOUT,KBUF)
      IF (IOK.LT.0)GO TO 940
      GO TO 40
   50 CONTINUE
      IF (NOUT.LT.10)GO TO 900
   53 CALL SDEST(' NO. OF REPORTS WRITTEN=',NOUT)
```

```
C      MDNO DATE TIME MDNC  TO VASTEXT
       IDOC(27)=(IDAY+ITIME) 100
       IDOC(36)=MDC
       IDOC(37)=ITIME
       IF (LUC(81).EQ.0)GO TO 55
       CALL DWRITE(LUN,ITERM,IDOC)
       GO TO 60
    55 CALL TSNIO(3,1,1,1,1,1,IDOC)
    60 CONTINUE
C      REWRITE ROW HEADER WITH NO OF REPORTS ADDED
       KBUF(3)=NCLT
       IOK=MDPUT(MDC,ITIME,0,KBUF)
       IF (IOK.NE.0)GO TO 940
       RETURN
   800 CALL SDEST(' FILE NOT AVAILABLE FOR PROCESSING',0)
       RETURN
   900 ITRY=ITRY+1
       IDHR=IDHR-100
       IHR=IHR-1
       IF (ITRY.LT.3)GO TO 10
       CALL SDEST(' INSUFFICIENT SURFACE DARA AVAILABLE..',0)
       GO TO 53
   940 CALL SDEST(' CANNOT WRITE OUTPUT FILE..',0)
       RETURN
   960 CALL SDEST(' OUTPUT MDFILE NOT DEFINED..',0)
       RETURN
   970 CALL SDEST(' UNABLE TO OPEN INPUT FILE..',0)
       RETURN
       END
```

```
//SRVA5600 JOB CLASS=E,MSGLEVEL=(0,0)
//  EXEC  MCPRG
//  VLSRVA   JPN  03/05/84: MEMBER UPDATED
//  VLSRVA   SBC  12/23/83; ENTERED USER MANUAL CARD
//FORT.SYSIN  DD
@PROCESS SC(TGMES,EMES,DMES,NCCD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(LWOPEN,LWMAKE,LWQLIT,ISFILE,WTOR,CPCOM,SGX,SGW)
@PROCESS SC(ISCHAR,DOPEN,DREAD,DWRITE,ISAN,MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAIN0
C     CALL SEQUENCE
C ** KEYIN
C ? PROGRAM TO PREPARE RETRIEVAL SURFACE GRID FILES
C ?   KEYIN: SRVA <PAR> <NGFS>        KEYWORDS
C ?     POSITIONAL PARMETERS:
C ?         <PAR>   'Z100' 'TSL' OR 'DD'
C ?         <NGFS>   OUTPUT GRID FILE FOR SURFACE DATA
C ?     KEYWORDS
C ?         LAT    MIN AND MAX LATS
C ?         LON    MIN AND MAX LONGITUDES
C ?         INC    INCREMENT IN DEG*10
C ?         SCL    BARNES SCALE FACTOR (DEFAULT 50)
C ?         GSS    GUESS OPTION 'G' TO USE VASGSS
C ?        MDNG    MD FILE FOR VASGSS (DEFAULT VASTEXT)
C ?         ERR    GROSS ERROR TOLERANCE
C ?         RPT    FORCE NO. OF REPORTS
C ?         EDIT   NON ZERO TURNS OFF AUTO EDIT OF DATA
C    SSEC/MCIDAS USERS MANUAL - CHAP12
      COMMON /NAV/FLAT,FLON,VZEN,SZEN,IL,IE,IRAS,IPIC,ITIME,JTIME,JDAY
      COMMON /DCC/IDCC(112)
      COMMON /DIMEN/NROWS,NCOLS
      COMMON/SRNAV/BETAIN,BETDOT,PTIME,INAV
      CHARACTER*8 MFILE
      CHARACTER 12 CGS,CKWP,CPP,CHAR
      DIMENSION KCLT(10),SDL(8),KBUF(200)
      DIMENSION LBUF(33)
      DIMENSION ITG(2400),IDL(2400)
      DIMENSION IUNI(4),DA(999),RW(999),CL(999)
      DIMENSION IREF(999),DAS(999),STLAT(999),STLON(999)
      DIMENSION FLD(2400),WT1(2400),WT2(2400)
C-----DESCRIPTION OF 64-WORD GRID HEADERS
C
      REAL*8 SUM,SUMS,FN
      INTEGER IGHD(64)
      INTEGER IGID(8)
C-----GIVE TOTAL SIZE (WORDS), # ROWS, # COLS. (IGSIZE=IGNR*IGNC)
      EQUIVALENCE (IGSIZE,IGHD(1)),(IGNR,IGHD(2)),(IGNC,IGHD(3))
C-----YYDDD, HHMMSS AND VALID-TIME (IF APPLICABLE) FOR GRID
      EQUIVALENCE (IGDAY,IGHD(4)),(IGTIME,IGHD(5)),(IGTIMV,IGHD(6))
C-----DESCRIPTION OF GRIDDED VARIABLE (IN MD-FILE TERMS):
C-----   NAME, SCALE, AND UNITS
      EQUIVALENCE (IGVNAM,IGHD(7)),(IGVSCA,IGHD(8)),(IGVUNI,IGHD(9))
C-----DESCRIPTION OF VERTICAL LEVEL: VALUE, SCALE, AND UNITS
```

```
      EQUIVALENCE (IGLEVL,IGHD(10)),(IGLSCA,IGHD(11)),(IGLUNI,IGHD(12)
C-----GRIDDED-VARIABLE TYPE: =1 (TIME DIF) 2(TIME AVG) 4 (LEVEL DIF)
C-----   8 (LEVEL AVG) OR ANY SUM OF THE FOREGOING
      EQUIVALENCE (IGVTYP,IGHD(13))
C-----FOLLOWING USED IF PARAMETER IS A VERTICAL (LEVEL) DIF OR AVG
C-----   (SAME SCALE AS IGLEVL)
      EQUIVALENCE (IGLDIF,IGHD(14))
C-----FOLLOWING USED IF PARAMETER IS A TIME DIF OR AVG (HHMMSS)
      EQUIVALENCE (IGTDIF,IGHD(15))
C-----GRID ORIGIN, TYPE (I.E. TYPE OF PROJECTION)
      EQUIVALENCE (IGORG,IGHD(33)),(IGTYPE,IGHD(34))
C-----SUBSEQUENT COORDS (IGLAMX,IGLOMX,IGLAMN,IGLOMN,IGINCR,
C-----   IGPOLR,IGPOLC,IGSP60,IGCLON) ALL HAVE 4 IMPLIED DEC. PLACES.
C-----   LAT GOES FROM -900000 TO 900000, LON GOES FROM -1800000
C-----   TO 1800000 (WEST IS +)
C-----TYPE 1 GRIDS ARE PSEUDO-MERCATOR
      EQUIVALENCE (IGLAMX,IGHD(35)),(IGLOMX,IGHD(36)),(IGLAMN,IGHD(37))
     *,(IGLOMN,IGHD(38)),(IGINCR,IGHD(39))
C-----TYPE 2 GRIDS ARE POLAR-STEREOGRAPHIC
C-----GIVE ROW # OF NORTH POLE, COL # OF N.P., COL SPACING AT 60 DEG N.
C-----   (DEG), LONGITUDE PARALLEL TO COLUMNS (DEG)
      EQUIVALENCE (IGPOLR,IGHD(35)),(IGPOLC,IGHD(36)),(IGSP60,IGHD(37))
     *,(IGCLON,IGHD(38))
C-----INITIALS OF USER AND PROJECT # UNDER WHICH GRID CREATED
      EQUIVALENCE (IGUSER,IGHD(41)),(IGPROJ,IGHD(42))
C-----CHARACTER ID SUPPLIED BY PROGRAM (ARBITRARY)
      EQUIVALENCE (IGID,IGHD(43))
C
C
      DATA MISS/Z80808080/,NMOD/4/,NSIZE/2400/,NBMAX/999/
      DATA LBUF/33*Z40404040/,IBLNK/Z40404040/
      DATA SCL/0.,0.,3.,5.,2*10.,3.,30./
C               X  X   T TD   U,V  TS   Z
      DATA MFILE,LUN,LEN/'VASTEXT ',20,100/
      DATA IUNI/'    ','Z   ','T   ','DD  '/,LPR/'MSL '/
      CALL SDEST(' BEGIN SFCPRG',0)
      IF (LUC(81).EQ.0)GO TO 1
C     FOLLOWING FOR VAS APPLICATIONS
C     PICK UP CONTEXT INFORMATION
      ITERM=LUC(-20)
      CALL DOPEN(MFILE,LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      GO TO 2
C     FOLLOWING FOR TOVS APPLICATIONS
    1 CALL TSNIC(1,1,1,1,1,1,IDOC)
    2 MDNS=IDOC(36)
      MDRS=IDOC(37)
      MDNG=IDOC(38)
      MDNG=IKWP('MDNG',1,MDNG)
      NGRFS=IPP(2,0)
      KBUG=IKWP('BUG',1,0)
      CGS=CKWP('GSS',1,' ')
```

```
      NRPT=0
      IF (MDOPEN(MDNS,2).LT.0)GO TO 920
C      GET NO. OF REPORTS FROM ROW HEADER
      ICK=MDGET(MDNS,MDRS,0,KBUF)
      ISYD=KBUF(1)
      IHMS=KBUF(2)
      IF (ICK.LT.0)GO TO 930
      NRPT=KBUF(3)
      NRPT=IKWP('RPT',1,NRPT)
    3 CALL SDEST(' TOTAL NO. REPORTS = ',NRPT)
C     CHECK OPERATOR FORCED BOUNDARIES
      LLNW=IDOC(25)
      LLNW=IDOC(25)
      LLSE=IDOC(26)
      LN=LLNW/1000
      LS=LLSE/1000
      LW=MOD(LLNW,1000)
      LE=MOD(LLSE,1000)
      L=IKWP('LAT',1,0)
      IF (L.NE.0)LS=L
      L=IKWP('LAT',2,0)
      IF (L.NE.0)LN=L
      L=IKWP('LON',1,0)
      IF (L.NE.0)LE=L
      L=IKWP('LON',2,0)
      IF (L.NE.0)LW=L
      LINC=IKWP('INC',1,0)
      DINC=.1*FLOAT(LINC)
C     USE  IDATA AS FLAG IN EVENT OF NO DATA
      IDATA=77777
      IP=0
      CHAR=CPP(1,' ')
      IF (CHAR.EQ.('TSL '))IP=3
      IF (CHAR.EQ.('DD  '))IP=4
      IF (CHAR.EQ.('U   '))IP=5
      IF (CHAR.EQ.('V   '))IP=6
      IF (CHAR.EQ.('TS  '))IP=7
      IF (CHAR.EQ.('Z100'))IP=8
      IF (IP.EQ.0)GO TO 500
C     SET OPTIONAL GROSS ERROR CHECK
      SDL(IP)=DKWP('ERR',1,SDL(IP))
C     INITIALIZE ANALYSIS FIELD
      DO 10 L=1,NSIZE
      IDL(L)=0
   10 ITG(L)=0
      ROWS=LN-LS
C     AVOID SO HEMISPHERE PROBLEMS
      LW=IABS(LW)
      LE=IABS(LE)
      IF (LW.GT.180)LW=LW-360
      IF (LE.GT.180)LE=LE-360
      IF (LW.LT.LE)LW=LW+360
```

```
      COLS=LW-LE
      PTS=(ROWS+1.)*(COLS+1.)
C     ESTABLISH GRID INCREMENT IN 10THS OF DEGREES
      FN=FLOAT(NSIZE)
      IF (LINC.EQ.0)DINC=SQRT(PTS/FN)
      INC=DINC 10.+0.5
      LAN=LN*10
      LAS=LS 10
      LOW=LW 10
      LOE=LE 10
   11 DINC=INC
      NROWS=(LAN-LAS)/INC+1
      NCOLS=(LOW-LOE)/INC+1
      CALL SDEST('    NROWS    NCOLS     INC',0)
      KOUT(1)=3
      KOUT(2)=NROWS
      KOUT(3)=NCOLS
      KOUT(4)=INC
      CALL OUTINT(KOUT)
      NPTS=NROWS NCOLS
      IF (LINC.NE.0)GO TO 13
      IF (NPTS.LE.NSIZE)GO TO 13
      INC=INC+1
      GO TO 11
   13 CONTINUE
      TLAT=LN
      SLAT=LS
      WLON=LW
      ELON=LE
      DINC=0.1 DINC
      LP=IP
      IF (LP.EQ.8)LP=2
C      USE LP IN INSTANCES WHERE GSS AND DOC ARE INVOLVED
      IF (LP.EQ.7)LP=3
C     CHECK GUESS OPTION..FILL FROM 'VASGSS'
      ICGS=LIT(CGS)
      IF (CGS.NE.('G   '))GO TO 120
      CALL SRGSS(IDL,ITG,NCOLS,NROWS,TLAT,WLON,DINC,LP,MDNG)
      DO 109 N=1,NPTS
  109 ITG(N)=IDL(N)
      GO TO 106
  120 CONTINUE
      IF (CGS.EQ.' ')GO TO 110
C      EXPLOIT GRIDFILE GUESS OPTION
      NGRF=ICGS/100
      NGR=MOD(ICGS,100)
      IOK=IGOPEN(NGRF,IFIL1)
      IF (IOK.NE.0)GO TO 995
      IOK=IGGET(NGRF,NGR,2400,IDL,NR,NC,IGHD)
      IF (NR.NE.NROWS.OR.NC.NE.NCOLS)GO TO 990
      IF (LP.NE.2)GO TO 106
      DO 105 N=1,NSIZE
```

```
      IDL(N)=IDL(N)*10
      ITG(N)=IDL(N)
  105 CONTINUE
  106 IF (IKWP('NCDAT',1,0).NE.0)GO TO 150
      FCK=SDL(IP)*3.
      ITOSS=FCK
      IF (KBUG.NE.0)CALL SDEST(' TOSSOUT IS ',ITOSS)
C     ABOVE IS GROSS ERROR CHECK FOR GUESS FIELDS
C     THIS CHECK IS PERFORMED ON 1ST ITERATION ONLY
  110 NB=0
      SUM=0.
      SUMS=0.
      DO 108 N=1,NPTS
  108 FLD(N)=.01*FLOAT(ITG(N))
C     NOTE..ITG IS USED TO SAVE ORIGINAL GUESS FOR ITERATION
      DO 148 N=1,NRPT
      IOK=MDGET(MDNS,MDRS,N,KBUF)
      IF (KBUF(NMOD).EQ.MISS)GO TO 148
      IF (IOK.LT.0)GO TO 148
      LAT=KBUF(5)
      LON=KBUF(6)
      FLAT=FLOAT(LAT)*.0001
      IF (FLAT.GT.TLAT.OR.FLAT.LT.SLAT)GO TO 148
      FLATJ=(TLAT-FLAT)/DINC+1.0
      FLON=FLOAT(LON)*.0001
      IF (FLON.GT.WLON.OR.FLON.LT.ELON)GO TO 148
      FLONI=(WLON-FLON)/DINC+1.0
      IF (KBUF(IP+6).EQ.MISS)GO TO 148
      TSL=.01*FLOAT(KBUF(9))
      DD=.01*FLOAT(KBUF(10))
      Z=KBUF(8)
C     ALL VARIABLES ARE IN TRUE UNITS
      IF (IP.EQ.7)FDAT=TSL-0.0065*Z
      IF (IP.EQ.4)FDAT=DD
      IF (IP.EQ.3)FDAT=TSL
      IF (IP.EQ.8)FDAT=KBUF(14)
      IF (CGS.NE.'G')GO TO 330
      IF (FCK.EQ.99999.)GO TO 330
C     GET VALUE FROM GUESS GRID
      CALL VALUE(LAN,LAS,LOW,LOE,INC,IDL,100.,VAL)
      IF (ABS(VAL-FDAT).LT.FCK)GO TO 330
      IVAL=FDAT
      IF (KBUG.NE.0)CALL SDEST(' REJECT DATA VALUE ',IVAL)
      KBUF(IP+6)=MISS
      IOK= MDPUT(MDNS,MDRS,N,KBUF)
      GO TO 148
  330 IDAT=FDAT*100.
      NB=NB+1
C     FILL BARNES ARRAYS
      SUM=SUM+FDAT
      SUMS=SUMS+FDAT*FDAT
      DAS(NB)=FDAT
```

```
          IREP(NB)=N
          STLAT(NB)=FLAT
          STLON(NB)=FLON
          DA(NB)=FDAT
          RW(NB)=FLATJ
          CL(NB)=FLONI
          IDATA=0
          IF (NB.LT.NBMAX) GO TO 148
          CALL SDEST(' TOO MUCH DARA...EXITING TO ANALYSIS',0)
          GO TO 140
      148 CONTINUE
      140 IF (IDATA.EQ.77777.AND.CGS.NE.('G    '))GO TO 400
          IF (IDATA.EQ.77777)GO TO 150
C         SET UP FOR BARNES ANLAYSIS
          IGB=0
          IF (CGS.NE.' ')IGB=LIT('GRD ')
          ISCL=1000/INC
C         SET BARNES SCALING
          ISCL=IKWP('SCL',1,ISCL)
C         CALL SDEST(' BEGIN BARNES WITH SCALING OF ',ISCL)
          CALL FBARN(TLAT,NROWS,NCOLS,FLD,WT1,WT2,DAS,RW,CL,
          NB,DINC,ISCL,IGB)
          DO 142 N=1,NPTS
C          AVOID NEGATIVE DEWPOINT DEPRESSION
          IF (IP.EQ.4.AND.FLD(N).LT.0.)FLD(N)=0.
      142 IDL(N)=100.*FLD(N)
C         CHECK ON THE EDIT OPTION
          IF (IKWP('EDIT',1,0).NE.0)GO TO 150
          FCK=999999.
          FN=NB
          SUM=SUM/FN
          SUMS=SUMS/FN
          SD=DSQRT(SUMS-SUM*SUM)
          ISD=SD*100.
          IF (KBUG.NE.0)CALL SDEST('DARA SD IS ',ISD)
C         LIMIT SIZE OF STANDARD DEVIATION
          SD=AMIN1(SD,SDL(IP))
          NOUT=0
          DO 130 NN=1,NB
          FLAT=STLAT(NN)
          FLON=STLON(NN)
C         GET VALUE FROM GRID
          CALL VALUE(LAN,LAS,LOW,LOE,INC,IDL,100.,VAL)
          IF (VAL.EQ.999999.)GO TO 130
          NVAL=VAL
          DAT =DA(NN)
          DIF=DAT-VAL
C         GROSS ERROR CHECK
C         ***** * *** *****
          ADIF=ABS(DIF)
          IF(ADIF.LT.SD) GO TO 130
          ICK=MDGET(MDNS,MDRS,IREP(NN),KBUF)
```

```
      NOUT=NOUT+1
      KBUF(IP+6)=MISS
      ICK=MDPUT(MDNS,MDRS,IREP(NN),KBUF)
      IF (KBUG.EQ.0)GO TO 130
      CALL ENKODE('(1X,I5,2F7.2,2F10.1/)',LBUF,NN,FLAT,FLON,DAT,VAL)
  130 CONTINUE
      IF (NOUT.GT.0)GO TO 110
  150 CONTINUE
C     OUTPUT FIELD
      IGVNAM=LIT(CHAR)
      IGVSCA=2
      IGVUNI=IUNI(IP)
      IGLEVL=LPR
      IGLSCA=2
      IGLUNI=1
      IGVTYP=8
      IGLDIF=1
      IGORG=0
      IGTYPE=1
      IGLAMX=LAN 1000
      IGLAMN=LAS 1000
      IGLOMX=LOW 1000
      IGLOMN=LOE 1000
      IGINCR=INC 1000
      IGDAY=MOD(ISYD,100000)
      IGTIME=IHMS
      IGSIZE=NPTS
      IGNR=NROWS
      IGNL=NCOLS
      IOK=IGOPEN(NGRFS,IFIL2)
      IF (IOK.NE.0)GO TO 995
      ICK=IGPUT(NGRFS,NGRNO,IDL,NROWS,NCOLS,IGHD,ISTAT)
      CALL TGMES(' ANALYSIS FILED AS GRID NO.$',ISTAT)
      IDOC(29)=NGRFS
      IDOC(28+LP)=ISTAT
      IF (LUC(81).EQ.0)GO TO 280
      CALL DWRITE(LUN,ITERM,IDOC)
      CALL DCLOSE(LUN)
      RETURN
  280 CALL TSNIC(3,1,1,1,1,1,IDOC)
      RETURN
  400 CALL SDEST(' NO DATA AVAILABLE FOR IMAGE',0)
      RETURN
  500 CALL SDEST(' IMPROPER PARAMETER KEYED',0)
      RETURN
  700 CALL SDEST(' (Z,T,D)  GRID  LLNW  LLSE  INC 10  G  GE  BSC',0)
      CALL SDEST(' GRID IS GRID FILE NO. FOR SFC GRIDS',0)
      CALL SDEST(' G=GUESS (0=NO GUESS,G=USE VASGSS,NNN=GRID NO..',0
      CALL SDEST(' GE=GROSS ERROR OVERRIDE (M 10,DEG*100)..',0)
      CALL SDEST(' BSC BARNES SCALE',0)
      RETURN
  920 CALL SDEST(' UNABLE TO OPEN MD DATA FILE NO.',MDNS)
```

```
      IF (CGS.EQ.'G')GO TO 3
      RETURN
  930 CALL SDEST(' NO DATA AVAILABLE FOR HOUR',MDRS)
      IF (CGS.EQ.'G')GO TO 3
      RETURN
  990 CALL SDEST(' GUESS GRID NOT COMPATIBLE..',0)
      RETURN
  995 CALL SDEST(' UNABLE TO OPEN GRID FILE..',0)
 1000 RETURN
      END
```

```
//XRVA7000 JOB CLASS=A,MSGLEVEL=(0,0)
//  VLXRVA   SBG  12/23/93: ENTERED USER MANUAL CARD
// EXEC MCPRG,MOD=XRVA
//FORT.SYSIN DD
      SUBROUTINE MAIN0
C ?  * MODE 1
C ?   EDIT VALUES IN MDFILE FOR REPORT AT CURSOR LOCATION
C ?       HIT SPACE TO DELETE. KEYIN "R END" TO TERMINATE
C ?   MD LINKAGE THRU MDSET AND VASTEXT FILE..THUS RESTRICTED TO
C ?   RETRIEVAL AND SURFACE MD FILES
C ?
C ?     KEYIN: XRVA <PARM> <LEVEL>
C ?       POSITIONAL PARAMETERS:
C ?           PARM CHARACTER KEYS FROM SCHEMA
C ?           LEVEL   PRESSURE LEVEL(IF UNSPECIFIED ALL LEVELS DELETED
C ?         WITH NO PARAMETERS ENTIRE REPORT IS DELETED
C ?  * MODE 2
C ?   EDIT SINGLE RETRIEVAL BY NUMBER
C ?     KEYIN: XRVA NRET,NN
C ?           NN IS NO. OF RETRIEVAL
C    SSEC/MCIDAS USERS MANUAL - CHAP12
      DIMENSION MF(64),IOUT(300)
      ,IRMAX(40),IRMIN(40),IPMAX(40),IPMIN(40)
      ,ILAMIN(40),ILOMIN(40)
      ,ILAMAX(40),ILOMAX(40)
      DIMENSION LIST(300),ISCL(300),IUN(300),LOCS(300)
      DIMENSION IADD(20),MDHD(64),FLA(4),FLO(4),KOUT(10)
      INTEGER 4 CBUF(20)
      REAL*8 DLIT
      CHARACTER 8 MFILE
      CHARACTER 12 CLIT,ICHR,CPP
      COMMON /DOC/IDOC(100)
      COMMON /NAV/FLAT,FLON,ZENLOC,SZEN,IL,IE,IRAS,IPIC,IHMS,JT,JD
      COMMON /IDENT/IYMD,JHMS,NROWM,NSAT
      COMMON /ANALS/NOAN,LTOP
      COMMON /ORBIT/NODE
      COMMON /TIGHT/ITOL
      COMMON /ENTRY/INIT
      COMMON /ORIENT/YCOORD,XCOORD
      COMMON /FLOONV/BETAIN,BETOCT,INAV,PTIME
      DATA MFILE,LUN,LEN/'VASTEXT ',20,100/
      DATA IOUT /300 Z80808080/
      DATA MISS/Z80808080/,NCOLS/56/
      MDNO=LUC(5)
      IWRM=LUC(-5)
      IFRM=LUC(-1)
C     FORCE HIGH RESOLUTION SRCH
      ITOL=1
      IF (LUC(81).EQ.0)GO TO 120
C     READ VASTEXT CONTEXT FILE
      ITERM=LUC(-20)
      CALL DOPEN(CLIT(MFILE),LUN,LEN)
```

```
      CALL DREAD(LUN,ITERM,IDOC)
C     INITIALIZE NAVIGATION
      IRAS=LUC(-11)
      IPIC=LUC(-12)
      CALL TVSAT(IFRM,IRAS,IPIC,IL,IE,JS,JD,JT)
      JD=JS*100000+JD
      CALL NVINIT(BETAIN,BETOCT,INAV,PTIME)
      GO TO 130
  120 CALL TSNIC(1,1,1,1,1,1,IDOC)
      NODE=IDOC(7)
      INIT=1
  130 CONTINUE
      DO 1 K=36,41,2
      MDR=IDOC(K+1)
      IF (MDNO.EQ.IDOC(K))GO TO 2
    1 CONTINUE
      GO TO 900
    2 IF (MDOPEN(MDNO,2).NE.0)GO TO 900
      IF (MDINFO(MDNO,MDHD).NE.0)GO TO 900
C     CHECK FOR INDIVIDUAL DELETION
      NRET=IKWP('NRET',1,0)
      IF (NRET.EQ.0)GO TO 99
      IF (MDHD(8).NE.0)IOK=MDPUT(MDNO,0,NRET,IOUT)
      IOK=MDPUT(MDNO,MDR,NRET,IOUT)
      IF (IOK.NE.0)GO TO 906
      CALL SDEST(' PURGED SOUNDING NO. ',NRET)
      GO TO 110
   99 CONTINUE
C     READ ROW HEADER RECORD
      IOK=MDGET(MDNO,MDR,0,IOUT)
      IF (IOK.LT.0)GO TO 902
      MREC=IOUT(3)
C     SAVE TOTAL TO DECREMENT AND UPDATE VASTEXT
      NREC=MREC
C     READ TEST RECORD
      M=0
  101 M=M+1
      IF (M.GT.100)GO TO 904
      IF(MDHD(8).NE.0)IOK=MDGET(MDNO,0,M,IOUT)
      IOK=MDGET(MDNO,MDR,M,IOUT)
      IF (IOK.LT.0)GO TO 101
      KBUG=IKWP('BUG',1,0)
      IF (KBUG.NE.0)CALL SDEST(' OPERATING WITH ROW= ',MDR)
C     READ IN KEYS
      NKEYS=MDKEYS(MDNO,-1,LIST,ISCL,IUN,LOCS)
C     FIND LAT/LON ADRESS IN KEYLIST
      DO 11 N=1,NKEYS
      IF (LIST(N).EQ.LIT('LAT '))NLAT=N
      IF (LIST(N).EQ.LIT('LON '))NLON=N
      IF (LIST(N).EQ.LIT('MOD '))NMOD=N
   11 CONTINUE
      IF (KBUG.NE.0)CALL SDEST(' NO OF KEYS IS ',NKEYS)
```

```
      CALL GETERM(IFRM,MF)
      MAG=MF(10)
      IF (MAG.LT.6)MAG=6
      CALL INITPL(IWRM,0)
      NX=0
      ICHR=CPP(1,' ')
      IF (ICHR.EQ.' ')GO TO 10
C     FALL THRU FOR SELECTIVE DELETION
      ILEV=IPP(2,0)*10
C     LEVELS IN MD FILES ARE MB 10
      IF (ILEV.NE.0)GO TO 4
C     SET UP TABLE TO DELETE ALL VALUES OF CHARACTER
      K=0
      DO 3 N=1,NKEYS
      IF (ICHR.NE.CLIT(LIST(N)))GO TO 3
      K=K+1
      IADD(K)=N
    3 CONTINUE
      IF (K.EQ.0)GO TO 920
      KTOT=K
      GO TO 8
    4 CONTINUE
C     CHECK ON SPECIFIC LEVEL AND CHARACTER
      DO 5 N=1,NKEYS
      IF (LIST(N).NE.LIT('P      '))GO TO 5
      I=LOCS(N)
      IF (ILEV.NE.ICUT(I))GO TO 5
      NS=N+1
      GO TO 6
    5 CONTINUE
      GO TO 930
    6 KTOT=1
      DO 7 N=NS,NKEYS
      IADD(KTOT)=LOCS(N)
      IF (ICHR.EQ.CLIT(LIST(N)))GO TO 8
    7 CONTINUE
      GO TO 920
    8 IF (KBUG.NE.0)CALL SDEST(' VARIABLE ADDRESS IS ',IADD(1))
   10 CONTINUE
      IF(ICURG(IRS,IPC,IRINC,IPINC).NE.0)GO TO 20
      NX=NX+1
      CALL WRBOX(IRS,IPC,IRINC,IPINC,2)
      IRAS=IRS-IRINC/2
      IPIC=IPC-IPINC/2
      IRMIN(NX)=IRAS
      IPMIN(NX)=IPIC
      CALL FLOC
      FLA(1)=FLAT
      FLO(1)=FLON
      IRAS=IRS+IRINC/2
      CALL FLOC
      FLA(2)=FLAT
```

```
      FLO(2)=FLON
      IPIC=IPIC+IPINC/2
      IRMAX(NX)=IRAS
      IPMAX(NX)=IPIC
      CALL FLOC
      FLA(3)=FLAT
      FLO(3)=FLON
      IRAS=IRS-IRINC/2
      CALL FLOC
      FLA(4)=FLAT
      FLO(4)=FLON
      FLAMAX=-999999.
      FLAMIN=999999.
      FLOMAX=-999999.
      FLOMIN=999999.
      DO 13 N=1,4
      IF (FLA(N).GT.FLAMAX)FLAMAX=FLA(N)
      IF (FLO(N).GT.FLOMAX)FLOMAX=FLO(N)
      IF (FLA(N).LT.FLAMIN)FLAMIN=FLA(N)
      IF (FLO(N).LT.FLOMIN)FLOMIN=FLO(N)
   13 CONTINUE
      KOUT(1)=4
      KOUT(2)=FLAMAX*10
      KOUT(3)=FLAMIN*10.
      KOUT(4)=FLOMAX*10.
      KOUT(5)=FLOMIN*10.
      IF (KBUG.NE.0)CALL CUTINT(KOUT)
      ILAMAX(NX)=FLAMAX*10000.
      ILOMAX(NX)=FLOMAX*10000.
      ILAMIN(NX)=FLAMIN*10000.
      ILOMIN(NX)=FLOMIN*10000.
C     NAV COMPLETE
      CALL ENDPLT
      IF(NX.EQ.40) GO TO 30
      GO TO 10
   20 IF(NX.NE.0) GO TO 30
      CALL SDEST('-*** NOTHING TO DELETE *** ',0)
      CALL ENDPLT
      RETURN
   30 CALL SDEST('  BEGINNING DELETIONS FROM FILE ... ',0)
      IF (LUC(81).NE.0)GO TO 31
C     SET UP AREA FOR IMAGE SPACE TO PLOT
      CALL TVSAT(IFRM,005,335,LTOP,LELE,ISS,ID,IT)
      CALL TVSAT(IFRM,495,335,LBOT,LELE,ISS,ID,IT)
      IF (LTOP.LT.1)LTOP=1
      IF (LBOT.GT.NROWM)LBOT=NROWM
      NROWS=LBOT-LTOP+1
      NPTS=NROWS*NCOLS
   31 DO 70 M=1,MREC
      IF (MDHD(8).NE.0)ICK=MDGET(MDNO,0,M,IOUT)
      ICK=MDGET(MDNO,MDR,M,IOUT)
C     SKIP SOUNDINGS ALREADY DELETED
```

```
      IF (IOUT(NMOD).NE.0)GO TO 70
      DO 40 K=1,NX
C     DELETE EVERYONE WITHIN CURSOR
      IF (IOUT(NLAT).LT.ILAMIN(K).OR.IOUT(NLAT).GT.ILAMAX(K))GO TO 4
      IF (IOUT(NLON).LT.ILOMIN(K).OR.IOUT(NLON).GT.ILOMAX(K))GO TO 4
C     THIS SOUNDING MUST GO
C     LOCATE RASTER AND PIXEL OF SOUNDING
      FLAT=IOUT(NLAT)*.0001
      FLON=IOUT(NLON)*.0001
      IF (LUC(81).EQ.0)GO TO 32
C     REVERT TO -W+E CONVENTION
      FLON=-FLON
      CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON,2,INAV,BETAIN,BETDOT,0.)
      IL=FLIN+0.5
      IE=FELE+0.5
      GO TO 33
C     USE TOVS NAVIGATION ROUTINE
   32 LATS=FLAT 100.
      LONGS=FLON 100.
      CALL SRCH(LATS,LONGS,IM,IL,IE,NPTS,NROWS)
      IF (IM.EQ.0)GO TO 40
C     GUARD AGAINST OVERLAP IN SRCH TO AVOID DUPLICATION
      IL=IL+LTOP-1
      IF (IL.LT.LTOP.OR.IL.GE.LBOT)GO TO 40
   33 CALL SATTV(IFRM,IL,IE,IRAS,IPIC,JS,JD,JT)
      IF (IRAS.LT.IRMIN(K).OR.IRAS.GT.IRMAX(K))GO TO 40
      IF (IPIC.LT.IPMIN(K).OR.IPIC.GT.IPMAX(K))GO TO 40
      CALL WRMAR(IRAS,IPIC,MAG,1,0)
      KOUT(1)=4
      KOUT(2)=IL
      KOUT(3)=IE
      KOUT(4)=ILALO(FLAT)
      KOUT(5)=ILALO(FLON)
      IF (KBUG.NE.0)CALL OUTINT(KOUT)
      GO TO 50
   40 CONTINUE
      GO TO 70
   50 IF (ICHR.NE.' ')GO TO 55
      IOUT(NMOD)=999999
      GO TO 60
C     REMOVE SELECTED VALUES
   55 DO 58 N=1,KTCT
      I=IADD(N)
      IOUT(I)=MISS
   58 CONTINUE
   60 CONTINUE
      IOK=MDPUT(MDNC,MDR,M,IOUT)
      IF (IOK.NE.0)GO TO 906
   70 CONTINUE
  100 CALL ENDPLT
      IF(LUC(81).NE.0)CALL DCLOSE(LUN)
  110 CALL SDEST(' ALL DONE DELETING ...',0)
```

```
      RETURN
  900 CALL SDEST(' CANNOT OPEN MDFILE NO. ',MDNO)
      RETURN
  902 CALL SDEST(' CANNOT LOCATE ROW NO. ',MDR)
      RETURN
  904 CALL SDEST(' TROUBLE READING DATA RECORD NO ',M)
      RETURN
  906 CALL SDEST(' TROUBLE WRITING DATA RECORD NO ',M)
      RETURN
  920 CALL SDEST(' REQUESTED PARAMETER DOES NOT EXIST IN SCHEMA',0)
      RETURN
  930 CALL SDEST(' CANNOT LOCATE DATA FOR LEVEL ',ILEV)
      RETURN
      END
```

```
//VTP25670 JOB CLASS=P,MSGLEVEL=(0,0)
//  VLVTPZ   S   05/03/84: MEMBER UPDATED
//  VLVTPZ   HMW  05/01/84; LATEST OPERATIONAL ... FORMERLY 'VTWZ'
//  VDVTWZ   HMW  04/10/84; ADD 'AUTO' FOR PROCESSING W/O IMAGE
//  VDVTWZ   HMW  03/30/84; OBTAIN 'BEG' & 'END' FROM VASTEXT
// EXEC MCPRG,MOD=VTPZ
//FORT.SYSIN DD
      SUBROUTINE MAIN0
C ?   VAS TOTAL-PRECIP-H2O + T(P) & W(P) RETRIEVAL (H.M.WOOLF)
C ?      KEYIN: VTPZ <KEYWORDS>
C ?         KEYWORDS (DEFAULTS IN PARENTHESES):
C ?            AUTO=RUN WITHOUT IMAGE, GET ALL PARAMS FROM VASTEXT (0)
C ?            ARA=NUMBER OF FIRST DIGITAL AREA FOR IMAGING (0)
C ?            ARP=NUMBER OF SECOND DIGITAL AREA FOR IMAGING (0)
C ?               'ARA' IS FOR TOTAL PRECIPITABLE WATER VAPOR
C ?               'ARP' IS FOR TOTAL-TOTALS
C ?             NOTE ... IF 'ARA' IS ZERO, 'ARP' IS FORCED TO BE ZERO
C ?            BOX=N (SIZE IN FOV'S OF SQR BOX (11)) *EVEN-LD,ODD-SD*
C ?            SPC=IL IE (LINE AND ELEMENT SPACING OF BOXES (BOX BOX))
C ?            BEG=IL IE(FIRST LINE AND ELEMENT TO DEFINE AREA (VASTEXT))
C ?            END=LL LE (LAST LINE AND ELEMENT TO DEFINE AREA (VASTEXT))
C ?            GAM=100 (NORMAL CONDITIONING)
C ?            GSS=C (CLIM) G (GRID,DEFAULT)
C ?            SFC=1 (NO SURFACE ANALYSIS (0))
C ?            TER=N (TERMINAL NUMBER (LOCAL))
C ?            BUG=DIAGNOSTICS: 1=SCREEN, 2=PRINTER (0)
C ?        THE FOLLOWING POINTERS MUST BE SET WITH 'SPVA':
C ?           MDNR,MDRR,NRET ... ALWAYS
C ?           MDNG,MDRG ... FOR GRID GUESS
C ?           MDNS,MDRS,NGFG,NGFS,ZGRID,TGRID,DGRID ... FOR SFC ANALYSIS
      DIMENSION IRET(246),IRETD(246),IAMES(8),IARET(660),IABUF(165),
     1 IAREA(2),IAREC(2)
      DIMENSION LBUF(33),NSPIN(13),ERM(13),VDAT(13),DTB(12),COEF(9)
      DIMENSION Z(40),U(40),TG(40),WG(40),TGS(20),DGS(11),
     1 TD(40),PST(15),TST(15),TDST(15),DIR(15),SPD(15),STABIL(12),
     2 LCHR(20),LMD(20)
      EQUIVALENCE (IRET(17),IRETD(1))
C     EQUIVALENCE TO IRET(ROWLEN-1)
C     ABOVE TO FACILITATE CHANGING LENGTH OF ROW HEADER
      INTEGER 2 IARRAY(11,11,60,2)
      CHARACTER 12 CGES,CKWP,CBLNK,CLETC,CLETG
      COMMON/ARENT/IDIR(64)
      COMMON/ATMOS/P(40),T(40),W(40)
      COMMON/AUTO/IBOX,ILL,ILL,IEL,IER,MAG
      COMMON/DANGLE/VLATS,VLONS
      COMMON/DBUG/LBUG
      COMMON/DOC/IDOC(100)
      COMMON/FILT/TDAT(13,11,11),IFLAG(2,11,11)
      COMMON/FUNC/PHI(40,5),XIT(9,12),ERS(12),TBC(10),NCH,NFT,NFW
      COMMON/GAM/GAMRET,ERTSFC,ERWSFC
      COMMON/GDE/GV(12),DV(12),EV(12)
      COMMON/GUESS/TGES(15),DGES(6)
```

```
      COMMON/LAST/LASLIN,LASELE,LELEV,ICHAR
      COMMON/MODE/IDET(13),ISPIN(13)
      COMMON/NAV/VLAT,VLON,VZEN,SZEN,IL,IE,IRAS,IPIC,ITIME,JTIME,JDAY
      COMMON/RADV/VRAD(13)
      COMMON/SDOC/ISDOC(6)
      COMMON/SIZE/NEXS
      COMMON/SURF/IZ10,ITSFC,IDSFC,IPSTA,IELEV,LSTA
      COMMON/TERMN/ITERM
      DATA DIR/15 0./,SPD/15 0./
      DATA IRET/246*Z80808080/,NSIZE/245/
      DATA LMD
     */0,40,39,38,37,36,35,34,31,28,26,25,24,23,20,18,16,15,13,11/
      DATA LCHR/'SFC ','1000','950 ','920 ','850 ','780 ','700 ',
     *'670 ','500 ','400 ','300 ','250 ','200 ','150 ','100 ',
     *'70  ','50  ','30  ','20  ','10  '/
      DATA CBLNK/' '/,CLETC/'C'/,CLETG/'G'/
      DATA GRAV/980.665/,TCTC/347./,NL/40/
      DATA VMISG/999999./,MISG/Z80808080/
      DATA MAXARA/9999/,MAXWV/255/
C *****
C     FOLLOWING IS PROGRAM VERSION DATE ... KEEP CURRENT!!!
      IVER=84123
      MCINIT=0
      IF(LUC(-25).EQ.1) MCINIT=LUC(-23)
      CALL CALDAY(IVER,IVY,IVM,IVD,IVMO)
      CALL ENCODE('(132X,T1,"BEGIN * VTPZ * VERSION OF ",
     * I2,1X,A4,I2," AT INIT ",I2/)',LBUF,IVD,IVMO,IVY,MCINIT)
C *****
C   LOOK FOR DIGITAL AREA NUMBER
      NARA=IKWP('ARA',1,0)
      IF(NARA.GT.MAXARA) GO TO 1280
      IAREA(1)=NARA
      NARA=IKWP('ARP',1,0)
      IF(NARA.GT.MAXARA) GO TO 1280
      IAREA(2)=NARA
      IF(IAREA(1).EQ.0) IAREA(2)=0
      NAREAS=0
      IF(IAREA(1).NE.0) NAREAS=NAREAS+1
      IF(IAREA(2).NE.0) NAREAS=NAREAS+1
      NARA=IAREA(1)
C   CHECK FOR OVERRIDE OF TERMINAL NUMBER
      IDEF=LUC(-20)
      ITERM=IKWP('TER',1,IDEF)
C   READ 'VASTEXT' DOCUMENTATION RECORD & EXTRACT NEEDED INFO
      CALL VRTIC(IRET,0,0)
      JDAY=IDOC(1)
      JTIME=IDOC(2)
      MENG=IDOC(38)
      MDMR=IDOC(40)
      MDRR=IDOC(41)
C   CHECK FOR 'NO-OUTPUT-MEDIUM'
      IF(NARA.EQ.0.AND.MDRR.LE.0) GO TO 1300
```

```
      LASRET=IRET(3)
      CGES=CKWP('GES',1,' ')
      IGES=0
      IF(CGES.EQ.CBLNK) GO TO 100
      IF(CGES.EQ.CLETG) GO TO 100
      IF(CGES.EQ.CLETC) IGES=1
  100 CONTINUE
      IDEF=0
      NOSFC=IKWP('SFC',1,IDEF)
      NFSAV=NOSFC
      IDEF=11
      NBXS=IKWP('BOX',1,IDEF)
C     NBXS SHOULD BE AN EVEN NUMBER FOR LARGE-DETECTOR DATA
      LDETR=0
      IF(MOD(NBXS,2).NE.0) GO TO 110
      LDETR=1
      NBXS=NBXS-1
  110 NBXS=MIN0(NBXS,11)
      IBOX=1
      IF(NBXS.EQ.1) IBOX=0
      IDEF=NBXS
      INCRL=IKWP('SFC',1,IDEF)
      INCRE=IKWP('SFC',2,IDEF)
C
      LLINE=IKWP('END',1,IDOC(55))
      LELEM=IKWP('END',2,IDOC(56))
      ILINE=IKWP('BEG',1,IDOC(57))
      IELEM=IKWP('BEG',2,IDOC(58))
C
      LBUG=IKWP('BUG',1,0)
      IF(LBUG.GT.1) CALL TGSET(2)
      NLP1=NL+1
      IFRM=LUC(-1)
      INRAS=LUC(-11)
      INPIC=LUC(-12)
      IGAM=IKWP('GAM',1,100)
      GAMRET=.01 FLOAT(IGAM)
      GAMSAV=GAMRET
      CALL GETDAY(IDPROS)
      IRET(1)=MOD(JDAY,100000)
      IRET(2)=JTIME
      IRET(3)=LASRET
      IF(ILINE IELEM.EQ.0) GO TO 140
      IF(LUC(16).EQ.0) GO TO 170
      IF(IKWP('ALTC',1,0).NE.0) GO TO 170
      CALL GETFRM(IFRM,IDIR)
      JSAT=IDIR(1)
      JDATE=IDIR(2)
      GO TO 160
  140 CALL TVSAT(IFRM,INRAS,INPIC,ILINE,IELEM,JSAT,JDATE,JTIME)
C     NEED 'OLDSTYLE' JDAY FOR NAVIGATION WITHIN VASCAT
  160 JDAY=JSAT 100000+JDATE
```

```
  170 IF((LLINE LELEM).NE.0) GO TO 180
      LLINE=ILINE
      LELEM=IELEM
  180 CONTINUE
      VDAT(1)=-1.
      ILINES=ILINE
      IELEMS=IELEM
C     VASDAT CHANGES ARGUMENTS 'ILINES' AND 'IELEMS'
      CALL VASDAT(ILINES,IELEMS,VDAT)
      VDAT(1)=VMISG
      JSAT=IDIR(3)
      ISAT=ISATNV(JSAT)
      JDATE=IDIR(4)
      JDAY=JSAT 100000+JDATE
      JTIME=IDIR(5)
      ILRES=IDIR(12)
      IERES=IDIR(13)
C     SET DEFAULT TO SQUARE SAMPLES ... I.E., EQUATE RESOLUTIONS
      INC=ILRES/IERES
      INCIL=ILRES INCRL
      IF(INCIL.EQ.0) INCIL=1
      INCIE=IERES INCRE*INC
      IF(INCIE.EQ.0) INCIE=1
C     DEFINE AREA PARAMETERS (NEEDED EVEN IF AREA NOT TO BE WRITTEN)
      INCLA=ILRES
      INCEA=IERES INC
      NBXH=NBXS/2
      ILINA=ILINE-NBXH*INCLA
      IELEA=IELEM-NBXH*INCEA
      LLINA=LLINE+NBXH INCLA
      LELEA=LELEM+NBXH*INCEA
C     EMPIRICAL ADJUSTMENT!
      ILINA=ILINA-1
      LLINA=LLINA-1
      MLIN=(LLINA-ILINA)/INCLA+1
      MELE=(LELEA-IELEA)/INCEA+1
      KLINES=NBXS
      KELEMS=NBXS
      KBOXES=MELE/KELEMS
      IF(MOD(MELE,KELEMS).NE.0) KBOXES=KBOXES+1
      IF(NARA.EQ.0) GO TO 200
C
      IF(MELE.GT.660) GO TO 1320
      DO 190 IA=1,NAREAS
      KARA=IAREA(IA)
      CALL ARASIZ(KARA,MLIN,MELE)
      CALL ENAREA(KARA,JSAT,JDATE,JTIME,ILINA,IELEA,INCLA,INCEA,IAMES)
      CALL OPNA(KARA)
      IAREC(IA)=0
  190 CONTINUE
      IAGAP=INCRL-NBXS
      KLINEA=KLINES
```

```
      IF(IAGAP.LT.0) KLINEA=KLINEA+IAGAP
      IF(LBUG.EQ.0) GO TO 200
      CALL ENKODE('("AREA ",I4," OPENED WITH NL =",I4,", NE =",I4/)',
     LBUF,NARA,MLIN,MELE)
  200 CONTINUE
C
C   BEGIN MAIN LINE LOOP
      DO 1260 IL=ILINE,LLINE,INCIL
      LASLIN=-1
      KBOX=0
C   BEGIN MAIN ELEMENT LOOP
      DO 1180 IE=IELEM,LELEM,INCIE
      KBOX=KBOX+1
      IF(LBUG.EQ.0) GO TO 220
      CALL ENKODE('("BEGIN LINE ",I4,", BOX ",I2/)',LBUF,IL,KBOX)
  220 LASELE=-1
C   INITIALIZE BOX PARAMETERS
      DO 240 KLIN=1,KLINES
      DO 240 KELE=1,KELEMS
      DO 230 I=1,13
  230 TDAT(I,KELE,KLIN)=VMISG
      IF(NARA.EQ.0) GO TO 240
      DO 235 KARA=1,NAREAS
      IARRAY(KELE,KLIN,KBOX,KARA)=MAXWV
  235 CONTINUE
  240 CONTINUE
C
C   ACQUIRE VAS DATA FOR BOX
      MSAM=0
      SELEV=0.
      LSTYPE=0
      DO 330 KLIN=1,KLINES
      DO 320 KELE=1,KELEMS
      DO 260 I=1,13
      VDAT(I)=VMISG
  260 NSPIN(I)=1
      ILINES=IL
      IELES=IE
      CALL VASDAT(ILINES,IELES,VDAT)
      IF(ABS(VLATS).GE.90.) GO TO 320
      IF(ABS(VZEN).GT.60.) GO TO 320
      IF(LELEV.EQ.999999) GO TO 320
      IF(VDAT(1).LT.0.) GO TO 320
      DO 300 I=1,12
      IF(VDAT(I).LT.180.OR.VDAT(I).GT.330.) VDAT(I)=VMISG
      IF(VDAT(I).EQ.VMISG) GO TO 300
      TDAT(I,KELE,KLIN)=VDAT(I)
C   SAVE REPRESENTATIVE SPIN BLOCK (ASSUME SPIN BUDGET INVARIANT)
      IF(ISPIN(I).NE.0) NSPIN(I)=ISPIN(I)
      IF(I.NE.8) GO TO 300
      MSAM=MSAM+1
      SELEV=SELEV+FLOAT(LELEV)
```

```
      IF(ICHAR.NE.0) ICHAR=1
      LSTYPE=LSTYPE+ICHAR
  300 CONTINUE
  320 CONTINUE
  330 CONTINUE
      IF(MSAM.EQ.0) GO TO 1180
C   OBTAIN MEAN ELEVATION
      SELEV=SELEV/FLOAT(MSAM)
      IELEV=IROUND(SELEV)
C   DETERMINE DOMINANT SURFACE TYPE; SET 'NO-SURFACE' FLAG OVER WATER
      MSAM2=MSAM/2
      NOSFC=NFSAV
C   DATA ACQUISITION COMPLETE
C
C   OBTAIN GUESS
      CALL GESPRO(IGES,NOSFC,MDNG)
      IF(TGES(1).LE.0.) GO TO 1180
C     IF(LSTYPE.LT.MSAM2)NOSFC=1
      PSTA=IPSTA
      TSTA=.01*FLOAT(ITSFC)
      TDSTA=0.01*FLOAT(IDSFC)
      DD=TSTA-TDSTA
      CALL WMIX(PSTA,TSTA,DD,WSTA,1)
C   FIND 'IS', FIRST LEVEL BELOW SURFACE (PRESSURE)
      IS=NL
      DO 340 J=1,20
      I=NLP1-J
      DP=P(I)-PSTA
      IF(DP.GE.0.) GO TO 340
      IS=I+1
      GO TO 360
  340 CONTINUE
  360 IS=MIN0(IS,NL)
      ILO=35
      IF(IGES.EQ.1) ILO=31
      IF(ILO.GE.IS) GO TO 400
      DTS=TSTA-T(IS)
      DTDP=DTS/(P(IS)-P(ILO))
      DO 380 I=ILO,IS
  380 T(I)=T(I)+DTDP*(P(I)-P(ILO))
  400 CONTINUE
      DO 420 I=IS,40
      T(I)=T(IS)
  420 W(I)=W(IS)
      DO 440 I=1,40
      W(I)=W(I)*WSTA/W(IS)
      WSATO=WSAT(P(I),T(I))
      W(I)=AMAX1(W(I),0.02)
      W(I)=AMIN1(W(I),WSATO)
      WG(I)=W(I)
      TG(I)=T(I)
  440 CONTINUE
```

```
      NLS=IS
      IF(LBUG.EQ.0) GO TO 460
      CALL ENKODE('(5X,"PRESSURE   ", 15F7.1/)',LBUF,P(26))
      CALL ENKODE('(5X,"GUESS TEMP", 15F7.1/)',LBUF,T(26))
      CALL ENKODE('(5X,"GUESS WVMR", 15F7.3/)',LBUF,W(26))
      CALL ENKODE('(1X,"Z1000 =",I8,5X,"TSFC =",I8,5X,"TDSFC =",I8/)',
     LBUF,IZ10,ITSFC,IDSFC)
  460 CONTINUE
C
C   GENERATE TBB'S FOR RETRIEVAL
      IF(NOSFC.NE.0) TSTA=AMIN1(TSTA,278.)
      CALL FILTER(VDAT,NSPIN,ERM,MSAM,TSTA,LDETR)
      IF(MSAM.EQ.0) GO TO 1180
      ICOUNT=0
      DO 480 I=1,10
      IF(VDAT(I).EQ.VMISG) GO TO 480
      ICOUNT=ICOUNT+1
  480 CONTINUE
      IF(ICOUNT.LT.7) GO TO 1180
      TSKIN=VSKINT(VDAT,0,1,JACT)
      IF(LBUG.NE.0) CALL ENKODE('("VSKINT ACTION =",I2/)',LBUF,JACT)
      IF(JACT.EQ.0) GO TO 1180
      ERTSFC=1.0
      ERWSFC=0.1
      ERMSAV=ERM(8)
      ERM(8)=1000.
      IF(NOSFC.EQ.0) GO TO 485
      TSTA=2. VDAT(8)-VDAT(7)
      WSTA=0.8*WSAT(PSTA,TSTA)
      TDSTA=DEWPT(PSTA,TSTA,WSTA)
      ERTSFC=4.0
      ERWSFC=0.25
      ERM(8)=ERMSAV
  485 CONTINUE
C
C   PREPARE GUESS INFO FOR MDFILE
      TGS(1)=TSTA
      DO 1000 I=2,20
      J=LMD(I)
 1000 TGS(I)=T(J)
      DGS(1)=TDSTA
      DO 1020 I=2,11
      J=LMD(I)
      DPT=DEWPT(P(J),T(J),W(J))
 1020 DGS(I)=DPT
C
C   MAKE SOUNDING FOR BOX
      IF(LBUG.NE.0) CALL ENKODE('("CALLING GVTWR, IS =",I3/)',LBUF,IS)
      GAMRET=GAMSAV
      CALL GVTWR(VDAT,ERM,TOTC,TSKIN,TSTA,WSTA,IS,ISAT,IFAIL)
      IF(IFAIL.NE.0) GO TO 1180
      DO 1030 I=1,NL
```

```
          TG(I)=T(I)
     1030 WG(I)=W(I)
          GAMRET=0.10 GAMSAV
          CALL GVTWR(VDAT,ERM,TOTC,TSKIN,TSTA,WSTA,IS,ISAT,IFAIL)
          IF(IFAIL.NE.0) GO TO 1180
          IF(MSAM.LT.5) GO TO 490
C
C    OBTAIN ADDITIONAL OUTPUT PARAMETERS
          CALL PRECW(P,W,U,IS)
          DO 1040 I=26,40
          TD(I)=DEWPT(P(I),T(I),W(I))
     1040 CONTINUE
          TCT=T(37)+TD(37)-2. T(31)
          CALL HTV(Z,IS)
C    SET UP FOR STABILITY CALCULATION
          NB=IS
          IF(PSTA.LT.P(NB)) NB=NB-1
          DO 1060 I=2,15
          PST(I)=P(NB)
          TST(I)=T(NB)
          TDST(I)=TD(NB)
     1060 NB=NB-1
          PST(1)=PSTA
          TST(1)=.01 ITSFC
          TDST(1)=.01*IDSFC
          IF(LBUG.NE.0)CALL ENKODE('("STAB. PRESS ", 10F8.2/)',LBUF,PST(1)
          CALL SNDANL(0.,15,PST,TST,TDST,DIR,SPD,STABIL)
          LIFT=IROUND(STABIL(8)*100.)
          IF(LBUG.EQ.0) GO TO 1080
          CALL ENKODE('(T6,"PRESSURE   ", 15F7.0/)',LBUF,P(26))
          CALL ENKODE('(T6,"T  PROFILE", 15F7.1/)',LBUF,T(26))
          CALL ENKODE('(T6,"DEW  POINT", 15F7.1/)',LBUF,TD(26))
          CALL ENKODE('(T6,"HEIGHT    ", 15F7.0/)',LBUF,Z(26))
          CALL ENKODE('(T6,"MIX  RATIO", 15F7.3/)',LBUF,W(26))
          CALL ENKODE('(T11,"TCT =",F7.2/)',LBUF,TCT)
     1080 CONTINUE
C    PREPARE OUTPUT BUFFER
          DO 1100 I=4,NSIZE
     1100 IRET(I)=MISG
          IRET(4)=LLC(-17)
          IRET(5)=IVER
          IRET(6)=INCRL
          IRET(7)=INCRE
          IGD=MOD(IDCC(33),1000)
          IRET(8)=IDCC(34)+IGD*10000
          IRET(9)=ISDCC(6)
          DO 1120 K=1,5
     1120 IRET(9+K)=ISDCC(K)
          IRET(15)=0
          IRET(16)=IDPROS
C    FILL IN USER MOD FLAG
          IRETD(3)=0
```

```
      IRETD(4)=LASRET+1
      IRETD(5)=IRCUND(VLAT 10000.)
      IRETD(6)=IRCUND(VLON 10000.)
      IRETD(7)=IDOC(2)
      IRETD(8)=NSAM
      DO 1140 N=1,12
      IF(VDAT(N).EQ.VMISG) GO TO 1140
      IRETD(N+8)=IRCUND(VDAT(N) 100.)
      IRETD(N+21)=IRCUND(ERM(N) 10000.)
 1140 CONTINUE
      IRETD(35)=IRCUND(U(IS) 1000.)
      IRETD(36)=IRCUND(TSKIN 100.)
      IRETD(39)=IRCUND(SZEN 100.)
      IRETD(40)=IRCUND(VZEN 100.)
      IF(PSTA.GE.850.) IRETD(41)=IRCUND(TCT 100.)
      IRETD(47)=LIFT
      IRETD(50)=21
C     STORE DATA BY LEVEL
      LS=NLP1-IS
      LS=MAX0(LS,2)
      DO 1160 K=2,20
      L=LMD(K)
      M=(K-1) 9
      IRETD(M+57)=Z(L)
      IRETD(M+51)=LCHR(K)
      IRETD(M+52)=P(L) 10.
      IF(P(L).GT.PSTA) GO TO 1160
C     BELOW SURFACE, STORE HEIGHTS ONLY
      IRETD(M+53)=IRCUND(T(L) 100.)
      IRETD(M+58)=IRCUND(TGS(K) 100.)
      IF(K.GT.11) GO TO 1160
      IRETD(M+54)=IRCUND(TD(L) 100.)
      IRETD(M+59)=IRCUND(DGS(K) 100.)
 1160 CONTINUE
C     ADD SURFACE VALUES
      IRETD(51)=LCHR(1)
      IRETD(52)=IPSTA 10
      IRETD(53)=ITSFC
      IRETD(54)=IDSFC
      IRETD(57)=IELEV
      IRETD(58)=IRCUND(TGS(1) 100.)
      IRETD(59)=IRCUND(DGS(1) 100.)
C     MD OUTPUT BUFFER COMPLETE
      LASRET=LASRET+1
      IDOC(100)=LASRET
      CALL VRTIO(IRET,LASRET,1)
C
  490 IF(NARA.EQ.0) GO TO 1180
C     OBTAIN SFOV T-P-W AND TOTAL-TOTALS FOR IMAGE
      NCHT=NCH+2
      NPWP=NPW+1
      NPTT=NPT+NPWP
```

```
      NFTM=NFTT-1
      DO 500 I=1,NL
      T(I)=TG(I)
  500 W(I)=WG(I)
      DO 560 KLIN=1,KLINES
      DO 540 KELE=1,KELEMS
      IF1=IFLAG(1,KELE,KLIN)
      IF2=IFLAG(2,KELE,KLIN)
      ICHECK=0
      IF(IF1.EQ.0.AND.IF2.EQ.0) ICHECK=10
      DO 502 I=1,13
  502 VDAT(I)=TDAT(I,KELE,KLIN)
      DO 504 I=1,NCH
  504 DTB(I)=VDAT(I)-TBC(I)
      DTB(NCH+1)=WSTA-WG(IS)
      DTB(NCH+2)=TG(IS)-TSTA
      DO 506 I=1,NCHT
      X=ABS(DTB(I))
      IF(X.GT.5.) DTB(I)=0.
      IF(ICHECK.EQ.0) GO TO 506
      IF(I.GT.2.AND.I.LT.9) DTB(I)=0
  506 DTB(I)=DTB(I)*ERS(I)
      DO 510 J=1,NFTT
      SUM=0.
      DO 508 K=1,NCHT
  508 SUM=SUM+XIT(J,K)*DTB(K)
  510 COEF(J)=SUM
      DO 514 I=21,IS
      SUM=0.
      DO 512 J=1,NFW
  512 SUM=SUM+COEF(J)*PHI(I,J)
  514 W(I)=WG(I)+GRAV*SUM
      CALL PRECW(P,W,U,IS)
      URET=U(IS)
      URET=CHOP(URET,1.,12.)
      IURET=IROUND(URET*20.)
      IARRAY(KELE,KLIN,KBOX,1)=IURET
      IF(IAREA(2).EQ.0) GO TO 540
      IF(PSTA.LT.850.) GO TO 540
      DO 518 I=21,IS
      SUM=0.
      DO 516 J=NFWP,NFTM
  516 SUM=SUM+COEF(J)*PHI(I,J)
      T(I)=TG(I)-SUM
  518 CONTINUE
      TD(37)=DEWPT(P(37),T(37),W(37))
      TOTL=T(37)+TD(37)-2.*T(31)
      ITOTL=IROUND(TOTL/3.)
      IARRAY(KELE,KLIN,KBOX,2)=ITOTL
  540 CONTINUE
  560 CONTINUE
C
```

```
 1180 CONTINUE
C
      IF(NARA.EQ.0) GO TO 1255
C   OUTPUT TO DIGITAL AREA, ONE LINE AT A TIME
      DO 1250 L=1,NAREAS
      KARA=IAREA(L)
      KREC=IAREC(L)
      DO 1220 J=1,KLINEA
      IA=0
      DO 1200 K=1,KBOXES
      DO 1200 I=1,KELEMS
      IA=IA+1
      IARET(IA)=IARRAY(I,J,K,L)
 1200 CONTINUE
      CALL PACK(MELE,IARET,IABUF)
      CALL WRITA(KARA,KREC,IABUF)
      KREC=KREC+1
 1220 CONTINUE
      IF(IAGAP.LE.0) GO TO 1245
C   FILL GAPS IN IMAGE WHEN BOX LINE-SPACING .GT. BOX SIZE
      DO 1230 M=1,MELE
 1230 IARET(M)=MAXWV
      CALL PACK(MELE,IARET,IABUF)
      DO 1240 I=1,IAGAP
      CALL WRITA(KARA,KREC,IABUF)
      KREC=KREC+1
 1240 CONTINUE
 1245 IAREC(L)=KREC
 1250 CONTINUE
C
 1255 CALL EDEST('FINISHED LINE ',IL)
 1260 CONTINUE
      IDOC(100)=LASRET
      IRET(3)=LASRET
      CALL VRTIO(IRET,0,1)
      IF(NARA.EQ.0) GO TO 1270
      DO 1265 L=1,NAREAS
      KARA=IAREA(L)
      CALL CLOSAC(KARA)
 1265 CONTINUE
 1270 CALL EDEST(' ** ALL DONE ** ',0)
      RETURN
 1280 CALL EDEST('INVALID AREA NUMBER ',NARA)
      GO TO 1340
 1300 CALL EDEST('NO OUTPUT MEDIUM (AREA OR MD) SPECIFIED ***',0)
      GO TO 1340
 1320 CALL EDEST('IMAGE TOO WIDE - MAXELE IS 660, ESIZ = ',MELE)
 1340 CALL ABORT(0)
      RETURN
      END
```

```
//PLVA5660 JOB CLASS=B,MSGLEVEL=(0,0)
//  VLPLVA    H     04/24/84: MEMBER UPDATED
//  VLPLVA    CMH   04/18/84; ADD GRA KEYWORD FOR GRAPHIC FRAME
// EXEC MCPRG,MOD=PLVA
//FORT.SYSIN DD
@PROCESS SC(TGMES,LIT,EMES,DMES,MCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,DOPEN,DCLOSE,DREAD,DWRITE,WTOR,CPCOM,SGX,SQW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAIN0
C ?    PLOT ALL VALUES IN MD FILE (VAS RETRIEVALS)
C ?    MD LINKAGE THRU VASTEXT OR KEYWORD
C ?   KEYIN: PLVA <PAR> <LEV> KEYWORDS
C ?          PAR MAY BE ANY PARAMETER IN THE MD SCHEMA
C ?          LEV MAY BE 'SFC', MB LEVEL, OR 'DIFF' FOR TBB DIFFERENCES
C ?      KEYWORDS:
C ?          'LAT'    MIN AND MAX LATS
C ?          'LON'    MIN AND MAX LONGITUDES
C ?          'LEV2'   2ND LEVEL FOR THICKNESS (LEV-LEV2)
C ?          'MDNR'   RETRIEVAL MD NO.
C ?          'MDRR'   RETRIEVAL MD ROW NO.
C ?          'SIZE'   SIZE OF PLOTTED DIGIT
C ?          'COLOR'  COLOR OF PLOTTED DIGIT
C ?          'ALL'    PLOT EVERYONE..EVEN THE REJECTED VALUES
C ?          'TYPE'   PLOT ONLY INDICATED TYPE (21,22, OR 23)
C ?          'GRA'    PLOT ON SPECIFIED GRAPHICS FRAME
      DIMENSION SATPOS(2)
      DIMENSION MF(64),LBUF(33),IOUT(300)
     ,ILAMIN(40),ILOMIN(40)
     ,ILAMAX(40),ILOMAX(40)
      DIMENSION LIST(400),ISCL(400),IUN(400),LOCS(400)
      DIMENSION KOLT(10),MDHD(64)
      CHARACTER 12 CCHP,LCHR,CPP,CLEV
      CHARACTER 8 MFILE
      CHARACTER 4 CLIT
      COMMON /DCC/IDCC(112)
      COMMON /NAV/FLAT,FLON,ZENLOC,SZEN,IL,IE,IRAS,IPIC,IHMS,JT,JD
      COMMON /IDENT/IYMD,JHMS,NROWM,NSAT
      COMMON /ANALS/NOAN,LTOP
      COMMON /CREIT/NODE
      COMMON /TIGHT/ITOL
      COMMON /ENTRY/INIT
      COMMON/ORIENT/YCOORD,XCOORD
      DATA SATPOS/135.,75./
      DATA COSI/.17365/
      DATA LBUF/33 Z40404040/
      DATA MFILE,LUN,LEN/'VASTEXT ',20,100/
      DATA MISS/Z80808080/
      DATA NCOLS/56/
C     CHECK DEBUG OPTION
      IHEL=IKWP('HELP',1,0)
      IF (IHEL.NE.0)GO TO 800
      KBUG=IKWP('BUG',1,0)
```

```
      ITYP=IKWP('TYPE',1,0)
      IWRM=LUC(-5)
      IWRM=IKWP('GRA',1,IWRM)
      IFRM=LUC(-1)
      IF (LUC(81).EQ.0)GO TO 110
C     READ VASTEXT CONTEXT FILE
      ITERM=LUC(-20)
      CALL DOPEN(MFILE,LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      CALL DCLOSE(LUN)
C     INITIALIZE NAVIGATION
      IRAS=LUC(-11)
      IPIC=LUC(-12)
      CALL TVSAT(IFRM,IRAS,IPIC,IL,IE,ISS,JD,JT)
      JD=ISS 100000+JD
      CALL NVINIT(BETAIN,BETDOT,INAV,PTIME)
      GO TO 120
  110 CALL TSNIC(1,1,1,1,1,1,IDOC)
      INIT=1
      NODE=IDOC(7)
C     SET UP AREA FOR IMAGE SPACE TO PLOT
      CALL TVSAT(IFRM,005,335,LTOP,LELE,ISS,ID,IT)
      CALL TVSAT(IFRM,495,335,LBOT,LELE,ISS,ID,IT)
      IF (LTOP.LT.1)LTOP=1
      IF (LBOT.GT.NROWM)LBOT=NROWM
      NROWS=LBOT-LTOP+1
      J=1
      N=1
      NPTS=NROWS NCOLS
  120 CONTINUE
      ISS=0
      IF (LUC(81).EQ.0)GO TO 125
      JD=IDOC(1)
      ISS=ISATNV(JD)
      SATLON=-SATPOS(ISS)
C     CHECK OPERATOR FORCED BOUNDARIES
  125 LLNW=IDOC(25)
      LLSE=IDOC(26)
      LAN=LLNW/1000
      LAS=LLSE/1000
      LOW=MOD(LLNW,1000)
      LOE=MOD(LLSE,1000)
      L=IKWP('LAT',1,0)
      IF (L.NE.0)LAS=L
      L=IKWP('LAT',2,0)
      IF (L.NE.0)LAN=L
      L=IKWP('LON',1,0)
      IF (L.NE.0)LOE=L
      L=IKWP('LON',2,0)
      IF (L.NE.0)LOW=L
      LOW=IABS(LOW)
      LOE=IABS(LOE)
```

```
      IF (LOW.GT.180)LOW=LOW-360
      IF (LOE.GT.180)LOE=LOE-360
      IF (LOW.LT.LOE)LOW=LOW+360
      LOX=0
      IF (IABS(LOW-LOE).GT.180)LOX=1
      IF (LOX.NE.0.AND.LOW.LT.0)LOW=LOW+360
      MDNO=IKWP('MDNR',1,IDOC(40))
      MDR=IKWP('MDRR',1,-1)
      IF (MDR.LT.0)MDR=IDOC(41)
      IF (MDINFO(MDNO,MDHD).NE.0)GO TO 900
      IF (MDOPEN(MDNO,2).NE.0)GO TO 900
C     HAVE TO CALL MDOPEN TO LOAD MDGET...
C     READ ROW HEADER RECORD
      IOK=MDGET(MDNO,MDR,0,IOUT)
      KOUT(1)=3
      KOUT(2)=IOUT(1)
      KOUT(3)=IOUT(3)
      KOUT(4)=IOUT(16)
      IF (KBUG.NE.0)CALL OUTINT(KOUT)
      IF (IOK.LT.0)GO TO 902
C     READ TEST RECORD
      M=1
   40 M=M+1
      IF (MDHD(8).NE.0)IOK=MDGET(MDNO,0,M,IOUT)
      MMAX=MDHD(5)
      IF (KBUG.NE.0)CALL SDEST(' MMAX IS ',MMAX)
C     MMAX IS MAX POSSIBLE DATA ENTRIES,USED TO AVOID
C     INFINITE LOOP AT SN 80
      IOK=MDGET(MDNO,MDR,M,IOUT)
      IF (M.GT.10)GO TO 904
      IF (IOK.LT.0)GO TO 40
      IF (KBUG.NE.0)CALL SDEST(' OPERATING WITH ROW= ',MDR)
C     READ IN KEYS
      NKEYS=MDKEYS(MDNO,-1,LIST,ISCL,IUN,LOCS)
      IF (KBUG.NE.0)CALL SDEST(' NO OF KEYS IS ',NKEYS)
      CALL GETFRM(IFRM,MF)
      MAG=MF(10)
      IF (MAG.LT.6)MAG=6
      MAG=IKWP('SIZE',1,MAG)
      CALL INITPL(IWRM,0)
      CCHR=CPP(1,'Z   ')
      LCHR=CCHR
      IF (LCHR.EQ.'WIN')CCHR='SPD'
      ISUB=0
      IF (CCHR.NE.('TDIF'))GO TO 20
      ISUB=5
      CCHR=('T   ')
   20 IF (CCHR.NE.('DDIF'))GO TO 30
      ISUB=5
      CCHR=('TD  ')
   30 CONTINUE
C     CHECK FOR SELECTIVE PRESSURE
```

```
      ILEV=IPP(2,0)*10
      CLEV=CPP(2,'  ')
      IF (KBUG.NE.0)CALL SDEST(' LEVEL IS ',ILEV)
C     SET DEFAULT TO 500 Z
      IF (CCHR.EQ.'Z'.AND.ILEV.EQ.0.AND.CLEV.EQ.'  ')ILEV=5000
C     LEVELS IN MD FILES ARE MB 10
C     CHECK FOR THICKNESS CALCULATION
      IL2=IKWP('LEV2',1,0)*10
C     CHECK FOR TBB DIFF
      IF (CLEV.EQ.'DIFF')ISUB=13
C     FIND WHERE LAT/LON INFO IS
      DO 18 N=1,NKEYS
      IF (LIST(N).EQ.LIT('MOD '))NMOD=N
      IF (LIST(N).EQ.LIT('CMAX'))NCA=N
      IF (LIST(N).EQ.LIT('NREC'))NCA=N
      IF (LIST(N).EQ.LIT('LAT '))NLAT=N
      IF (LIST(N).EQ.LIT('LON '))NLON=N
      IF (LIST(N).EQ.LIT('RT  '))NTYP=N
C     SAVE REPRESENTATIVE ENTRIES FOR NODATA CASE
      IF (CLIT(LIST(N)).EQ.CCHR)NADD=N
   18 CONTINUE
      ISC=10.**ISCL(NADD)
      MREC=500
      IF (NCA.NE.0)MREC=IOUT(NCA)
      MREC=IKWP('LAST',1,MREC)
      IF (KBUG.NE.0)CALL SDEST(' NO. OF REPORTS IS ',MREC)
      IALL=IKWP('ALL',1,0)
C     BEGIN IMPLICIT DO LOOP ON REPORTS
      M=1
      MM=1
   15 CONTINUE
      IF (MDHD(8).EQ.0)GO TO 115
C     READ COLUMN HEADER FOR RAOB
      IOK=MDGET(MDNO,0,MM,IOUT)
  115 CONTINUE
      IOK=MDGET(MDNO,MDR,MM,IOUT)
      IF (IOK.LT.0)GO TO 75
C     CHECK ON PLOT OF EDITED VALUES
      IF (IOUT(NMOD).NE.0.AND.IALL.EQ.0)GO TO 70
      IF (IOUT(NTYP).NE.ITYP.AND. ITYP.NE.0)GO TO 70
      NS=1
      IF (ILEV.NE.0)GO TO 44
      DO 43 N=1,NKEYS
      IF (CLIT(LIST(N)).NE.CCHR)GO TO 43
      IADD=LOCS(N)
      GO TO 56
   43 CONTINUE
      GO TO 70
   44 CONTINUE
C     PICK ON SPECIFIC LEVEL AND CHARACTER
      DO 45 N=1,NKEYS
      IF (LIST(N).NE.LIT('P   '))GO TO 45
```

```
      I=LOCS(N)
      IF (ILEV.EQ.ICUT(I))GO TO 31
      IF (CLEV.EQ.CLIT(ICUT(I)))GO TO 31
      GO TO 45
   31 CONTINUE
      NS=N+1
      GO TO 46
   45 CONTINUE
      GO TO 70
   46 CONTINUE
      DO 47 N=NS,NKEYS
      IADD=LOCS(N)
      IF (CLIT(LIST(N)).EQ.CCHR)GO TO 48
   47 CONTINUE
      GO TO 70
   48 CONTINUE
C     LOOK FOR THICKNESS
      IAD2=0
      IF (IL2.EQ.0)GO TO 56
      DO 52 N=1,NKEYS
      IF (LIST(N).NE.LIT('P   '))GO TO 52
      I=LOCS(N)
      IF (IL2.EQ.ICUT(I))GO TO 51
      IF (CLEV.EQ.CLIT(ICUT(I)))GO TO 51
      GO TO 52
   51 CONTINUE
      NS=N+1
      GO TO 53
   52 CONTINUE
      GO TO 70
   53 CONTINUE
      DO 54 N=NS,NKEYS
      IAD2=LOCS(N)
      IF (CLIT(LIST(N)).EQ.CCHR)GO TO 55
   54 CONTINUE
      GO TO 70
   55 IF (IOUT(IAD2).EQ.MISS)GO TO 70
   56 IF (ICUT(IADD).EQ.MISS)GO TO 70
C     LOCATE RASTER AND PIXEL OF SOUNDING
      FLAT=ICUT(NLAT)*.0001
      IF (FLAT.GT.LAN.OR.FLAT.LT.LAS)GO TO 70
      FLON=ICUT(NLON)*.0001
      IF (FLON.GT.LOW.OR.FLON.LT.LOE)GO TO 70
      IF (LUC(81).EQ.0)GO TO 12
C     REVERT TO -W+E CONVENTION
      SLON=-FLON
      CALL SATEAR(FTIME,FLIN,FELE,FLAT,SLON,2,INAV,BETAIN,BETDCT,0.)
      IL=FLIN+0.5
      IE=FELE+0.5
      GO TO 13
C     USE TOVS NAVIGATION ROUTINE
   12 LATS=FLAT*100.
```

```
      LONGS=FLON*100.
      CALL SRCH(LATS,LONGS,IM,IL,IE,NPTS,NROWS)
      IF (IM.EQ.0)GO TO 70
C     GUARD AGAINST OVERLAP IN SRCH TO AVOID DUPLICATION
      IL=IL+LTOF-1
      IF (IL.LT.LTOF.OR.IL.GE.LBOT)GO TO 70
   13 KOUT(1)=4
      KOUT(2)=IL
      KOUT(3)=IE
      KOUT(4)=ILALO(FLAT)
      KOUT(5)=ILALO(FLON)
C     IF (KBUG.NE.0)CALL OUTINT(KOUT)
      CALL SATTV(IFRM,IL,IE,IRAS,IPIC,JS,JD,JT)
C     DO PLOTTING
      KOLOR=MOD(IOUT(NTYP),10)+1
      IF (KOLOR.GT.3)KOLOR=1
      IF (KOLOR.LT.1)KOLOR=1
      IDAT=IOUT(IADD)/ISC
      IF (IAD2.NE.0)IDAT=IDAT-IOUT(IAD2)/ISC
      KOLOR=IKWP('COLOR',1,KOLOR)
      IF (LCHR.NE.'WIN')GO TO 60
      SPD=FLOAT(IDAT)
      IDIR=IOUT(IADD-1)
      DIR=FLOAT(IDIR)
      YP=IRAS+MAG
      XP=IPIC+MAG
      IF (ISS.EQ.0)GO TO 57
      FLON=-FLON
      ADJ=DIR
      CALL DIRADJ(FLAT,FLON,SATLON,DIR,ADJ)
      GO TO 58
   57 SEC=1./COS(FLAT*.0174533)
      ADD=ARSIN(COSI*SEC)/.0174533
      IF (NODE.EQ.2)ADD=-ADD
      DIR=DIR+ADD
      IF (DIR.GT.360.)DIR=DIR-360.
      IF (DIR.LT.0.)DIR=DIR+360.
      ADJ=DIR
   58 SZ=MAG
      IF (FLAT.LT.0)SZ=-MAG
      CALL BARB(ADJ,SPD,XP,YP,KOLOR,SZ)
      GO TO 70
   60 IF (ISUB.EQ.0)GO TO 65
      IF (IOUT(IADD+ISUB).EQ.MISS)GO TO 70
C     FORM DIFFERENCE QUANTITY
      IDAT=IDAT-IOUT(IADD+ISUB)/ISC
      GO TO 67
   65 IF (CCHR.EQ.'T'.OR.CCHR.EQ.'TD')IDAT=IDAT-273
C     ABOVE TO CONVERT TO CENTI
      IF (CCHR.EQ.'Z')IDAT=(IDAT+5)/10
      IF (CLEV.EQ.'DIFF')IDAT=IDAT*10
   67 CONTINUE
```

```
      IF (CCHR.NE.'FCT')GO TO 68
      KOLOR=MOD(IDAT,10)
      IDAT=IDAT/10
   68 CONTINUE
      IDAT=MOD(IDAT,1000)
      CALL PLTDIG(IRAS+MAG,IPIC+MAG,IDAT,MAG,1,KOLOR)
   70 M=M+1
      IF (M.GT.MREC)GO TO 80
   75 MM=MM+1
      IF (MM.LE.MMAX)GO TO 15
   80 CALL ENDPLT
      RETURN
  800 CALL SDEST(' (PARAM)  (P-LEVEL MB)',0)
      RETURN
  900 CALL SDEST(' CANNOT OPEN MDFILE NO. ',MDNO)
      RETURN
  902 CALL SDEST(' CANNOT LOCATE ROW NO. ',MDR)
      RETURN
  904 CALL SDEST(' TROUBLE READING DATA RECORD NO ',M)
      RETURN
      END
```

```
//BNVA5670 JOB  CLASS=B,MSGLEVEL=(0,0)
//  EXEC  MCPROG
//  VLBNVA   RJP  04/24/84: MEMBER UPDATED
//FORT.SYSIN  DD
      SUBROUTINE MAIN0
C     CALL SEQUENCE
C ** KEYIN
C ?   PROGRAM TO PREPARE RETRIEVAL  GRID FILES
C ?   KEYIN: BNVA <CPAR> <LEV> <NGG> KEYWORDS
C ?     POSITIONAL PARAMETERS
C ?           <CPAR>  'Z' 'T' ETC
C ?           <LEV>   MB LEVEL (OR 'SFC' OR 'DIFF' FOR TBB DIFF)
C ?           <NGG>   GRID NO OF GUESS GRID
C ?     KEYWORDS
C ?           LAT   MIN AND MAX LATS
C ?           LON   MIN AND MAX LONGITUDES
C ?           INC   GRID SPACING IN DEG*10
C ?           LEV2  2ND LEVEL FOR THICKNESS (LEV-LEV2)
C ?           SCL   BARNES SCALE FACTOR (DEFAULT 50)
C ?           GSS   GUESS OPTION 'G' TO USE VASGSS
C ?           ERR   GROSS ERROR CHECK (REAL UNITS) VALID WITH GSS/EDIT
C ?           SFC   NON ZERO FOR NO SURFACE GUESS OPTION
C ?           MDNR  MD FILE FOR DATA (DEFAULT TO VASTEXT)
C ?           MDRR  MD ROW NO (DEFAULT TO VASTEXT)
C ?           NGFG  GRID FILE FOR GUESS (DEFAULT TO VASTEXT)
C ?           MDNG  GUESS MD FILE (DEFAULT TO VASTEXT)
C ?           WGT   NON ZERO WRITES GRID OF PASS  WEIGHTS
C ?           BIAS   VALUE TO BE SUBTRACTED FROM OBS
C ?           EDIT  >0 EDITS MDNR BY DATA/ANAL COMP
C ?                 <0 SKIPS DATA/GUESS GROSS ERROR CHECK
C ?           NODAT NON ZERO FORCES NO DATA (GUESS ONLY)
C ?           ALL   NON ZERO KEEPS ALL REPORTS
      COMMON /DOC/IDOC(112)
      COMMON /NAV/FLAT,FLON,VZEN,SZEN,IL,IE,IRAS,IPIC,ITIME,JTIME,JDAY
      COMMON /DBUG/KBUG
      COMMON /DIMEN/NROWS,NCOLS
      COMMON /THICK/LPR2
      CHARACTER*8 MFILE
      CHARACTER*12 CPP,CLIT,CLEV,CKWP,ICHR,LCHR,CGS
      REAL*8 DLIT
      REAL*8 FN,SUM,SUMS
      DIMENSION KOLT(9),MDHD(64),KBUF(400)
      DIMENSION LBUF(33)
      DIMENSION ITG(2400),IDL(2400)
      DIMENSION ILNI(4),IREP(3000),DA(3000),RW(3000),CL(3000)
      DIMENSION DAS(3000),STLAT(3000),STLON(3000)
      DIMENSION FLD(2400),WT1(2400),WT2(2400)
      DIMENSION LIST(400),ISCL(400),IUN(400),LOCS(400)
C-----DESCRIPTION OF 64-WORD GRID HEADERS
C
      INTEGER IGHD(64)
      INTEGER IGID(8)
```

```
C-----GIVE TOTAL SIZE (WORDS), # ROWS, # COLS. (IGSIZE=IGNR*IGNC)
      EQUIVALENCE (IGSIZE,IGHD(1)),(IGNR,IGHD(2)),(IGNC,IGHD(3))
C-----YYDDD, HHMMSS AND VALID-TIME (IF APPLICABLE) FOR GRID
      EQUIVALENCE (IGDAY,IGHD(4)),(IGTIME,IGHD(5)),(IGTIMV,IGHD(6))
C-----DESCRIPTION OF GRIDDED VARIABLE (IN MD-FILE TERMS):
C-----   NAME, SCALE, AND UNITS
      EQUIVALENCE (IGVNAM,IGHD(7)),(IGVSCA,IGHD(8)),(IGVUNI,IGHD(9))
C-----DESCRIPTION OF VERTICAL LEVEL: VALUE, SCALE, AND UNITS
      EQUIVALENCE (IGLEVL,IGHD(10)),(IGLSCA,IGHD(11)),(IGLUNI,IGHD(12)
C-----GRIDDED-VARIABLE TYPE: =1 (TIME DIF) 2(TIME AVG) 4 (LEVEL DIF)
C-----   8 (LEVEL AVG) OR ANY SUM OF THE FOREGOING
      EQUIVALENCE (IGVTYP,IGHD(13))
C-----FOLLOWING USED IF PARAMETER IS A VERTICAL (LEVEL) DIF OR AVG
C-----   (SAME SCALE AS IGLEVL)
      EQUIVALENCE (IGLDIF,IGHD(14))
C-----FOLLOWING USED IF PARAMETER IS A TIME DIF OR AVG (HHMMSS)
      EQUIVALENCE (IGTDIF,IGHD(15))
C-----GRID ORIGIN, TYPE (I.E. TYPE OF PROJECTION)
      EQUIVALENCE (IGORG,IGHD(33)),(IGTYPE,IGHD(34))
C-----SUBSEQUENT COORDS (IGLAMX,IGLOMX,IGLAMN,IGLOMN,IGINCR,
C-----   IGPOLR,IGPOLC,IGSP60,IGCLON) ALL HAVE 4 IMPLIED DEC. PLACES.
C-----   LAT GOES FROM -900000 TO 900000, LON GOES FROM -1800000
C-----   TO 1800000 (WEST IS +)
C-----TYPE 1 GRIDS ARE PSEUDO-MERCATOR
      EQUIVALENCE (IGLAMX,IGHD(35)),(IGLOMX,IGHD(36)),(IGLAMN,IGHD(37)
     *,(IGLOMN,IGHD(38)),(IGINCR,IGHD(39))
C-----TYPE 2 GRIDS ARE POLAR-STEREOGRAPHIC
C-----GIVE ROW # OF NORTH POLE, COL # OF N.P., COL SPACING AT 60 DEG N
C-----   (DEG), LONGITUDE PARALLEL TO COLUMNS (DEG)
      EQUIVALENCE (IGPOLR,IGHD(35)),(IGPOLC,IGHD(36)),(IGSP60,IGHD(37)
     *,(IGCLON,IGHD(38))
C-----INITIALS OF USER AND PROJECT # UNDER WHICH GRID CREATED
      EQUIVALENCE (IGUSER,IGHD(41)),(IGPROJ,IGHD(42))
C-----CHARACTER ID SUPPLIED BY PROGRAM (ARBITRARY)
      EQUIVALENCE (IGID,IGHD(43))
C
      EQUIVALENCE   (WT2(1),ITG(1))
C
      DATA MISS/Z80808080/,NSIZE/2400/,NBMAX/5000/
      DATA LBUF/33*Z40404040/,IBLNK/Z40404040/
      DATA MFILE,LUN,LEN/'VASTEXT ',20,100/
      IALL=IKWP('ALL',1,0)
      NGOUT=IKWP('NGOUT',1,0)
C     159 IS MAX NO. OF GRIDS PERMITTED BY IG SOFTWARE
      IF (NGOUT.NE.0)NGOUT=-NGOUT
C     ABOVE FORCES OUTPUT GRID NO.
      BIAS=DKWP('BIAS',1,0.)
      KBUG=IKWP('BUG',1,0)
      IF(KBUG.GE.0)CALL SDEST(' GO!!!!!!!',0)
C     PICK UP CONTEXT INFORMATION
      ITERM=LUC(-20)
      IF (LUC(81).NE.0)GO TO 111
```

```
      CALL TSNIC(1,1,1,1,1,IDOC)
      GO TO 112
  111 CALL DOPEN(DLIT(MFILE),LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      CALL DCLOSE(LUN)
  112 MDNO=IKWP('MDNR',1,0)
      IF(MDNO.EQ.0)MDNO=IDOC(40)
      NGFG=IKWP('NGFG',1,0)
      IF (NGFG.EQ.0)NGFG=IDOC(42)
      MDR=IKWP('MDRR',1, -1)
      IF(MDR.LT.0)MDR=IDOC(41)
      MDNG=IDOC(38)
      MDNG=IKWP('MDNG',1,MDNG)
      ICHR=CPP(1,'Z')
      LCHR=ICHR
      ILEV=IPP(2,0)*10
      LPR=ILEV/10
      IF (MDINFO(MDNO,MDHD).NE.0)GO TO 900
      IF (MDOPEN(MDNO,2).NE.0)GO TO 900
C     READ ROW HEADER RECORD
      IOK=MDGET(MDNO,MDR,0,KBUF)
      IF (IOK.LT.0)GO TO 902
      ISYD=KBUF(1)
      IHMS=KBUF(2)
C     READ TEST RECORD
      M=0
    1 M=M+1
      IF (M.GT.100)GO TO 904
C     READ COLUMN HEADER FOR RAOB
      IF (MDHD(8).NE.0)IOK=MDGET(MDNO,0,M,KBUF)
      IOK=MDGET(MDNO,MDR,M,KBUF)
      IF (IOK.LT.0)GO TO 1
      MMAX=MDHD(5)
      IF (KBUG.GT.0)CALL SDEST(' MMAX IS ',MMAX)
C     MMAX IS MAX POSSIBLE DATA ENTRIES,USED TO AVOID
C     INFINITE LOOPING AT SN 148
C     FIND ADRESS FOR LAT AND LON
      NKEYS=MDKEYS(MDNO,-1,LIST,ISCL,IUN,LOCS)
      NCA=0
      DO 2  N=1,NKEYS
      IF (LIST(N).EQ.LIT('MOD ')) NMOD=N
      IF (LIST(N).EQ.LIT('CMAX')) NCA=N
      IF (LIST(N).EQ.LIT('NREC')) NCA=N
      IF (LIST(N).EQ.LIT('LAT ')) NLAT=N
      IF (LIST(N).EQ.LIT('LON ')) NLON=N
C     SAVE REPRESENTATIVE ENTRIES FOR NODATA CASE
      IF (LIST(N).EQ.LIT(ICHR))NADD=N
    2 CONTINUE
      NRPT=500
      IF (NCA.NE.0)NRPT=KBUF(NCA)
      NRPT=IKWP('LAST',1,NRPT)
      IF (LCHR.EQ.'U'.OR.LCHR.EQ.'V')ICHR='SPD'
```

```
      IBIAS=0
      IF (ICHR.EQ.('T   ').OR.ICHR.EQ.('TD  '))
       IBIAS=27312
C     CHECK ON KEYIN OVERRIDE
      IF (BIAS.NE.0)IBIAS=BIAS
      ISUB=0
      IF (ICHR.NE.('TDIF'))GO TO 20
      ISUB=5
      ICHR=('T   ')
   20 IF (ICHR.NE.('DDIF'))GO TO 30
      ISUB=5
      ICHR=('TD  ')
   30 CONTINUE
      NS=1
C     CHECK FOR SELECTIVE PRESSURE
      CLEV=CPP(2,' ')
      IF (ICHR.EQ.'Z'.AND.ILEV.EQ.0.AND.CLEV.EQ.' ')ILEV=5000
      LPR=ILEV/10
      MMOD=ILEV
      IF (ILEV.EQ.0)MMOD=999
C     CHECK FOR THICKNESS CALCULATION
      IL2=IKWP('LEV2',1,0)*10
      LPR2=IL2/10
C     CHECK FOR TBB DIFF
      IF (CLEV.NE.'DIFF')GO TO 10
      ISUB=13
C     CHECK FOR GUESS GRID
   10 NGG=IPP(3,0)
      IF (NGG.NE.0)GO TO 14
C     CHECK OPERATOR FORCED BOUNDARIES
      LLNW=IDOC(25)
      LLSE=IDOC(26)
      LN=LLNW/1000
      LS=LLSE/1000
      LW=MOD(LLNW,1000)
      LE=MOD(LLSE,1000)
      L=IKWP('LAT',1,0)
      IF (L.NE.0)LS=L
      L=IKWP('LAT',2,0)
      IF (L.NE.0)LN=L
      L=IKWP('LON',1,0)
      IF (L.NE.0)LE=L
      L=IKWP('LON',2,0)
      IF (L.NE.0)LW=L
      LW=IABS(LW)
      LE=IABS(LE)
      IF (LW.GT.180)LW=LW-360
      IF (LE.GT.180)LE=LE-360
      IF (LW.LT.LE)LW=LW+360
      LINC=IKWP('INC',1,0)
      DINC=.1 FLOAT(LINC)
      ROWS=LN-LS
```

```
      COLS=LW-LE
      PTS=(ROWS+1.)*(COLS+1.)
C     ESTABLISH GRID INCREMENT IN 10THS OF DEGREES
      FN=FLOAT(NSIZE)
      IF (LINC.EQ.0)DINC=SQRT(PTS/FN)
      INC=DINC*10.+0.5
      LAN=LN*10
      LAS=LS*10
      LOW=LW*10
      LOE=LE*10
   13 DINC=INC
      DLAT=DINC
      DLON=DINC
      NROWS=(LAN-LAS)/INC+1
      NCOLS=(LOW-LOE)/INC+1
      NPTS=NROWS*NCOLS
      IF (LINC.NE.0)GO TO 14
      IF (NPTS.LE.NSIZE)GO TO 14
      INC=INC+1
      GO TO 13
   14 CONTINUE
C     USE  IDATA AS FLAG IN EVENT OF NO DATA
      IDATA=77777
C     INITIALIZE ANALYSIS FIELD
      DO 12 L=1,NSIZE
      IDL(L)=0
   12 ITG(L)=0
      IGB=0
      IF (NGG.EQ.0)GO TO 105
      IOK=IGGET(NGFG,NGG,2400,IDL(1),NROWS,NCOLS,IGHD)
C     IF (LIT(ICHR).NE.IGVNAM)GO TO 940
C *** ABOVE COMMENTED OUT UNTIL CALLAN JOINS THE REST OF US
     CALL SDEST(' LPR IS ',LPR)
     CALL SDEST(' IGLEVL IS ',IGLEVL)
C    IF (LPR.NE.IGLEVL)GO TO 940
C    NOTE..ABOVE WILL DEFINE THE GRID SIZE
     IF (IOK.LT.0)GO TO 300
     NPTS=NROWS*NCOLS
     SCL=10.**IGVSCA
     DO 102 N=1,NPTS
 102 FLD(N)=FLOAT(IDL(N))/SCL
     LAN=IGLAMX/1000
     LAS=IGLAMN/1000
     LOW=IGLOMX/1000
     LOE=IGLOMN/1000
     INC=IGINCR/1000
     DLAT=INC
     DLON=INC
     IGB=1
 105 TLAT=0.1*FLOAT(LAN)
     SLAT=0.1*FLOAT(LAS)
     WLON=0.1*FLOAT(LOW)
```

```
      ELON=0.1*FLOAT(LOE)
      DINC=DINC*0.1
      DLAT=DLAT*0.1
      DLON=DLON*0.1
      IF(KBUG.GE.0)CALL SDEST('    NROWS    NCOLS    INC',0)
      KOUT(1)=3
      KOUT(2)=NROWS
      KOUT(3)=NCOLS
      KOUT(4)=INC
      IF(KBUG.GE.0)CALL OUTINT(KOUT)
C     CHECK GUESS OPTION..FILL FROM 'VASGSS'
      CGS=CKWP('GSS',1,' ')
      IF (CGS.NE.('G   '))GO TO 110
      IGB=1
      NOSFC=IKWP('SFC',1,0)
      CALL ANGSS(IDL,ITG,NCOLS,NROWS,TLAT,WLON,DINC,LFR,
     ICHR,MDNG,NOSFC)
  110 NB=0
C     NB IS USED TO COUNT REPORTS FOR BARNES ANALYSIS
      IF (IKWP('NODAT',1,0).NE.0)GO TO 150
      IF (KBUG.GT.0)CALL SDEST(' NO. OF REPORTS IN FILE IS ',NRPT)
      SUM=0.
      SUMS=0.
C     BEGIN IMPLICIT DO LOOP ON REPORTS
      NN=1
C     NN IS USED TO COUNT ALL REPORTS
      LL=0
C     LL IS USED TO COUNT ALL ACCEPTED REPS (INCLUDING GROSS ERROR EDI
C     FOR THE FINAL EDIT STEP,IF REQUESTED
      KK=0
C     KK IS USED TO COUNT ALL VALID REPORTS (WITH DARA)
C     (IN RAOB FILES MANY "COLUMNS" DO NOT IN FACT REPORT)
      IADD=0
      ERR=999999.
      IF (ICHR.EQ.'TBB')ERR=1.5
      IF (ICHR.EQ.'Z')ERR=30.
      IF (ICHR.EQ.'T')ERR=3.
      IF (ICHR.EQ.'TD')ERR=3.
      ERR=DKWP('ERR',1,ERR)
      FCK=ERR*3.
C     FCK IS GROSS ERROR CHECK AGAINST GUESS FIELD
   15 CONTINUE
      IF (MDHD(8).EQ.0)GO TO 115
C     READ COLUMN HEADER FOR RAOB
      IOK=MDGET(MDNO,0,NN,KBUF)
      IF (IOK.LT.0)GO TO 148
  115 CONTINUE
      IOK=MDGET(MDNO,MDR,NN,KBUF)
      IF (IOK.LT.0)GO TO 148
      IF (KBUF(NMOD).NE.0.AND.IALL.EQ.0)GO TO 148
C     SKIP REPORTS WHICH HAVE BEEN PREVIOUSLY REJECTED
      NS=1
```

```
      IF (ILEV.NE.0)GO TO 44
      DO 43 N=1,NKEYS
      IF (LIST(N).NE.LIT(ICHR))GO TO 43
      NS1=N
      GO TO 48
   43 CONTINUE
      GO TO 148
   44 CONTINUE
C     PICK ON SPECIFIC LEVEL AND CHARACTER
      DO 45 N=1,NKEYS
      IF (LIST(N).NE.LIT('P   '))GO TO 45
      I=LOCS(N)
      IF (ILEV.EQ.KBUF(I))GO TO 31
      IF (CLEV.EQ.CLIT(KBUF(I)))GO TO 31
      GO TO 45
   31 CONTINUE
      NS=N+1
      GO TO 46
   45 CONTINUE
      GO TO 148
   46 CONTINUE
      DO 47 N=NS,NKEYS
      NS1=N
      IF (LIT(ICHR).EQ.LIST(N))GO TO 48
   47 CONTINUE
      GO TO 148
   48 CONTINUE
      KK=KK+1
C     LOOK FOR THICKNESS
      IAD2=0
      IF (IL2.EQ.0)GO TO 56
      DO 52 N=1,NKEYS
      IF (LIST(N).NE.LIT('P   '))GO TO 52
      I=LOCS(N)
      IF (IL2.EQ.KBUF(I))GO TO 51
      IF (CLEV.EQ.CLIT(KBUF(I)))GO TO 51
      GO TO 52
   51 CONTINUE
      NS=N+1
      GO TO 53
   52 CONTINUE
      GO TO 148
   53 CONTINUE
      DO 54 N=NS,NKEYS
      IAD2=LOCS(N)
      IF (LIT(ICHR).EQ.LIST(N))GO TO 55
   54 CONTINUE
      GO TO 148
   55 IF (KBUF(IAD2).EQ.MISS)GO TO 148
   56 CONTINUE
      IADD=LOCS(NS1)
      KCUT(1)=4
```

```
      KOUT(2)=NN
      KOUT(3)=KBUF(NLAT)
      KOUT(4)=KBUF(NLON)
      KOUT(5)=KBUF(IADD)
      IF (KBUG.GT.1)CALL OUTINT(KOUT)
      IF (KBUF(IADD).EQ.MISS)GO TO 148
      IS=ISCL(NS1)
      FSCL=10.**IS
      IDAT=KBUF(IADD)
      IF (IAD2.NE.0)IDAT=IDAT-KBUF(IAD2)
      FLAT=KBUF(NLAT)*.0001
      FLON=KBUF(NLON)*.0001
      IF (FLAT.GT.TLAT.OR.FLAT.LT.SLAT)GO TO 148
      FLATJ=(TLAT-FLAT)/DLAT+1.0
      IF (FLON.LT.ELON.OR.FLON.GT.WLON)GO TO 148
      FLONI=(WLON-FLON)/DLON+1.0
      IDAT=IDAT-IBIAS
      IF (ISUB.NE.0)IDAT=IDAT-KBUF(IADD+ISUB)
      DAT=FLOAT(IDAT)/FSCL
      LL=LL+1
      DA(LL)=DAT
      STLAT(LL)=FLAT
      STLON(LL)=FLON
      IREF(LL)=NN                                  IGB.EQ.Ø
C     ABOVE SAVES ORIGINAL SUBSCRIPT FOR EDITING
C     ABOVE ARE SAVED FOR EDITING
      IF (GGS.NE.('G   '))GO TO 146
      IF (IKWP('EDIT',1,0).LT.0)GO TO 146
      CALL VALUE(LAN,LAS,LOW,LOE,INC,IDL,FSCL,VAL)
      IF (ABS(VAL-DAT).LT.FCK)GO TO 146
C     ABOVE IS GROSS ERROR CHECK
      KOUT(1)=4
      KOUT(2)=FLAT
      KOUT(3)=FLON
      KOUT(4)=DAT
      KOUT(5)=VAL
      CALL OUTINT(KOUT)
      GO TO 148
  146 IF (ICHR.EQ.'DIR'.AND.DAT.LT.0.)DAT=DAT+360.
      IF (ICHR.NE.'SPD')GO TO 70
      IF (LCHR.EQ.'SPD')GO TO 70
C     BREAK OUT APPROPRIATE WIND COMPONENT
      SPD=DAT
      IDIR=KBUF(IADD-1)
      LS=ISCL(NS1-1)
      SSCL=10.**LS
      DIR=FLOAT(IDIR)/SSCL*.0174533
      U=-SPD*SIN(DIR)
      V=-SPD*COS(DIR)
      DAT=U
      IF (LCHR.EQ.'V')DAT=V
   70 CONTINUE
```

```
C       ADD ARRAY FOR BARNES
C       FILL BARNES ARRAYS
        NB=NB+1
        DAS(NB)=DAT
C       THE VALUES OF DAS WILL BE CHANGED IN FBARN
        RW(NB)=FLATJ
        CL(NB)=FLONI
C       COMPUTE STATISTICS
        SUM=SUM+DAT
        SUMS=SUMS+DAT*DAT
        IDATA=0
        IF (NB.LT.NBMAX) GO TO 148
        CALL SDEST(' TOO MUCH DARA...EXITING TO ANALYSIS',0)
        GO TO 140
    148 NN=NN+1
        IF (KK.GT.NRPT)GO TO 140
        IF (NN.LE.MMAX)GO TO 15
    140 IF (KBUG.GT.0)CALL SDEST(' ACTUAL NO. OF RPTS IS ',NB)
        IF (IDATA.EQ.77777.AND.IGB.NE.1)GO TO 400
        IF (IDATA.EQ.77777)GO TO 150
        IF (FN.LT.2)GO TO 145
        FN=NB
        SUM=SUM/FN
        SUMS=SUMS/FN
        SD=DSQRT(SUMS-SUM*SUM)
        SD=AMIN1(ERR,SD)
        SD=AMAX1(1.,SD)
    145 ISD=SD*100.
        IF (ERR.EQ.999999.)ERR=SD
        IF (KBUG.GT.0)CALL SDEST('DARA TOSSOUT 100 IS ',ISD)
C       SET BARNES SCALING
        ISC=IKWP('SCL',1,50)
        IF (CGS.NE.('G    '))GO TO 149
        DO 135 N=1,NPTS
    135 FLD(N)=FLOAT(IDL(N))/FSCL
    149 CALL FBARN(TLAT,NROWS,NCOLS,FLD,WT1,WT2,DAS,RW,CL,
        NB,DLAT,ISC,IGB)
        DO 155 N=1,NPTS
    155 IDL(N)=FLD(N)*FSCL+0.5
        II=FSCL*100.
        KOUT(1)=5
        KOUT(2)=IDL(10)
        KOUT(3)=IDL(20)
        KOUT(4)=IDL(30)
        KOUT(5)=IDL(40)
        KOUT(6)=IDL(50)
        IF (KBUG.GT.1)CALL OUTINT(KOUT)
        IF (KBUG.GT.1)CALL SDEST(' SCALE 100 IS ',II)
C       CHECK ON THE EDIT OPTION
        IF (IKWP('EDIT',1,0).LE.0)GO TO 150
C       NOTE..WE EDIT ALL REPORTS THAT ARE CURRENTLY VALID EVEN IF
C       THEY DID NOT TAKE PART IN THE ANALYSIS
```

```
      DO 130 NN=1,LL
      FLAT=STLAT(NN)
      FLON=STLON(NN)
C     GET VALUE FROM GRID
      CALL VALUE(LAN,LAS,LOW,LOE,INC,IDL,FSCL,VAL)
      NVAL=VAL
      DAT =DA(NN)
      DIF=DAT-VAL
C     GROSS ERROR CHECK
C     ***** ***** *****
      ADIF=ABS(DIF)
      IF(ADIF.LT.ERR) GO TO 130
      ICK=MDGET(MDNO,MDP,IREP(NN),KBUF)
      KBUF(NMOD)=MMOD
      IOK=MDPUT(MDNO,MDP,IREP(NN),KBUF)
      IF (KBUG.LE.0)GO TO 130
      D=DAT-VAL
      CALL ENKODE('(1X,I5,2F7.2,3F10.1/)',LBUF,NN,FLAT,FLON,DAT,VAL,D)
  130 CONTINUE
      RETURN
C     OUTPUT FIELD
  150 CONTINUE
      IF (IADD.EQ.0)IADD=NADD
      IGVNAM=LIT(LCHR)
      IGVSCA=ISCL(IADD)
      IGVUNI=IUN(IADD)
      IGLEVL=LPR
      IGLSCA=1
      IGLUNI=1
      IGVTYP=8
      IGLDIF=1
      IGORG=0
      IGTYPE=1
      IGLAMX=LAN 1000
      IGLAMN=LAS 1000
      IGLOMX=LOW 1000
      IGLOMN=LOE 1000
      IGINCR=INC 1000
      IGDAY=MOD(ISYD,100000)
      IGTIME=IHMS/100
      IGSIZE=NPTS
      IGNR=NROWS
      IGNL=NCOLS
      NGRF=LLC(6)
C     FOLLOWING IS TO STORE RESULT IN SAME GRID FILE AS GUESS GRID
C     AS EXPECTED BY "UCVA"
      IF (NGG.NE.0)NGRF=NGFG
      ICK=IGPUT(NGRF,NGOUT,IDL,NROWS,NCOLS,IGHD,ISTAT)
      IF(KBUG.GE.0)CALL SDEST(' ANALYSIS FILED AS GRID NO.',ISTAT)
      IGVUNI=IBLNK
      IGVSCA=2
      IF (IKWP('WGT ',1,0).EQ.0)GO TO 296
```

```
      IGVNAM=LIT('WGT1')
      DO 295 N=1,NPTS
      IDL(N)=WT1(N)*10000.
  295 CONTINUE
      IF (NGOUT.NE.0)NGOUT=NGOUT-1
      IOK=IGPUT(NGRF,NGOUT,IDL,NROWS,NCOLS,IGHD,ISTAT)
      IF (KBUG.GE.0)CALL SDEST(' WEIGHT1 FILED AS GRID NO.',ISTAT)
  296 IF (IKWP('WGT ',2,0).EQ.0)RETURN
      IGVNAM=LIT('WGT2')
      DO 298 N=1,NPTS
      IDL(N)=WT2(N)*10000.
  298 CONTINUE
      IF (NGOUT.NE.0)NGOUT=NGOUT-1
      IOK=IGPUT(NGRF,NGOUT,IDL,NROWS,NCOLS,IGHD,ISTAT)
      IF (KBUG.GE.0)CALL SDEST(' WEIGHT2 FILED AS GRID NO.',ISTAT)
      RETURN
  300 CALL SDEST(' CANNOT READ GUESS GRID NO. ',NGG)
      RETURN
  400 CALL SDEST(' NO DATA AVAILABLE FOR IMAGE',0)
      RETURN
  900 CALL SDEST(' CANNOT OPEN MDFILE NO. ',MDNO)
      RETURN
  902 CALL SDEST(' CANNOT LOCATE ROW NO. ',MDR)
      RETURN
  904 CALL SDEST(' TROUBLE READING DATA RECORD NO ',N)
      RETURN
      END
```

```
//GWVA5660 JOB CLASS=B,MSGLEVEL=(0,0)
//* VLGWVA  MF  02/28/84: MEMBER UPDATED
//* VLGWVA  SPG  12/23/83; ENTERED USER MANUAL CARD
// EXEC MCPRG,MOD=GWVA
//FORT.SYSIN DD *
@PROCESS SC(TGMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTOR,OPCOM,SGX,SGW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW)
@PROCESS SC(DOPEN,DREAD,DWRITE,ISFILE,WTOR,OPCOM,SGX,SGW)
@PROCESS SC(MDOPEN,MDGET,MDPUT,ISCHAR,ISAN,MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAIN0
C ? PROGRAM TO PRODUCE WINDS FOR  MD USING GRID HEIGHTS
C ?    MD LINKAGE THRU  VASTEXT FILE
C ? GWVA NG LEV <KEYWORDS>
C ?   NG - GRID NO (GRID FILE LINKAGE IS THRU "IGU SET NN")
C ?   LEV- LEVEL(MB)
C ?        IF LEVEL NOT SPECIFIED WINDS ARE NOT WRITTED TO OUTPUT
C ?   KEYWORDS:
C ?    COLOR = COLOR OF WIND BARB PLOTTED ON IMAGE (1,2,3)
C ?    DELT  = NO OF HOURS BETWEEN NG (LATER TIME) AND NGOLD
C ?    NGOLD = GRID NO OF PREVIOUS ANALYSIS
C ?    TYPE  = CONTROLS WIND DERIVATION (G,GR(DEFAULT),AG,IS)
C ?    SFAC  = FACTOR TO MULTIPLY WIND SPEED (DEFAULT=1.0)
C ?    PLT   = NON ZERO SKIPS PLOT
C   SSEC/MCIDAS USERS MANUAL - CHAP12
      DIMENSION KOUT(10),MDHD(64)
      DIMENSION JHITE(2400),JHIT2(2400),JHIT3(2400),ITAB(16)
      DIMENSION IBUF(500),IPRESS(10),SATPOS(2),INDX(10)
      DIMENSION LIST(500),ISCL(500),IUN(500),LOCS(500)
      CHARACTER*8 MFILE
      CHARACTER*12 CKWP,CTYP
      COMMON /NAV/FLAT,FLON,ZENLOC,SZEN,IL,IE,IRAS,IFIC,IHMS,JT,JD
      COMMON /WTYPE/KODE
      COMMON /SAT/DX,DY,SLAT,DINC
      COMMON /SIZE/NPT,NOFF,DXFAC
      COMMON /DIMG/NR,NC
      COMMON /DOC/IDOC(112)
      COMMON /TERMN/ITERM
      COMMON /ANALS/NOAN,LTOP
      COMMON /CREIT/MODE
      COMMON /TIGHT/ITOL
      COMMON /ENTRY/INIT
      EQUIVALENCE (IDOC(40),MDNR),(IDOC(41),MDRR)
      INTEGER IGHD(64)
      INTEGER IGID(8)
C-----GIVE TOTAL SIZE (WORDS), # ROWS, # COLS. (IGSIZE=IGNR*IGNC)
      EQUIVALENCE (IGSIZE,IGHD(1)),(IGNR,IGHD(2)),(IGNC,IGHD(3))
C-----YYDDD, HHMMSS AND VALID-TIME (IF APPLICABLE) FOR GRID
      EQUIVALENCE (IGDAY,IGHD(4)),(IGTIME,IGHD(5)),(IGTIMV,IGHD(6))
C-----DESCRIPTION OF GRIDDED VARIABLE (IN MD-FILE TERMS):
C-----   NAME, SCALE, AND UNITS
      EQUIVALENCE (IGVNAM,IGHD(7)),(IGVSCA,IGHD(8)),(IGVUNI,IGHD(9))
```

```
C-----DESCRIPTION OF VERTICAL LEVEL: VALUE, SCALE, AND UNITS
      EQUIVALENCE (IGLEVL,IGHD(10)),(IGLSCA,IGHD(11)),(IGLUNI,IGHD(12))
C-----GRIDDED-VARIABLE TYPE: =1 (TIME DIF) 2(TIME AVG) 4 (LEVEL DIF)
C-----    8 (LEVEL AVG) OR ANY SUM OF THE FOREGOING
      EQUIVALENCE (IGVTYP,IGHD(13))
C-----GRID ORIGIN, TYPE (I.E. TYPE OF PROJECTION)
      EQUIVALENCE (IGORG,IGHD(33)),(IGTYPE,IGHD(34))
      EQUIVALENCE (IGLAMX,IGHD(35)),(IGLOMX,IGHD(36)),(IGLAMN,IGHD(37))
     *,(IGLOMN,IGHD(38)),(IGINCR,IGHD(39))
      DATA SATPOS/135.,75./
      DATA SIZE/8./
      DATA IPRESS/1000,850,700,500,400,300,250,200,150,100/
C     ABOVE EQUATES PRESS TO RETRIEVAL LEVELS
      DATA IRLEN/18/
C     ABOVE IS THE NO. OF WORDS IN THE ROW HEADER
      DATA MFILE/'VASTEXT '/,LUN/20/,LEN/100/
      DATA SFAC/1./,IPP/0/
C
      CALL INITPL(0,0)
C     FORCE HIGH RESOLUTION SRCH
      ITOL=1
      JFR=LUC(-1)
      IPLT=IKWP('PLT',1,0)
      IF (LUC(81).EQ.0)GO TO 120
C     READ VASTEXT CONTEXT FILE
      ITERM=LUC(-20)
      CALL DOPEN(MFILE,LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      IF (IPLT.NE.0)GO TO 130
C     INITIALIZE NAVIGATION
      IRAS=LUC(-11)
      IPIC=LUC(-12)
      CALL TVSAT(JFR,IRAS,IPIC,IL,IE,ISS,JD,JT)
      JD=ISS 100000+JD
      CALL NVINIT(BETAIN,BETDOT,INAV,PTIME)
      GO TO 130
  120 CALL TSNIO(1,1,1,1,1,1,IDOC)
      NODE=IDOC(7)
      INIT=1
      IF (IPLT.NE.0)GO TO 130
C     SET UP AREA FOR IMAGE SPACE TO PLOT
      CALL TVSAT(JFR,005,335,LTOP,LELE,ISS,ID,IT)
      CALL TVSAT(JFR,495,335,LBOT,LELE,ISS,ID,IT)
      IF (LTOP.LT.1)LTOP=1
      IF (LBOT.GT.NROWM)LBOT=NROWM
      NROWS=LBOT-LTOP+1
      NPT=NROWS 56
  130 CONTINUE
      MDNO=IDOC(40)
      MDR=IDOC(41)
    2 IF (MDOPEN(MDNO,2).NE.0)GO TO 900
C     DO INITIALIZATION
```

```
      CTYP=CKWP('TYPE',1,'GR')
C     ABOVE DEFAULTS TO GEOSTROPHIC+GRADIENT ADJUSTMENT
C
C     SET UP TYPE CHOICES THROUGH KGEO
      KGEO=11
      IF (CTYP.EQ.'G' )KGEO=0001
      IF (CTYP.EQ.'AG')KGEO=1111
      IF (CTYP.EQ.'IS')KGEO=1100
      IF (CTYP.EQ.'CU')KGEO=0100
      IF (CTYP.EQ.'MG')KGEO=1000
      NSAT=0
      IF (LUC(81).EQ.0)GO TO 8
      NSAT=ISATNV(IDOC(1))
      SATLON=-SATPOS(NSAT)
    8 KOL=IKWP('COLOR',1,1)
      NGRF=LUC(6)
      NGR=IPP(1,0)
      IOK=IGGET(NGRF,NGR,2400,JHITE,NR,NC,IGHD)
      IF (IOK.LT.0)GO TO 995
      LEVEL=IGLEVL
      NLV=10
      LEVP=IPP(2,0)
C     ABOVE WILL ALLOW ALL WINDS TO PLOT WHEN NOT STORING
      IF (LEVP.EQ.0)GO TO 15
      IF (LEVEL.EQ.LEVP)GO TO 12
      CALL SDEST(' GRID LEVEL DOES NOT AGREE WITH  KEYED IN..',LEVP)
      RETURN
   12 CONTINUE
      DO 14 I=1,10
      IF (LEVP.EQ.IPRESS(I))NLV=I
   14 CONTINUE
   15 CONTINUE
      LAN=IGLAMX/1000
      LOW=IGLOMX/1000
      LAS=(IGLAMX-IGINCR*NR)/1000
      LOE=(IGLOMX-IGINCR*NC)/1000
      INC=IGINCR/1000
C     INC IS GRIDLENGTH IN TENTHS OF DEGREES LAT
C     THIS PROGRAM ASSUMES THAT INCREMENTS ARE NEVER FINER
      INC2=INC/2
C     GET HORIZONTAL INCREMENT IN METERS
      DINC=FLOAT(INC)*11100.
      DX=DINC
      DY=DINC
  150 LATMAX=(LAN-2*INC-INC2)*10
      LONMAX=(LOW-2*INC-INC2)*10
      LATMIN=(LAS+2*INC+INC2)*10
      LONMIN=(LOE+2*INC+INC2)*10
      NPTS=NR*NC
      DO 160 J=1,NPTS
      JHIT2(J)=JHITE(J)
  160 CONTINUE
```

```
      IFIND=0
      NWIN=0
C     READ IN KEYS
      IF (MDINFO(MDNO,MDHD).NE.0)GO TO 900
      NKEYS=MDKEYS(MDNO,-1,LIST,ISCL,ILN,LOCS)
C     READ TEST RECORD
      M=0
  101 M=M+1
      IF (M.GT.100)GO TO 904
C     READ ROW HEADER
      ICK=MDGET(MDNO,MDR,0,IBUF)
C     READ COLUMN HEADER IF IT EXISTS
      IF (MDHD(9).NE.0)ICK=MDGET(MDNO,0,M,IBUF)
      ICK=MDGET(MDNO,MDR,M,IBUF)
      IF (IOK.LT.0)GO TO 101
C     FIND LAT/LON ADRESS IN KEYLIST
      NCA=0
      DO 11 N=1,NKEYS
      IF (LIST(N).EQ.LIT('CMAX'))NCA=N
      IF (LIST(N).EQ.LIT('NREC'))NCA=N
      IF (LIST(N).EQ.LIT('LAT '))NLAT=N
      IF (LIST(N).EQ.LIT('LON '))NLON=N
   11 CONTINUE
      LAST=500
      IF (NCA.NE.0)LAST=IBUF(NCA)
      LAST=IKWP('LAST',1,LAST)
      KBUG=IKWP('BUG',1,0)
      KOUT(1)=3
      KOUT(2)=NCA
      KOUT(3)=NLAT
      KOUT(4)=NLON
      IF (KBUG.NE.0)CALL OUTINT(KOUT)
      IF (KBUG.NE.0)CALL SDEST(' NO. OF REPORTS IS ',LAST)
      IF (LAST.NE.0)GO TO 10
      CALL SDEST(' NO DARA AVAILABLE',0)
      RETURN
   10 DELT=DKWP('DELT',1,0.)
      NGR=IKWP('NGOLD',1,0)
      IF(DELT.NE.0.)ICK=IGGET(NGRF,NGR,2400,JHIT2,NR,NC,IGHD)
      IF (ICK.LT.0)GO TO 995
      SFAC=DKWP('SFAC',1,1.)
      IPR=0
C     CONVERT HITES TO WHOLE METERS
      ISC=10**IGVSCA
      DO 165 J=1,NPTS
      JHITE(J)=(JHITE(J)+5)/ISC
      JHIT2(J)=(JHIT2(J)+5)/ISC
      JHIT3(J)=(JHITE(J)+JHIT2(J))/2
C     ABOVE IS FOR COMPUTING TRAJECTORY
  165 CONTINUE
      JD=IDOC(1)
      JT=IDOC(2)
```

```
      IFIND=0
      NWIN=0
      DO 200 N=1,LAST
      IFIND=0
      IF (MOHD(8).NE.0)ICK=MDGET(MDNO,0,N,IBUF)
      IOK=MDGET(MDNO,MDR,N,IBUF)
      IF (IOK.LT.0)GO TO 200
      DO 110 L=1,NKEYS
      IF (LIST(L).NE.LIT('P   '))GO TO 110
      IF (IFIND.NE.0)GO TO 13
C     SAVE SFC PRESSURE
      NSFP=L
      IFIND=1
   13 I=LOCS(L)
C     CHECK ON SPECIFIC LEVEL
      IF (IBUF(I)/10.NE.LEVEL)GO TO 110
      NWIN=L+3
  110 CONTINUE
      IF (NWIN.EQ.0.AND.LEVP.NE.0)GO TO 200
C     AVOID WINDS BELOW GROUND
      IF (IBUF(NSFP).LT.IPRESS(NLV))GO TO 200
      LAT=IBUF(NLAT)/100
      JLAT=LAT
      IF (LAT.GT.LATMAX.OR.LAT.LE.LATMIN)GO TO 200
      LON=IBUF(NLON)/100
      IF (LON.GE.LONMAX.OR.LON.LT.LONMIN)GO TO 200
      FLAT=IBUF(NLAT)*.0001
      SLAT=FLAT
      IF (IPLT.NE.0)GO TO 168
      FLON=IBUF(NLON)*.0001
      IF (LUC(81).EQ.0)GO TO 32
C     REVERT TO -W+E CONVENTION
      SLON=-FLON
      CALL SATEAR(PTIME,FLIN,FELE,FLAT,SLON,2,INAV,BETAIN,BETDOT,0.)
      IL=FLIN+0.5
      IE=FELE+0.5
      GO TO 33
C     USE TOVS NAVIGATION ROUTINE
   32 LATS=FLAT*100.
      LONGS=FLON*100.
      CALL SRCH(LATS,LONGS,IM,IL,IE,NPT,NROWS)
      IF (IM.EQ.0)GO TO 200
C     GUARD AGAINST OVERLAP IN SRCH TO AVOID DUPLICATION
      IL=IL+LTOP-1
      IF (IL.LT.LTOP.OR.IL.GE.LBOT)GO TO 200
   33 CALL SATTV(JFR,IL,IE,JF,IF,JS,JD,JT)
  168 I=(LOW-LON/10-INC2)/INC+1
      J=(LAN-LAT/10-INC2)/INC+1
      CALL ZWIND(JHITE,I,J,JLAT,SPEED,DIREC,JHIT2,DELT,JHIT3)
      ISPD=SPEED
      IF (DIREC.LT.0.)DIREC=DIREC+360.
      SPEED=SPEED*SFAC
```

```
      IF (IPLT.NE.0)GO TO 170
      XP=IP
      YP=JP
      FLON=-.01 FLOAT(LON)
      ADJ=DIREC
      IF(NSAT.NE.0)CALL DIRADJ(FLAT,FLON,SATLON,DIREC,ADJ)
      SZ=SIZE
      IF (FLAT.LT.0.)SZ=-SIZE
      CALL BARB(ADJ,SPEED,XP,YP,KOL,SZ)
C     CHECK HEIGHT LEVEL FOR WRITING WIND OB
  170 IF (LEVP.NE.0)GO TO 175
      IF (IPR.NE.0)GO TO 200
      IPR=1
      CALL SDEST(' WINDS NOT WRITTEN TO OUTPUT FILE...',0)
      GO TO 200
  175 CONTINUE
      IDIR=DIREC+0.5
      ISPD=SPEED+0.5
      IBUF(NWIN)=IDIR
      IBUF(NWIN+1)=ISPD
      IOK=MDPUT(MDNO,MDR,N,IBUF)
      IF (IOK.LT.0)GO TO 906
  200 CONTINUE
  300 CONTINUE
      CALL ENDPLT
      CALL SDEST(' DUN...',0)
C
      RETURN
C
  900 CALL SDEST(' CANNOT OPEN MDFILE NO. ',MDNO)
      RETURN
  904 CALL SDEST(' TROUBLE READING DATA RECORDS   ',0)
      RETURN
  906 CALL SDEST(' TROUBLE WRITING DATA RECORD NO ',N)
      RETURN
  995 CALL SDEST(' UNABLE TO READ GRID  ICK= ',ICK)
      RETURN
C
      END
```

```
//UGVA5670 JOB CLASS=A,MSGLEVEL=(0,0)
//* VLUGVA   L    03/27/84: MEMBER UPDATED
//* VLUGVA   SEG  12/23/83; ENTERED USER MANUAL CARD
// EXEC MCPRG,MOD=UGVA
//FORT.SYSIN DD *
@PROCESS SC(TGMES,LIT,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,TQ,DOPEN,DCLOSE,DREAD,DWRITE,WTCR,CPCOM,SQX,SQW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAIN0
C ?     UPDATE THE CURRENT GUESS GRID FILE USING RETRIEVALS
C ?     GRID FILE POINTER DEFAULTS TO VASTEXT
C ?     KEYIN: UGVA <NGRID>
C ?          POSITIONAL PARAMETERS:
C ?          <NGRID> BEGINNING NO. OF GRIDS TO BE UPDATED (NO DEFAULT)
C ?                  GRID FILE DEFAULTS TO GUESS GRID OF VASTEXT
C    SSEC/MCIDAS USER MANUAL - CHAP12
      COMMON /DCC/IDCC(100)
      DIMENSION LIST(26),KOUT(10),ISCALE(26),IUNIT(26),LOCS(26)
C     ABOVE TO RECEIVE MDKEY INFO
      INTEGER*4 DBUF(20)
      DIMENSION IGRID(3200),MES(64),JREC(26),IREC(26),FISC(22)
      DIMENSION ICHAR(22),KCHAR(22),JCHAR(22),IPRESS(22),IOCT(22)
      DIMENSION LCHAR(22)
      INTEGER IGHD(64),IGHDT(64),IGID(8),ROW(3),COL
C-----DESCRIPTION OF 64-WORD GRID HEADERS
C
C-----GIVE TOTAL SIZE (WORDS), # ROWS, # COLS. (IGSIZE=IGNR*IGNC)
      EQUIVALENCE (IGSIZE,IGHD(1)),(IGNR,IGHD(2)),(IGNC,IGHD(3))
C-----YYDDD, HHMMSS AND VALID-TIME (IF APPLICABLE) FOR GRID
      EQUIVALENCE (IGDAY,IGHD(4)),(IGTIME,IGHD(5)),(IGTIMV,IGHD(6))
C-----DESCRIPTION OF GRIDDED VARIABLE (IN MD-FILE TERMS):
C-----    NAME, SCALE, AND UNITS
      EQUIVALENCE (IGVNAM,IGHD(7)),(IGVSCA,IGHD(8)),(IGVUNI,IGHD(9))
C-----DESCRIPTION OF VERTICAL LEVEL: VALUE, SCALE, AND UNITS
      EQUIVALENCE (IGLEVL,IGHD(10)),(IGLSCA,IGHD(11)),(IGLUNI,IGHD(12))
C-----GRIDDED-VARIABLE TYPE: =1 (TIME DIF) 2(TIME AVG) 4 (LEVEL DIF)
C-----    8 (LEVEL AVG) OR ANY SUM OF THE FOREGOING
      EQUIVALENCE (IGVTYP,IGHD(13))
C-----FOLLOWING USED IF PARAMETER IS A VERTICAL (LEVEL) DIF OR AVG
C-----    (SAME SCALE AS IGLEVL)
      EQUIVALENCE (IGLDIF,IGHD(14))
C-----FOLLOWING USED IF PARAMETER IS A TIME DIF OR AVG (HHMMSS)
      EQUIVALENCE (IGTDIF,IGHD(15))
C-----GRID ORIGIN, TYPE (I.E. TYPE OF PROJECTION)
      EQUIVALENCE (IGORG,IGHD(33)),(IGTYPE,IGHD(34))
C-----SUBSEQUENT COORDS (IGLAMX,IGLOMX,IGLAMN,IGLOMN,IGINCR,
C-----    IGPOLR,IGPOLC,IGSP60,IGCLON) ALL HAVE 4 IMPLIED DEC. PLACES.
C-----    LAT GOES FROM -900000 TO 900000, LON GOES FROM -1800000
C-----    TO 1800000 (WEST IS +)
C-----TYPE 1 GRIDS ARE PSEUDO-MERCATOR
      EQUIVALENCE (IGLAMX,IGHD(35)),(IGLOMX,IGHD(36)),(IGLAMN,IGHD(37))
     *,(IGLOMN,IGHD(38)),(IGINCR,IGHD(39))
```

```
C-----TYPE 2 GRIDS ARE POLAR-STEREOGRAPHIC
C-----GIVE ROW # OF NORTH POLE, COL # OF N.P., COL SPACING AT 60 DEG N
C-----     (DEG), LONGITUDE PARALLEL TO COLUMNS (DEG)
      EQUIVALENCE (IGPOLR,IGHD(35)),(IGPOLC,IGHD(36)),(IGSP60,IGHD(3 )
     ,(IGCLON,IGHD(38))
C-----INITIALS OF USER AND PROJECT # UNDER WHICH GRID CREATED
      EQUIVALENCE (IGUSER,IGHD(41)),(IGPROJ,IGHD(42))
C-----CHARACTER ID SUPPLIED BY PROGRAM (ARBITRARY)
      EQUIVALENCE (IGID,IGHD(43))
      EQUIVALENCE (ROW(1),IREC(1))
      EQUIVALENCE (COL,IREC(4))
      DATA OBUF/20*'    '/
      DATA IPRESS/1000,850,700,500,400,300,250,200,150,100,70,50,30, 0
     10,1000,850,700,500,400,300,1000/
C     FOLLOWING 2 SPECIAL CASES FOR CALLAN'S ANMRC AND ECMWF FIASCOS
      DATA LCHAR/15*' TMP',6*' DPT',' HGT'/
      DATA ICHAR/15*'TEMP',6*'TDPT','HGHT'/
      DATA KCHAR/15*'   T',6*'  TD','   H'/
      DATA JCHAR/15*'T   ',6*'TD  ','Z   '/
      DATA IGOT/22*0/
      DATA MISG/Z80808080/
      DATA LUN/20/,LEN/100/
      DATA IBIAS /0/
C
C
C
      NGB=IPP(1,0)
      IF (NGB.EQ.0)GO TO 900
      ITERM=LUC(-20)
      CALL DOPEN('VASTEXT ',LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      NGFG=IDOC(42)
  100 NGE=NGB+70
      L=0
      DO 10 N=NGE,NGE
      IOK=IGGET(NGFG,N,3200,IGRID(1),NR,NC,IGHDT)
C     FOLLOWING NECESSARY BECAUSE OF GRIDS WITH HOLES
      ITVNAM=IGHDT(7)
      ITLEVL=IGHDT(10)
      IF (IOK.EQ.0)GO TO 2
      IF (IOK.LT.-1)GO TO 930
      GO TO 10
    2 CONTINUE
      NPT=NR NC
      DO 3 I=1,22
      JE=I
      IF (LCHAR(I).EQ.ITVNAM)GO TO 4
      IF (ICHAR(I).EQ.ITVNAM)GO TO 4
      IF (KCHAR(I).EQ.ITVNAM)GO TO 4
      IF (JCHAR(I).EQ.ITVNAM)GO TO 4
    3 CONTINUE
      GO TO 10
```

```
    4 DO 5 J=JB,22
      IF (IGOT(J).EQ.1)GO TO 5
      L=J
      IF (IPRESS(J).EQ.ITLEVL)GO TO 6
    5 CONTINUE
      GO TO 10
    6 CONTINUE
      DO 80 I=1,64
   80 IGHD(I)=IGHCT(I)
      KOUT(1)=6
      KOUT(2)=IGDAY
      KOUT(3)=IGTIME
      KOUT(4)=IGLEVL
      KOUT(5)=L
      KOUT(6)=IGNR
      KOUT(7)=IGNC
C     CALL OUTINT(KOUT)
      IGOT(L)=1
C     SET UP TG FOR GUESS UPDATE
      CALL ENKODE(' (A4,8X,K5,7X,K5,7X,K5,7X)',OBUF,JCHAR(JB),ITLEVL,N,
     *IBIAS)
      CALL TG(OBUF)
      CALL JSGX('ENVA       ',OBUF(1),OBUF(4),OBUF(7),OBUF(10))
      NGOT=0
      DO 11 J=1,22
   11 NGOT=NGOT+IGOT(J)
      IF(NGOT.EQ.22) GO TO 12
   10 CONTINUE
      IF (L.EQ.0)GO TO 950
   12 CONTINUE
      CALL SDEST(' UPDATING GUESS FOR DAY ',IGDAY)
      CALL SDEST(' HOUR =   ',IGTIME)
      CALL SDEST(' VALID TIME = ',IGTIMV)
      RETURN
  900 CALL SDEST( ' <FIRST GRID NO.>  ',0)
      RETURN
  930 CALL SDEST(' UNABLE TO OPEN GRID FILE NO. ',NGFG)
      RETURN
  950 CALL SDEST(' CANNOT FIND ONE SINGLE LOUSY GRID',0)
      RETURN
      END
```

```
//EXVA7000 JOB  CLASS=B,MSGLEVEL=(0,0)
//  EXEC  MCPROC
//  VLEXVA   SBG  12/23/83; ENTERED USER MANUAL CARD
//FORT.SYSIN  DD  *
@PROCESS SC(TCMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(DOPEN,DCLOSE,DREAD,DWRITE,ISFILE,WTOR,CFCOM,SCX,SQW)
@PROCESS SC(MDINFO,MDGET,ISCHAR,ISAN,MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAIN0
C ?   PROGRAM TO EXAMINE RETRIEVAL MDFILES USING CURSOR ON TV IMAGE
C ?      KEYIN:  EXVA 'KEYWORD'
C ?            'SLOP'  SEARCH RADIUS IN DEG LAT TO FIND REPORT
C   SSEC/MCIDAS USERS MANUAL - CHAP12
      COMMON /NAV/FLAT,FLON,ZENLOC,SZEN,IL,IE,IRAS,IPIC,IHMS,JT,JD
      DIMENSION MDD(64),IOUT(300),KOUT(10),NOUT(10)
      DIMENSION LIST(300),ISCL(300),IUN(300),LOCS(300)
      DIMENSION ILV(8)
      CHARACTER 8 MFILE
      COMMON /DOC/IDOC(112)
      DATA MFILE,LUN,LEN/'VASTEXT ',20,100/
      DATA MISS/Z80808080/
      DATA ILV/1,5,7,9,10,11,12,13/
      IHEL=IKWP('HELP',1,0)
      IF(IHEL.EQ.LIT('HELP')) GO TO 270
      IFRM=LUC(-1)
      IRAS=LUC(-11)
      IPIC=LUC(-12)
      CALL TVSAT(IFRM,IRAS,IPIC,IL,IE,JS,JD,JT)
C     READ IN DOCUMENTATION
      IF (LUC(81).EQ.0)GO TO 120
      ITERM=LUC(-20)
      CALL DOPEN(MFILE,LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      CALL DCLOSE(LUN)
C     INITIALIZE NAVIGATION
      JD=JS*100000+JD
      CALL NVINIT(BETAIN,BETOUT,INAV,PTIME)
      GO TO 122
  120 CALL TSNIC(1,1,1,1,1,1,IDOC)
  122 MDNO=IDOC(40)
      MDR=IDOC(41)
C     MUST CALL MDOPEN TO USE MDGET
      IF (MDOPEN(MDNO,1).NE.0)GO TO 900
C     READ IN KEYS
      NKEYS=MDKEYS(MDNO,-1,LIST,ISCL,IUN,LOCS)
C SET UP INDICES
      DO 11 N=1,NKEYS
      IF (LIST(N).EQ.LIT('LAT '))NLAT=N
      IF (LIST(N).EQ.LIT('LON '))NLON=N
      IF (LIST(N).NE.LIT('T   '))GO TO 11
      NDX=N
      GO TO 12
   11 CONTINUE
```

```
   12 CONTINUE
      ICK=MDINFO(MDNO,MDC)
      IF (ICK.LT.0)GO TO 500
      ICK=MDGET(MDNO,MDR,0,IOUT)
      MREC=IOUT(3)
      SLOP=DKWP('SLOP',1,0.75)
      N=IPP(1,0)
      M=IPP(2,0)
      IF (M.EQ.0)M=N
      ICUR=0
      IF (N.EQ.0)GO TO 160
  130 IF (N.GT.MREC)GO TO 280
      NN=N
      ICK=MDGET(MDNO,MDR,NN,IOUT)
      FLATR=.0001 IOUT(NLAT)
      FLONR=.0001 IOUT(NLON)
      IF (ICK.LT.0)GO TO 200
      GO TO 210
  160 CONTINUE
      ICUR=1
      IF (LUC(81).EQ.0)GO TO 162
      FLIN=FLOAT(IL)
      FELE=FLOAT(IE)
      CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON,1,INAV,BETAIN,BETDOT,0.)
      GO TO 164
  162 CALL TSNIO(0,1,IL,IE,1,1,ILAT)
      CALL TSNIO(0,2,IL,IE,1,1,ILON)
      FLAT=.01 FLOAT(ILAT)
      FLON=.01*FLOAT(ILON)
  164 CONTINUE
C     CONVERT TO +W-E CONVENTION
      FLON=-FLON
C     NAV COMPLETE
      DO 200 N=1,MREC
      ICK=MDGET(MDNO,MDR,N,IOUT)
      IF (IOK.LT.0)GO TO 200
      FLATR=.0001 FLOAT(IOUT(NLAT))
      FLONR=.0001 FLOAT(IOUT(NLON))
      FCK=ABS(FLAT-FLATR)+ABS(FLON-FLONR)
      IF (FCK.GT.SLOP)GO TO 200
      NN=N
      GO TO 210
  200 CONTINUE
      CALL SDEST(' NO SOUNDING AT THIS LOCATION ...',0)
      GO TO 290
  210 CALL SDEST('       LAT      LON     NMBR (RE-EXAMINE)',0)
      KOUT(1)=3
      KOUT(2)=FLATR*100.
      KOUT(3)=FLONR*100.
      KOUT(4)=NN
      CALL OUTINT(KOUT)
      CALL SDEST('      TSFC      T850      T700      T500      T400      T300
```

```
      250    T200            ',0)
      KOUT(1)=8
      MOUT(1)=8
      L=1
      DO 220 K=1,8
      L=L+1
      MM=(ILV(K)-1)*9+NDX
      KOUT(L)=999
      MOUT(L)=999
      IF (IOUT(MM).EQ.MISS)GO TO 220
      KOUT(L)=IOUT(MM)
      MOUT(L)=IOUT(MM)-IOUT(MM+5)
  220 CONTINUE
      CALL OUTINT(KOUT)
      CALL OUTINT(MOUT)
      IF(ICUR.NE.0) GO TO 290
      IF(N.EQ.0) GO TO 290
      N=N+1
      IF(N.GT.M) GO TO 290
      GO TO 130
  270 CALL SDEST(' (NUM-STRT) (NUM-STOP)',0)
      CALL SDEST(' IF NO NUM, PICK UP RPT AT CURSOR LOCATION',0)
      GO TO 290
  280 CALL SDEST(' NO SOUNDING AT RECORD NUMBER',N)
  290 RETURN
  900 CALL SDEST(' CANNOT OPEN MD FILE ..USE MDU SET',0)
      RETURN
      END
```

```
//VDVA7000 JOB CLASS=A,MSGLEVEL=(0,0)
//  VLVDVA   [illegible]  12/23/83; ENTERED USER MANUAL CARD
// EXEC MCPROC,MOD=VDVA
//FORT.SYSIN DD
@PROCESS SC(TGMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTCR,CPCOM,SQX,SQW)
@PROCESS SC(DOPEN,DREAD,DWRITE,MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAIN0
C ?   ROUTINE TO PICK UP VAS DATA ACCORDING TO CURSOR POSITION
C ?    AND SOUNDER-AREA POINTER   (SET BY VPVA)
C ?
C ?   PARAMETERS ARE:
C ?   (B/R/F/S/N..DEFAULT B)  (LINE)  (ELEMENT)  (DIAGNOSTIC)
C ?   BRIGHTNESS T; RADIANCE; FILTER; SPIN BUDGET; NAVIGATION
C    SSEC/MCIDAS USERS MANUAL - CHAP12
      DIMENSION IBUF(400),VDAT(13),MIN(10),KOUT(10)
      DIMENSION MOUT(10),NOUT(10),KKOUT(10,3)
      COMMON /DLIS/LISDAT
      COMMON /LAST/LASLIN,LASELE,LELEV,ICHAR
      COMMON /MODE/IFIL(13),ISPN(13)
      COMMON /NAV/FLAT,FLON,ZENLOC,SZEN,IL,IE,IRAS,IPIC,IHMS,JT,JD
      COMMON /RADV/RAD(13)
      COMMON /SFCGNC/NG(3)
      EQUIVALENCE (KOUT(1),KKOUT(1,1)),(MOUT(1),KKOUT(1,2)),
     *(NOUT(1),KKOUT(1,3))
      DIMENSION SCALE(12)
      DATA SCALE/5*10000.,100000.,4*10000.,2*100000./
      CALL IQ(MIN)
      IF (MIN(1).EQ.LIT('HEL ').OR.MIN(1).EQ.LIT('HELP'))GO TO 190
      LISDAT=MIN(4)
      IP=1
      IF (MIN(1).EQ.LIT('F   '))IP=2
      IF (MIN(1).EQ.LIT('S   '))IP=3
      IL=MIN(2)
      IE=MIN(3)
      IF (MIN(2).NE.0)GO TO 110
      IFRAME=LUC(-1)
      IRAS=LUC(-11)
      IPIC=LUC(-12)
      CALL TVSAT(IFRAME,IRAS,IPIC,IL,IE,IS,JD,JT)
      JD=IS*100000+JD
  110 IF (IP.NE.1)GO TO 130
      IF (MIN(1).EQ.LIT('B   ').OR.MIN(1).EQ.LIT('R   '))GO TO 130
      IF (MIN(1).EQ.LIT('N   '))VDAT(1)=-1.
      CALL VASDAT(IL,IE,VDAT)
      CALL SDEST(' JDAY =',JD)
      CALL SDEST('        LINE    ELEM        LAT       LON     LZEN      SZEN
     LEV     SURF',0)
      KOUT(1)=8
      KOUT(2)=IL
      KOUT(3)=IE
      KOUT(4)=IROUND(FLAT*100.)
```

```
      KOUT(5)=IROUND(FLON*100.)
      KOUT(6)=IROUND(ZENLOC*100.)
      KOUT(7)=IROUND(SZEN*100.)
      LATT=KOUT(4)
      LONN=-KOUT(5)
      CALL HRTOPO(LATT,LONN,LELEV,ICHAR)
      KOUT(8)=LELEV
      ISURF=0
      IF(ICHAR.NE.0) ISURF=1
      KOUT(9)=ISURF
      CALL OUTINT(KOUT)
      IF(MIN(1).NE.LIT('N   ')) GO TO 140
      RETURN
  130 CONTINUE
C     PICK UP DATA WITHOUT NAVIGATION
      JD=0
      CALL VASDAT(IL,IE,VDAT)
  140 CONTINUE
      CALL SDEST('   CHAN1   CHAN2   CHAN3   CHAN4   CHAN5   CHAN6   CH
     N7   CHAN8',0)
      SC=100.
      DO 160 K=1,8
      KOUT(K+1)=VDAT(K)
      IF (MIN(1).NE.LIT('R   '))GO TO 150
      VDAT(K)=RAD(K)
      SC=SCALE(K)
  150 IF (VDAT(K).NE.999999.)KOUT(K+1)=VDAT(K)*SC
      MOUT(K+1)=IFIL(K)
      NOUT(K+1)=ISPN(K)
  160 CONTINUE
      KKOUT(1,IP)=8
      CALL OUTINT(KKOUT(1,IP))
      CALL SDEST('    CHAN9   CHAN10  CHAN11  CHAN12 ',0)
      DO 180 K=1,4
      KOUT(K+1)=VDAT(K+8)
      IF (MIN(1).NE.LIT('R   '))GO TO 170
      VDAT(K+8)=RAD(K+8)
      SC=SCALE(K+8)
  170 IF(VDAT(K+8).NE.999999.) KOUT(K+1)=VDAT(K+8)*SC
      MOUT(K+1)=IFIL(K+8)
      NOUT(K+1)=ISPN(K+8)
  180 CONTINUE
      KKOUT(1,IP)=4
      CALL OUTINT(KKOUT(1,IP))
      RETURN
  190 CONTINUE
      CALL SDEST('(B/R/F/S/N...DEFAULT=B) (LINE) (ELEM) (VASDAT DIAGNOST
     IC DISPLAY',0)
C     CALL SDEST('... N (NAV) GIVES SFC DATA IF AVAIL ... RTVL-FILE P R
C    *MUST BE SET',0)
      RETURN
      END
```

```
//SRAD5670 JOB CLASS=B,MSGLEVEL=(0,0)
//  VLSRAD    N     02/13/84: MEMBER UPDATED
// EXEC MCPRG
//FORT.SYSIN DD *
      SUBROUTINE MAINC
C ?   VAS RADIANCE PROGRAM;USES KEYWORDS OR 'VASTEXT' DEFAULTS
C ?      KEYIN: SRAD <ACTION>
C ?         POSITIONAL PARAMETER:
C ?         ACTION: 'CLE' TO CLEAR KEYWORD LIST
C ?                 'GO'  TO BEGIN MAKING RETRIEVALS
C ?            KEYWORDS:
C ?            TYP=F (FORCE) N (NSTAR) C (CLEAR) O (OVCST) B (DEFAULT)
C ?            SPC=IL IE (LINE AND ELEMENT SPACING OF RETRIEVALS (10*10))
C ?            SIZ=N (FOV AVERAGED BOX,SYMMETRIC (5*5))
C ?            SFC=1 (NO SURFACE ANALYSIS OPTION) (DEFAULT=0)
C ?            END=LL LE (LAST LINE AND ELEMENT TO DEFINE AREA (VASTEXT))
C ?            BEG=IL IE (FIRST LINE AND ELEMENT TO DEFINE AREA (CURSCR))
C ?            TER=N (TERMINAL NUMBER TO DEFINE 'VASTEXT' (LOCAL))
C ?            BUG=1 (FOR DEBUG DIAGNOSTICS (0))
C ?            PLT=1 (PLOT ON GRAPHICS (0))
C ?            ICB=1 (MAKE CALIBRATION RUN USING GUESS)
C ?            ALTC=1 (TAKE IL,IE,LL,LE FROM VASTEXT)
C ?            OLD=1  DATA IS BEFORE JAN 15 1982
C ?            NOPC=1  DO NOT USE THE NSTAR OPTION
      DIMENSION IFILNM(2),KOUT(10),ICU(3),ICT(6),IHLM(3)
     ,VDAT(13),BDAT(13)
     ,NSAM(13),AVG(13),RADS(121,13),NRAS(121)
     ,MF(64),LBLF(33)
      DIMENSION LC(13),NSPIN(13),ICZ(15),IUSE(13),DELS(6),HUM(3),TD(40)
      DIMENSION KUSE(12),TSAV(40),IRET(246),IRETD(246),EX(12)
      DIMENSION TGS(40),LMD(20),LCHR(20),DGS(11)
      DIMENSION PST(15),TST(15),TDST(15),SPD(15),DIR(15),STABIL(12)
      CHARACTER 12 CGES,CFP,CKWP,CTYP,CTYPS
      COMMON /DANGLE/PLAT,PLON
      COMMON/SDOC/ISDOC(6)
      COMMON/DOC/IDOC(100)
      COMMON/TERMN/ITERM
      COMMON/AUTO/IBOX,ILU,ILL,IEL,IER,MAG
      COMMON/SIZE/NBXS
      COMMON/SFC/IS,PWR,DEL7,DEL8
      COMMON/ATMOS/P(40),T(40),W(40)
      COMMON/ARENT/IDIR(64)
      COMMON/GUESS/TGES(15),DGES(6)
      COMMON/LAST/LASLIN,LASELE,LELEV,ICHAR
      COMMON/RADV/SRAD(13)
      COMMON/SURF/IZ10,ITSFC,IDSFC,IPSTA,IELEV,LSTA
      COMMON/NAV/VLAT,VLON,VZEN,SZEN,IL,IE,IRAS,IPIC,ITIME,JTIME,JDAY
      COMMON/MODE/IDET(13),ISPIN(13)
      COMMON/GDE/GV(12),DV(12),EV(12)
      COMMON/USE/ILCH(12,2)
      COMMON/DBUG/KBUG
      COMMON /JAN/IOLD
```

```
C      IF IOLD NE 0. WE USE THE FIXED DELTA F RATHER THAN LINE
C      DOCUMENTATION (DATA BEFORE 15 JAN 82
       EQUIVALENCE (IRET(17),IRETD(1))
C      EQUIVALENCE TO IRET(ROWLEN-1)
C      ABOVE TO FACILITATE CHANGING LENGTH OF ROW HEADER
       DATA DIR/15 0./,SPD/15 0./
       DATA IRET/246 Z80808080/,NSIZE/245/
       DATA LMD/0,40,39,38,37,36,35,34,31,28,26,25,24,23,20,
        18,16,15,13,11/
       DATA LCHR/'SFC ','1000','950 ','920 ','850 ','780 ','700 ',
           '670 ','500 ','400 ','300 ','250 ','200 ','150 ','100 ',
           '70  ','50  ','30  ','20  ','10  '/
       DATA ICT/1,2,3,4,5,6/
       DATA ICL/7,9,10/
       DATA ICW/8/
       DATA ICZ/40,37,35,31,28,26,25,24,23,20,18,16,15,13,11/
       DATA ITCH/6/
       DATA RCG/29.2898/,GRAV/980.665/,TCTC/347./
       DATA VMISG/999999./,MISG/Z80808080/
       DATA NL/40/,NSMAX/121/,MAXRET/699/
       DATA ILINE/0/,IELEM/0/
C
C      FOLLOWING IS PROGRAM VERSION DATE ... KEEP CURRENT!!!
       DATA IVER/84023/
C *****
       CALL CALDAY(IVER,IVY,IVM,IVD,IVMO)
       CALL ENKODE('(132X,T1,"BEGIN SRAD, VERSION OF ",
        I2,1X,A4,I2/)',LBUF,IVD,IVMO,IVY)
C
C      READ IN VASTEXT AND ROW HEADER
       CALL VRTIO(IRET,0,0)
       CALL MDNAME(IDOC(40),IFILNM)
C      ABOVE SETS UP FILE NAME TO FORCE OUTPUT AT END OF EACH LINE
       IF (CPP(1,' ').NE.'CLE')GO TO 100
       DO 50 I=50,60
    50 IDOC(I)=0
       IDOC(52)=10
       IDOC(53)=5
       CALL VRTIO(IRET,0,1)
       RETURN
   100 CONTINUE
       LLNW=IDOC(25)
       LLSE=IDOC(26)
       LN=LLNW/1000
       LS=LLSE/1000
       LW=MOD(LLNW,1000)
       LE=MOD(LLSE,1000)
C       ABOVE WILL LIMIT AREA OF RETRIEVALS
C      TEST FOR CALIBRATION RUN
       ICB=IKWP('ICB',1,0)
       ITYP=IDOC(50)
       CTYP=CKWP('TYP',1,' ')
```

```
      IF (CTYP.NE.' ')GO TO 105
      IF (ITYP.EQ.21)CTYP='C'
      IF (ITYP.EQ.22)CTYP='N'
      IF (ITYP.EQ.23)CTYP='F'
  105 FORCE=0.
      IF(CTYP.EQ.'F')FORCE=10.
      IF (CTYP.EQ.'C')FORCE=10.
      IF (CTYP.EQ.'C')FORCE=10.
C     'FORCE' USED AS NON ZERO FLAG TO SKIP GROSS ERROR CHECKS
  110 CONTINUE
      ITYP=0
      IF (CTYP.EQ.'C')ITYP=21
      IF (CTYP.EQ.'N')ITYP=22
      IF (CTYP.EQ.'F')ITYP=23
      IDOC(50)=ITYP
      IDEF=IDOC(54)
      NOSFC=IKWP('SFC',1,IDEF)
      IDOC(54)=NOSFC
      IDEF=IDOC(52)
      INCRL=IKWP('SPC',1,IDEF)
      IDOC(52)=INCRL
      INCRE=IKWP('SPC',2,INCRL)
      IDEF=IDOC(53)
      NBXS=IKWP('SIZ',1,IDEF)
      IF(MOD(NBXS,2).EQ.0) NBXS=NBXS-1
      IF (NBXS.GT.11)NBXS=11
C     MAX SIZE FOR RETRIEVAL IS 11 11
      IF(NBXS.LT.3) NBXS=3
      IDOC(53)=NBXS
      NSMAX=NBXS*NBXS
      IOLD=IKWP('OLD',1,0)
      KBUG=IKWP('BUG',1,0)
      IF (KBUG.NE.0)CALL TQSET(KBUG)
      IDEF=ITERM
C     USE KEYWORD FOR TERM WHEN RUNNING FROM BACKGROUND
      ITERM=IKWP('TER',1,IDEF)
      IDOC(59)=ITERM
      IDEF=IDOC(60)
      IPLT=IKWP('PLT',1,IDEF)
      IDOC(60)=IPLT
      NLP1=NL+1
C     SET INDICATOR OR SQUARE SAMPLES (IN VASCAT PROCESSING)
      IBOXS=1
C     SET UP PROCESSING DAY FOR DOCUMENTATION IN PCW HEADER
      CALL GETDAY(IDPROS)
      MDNG=IDOC(38)
      MDNR=IDOC(40)
      LASRET=IRET(3)
      IFRM=LUC(-1)
      INRAS=LUC(-11)
      INPIC=LUC(-12)
C     SEE IF THIS IS AN AUTO MODE RUN
```

```
      IF (IKWP('ALTO',1,0).EQ.0)GO TO 112
      LLINE=IDOC(55)
      LELEM=IDOC(56)
      ILINE=IDOC(57)
      IELEM=IDOC(58)
  112 CONTINUE
      LLINE=IKWP('END',1,LLINE)
      LELEM=IKWP('END',2,LELEM)
      ILINE=IKWP('BEG',1,ILINE)
      IELEM=IKWP('BEG',2,IELEM)
      JTIME=IDOC(2)
      JDAY=IDOC(1)
      CALL VRTIO(IRET,0,1)
      IF (CPP(1,' ').NE.'GO')RETURN
      IF (ILINE IELEM.NE.0)GO TO 218
      CALL TVSAT(IFRM,INRAS,INPIC,ILINE,IELEM,JSAT,JDAY,JTIME)
C     RESET JDAY FOR NAVIGATION WITHIN VASDAT
      JDAY=JSAT 100000+JDAY
  218 IF((LLINE LELEM).NE.0) GO TO 220
      LLINE=ILINE
      LELEM=IELEM
  220 CONTINUE
      VDAT(1)=-1.
      ILINES=ILINE
      IELEMS=IELEM
      LASLIN=0
C **  VASDAT CHANGES ARGUMENTS 'ILINES' AND 'IELEMS'
      CALL VASDAT(ILINES,IELEMS,VDAT)
      VDAT(1)=VMISG
      ISAT=ISATNV(IDIR(3))
      ILRES=IDIR(12)
      IERES=IDIR(13)
C     SET DEFAULT TO SQUARE SAMPLES..IE EQUATE RESOLUTIONS
      INC=ILRES/IERES
      INCIL=ILRES INCRL
      IF (INCIL.EQ.0)INCIL=1
      INCIE=IERES INCRE*INC
      IF (INCIE.EQ.0)INCIE=1
      CALL GETFRM(IFRM,MF)
      MAG=MF(10)
      MAG2=MAG/2
      MAG=MAX0(MAG,6)
      IDEL=INCIE/2
      CTYPS=CTYP
      ITYPS=ITYP
      DO 2400 IL=ILINE,LLINE,INCIL
      LASLIN=-1
C     ABOVE IS NECESSARY FOR REPEATED CALLS TO VASDAT
      LBEG=0
      NOGO=0
      IEP=IELEM-INCIE
C   **BEGIN IMPLICIT DO LOOP FOR LINE
```

```
    230 IEP=IEP+INCIE
        IF(IEP.GT.LELEM)GO TO 2385
        CTYP=CTYPS
        ITYP=ITYPS
        LASELE=-1
        IFLG=0
C       BACK UP HALF INCREMENT IF PREVIOUS ATTEMPT FAILED
        IF (NOGO.NE.0)IEP=IEP-IDEL
        DO 300 I=1,13
        DO 280 N=1,NSMAX
    280 RADS(N,I)=VMISG
        IE=IEP
        RDAT(I)=VMISG
        AVG(I)=0.
        VDAT(I)=VMISG
        IUSE(I)=0
        NSPIN(I)=0
    300 NSAM(I)=0
        MSAM=0
        SELEV=0.
        SLAT=0.
        SLON=0.
        IBOX=IBOXS
        N=0
C
C ***   BEGIN IMPLICIT LOOP TO OBTAIN ALL DATA FOR RETRIEVAL
C       COMPLETION IS FLAGGED BY IBOX=3 SET IN VASDAT
    305 N=N+1
        VDAT(1)=0.
        ILINES=IL
        IELES=IE
        CALL VASDAT(ILINES,IELES,VDAT)
        IF(N.GT.1) GO TO 330
C       CHECK CENTER LAT/LON/ZENITH
        LA=VLAT
        IF (LA.GT.LN.OR.LA.LT.LS)GO TO 2400
        TLON=VLON
        IF (TLON.LT.0)TLON=360.+TLON
        LO=TLON
        IF (LO.GT.LW)GO TO 2190
        IF (LO.LT.LE)GO TO 2390
        IF(ABS(VZEN).LE.60.) GO TO 330
        IF (LBEG.EQ.1)GO TO 2390
        GO TO 2190
    330 IF(VDAT(1).LT.0.) GO TO 420
        IF (KBUG.EQ.9)
       CALL ENKODE('(T2,"N,LAT,LON,EL,CHAN 5-8 ",I3,2F7.2,I5, 4F7.2/)',
        LBUF,N,PLAT,PLON,LELEV,VDAT(5))
C       FOLLOWING STATEMENTS ARE COMMENTED BECAUSE OF SMALL DETECTOR
CCCCCCIF(VDAT(ICW).EQ.VMISG)CGOCTOC420CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCDOC340CI=1,3CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCICV=ICL(I)CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCIF(VDAT(ICV).NE.VMISG)GOCTOC360CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC340CCONTINUECCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCGOCTOC420CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC360CCONTINUECCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMF=1
      DO 400 I=1,13
      IF(VDAT(I).LT.180.OR.VDAT(I).GT.330.) VDAT(I)=VMISG
      IF(VDAT(I).EQ.VMISG) GO TO 400
      IF(IMF.EQ.0) GO TO 380
      MSAM=MSAM+1
      IMF=0
  380 CONTINUE
      RADS(N,I)=VDAT(I)
C     SAVE REPRESENTATIVE SPIN BLOC..WE ASSUME SPIN BUDGET DO NOT CH
      NSPIN(I)=ISPIN(I)
  400 CONTINUE
      SELEV=SELEV+LELEV
      SLAT=SLAT+PLAT
      IF (PLON.LT.0.)PLON=360.+PLON
      SLON=SLON+PLON
  420 CONTINUE
      IF (IBOX.NE.3)GO TO 305
      IF(MSAM.EQ.0) GO TO 2190
      LBEG=1
      FSAM=FLOAT(MSAM)
      SELEV=SELEV/FSAM
      SLAT=SLAT/FSAM
      SLON=SLON/FSAM
      IELEV=SELEV+.5
      LAT=SLAT*100.+0.5
      LON=SLON*100.+0.5
      IVZ=IROUND(VZEN*100.)
      ISZ=IROUND(SZEN*100.)
      IF(KBUG.EQ.0) GO TO 430
      CALL ENKODE('(T5,"LINE",4X,"ELEM",5X,"LAT",5X,"LON",4X,
     * "LZEN",4X,"SZEN",4X,"IELE"/)',LBUF)
      CALL ENKODE('(7I8/)',LBUF,IL,IE,LAT,LON,IVZ,ISZ,IELEV)
      CALL SDEST('  SAMPLE = ',MSAM)
  430 CONTINUE
      IPSTA=MISG
      ITSTA=MISG
      IDSTA=MISG
      CALL SURGES(NCSFC)
      TSTA=0.01*FLOAT(ITSFC)
      TMN=TSTA-10.
      IF (NCSFC.NE.0)TMN=TMN-5.
C     GENERATE CLEAREST RADIANCES.
      DO 580 J=1,NSMAX
      DO 560 K=1,12
      IF(RADS(J,K).EQ.VMISG) GO TO 560
      IUSE(K)=IUSE(K)+1
      NSAM(K)=NSAM(K)+1
```

```
      AVG(K)=AVG(K)+RADS(J,K)
  560 CONTINUE
  580 CONTINUE
      LIM=3
      IF(IUSE(8).LT.LIM) GO TO 2190
C     INSIST ON RAW SAMPLE OF 3 OBS
      DO 590 K=1,12
      KUSE(K)=1
      IF(IUSE(K).LT.LIM) IUSE(K)=0
  590 CONTINUE
      IF (CTYP.EQ.'N')GO TO 770
C     SEEK WARMEST WHEN SAMPLE IS ADEQUATE FOR ALL CHANNELS
      TWMAX=0.
      T6MAX=0.
      JSAV6=0
      JSAV=0
      DO 600 J=1,NSMAX
      DO 595 K=3,10
      IF(IUSE(K).EQ.0) GO TO 600
  595 CONTINUE
      T6=RADS(J,6)
      IF(T6.EQ.VMISG) GO TO 597
      IF(T6.LT.T6MAX) GO TO 597
      T6MAX=T6
      JSAV6=J
  597 CONTINUE
      TW=RADS(J,8)
      IF(TW.EQ.VMISG) GO TO 600
      IF(TW.LT.TWMAX) GO TO 600
      TWMAX=TW
      JSAV=J
  600 CONTINUE
      IF (TWMAX.LT.TMN)GO TO 2190
      IF(KBUG.NE.0)
      CALL ENKODE('(" JSAV =",I3,", JSAV6 =",I3/)',LBUF,JSAV,JSAV6)
      IF(JSAV.EQ.0) GO TO 2190
      IF(JSAV6.EQ.0) GO TO 2190
      RLO8=RADS(JSAV,8)-2.0
      IF(NOSFC.EQ.0) RLO8=AMIN1(RLO8,TSTA)
      RLO6=RADS(JSAV6,6)-1.5
      RLO9=RLO6-20.
      RLO5=RADS(JSAV,5)-1.0
      RLO4=RADS(JSAV,4)-1.0
      DO 650 K=3,12
      NSAM(K)=0
  650 AVG(K)=0.
      DO 720 I=1,NSMAX
      DO 660 J=3,12
  660 LC(J)=1
      IF(RADS(I,6).NE.VMISG.AND.RADS(I,6).GE.RLO6) GO TO 670
      LC(6)=0
      LC(12)=0
```

```
      IF(RADS(I,6).NE.VMISG.AND.RADS(I,6).GE.RLC9) GC TC 670
      LC(9)=0
      LC(10)=0
  670 CONTINUE
      IF(RADS(I,8).NE.VMISG.AND.RADS(I,8).GE.RLO8) GC TC 680
      LC(8)=0
      LC(7)=0
      LC(12)=0
      LC(5)=0
      LC(6)=0
      IF(RADS(I,5).NE.VMISG.AND.RADS(I,5).GE.RLC5) GC TC 680
      LC(4)=0
      LC(9)=0
      IF(RADS(I,4).NE.VMISG.AND.RADS(I,4).GE.RLO4) GC TC 680
      LC(3)=0
      LC(10)=0
  680 DC 700 J=3,12
      IF(LC(J).EG.0.CR.RADS(I,J).EG.VMISG) GC TO 70C
      NSAM(J)=NSAM(J)+1
      AVG(J)=AVG(J)+RADS(I,J)
  700 CONTINUE
  720 CONTINUE
  740 CONTINUE
      DC 760 I=1,12
      VDAT(I)=VMISG
      SRAD(I)=VMISG
      IF(NSAM(I).EG.0.OR.IUSF(I).EG.0) GO TO 760
      VDAT(I)=AVG(I)/NSAM(I)
  760 CONTINUE
C     DUE TC SUBMCDE 1 WE HAVE TO CHECK CN CHANNEL 7 SAMPLE INSTEAD
C     OF THE WINDCW CHANNEL
      MSAM=NSAM(7)
CCC   MSAM=NSAM(8)
      IF (MSAM.GE.5)GC TO 790
      IF(FORCE.NE.0..AND.CTYP.NE.'N')GC TO 790
      CTYP='N'
C     EXERCISE 'N' CPTION
  770 DC 780 I=1,12
  780 IUCH(I,1)=ILSE(I)
C      CHECK FCR NC NSTAR CPTICN FOR KC
      IF (IKWP('NCFC',1,0).NE.0)GO TO 2190
      CALL NSTAR(RADS,NRAS,VDAT,IFAIL,NSMAX)
      IF(IFAIL.EG.C) GO TC 790
      IF(KBUG.NE.0)CALL SDEST(' NSTAR FAILURE NC. ',IFAIL)
      GO TO 2190
  790 CCNTINUE
      IF(KBUG.EG.0) GC TC 800
      CALL ENKCDE('(T2,"BRIGHT-TEMP", 12F8.2/)',LBUF,VDAT)
  800 CCNTINUE
      DC 830 I=1,12
      EX(I)=100.
      IF(VDAT(I).EG.VMISG) GO TO 830
```

```
          SRAD(I)=VPLANC(VDAT(I),I)
          DBDE=VDBDTE(VDAT(I),I)
          SPIN=NSPIN(I)*NSAM(I)
          IF(SPIN.EQ.0..OR.DBDE.EQ.0.) GO TO 830
          EX(I)=(EV(I)/DBDE)/SQRT(SPIN)
          IF(I.EQ.1) EX(I)=0.5*EX(I)
          IF(I.EQ.5) EX(I)=2.0*EX(I)
      830 IF(EX(I).EQ.100.)IUSE(I)=0
C     DATA ACQUISITION COMPLETE
          TS=VSKINT(VDAT,0,0,JDUM)
          IF(TS.GT.310.) TS=2.*VDAT(8)-VDAT(7)
          NOGO=0
          GO TO 2200
     2190 NOGO=1
          GO TO 2380
     2200 CONTINUE
C         PREPARE OUTPUT BUFFER
          DO 2201 K=1,NSIZE
     2201 IRET(K)=MISG
          IRET(4)=LUC(-17)
          IRET(5)=IVER
          IRET(6)=INCRL
          IRET(7)=INCRE
          IGD=MOD(IDCC(33),1000)
          IRET(8)=IDCC(34)+IGD*10000
          IRET(9)=ISDCC(6)
          DO 2202 K=1,5
     2202 IRET(9+K)=ISDCC(K)
          IRET(15)=0
          IRET(16)=IDPROS
C         FILL IN USER MOD FLAG
          IRETD(3)=-999
          IRETD(4)=LASRET+1
          IRETD(5)=LAT*100
          IRETD(6)=LON*100
          IRETD(7)=IDCC(2)
          IRETD(8)=MSAM
          DO 2203 N=1,12
          IF (IUSE(N).EQ.0)GO TO 2203
          IRETD(8+N)=100.*VDAT(N)+0.5
          IRETD(21+N)=EX(N)*10000.
     2203 CONTINUE
          IRETD(36)=100.*TS
          IRETD(39)=SZEN*100.
          IRETD(40)=VZEN*100.
     2218 ITYP=21
          IF (IFLG.NE.0)ITYP=23
          IF (CTYP.EQ.'N')ITYP=22
          IRETD(50)=ITYP
          IRETD(57)=IELEV
C         MD OUTPUT BUFFER COMPLETE
          LASRET=LASRET+1
```

```
      IDCC(100)=LASRET
      CALL VRTIO(IRET,LASRET,1)
      IF (IPLT.EQ.0)GO TO 2360
      CALL SATTV(IFRM,IL,IE,IRAS,IPIC,JSAT,JDAY,JTIME)
      JDAY=JSAT*100000+JDAY
      KOLOR=2
      IF (CTYP.EQ.'N')KOLOR=3
      IF(IFLG.NE.0) KOLOR=1
      ITEM=VDAT(8)
      CALL VASDIG(IRAS+MAG2,IPIC+MAG2,ITEM,MAG,1,KOLOR)
 2360 CONTINUE
      IF(LASRET.EQ.MAXRET) GO TO 2390
 2380 CONTINUE
      GO TO 230
 2385 CONTINUE
C     WRITE ROW HEADER AND VASTEXT
 2390 IRET(1)=MOD(JDAY,100000)
      IRET(2)=IDCC(2)
      IRET(3)=LASRET
      CALL VRTIO(IRET,0,1)
      IF(LASRET.EQ.MAXRET) GO TO 2500
 2400 CONTINUE
 2500 CALL SDEST(' NO. OF SNDGS. IN RTVL. FILE = ',LASRET)
      RETURN
      END
```

```
//SRET5600 JOB CLASS=I,MSGLEVEL=(0,0)
//   VLSRET   JPN  03/16/84: MEMBER UPDATED
// EXEC MCPRG
//FORT.SYSIN DD
@PROCESS SC(TGMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTCR,OFCOM,SQX,SGW)
@PROCESS SC(DOPEN,DREAD,DWRITE,MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAINC
C ?   VAS RETRIEVAL PROGRAM; USES MDFILE DRIVER 'VASTEXT' DEFAULTS
C ?      KEYIN: SRET N1 N2 <KEYWORDS>
C ?        POSITIONAL PARAMETER:
C ?           N1 FIRST RETRIEVAL IN FILE (DEFAULT TO 1)
C ?           N2  LAST RETRIEVAL IN FILE (DEFAULT TO VASTEXT NRET)
C ?          KEYWORDS:
C ?             TYP=F (FORCE)
C ?             GSS=C (CLIM) P (PROFILE) G (GRID=DEFAULT)
C ?             SFC=1 (NO SURFACE ANALYSIS OPTION) (DEFAULT=0)
C ?             TER=N (TERMINAL NUMBER TO DEFINE 'VASTEXT' (LOCAL))
C ?             BUG=1 (FOR DEBUG DIAGNOSTICS (0))
C ?             PLT=1 (PLOT ON GRAPHICS (0))
C ?             ENH=1 (TURN OFF ENHANCEMENT)
C ?             ICB=1 (MAKE CALIBRATION RUN USING GUESS)
C ?             ITR=N (MAX NO OF ITERATIONS) (DEFAULT=3)
C ?             BIAS  IF NON-ZERO USE BIAS VECTOR IN VASTEXT
      DIMENSION IFILNM(2),KOUT(10),ICU(3),ICT(6),IHLM(3)
     ,B(40),S(40),L(40),TAU(40),DELT(6),VDAT(13),BDAT(13)
     ,MF(64)
     ,LBUF(33),TAUS(40,6),TAUW(40,4),WG(40,6)
      DIMENSION LC(13),IUSE(13),DELS(6),HUM(3),TD(40),RRHO(3),RHO(3)
      DIMENSION KUSE(13),TSAV(40),IRET(246),IRETD(246),EX(13)
      DIMENSION TGS(40),LMD(20),LCHR(20),DGS(11)
      DIMENSION PST(15),TST(15),TDST(15),SPD(15),DIR(15),STABIL(12)
      CHARACTER 12 CGES,CPP,CKWP,CTYP
      COMMON/DBUG/KBUG
      COMMON/SDOC/ISDOC(6)
      COMMON /GESTBE/TBG(12),TRN(40,12)
      COMMON/DOC/IDOC(100)
      COMMON/TERMN/ITERM
      COMMON/SFC/IS,PWR,DEL7,DEL8
      COMMON/ATMOS/P(40),T(40),W(40)
      COMMON/GUESS/TGES(15),PGES(6)
      COMMON/RADV/VRAD(13)
      COMMON/SURF/IZ10,ITSFC,IDSFC,IPSTA,IELEV,LSTA
      COMMON/NAV/VLAT,VLON,VZEN,SZEN,IL,IE,IRAS,IPIC,ITIME,JTIME,JDAY
      COMMON/MODE/IDET(13),ISPIN(13)
      COMMON/GDE/GV(12),DV(12),EV(12)
      COMMON /USE/ILCH(12,2)
      EQUIVALENCE (IRET(17),IRETD(1))
      EQUIVALENCE (TAUS(1,1),TRN(1,1)),(TAUW(1,1),TRN(1,7))
C     EQUIVALENCE TO IRET(ROWLEN-1)
C     ABOVE TO FACILITATE CHANGING LENGTH OF ROW HEADER
      DATA DIR/15*0./,SPD/15*0./
```

```
      DATA IRET/246*Z80808080/,IMISS/Z80808080/,NSIZE/245/
      DATA LIC/0,40,39,38,37,36,35,34,31,28,26,25,24,23,20,
     18,16,15,13,11/
      DATA LCHR/'SFC ','1000','950 ','920 ','850 ','780 ','700 ',
     '670 ','500 ','400 ','300 ','250 ','200 ','150 ','100 ',
     '70  ','50  ','30  ','20  ','10  '/
      DATA ICT/1,2,3,4,5,6/
      DATA ICU/7,9,10/
      DATA ITCH/6/
      DATA RCG/29.2899/,GRAV/980.665/,TCTC/347./
      DATA VMISG/999999./,MISG/Z80808080/
      DATA NL/40/
C
C     FOLLOWING IS PROGRAM VERSION DATE ... KEEP CURRENT!!!
      DATA IVER/84003/
C
      CALL CALDAY(IVER,IVY,IVM,IVD,IVMO)
      CALL ENKODE('(132X,T1,"BEGIN SRET, VERSION OF ",
     I2,1X,A4,I2/)',LBLF,IVD,IVMO,IVY)
C
C     READ IN VASTEXT AND ROW HEADER
      CALL VRTIO(IRET,0,0)
      CALL MDNAME(IDOC(40),IFILNM)
C     ABOVE SETS LF FILE NAME TO FORCE OUTPUT AT END OF EACH LINE
      ISATEL=IDOC(1)/100000
      ISAT=ISATNV(ISATEL)
      CALL FLNKIV(ISAT)
C     CHECK FOR CALCULATED BIAS
      IF (IKWP('BIAS',1,0).EQ.0)GO TO 8
      DO 6 K=1,6
    6 DV(K)=FLOAT(IDOC(60+K))*.0001
C     TEST FOR CALIBRATION RUN
    8 ICB=IKWP('ICB',1,0)
      IF (ICB.EQ.0)GO TO 10
C     TURN OFF BIAS VALUES OBTAINED IN FLNKIV OR FROM VASTEXT
      DO 9 K=1,12
    9 DV(K)=0.
   10 CGES=CKWP('GSS',1,' ')
      IGES=IDOC(51)
      IF (CGES.EQ.'G')IGES=0
      IF (CGES.EQ.'C')IGES=1
      IF (CGES.EQ.'P')IGES=2
      IDOC(51)=IGES
      IDEF=IDOC(54)
      NOSFC=IKWP('SFC',1,IDEF)
      IDOC(54)=NOSFC
      CTYP=CKWP('TYP',1,' ')
  105 FORCE=0.
      IF(CTYP.EQ.'F')FORCE=10.
      IF (CTYP.EQ.'C')FORCE=10.
      IF(CTYP.EQ.'C')FORCE=10.
C     'FORCE' USED AS NON ZERO FLAG TO SKIP GROSS ERROR CHECKS
```

```
  110 CONTINUE
      PREF=DKWP('PREF',1,1050.)
C     PREF IS USED TO BLEND CLESS PROFILE WITH RETRIEVAL AT
C     EACH ITERATION..BLENDING IS A LEVELS BELOW PREF
      LCRT=IKWP('BUG',1,0)
      IF (LCRT.NE.0)CALL TGSET(LCRT)
C GET TERMINAL NO..THIS HAS ALREADY BEEN LOADED IN VRTIC.OVERRIDE WITH
C     VASTEXT IF THAT VALUE NOT ZERO
      IDEF=IDOC(59)
      IF (IDEF.EQ.0)IDEF=ITERM
      ITERM=IKWP('TER',1,IDEF)
      IDOC(59)=ITERM
      KBUG=LCRT
      IDEF=IDOC(60)
      IPLT=IKWP('PLT',1,IDEF)
      IDOC(60)=IPLT
      IF (IPLT.EQ.0)GO TO 120
C     INITIALIZE NAVIGATION
      IRAS=LUC(-11)
      IPIC=LUC(-12)
      IFRM=LUC(-1)
      CALL TVSAT(IFRM,IRAS,IPIC,IL,IE,JS,JDAY,JTIME)
      JDAY=JS 100000+JDAY
      CALL NVINIT(BETAIN,BETDCT,INAV,PTIME)
      CALL GETFRM(IFRM,MF)
      MAG=MF(10)
      MAG=MAX0(MAG,6)
      MAG2=MAG/2
  120 NLP1=NL+1
      IBOX=1
C     SET UP PROCESSING DAY FOR DOCUMENTATION IN ROW HEADER
      CALL GETDAY(IDPROS)
      MDNG=IDOC(38)
      MDNR=IDOC(40)
      LASRET=IRET(3)
      IENH=IKWP('ENH',1,0)
      JDAY=IDOC(1)
      JTIME=IDOC(2)
      NB=IPP(1,1)
      LASRET=IPP(2,LASRET)
      NRET=0
      DO 2400 NN=NB,LASRET
      CALL VRTIC(IRET,NN,0)
C     LOOK AT USER MOD FLAG; LT 0 MEANS IT HASN'T BEEN RETRIEVED
C     EQ 0 MEANS IT HAS BEEN RETRIEVED
C     GT 0 MEANS ITS EITHER FAILED OR BEEN EDITED
      IF (IRETD(3).GE.0)GO TO 2400
      IF (KBUG.NE.0)CALL SDEST(' BEGIN SNDG NO. ',NN)
      DO 700 N=1,12
      IUSE(N)=0
      VDAT(N)=999.
      IF (IRETD(8+N).EQ.MISG)GO TO 700
```

```
      EX(N)=.0001 FLOAT(IRETD(21+N))
      VDAT(N)=.01 FLOAT(IRETD(8+N))
C      CHECK TO SEE IF THIS CHANNEL IS INTENTIONALLY AVOIDED
      IUSE(N)=IUCH(N,1)+IUCH(N,2)
      VRAD(N)=VPLANC(VDAT(N),N)
  700 CONTINUE
C     **THERE ARE TWO SETS OF "USE" FLAGS IN THIS PROGRAM
C     IUSE IS SET TO Ø IF DATA IS MISSING OR CHANNEL IS INTENTIONALLY
C     SET ASIDE.  KUSE IS SET TO Ø IF CHANNEL IS BELIEVED TO BE
C     CLOUD CONTAMINATED: IN THIS CASE GUESS PROFILE IS IMPLICITLY
C     BELIEVED UNLESS IT IS A NO SURFACE.CLIMATOLOGY CASE.
      IF (KBUG.NE.0)
      CALL ENKODE('(T2,"BRIGHT-TEMP", 12F7.2/)',LBUF,VDAT)
      IF (KBUG.NE.0)
      CALL ENKODE('(T2,"CHAN FLAGS ", 12I7/)',LBUF, IUSE)
      TS=.01 FLOAT(IRETD(36))
      SZEN=.01 FLOAT(IRETD(39))
      VZEN=.01 FLOAT(IRETD(40))
      VLAT=.0001 FLOAT(IRETD(5))
      VLON=.0001 FLOAT(IRETD(6))
      ITYP=IRETD(50)
      IELEV=IRETD(57)
C     ZERO OUT PREVIOUSLY RETRIEVED QUANTITIES
      IRETD(35)=MISG
      DO 380 I=41,NSIZE
  380 IRETD(I)=MISG
C     RESTORE TYPE AND ELEVATION
      IRETD(50)=ITYP
      IRETD(57)=IELEV
      NOGC=1
      CALL GESPRO(IGES,NOSFC,MONG)
      IF(TGES(1).LT.0.) GO TO 2200
      PSTA=IPSTA
      TSTA=0.01 FLOAT(ITSFC)
      TCK=2. VDAT(8)-VDAT(7)
C      CHECK ON THE HOPELESSLY COLD,PRESUMED OVERCAST
      IF ((TSTA-TCK).GT.15.)GO TO 2200
C      LET HOT GROUND GO FOR NOW
CCC   IF (ABS(TCK-TSTA).GT.15.)GO TO 2200
C     FIND "IS" 1ST LEVEL BELOW SURFACE (PRESSURE)
      IS=NL
      DO 400 J=1,20
      I=NLP1-J
      DP=P(I)-PSTA
      IF(DP.GE.0.) GO TO 400
      IS=I+1
      GO TO 420
  400 CONTINUE
  420 IS=MINO(IS,NL)
C     BLEND GUESS WITH SURFACE DATA (IF THEY EXIST)
      IF(NOSFC.NE.0) GO TO 460
C     BEGIN AT 850MB FOR NORMAL GUESS,500MB FOR CLIMATOLOGY
```

```
      ILO=37
      IF(IGES.EQ.1) ILO=31
      IF (ILO.GE.IS)GO TO 460
      DTS=TSTA-T(IS)
      DTDP=DTS/(P(IS)-P(ILO))
      DO 440 I=ILO,IS
  440 T(I)=T(I)+DTDP*(P(I)-P(ILO))
  460 CONTINUE
      DO 425 I=1,40
  425 TSAV(I)=T(I)
C     PREPARE GUESS INFO FOR MDFILE
      DO 480 I=2,20
      J=LMD(I)
  480 TGS(I)=T(J)
      TGS(1)=.01*FLOAT(ITSFC)
      DO 485 I=2,11
      J=LMD(I)
      DPT=DEWPT(P(J),T(J),W(J))
  485 DGS(I)=DPT
      DGS(1)=.01*FLOAT(IDSFC)
      IF(KBUG.EQ.0) GO TO 540
      CALL ENKODE('(132X,T6,"PRESSURE  ",15F7.1/)',LBUF,P(26))
      CALL ENKODE('(T6,"GUESS TEMP", 15F7.1/)',LBUF,T(26))
      CALL ENKODE('(T6,"GUESS WVMR", 15F7.3/)',LBUF,W(26))
      CALL ENKODE('(T2,"Z1000 =",I8,", TSFC =",I8,", TDSFC =",I8/)'
     ,LBUF,IZ10,ITSFC,IDSFC)
  540 CONTINUE
      TDSTA=.01*FLOAT(IDSFC)
      DD=TSTA-TDSTA
      PSTA=IPSTA
      CALL WMIX(PSTA,TSTA,DD,WSTA,1)
      IF (CTYP.EQ.'C')GO TO 655
C
C *** DETERMINE A TENTATIVE CLOUD FLAG (IFLG).  THIS WILL BE USED
C     TO ESTABLISH RH BOUNDARY AND IN T RETRIEVAL.  IF FLAG IS SET
C     TO NON-ZERO WE WILL NOT DO ENHANCEMENT AND WE WILL NOT
C     BELIEVE SKIN TEMPERATURE PREVIOUSLY DEFINED.  HOWEVER, WE
C     MAY STILL USE ALL CHANNELS (1550 LOOP) AND WE MAY STILL DO
C     RTE DOWN TO SURFACE (1463 LOOP)
C
      IFLG=99
      IF (CTYP.EQ.'C')GO TO 660
      IF(IGES.EQ.1.AND.NOSFC.NE.0) GO TO 660
      IF (IUSE(12).NE.1)GO TO 550
C     CHECK REFLECTED SUNLIGHT. BUT FOR HIGH WINDOW VALUES GIVE
C     THE NOD TO CLEAR
      WDIF=VDAT(12)-TS
      IF (WDIF.GT.5.)GO TO 660
      IF (VDAT(8).LE.273.AND.WDIF.GT.2.)GO TO 660
C      CHECK SURFACE AIR AGAINST SKIN ESTIMATE. LOOSEN CHECK
C     FOR WARM SKIN (PROBABLY CLEAR) OR FOR VERY COLD SURFACE
C     AIR (PROBABLE INVERSION CONDITION) OR FOR POOR SURFACE AIR EST.
```

```
  550 TMN=TSTA-3.
      IF(TSTA.LT.273.OR.NOSFC.NE.0.OR.TS.GT.290.) TMN=TSTA-6.
      IF(TS.LT.TMN) GO TO 660
C     OK SET FLAG TO CLEAR
  655 IFLG=0
  660 CONTINUE
      LS=NL
      IF (IFLG.EQ.0)GO TO 980
C     FIND "LS" 1ST LEVEL BELOW 1ST COLDER THAN SFC SKIN
C     FOR BOUNDARY IN RTE FOR RELATIVE HUMIDITY.
      DO 920 J=1,20
      I=NLP1-J
      DT=T(I)-TS
      IF(DT.GE.0.) GO TO 920
      LS=I+1
      LS=MIN0(LS,NL)
      GO TO 980
  920 CONTINUE
  980 NLS=MIN0(IS,LS)
      IF (KBUG.NE.0)CALL SDEST(' RH LOWER BOUNDARY LEVEL NO. IS ',NLS)
      ST=TS
C     TEST TO SEE IF THIS IS A GUESS-ONLY CALIBRATION RUN
      IF (ICB.EQ.1)GO TO 1325
      DO 860 I=1,25
  860 S(I)=.001
      DO 1100 KC=1,3
      IHUM(KC)=MISG
      ICV=ICU(KC)
      IF (IUSE(ICV).EQ.0)GO TO 1100
      RHO(1)=0.1
      RHO(2)=0.5
      RHO(3)=0.9
      RWV=VRAD(ICV)
      IF(RWV.EQ.VMISG)GO TO 1100
      DO 880 I=1,NLS
  880 B(I)=VPLANC(T(I),ICV)
      BS=VPLANC(ST,ICV)
      DO 1080 ISTEP=1,3
      IF(ISTEP.LT.3)GO TO 885
C     IF 2ND GUESS FOR RH IS TOO HIGH, BASE FINAL ESTIMATE ON 1 AND
      IF(RWV.LT.RRHO(2))GO TO 885
      TGR=(RWV-RRHO(1))/(RRHO(2)-RRHO(1))
      TGR=AMAX1(TGR,0.)
      TGR=AMIN1(TGR,5.)
      RH=RHO(1)*((RHO(2)/RHO(1))**TGR)
      GO TO 1090
  885 CONTINUE
C     SET UP MIXING RATIO PROFILE, FIND PREC WATER AND CALCULATE
C     TRANSMITTANCE AND RADIANCE
      DO 890 I=26,NL
  890 S(I)=RHO(ISTEP)*WSAT(P(I),T(I))
      CALL PRECW(P,S,U,NL)
```

```
      CALL VASTAL(T,S,TCTC,VZEN,TAU,ISAT,ICV)
 1080 RRHO(ISTEP)=VBDTAL(TAU,B,BS,ICV,NLS)
      TGR=(RWV-RRHO(2))/(RRHO(3)-RRHO(2))
      TGR=AMAX1(TGR,-4.)
      TGR=AMIN1(TGR,2.)
      RH=RHO(2)*((RHO(3)/RHO(2))**TGR)
 1090 CONTINUE
      R50=RRHO(2)
      IF(IGES.EQ.1.AND.RH.GT.1.2) GO TO 1100
      RH=AMIN1(RH,1.)
      RH=AMAX1(RH,0.10)
      IRH=IROUND(RH*100.)
      IHUM(KC)=IROUND(RH*10000.)
      IF(LCRT.EQ.0) GO TO 1100
      CALL ENKODE('(T5,"CHAN RWV 100 R50 100 TGR*100  RH*100")',
     LBUF)
      CALL VTG(LBUF)
      IRWV=IROUND(RWV*100.)
      IR50=IROUND(R50*100.)
      ITGR=IROUND(TGR*100.)
      CALL ENKODE('(T1,T1,5I8)',LBUF,ICV,IRWV,IR50,ITGR,IRH)
      CALL VTG(LBUF)
 1100 CONTINUE
      IF(IHUM(3).EQ.MISG) GO TO 2380
      IF(IHUM(1).EQ.MISG)IHUM(1)=10000.*WSTA/WSAT(PSTA,TSTA)
      IF(IHUM(2).EQ.MISG) IHUM(2)=(IHUM(1)+IHUM(3))/2
      RHS=IHUM(1)*.0001
C     APPORTION THE RELATIVE HUMIDITY TO LAYERS
      IF(NOSFC.NE.0) GO TO 1160
      RHS=WSTA/WSAT(PSTA,TSTA)
      IF(PSTA.LT.800.) IHUM(1)=IROUND(10000.*RHS)
 1160 CONTINUE
      DO 1180 I=1,3
 1180 HUM(I)=0.0001*FLOAT(IHUM(I))
      IF (FORCE.NE.0.)GO TO 1190
C     IF(HUM(1).GT.0.95) GO TO 2200
C   **** I IS LETTING HIGH RH GO HEAH
C     RESTORE SATURATED MIXING RATIO PROFILE
C     AND ESTABLISH MIXING RATIO PROFILE
 1190 DO 1300 I=26,NL
      S(I)=WSAT(P(I),T(I))
      IF(P(I).LT.800.) GO TO 1260
      DRDP=(RHS-HUM(1))/(1000.-800.)
      RHM=HUM(1)+DRDP*(P(I)-800.)
      W(I)=RHM*S(I)
      IF(P(I).GE.P(IS)) W(I)=RHS*S(I)
      GO TO 1300
 1260 IF(P(I).LT.600.) GO TO 1280
      DRDP=(HUM(1)-HUM(2))/(800.-600.)
      RHM=HUM(2)+DRDP*(P(I)-600.)
      W(I)=RHM*S(I)
      GO TO 1300
```

```
 1280 DRDP=(HUM(2)-HUM(3))/(600.-300.)
      RHM=HUM(3)+DRDP*(P(I)-300.)
      W(I)=RHM*S(I)
 1300 CONTINUE
      IF(KBUG.EQ.0) GO TO 1320
      CALL ENKODE('(T11,"W =",-15F7.3/)',LBUF,W(26))
 1320 CONTINUE
      DO 1322 K=1,4
      KC=K+6
      CALL VASTAL(T,W,TCTO,VZEN,TAUW(1,K),ISAT,KC)
 1322 CONTINUE
      CALL PRECW(P,W,U,NL)
C
C     BEGIN RETRIEVAL LOOPS
 1325 ITLIM=3
      IF (IGES.EQ.1)ITLIM=10
      ITLIM=IKWF('ITR',1,ITLIM)
      ITER=0
      IPASS=0
      IOK=0
C     IOK IS SET TO -1 FOR FAILED ENHANCEMENT,+1 FOR SUCCESS IN VTRET
C
      INIT=0
C     INIT IS SET TO FIND LOWEST LEVEL JUST ONCE
C
C ....BEGIN IMPLICIT ITERATION LOOP
C     IPASS=0   CALC TAU;DO T ITERATION;ENHANCE H2O..CONVERGENCE
C          TEST IS 6 1E20
C           =1   ENHANCE TEMP, LAYER 1
C           =2                 LAYER 2
C           =3                 LAYER 3
C           =4   REPEAT T ITERATION IF ENHANCE FAILED,REPEAT H2O,EXIT
C     HEAVE ANY CLOUDY/CLIMATOLOGY SOUNDING
C     IF (IGES.EQ.1.AND.IFLG.NE.0)GO TO 2200
C   CHECK ON CALIBRATION AGAIN
      KC2=6
      KC1=1
      IA=IS
      DO 1380 I=1,6
      KUSE(I)=1
      J=7-I
      IF(KC1.EQ.I.AND.IUSE(I).EQ.0) KC1=I+1
      IF(KC2.EQ.J.AND.IUSE(J).EQ.0) KC2=J-1
 1380 CONTINUE
      IF (ICB.NE.1)GO TO 1385
      IF (IFLG.NE.0)GO TO 2200
      GO TO 2190
 1385 IF(KC1.GE.3.AND.KC2.LE.3) GO TO 2200
 1390 DO 1520 KC=KC1,KC2
      ICV=ICT(KC)
C
C  ** CALCULATE TAU UNDER THE FOLLOWING CONDITIONS:
```

```
C          INITIAL ENTRY
C
C          ENHANCEMENT OR 2ND ITERATIVE PASS (ENHANCEMENT FAILED)
C          BUT ONLY FOR CHAN 4 AND 5 UNLESS GUESS IS CLIMATOLOGY
C
      IF(ITER.GT.0.AND.IPASS.NE.1) GO TO 1420
      DELT(KC)=0.
      IF (IPASS.EQ.1.AND.KC.NE.4.AND.KC.NE.5.AND.IGES.NE.1)GO TO 1420
      CALL VASTAU(T,W,TOTC,VZEN,TAU,ISAT,ICV)
      DO 1400 IK=1,40
      WG(IK,KC)=0.
 1400 TAUS(IK,KC)=TAU(IK)
      GO TO 1460
 1420 DO 1440 IK=1,40
 1440 TAU(IK)=TAUS(IK,KC)
 1460 CONTINUE
      IF(INIT.GT.0) GO TO 1470
      INIT=1
C
C   ***  IN THE FOLLOWING WE DETERMINE THE LOWEST LEVEL TO WHICH WE
C      WILL DO RADIATIVE TRANSFER.
      TA=TS
      IF(IFLG.EQ.0) GO TO 1465
      TA=VDAT(8)
      IF(IGES.EQ.1.AND.NCSFC.NE.0) GO TO 1465
C     LOOK FOR NEW CLOUD LEVEL FOR SURFACE TERM
      DO 1463 J=1,20
      I=NLP1-J
      DT=T(I)-TA
      IF(DT.GE.0.) GO TO 1463
      IA=I+1
      IA=MIN0(IA,IS)
      GO TO 1465
 1463 CONTINUE
 1465 CONTINUE
      IF (KBUG.NE.0)CALL ENKODE('(1X,"SKIN LEVEL AND TEMP",I5,
     F7.2/)',LBUF,IA,TA)
 1470 CALL VASRTE(TAU,T,TA,CRAD,B,TBB,DBDB,ICV,IA)
      B1=B(1)
      T1=TAU(1)
      DO 1480 L=2,IS
      B2=B(L)
      T2=TAU(L)
      WGL=.5*(B1+B2)*(T1-T2)
      WG(L,KC)=WGL/DBDB
      B1=B2
 1480 T1=T2
C     SAVE PREVIOUS CONVERGENCE STATE
      IF(KUSE(KC).EQ.0) GO TO 1520
      IF(IUSE(KC).EQ.0) GO TO 1520
      DELS(KC)=DELT(KC)
      DELT(KC)=VDAT(ICV)-TBB
```

```
 1520 CONTINUE
      IF(ITER.GT.0) GO TO 1580
      IF (IUSE(5).EQ.0.OR.IUSE(6).EQ.0)GO TO 1540
      IF (IFLG.NE.0)GO TO 1540
C     WATCH OUT FOR SHAKEY CHANNEL 6 WHICH WE DONT UNDERSTAND
      AB5=ABS(DELT(5))
      AB6=ABS(DELT(6))
      IF (AB6.LT.AB5)GO TO 1530
      AB4=ABS(DELT(4))
      IF (AB6.LT.AB4)GO TO 1530
      IUSE(6)=0
C     TEST CHANNEL 5 VS 6 RESIDUAL AND ASSUME THAT A LARGE
C     DISCREPANCY INDICATES CLOUD PROBLEM (HIGH RH1 AS WELL)
 1530 CCK=DELT(5)-DELT(6)
      ICCK=100. CCK
      IF (KBUG.NE.0)CALL SDEST(' CHAN5-CHAN6= ',ICCK)
      RHCK=AMAX1(HUM(1),HUM(2))
      IF (ABS(CCK).GT.2.5.AND.RHCK.GT.0.95)IFLG=1
 1540 CONTINUE
C
C     FOR HIGH GROUND WE THROW OUT SHORT WAVE CHANNEL
      IF(PSTA.LE.850.) KUSE(6)=0
C
C     SET ASIDE ONE CHANNEL FOR TESTING SUCCESS OF RETRIEVAL(@1960)
C
      ITCH=5
      IF(IFLG.NE.0.OR.PSTA.LE.850.) ITCH=6
      IF(CTYP.EQ.'C') GO TO 1580
      IF(IGES.EQ.1.AND.NOSFC.NE.0) GO TO 1580
      IF(IFLG.EQ.0) GO TO 1580
C     ENTER FOR SUSPECTED CLOUDY SOUNDING WITH REASONABLY GOOD GUESS
C     CHECK OBS-CALC FOR INCREASINGLY OPAQUE CHANNELS AND DELETE THOSE
C     COLDER THAN GUESS
      DO 1560 I=KC1,KC2
      JJ=KC2+1-I
      J=JJ
C     REVERSE ORDER BECAUSE CHAN 5 IS MORE TRANSPARENT
      IF(JJ.EQ.6) J=5
      IF(JJ.EQ.5) J=6
      DELTT=DELT(J)
      IF(DELTT.GE.-0.25) GO TO 1550
      IF (KBUG.NE.0)CALL SDEST(' REMOVING CLOUDY CHANNEL ',J)
      KUSE(J)=0
C     SET RETRIEVAL TYPE TO OVERCAST AT THIS POINT
      ITYP=23
      GO TO 1560
C     IF 5 PASSES SKIP OUT,BUT CHECK 4 EVEN IF 6 PASSES
 1550 IF(J.NE.6) GO TO 1580
 1560 CONTINUE
 1580 CONTINUE
      IF(KUSE(4).EQ.0) GO TO 2200
      IF(KBUG.EQ.0) GO TO 1660
```

```
      CALL ENKODE('(T11,"T =",15F7.1/)',LBUF,T(26))
      CALL ENKODE('(T11,"W =",15F7.3/)',LBUF,W(26))
      CALL ENKODE('(T11,"DTDB",6F8.2/)',LBUF,DELT)
 1660 CONTINUE
C     IF RECALCULATION ON SECOND PASS SHOWS LARGE RESIDUAL ASSUME
C     THAT SOMETHING HAS GONE WRONG
      DO 1661 KC=KC1,KC2
      IF (IUSE(KC).EQ.0)GO TO 1661
      IF (KUSE(KC).EQ.0)GO TO 1661
      IF (FORCE.GT.0.)GO TO 1661
      IF (IPASS.NE.0.AND.ABS(DELT(KC)).GT.2.5)GO TO 2200
 1661 CONTINUE
      IF (IPASS.LE.3)GO TO 1662
      IF (ICK.GT.0)GO TO 1940
C     ABOVE JUMPS OUT WHEN ENHANCEMENT OF T COMPLETED SUCCESSFULLY
C     OTHERWISE CONTINUE THE ITERATIVE WITH NEW TAU UNLESS CONVERGED
      GO TO 1670
 1662 IF (IPASS.EQ.0)GO TO 1670
C     CHECK CONVERGENCE OF ALL CHANNELS WITHIN NOISE
      DO 1665 I=KC1,KC2
      DTABS=ABS(DELT(I))
      IF(DTABS.GT.EX(I)) GO TO 1668
 1665 CONTINUE
      GO TO 1951
 1668 CONTINUE
      IF (IFLG.NE.0)GO TO 1940
      ICK=-1
      IF (IENH.NE.1)
      CALL VTRET(KC1,KC2,DELT,KUSE,WG,IS,IPASS,EX,ICK)
      IF (ICK.LT.0)GO TO 1882
      GO TO 1730
 1670 CONTINUE
C     *********
C     APPLY CORRECTION TO TEMP PROFILE
      DO 1720 I=2,IS
      FNUM=0.
      FDEN=0.
      DO 1700 J=KC1,KC2
      IF(J.EQ.ITCH) GO TO 1700
      IF(IUSE(J).EQ.0) GO TO 1700
      WGHT=WG(I,J)/EX(J)
      IF(KUSE(J).EQ.0) GO TO 1680
      FNUM=FNUM+DELT(J)*WGHT
 1680 FDEN=WGHT+FDEN
 1700 CONTINUE
      IF(ABS(FDEN).LT.1.E-8) GO TO 1720
      T(I)=T(I)+FNUM/FDEN
C     CHECK FOR GUESS T(P) BLENDING
      IF (PREF.GE.P(IS))GO TO 1720
      WPROF=((P(I)-PREF)/(P(IS)-PREF))
      T(I)=WPROF TSAV(I)+(1.-WPROF)*T(I)
 1720 CONTINUE
```

```
C
 1730 CONTINUE
C      ADJUST LOW LEVEL T(P) USING SURFACE AIR TEMP
       IF(NOSFC.NE.0) GO TO 1840
       DTS=TSTA-T(IS)
       IF(IS.LE.37) GO TO 1800
C      IF SURF AT OR ABOVE 850 EXIT,OTHERWISE BLEND LOWER LEVELS
C      WITH SURFACE AIR TEMP RETAINING SHAPE OF PROFILE
       DTDP=DTS/(P(IS)-P(37))
       DO 1780 I=37,IS
 1780  T(I)=T(I)+DTDP*(P(I)-P(37))
 1800  DO 1820 I=IS,40
 1820  T(I)=TSTA
 1840  CONTINUE
       DO 1860 I=26,40
       IF(S(I).LT.0.02) GO TO 1880
       SNEW=WSAT(P(I),T(I))
       IF(SNEW.LT.0.02) SNEW=0.02
       W(I)=W(I)*SNEW/S(I)
       S(I)=SNEW
 1860  TD(I)=DEWPT(P(I),T(I),W(I))
 1880  CONTINUE
       IF(IPASS.EQ.0.OR.IPASS.GT.3) GO TO 1885
 1882  IPASS=IPASS+1
       IF (IOK.GE.0)GO TO 1390
       IF (IPASS.LE.3)GO TO 1668
C      ABOVE AVOIDS REPEATING RAD TRAN WHEN ENHANCEMENT FAILED
C       FOR FAILED ENHANCEMENT APPLY NORMAL ITERATIVE CORRECTION
       ICK=0
       ITLIM=ITLIM+3
C      RAISE ITERATION LIMIT WHEN WE GO TO SECOND PASS OF ITERATIVE
       GO TO 1670
 1885  CONTINUE
C  *   CHECK CONVERGENCE
       ICK=0
       D2=0.
       SUM=0.
       DO 1920 KC=KC1,KC2
       IF(KUSE(KC).EQ.0) GO TO 1920
       IF(ICT(KC).NE.ITCH) GO TO 1900
       KSAV=KC
       GO TO 1920
 1900  D1=ABS(DELT(KC)-DELS(KC))
       D2=D2+1.
       SUM=SUM+D1*D1
 1920  CONTINUE
       SUM=SQRT(SUM/D2)
       ITER=ITER+1
       IF(SUM.LT.0.025) GO TO 1940
       IF (KBUG.NE.0)CALL SCEST(' *** END ITER =',ITER)
       IF(ITER.GE.ITLIM) GO TO 1940
       GO TO 1390
```

```
 1940 CONTINUE
      IF (IENH.NE.0)GO TO 1949
C
C ***** BEGIN MOISTURE ENHANCEMENT
C     REDEFINE CLOUD LEVEL
      LS=NL
      DO 1942 J=1,20
      I=NLP1-J
      DT=T(I)-TS
      IF(DT.GE.0.) GO TO 1942
      LS=I+1
      LS=MINO(LS,NL)
      GO TO 1943
 1942 CONTINUE
 1943 CONTINUE
      NLS=MINO(IS,LS)
      DO 1944 K=1,4
      KC=K+6
      CALL VASTAU(T,W,TOTO,VZEN,TAUW(1,K),ISAT,KC)
 1944 CONTINUE
      CALL VWRET(TS,TAUW,U,NL,NLS,TSP,USP)
 1949 CALL PRECW(P,W,U,NL)
C
      DO 1950 I=26,40
 1950 TD(I)=DEWPT(P(I),T(I),W(I))
      IF(IPASS.NE.0) GO TO 1951
      IF(IPASS.EQ.0) IPASS=1
C
C     RETURN TO RECALCULATE TRANSMITTANCE AND ATTEMPT ENHANCEMENT
      GO TO 1390
 1951 CONTINUE
      TOT=T(37)+TD(37)-2*T(31)
      IF (FORCE.NE.0.)GO TO 1960
      GTST=2.5
      IF(ITCH.EQ.6) GTST=2.0
      IF(TS.GT.305.) GTST=5.0
      IF(KUSE(ITCH).EQ.0) GO TO 1960
      IF(VDAT(ITCH).EQ.VMISG) GO TO 1960
      TEST=ABS(DELT(KSAV))
      IF(TEST.GT.GTST) GO TO 2200
 1960 CONTINUE
      CALL HTX(S,IS)
C     SET UP FOR STABILITY CALCULATION
      NB=IS
      IF (PSTA.LT.P(NB))NB=NB-1
      DO 2010 I=2,15
      PST(I)=P(NB)
      TST(I)=T(NB)
      TDST(I)=TD(NB)
 2010 NB=NB-1
      PST(1)=PSTA
      TST(1)=.01*ITSFC
```

```
      TDST(1)=.01 IDSFC
      IF(KBUG.NE.0)CALL ENKODE('("STAB. PRESS ", 10F8.2/)',LBUF,PST(1)
      CALL SADANL(0.,15,PST,TST,TDST,DIR,SPD,STABIL)
      LIFT=100. STABIL(8)
      DO 2020 I=1,NL
      J=NLP1-I
      U(I)=S(I)
C     ABOVE S(I) GOES WITH NEW HTX CALL
 2020 CONTINUE
      DELZ1=U(31)-U(37)
      DELZ2=U(24)-U(37)
      IF(KBUG.EQ.0) GO TO 2060
      CALL ENKODE('(T11,"DELZ1 =",F10.2,5X,"DELZ2 =",F10.2/)',
     LBUF,DELZ1,DELZ2)
 2060 CONTINUE
      IF(KBUG.EQ.0) GO TO 2180
      CALL ENKODE('(T6,"PRESSURE  ", 15F7.0/)',LBUF,P(26))
      CALL ENKODE('(T6,"T  PROFILE", 15F7.1/)',LBUF,T(26))
      CALL ENKODE('(T6,"DEW  POINT", 15F7.1/)',LBUF,TD(26))
      CALL ENKODE('(T6,"HEIGHT     ", 15F7.0/)',LBUF,U(26))
      CALL ENKODE('(T6,"MIX  RATIO", 15F7.3/)',LBUF,W(26))
      CALL ENKODE('(T11,"TCT =",F7.2/)',LBUF,TCT)
 2180 CONTINUE
C     COMPUTE AN AVERAGED TOTALS...
      C0=.3337*DELZ2-.8457*DELZ1
      C1=.14044*DELZ1-.0515*DELZ2
      T500=C0+C1*ALOG(500.)
      T850=C0+C1*ALOG(850.)
      R850=.0001 FLOAT(IHUM(1))
      P850=850.
      W850=R850 WSAT(P850,T850)
      D850=DEWPT(P850,T850,W850)
      TOTLS=T850+D850-2.*T500
      IT850=IROUND(T850*100.)
      ID850=IROUND(D850*100.)
      IT500=IROUND(T500*100.)
      ITOTLS=IROUND(TOTLS 100.)
      IDZ1=IROUND(DELZ1*10.)
      KDZ1=IROUND(DELZ1)
      IDZ2=IROUND(DELZ2)
      IF(LCRT.EQ.0) GO TO 2190
      CALL  ENKODE  ('(132X,T5,"T850",4X,"D850",4X,"T500  TOTALS 850-50
     850-200")',LBUF)
      CALL VTG(LBUF)
      CALL ENKODE('(6I8)',LBUF,IT850,ID850,IT500,ITOTLS,KDZ1,
     IDZ2)
      CALL VTG(LBUF)
 2190 NOGO=0
 2200 CONTINUE
C     PREPARE OUTPUT BUFFER
      IRETD(3)=7777
      IF (NOGO.NE.0)GO TO 2360
```

```
      IRETD(3)=0
C     ABOVE IS USER MOD FLAG TO CONTROL PACKING
C      CALCULATE FINAL TBB
C *** THESE CALCULATIONS ARE DEACTIVATED TO SAVE TIME...
      DO 2205 K=KC1,KC2
      IF (KUSE(K).EQ.0)GO TO 2205
      IF (ICB.EQ.0)GO TO 2205
      CALL VASTAU(T,W,TOTO,VZEN,TAU,ISAT,K)
      CALL VASRTE(TAU,T,TS,CRAD,B,TBB,DBDB,K,IA)
      IRETD(21+K)=100.*TBB+0.5
 2205 CONTINUE
      DO 2210 K=7,10
      J=K-6
      IF (ICB.EQ.0)GO TO 2210
      CALL VASTAU(T,W,TOTO,VZEN,TAUW(1,J),ISAT,K)
      CALL VASRTE(TAUW(1,J),T,TS,CRAD,B,TBB,DBDB,K,IA)
      IRETD(21+K)=100.*TBB+0.5
 2210 CONTINUE
      CALL PRECW(P,W,U,IS)
      IRETD(35)=U(IS)*1000+0.5
      IRETD(36)=100.*TS
      IF (ICB.EQ.1)GO TO 2218
      IF(PSTA.GE.850)IRETD(41)=TOT*100.
      IRETD(42)=ITOTLS
      IRETD(43)=LIFT
      DO 2215 K=1,3
      IF(IHUM(K).EQ.10000) IHUM(K)=MISG
      IF(IHUM(K).EQ.200) IHUM(K)=MISG
 2215 IRETD(43+K)=IHUM(K)
 2218 CONTINUE
      IRETD(50)=ITYP
      LS=NLP1-IS
      IF (LS.LT.2)LS=2
      DO 2220 K=2,20
      L=LMD(K)
      M=(K-1)*9
      N=NLP1-L
      IRETD(57+M)=S(L)
C     ABOVE CHANGED FROM N TO L TO CONFORM WITH HTX
      IRETD(51+M)=LCHR(K)
      IRETD(52+M)=P(L)*10.
      IF (P(L).GT.PSTA)GO TO 2220
C      STORE ONLY HEIGHTS BELOW SURFACE
      IRETD(53+M)=T(L)*100.
      IRETD(58+M)=TGS(K)*100.
      IF (K.GT.11)GO TO 2220
      IRETD(54+M)=TD(L)*100.
      IRETD(59+M)=DGS(K)*100.
 2220 CONTINUE
C     ADD SURFACE VALUES
      IRETD(51)=LCHR(1)
      IRETD(52)=IPSTA 10
```

```
      IRETD(53)=ITSFC
      IRETD(54)=IDSFC
      IRETD(57)=IELEV
      IRETD(58)=TGS(1)*100.
      IRETD(59)=DGS(1)*100.
C     MD OUTPUT BUFFER COMPLETE
      NRET=NRET+1
      IF (IPLT.EQ.0)GO TO 2360
C     REVERT TO -W+E CONVENTION
      SLON=-VLON
      CALL SATEAR(FTIME,FLIN,FELE,VLAT,SLON,2,INAV,EETAIN,RETDOT,0.)
      IL=FLIN+0.5
      IE=FELE+0.5
      CALL SATTV(IFRM,IL,IE,IRAS,IPIC,JSAT,JDAY,JTIME)
      JDAY=JSAT*100000+JDAY
      KOLOR=2
      IF (ITYP.EQ.22)KOLOR=3
      IF(IFLG.NE.0) KOLOR=1
      ITOTLS=(ITOTLS+50)/100
      CALL VASDIG(IRAS+MAG2,IPIC+MAG2,ITOTLS,MAG,1,KOLOR)
 2360 CONTINUE
      CALL VRTIC(IRET,NN,1)
 2380 CONTINUE
 2400 CONTINUE
 2500 CALL SDEST(' LAST RETRIEVAL NO. PROCESSED WAS ',NN)
      RETURN
      END
```

```
//GPVA7000 JOB  CLASS=A,MSGLEVEL=(0,0)
//* VLGPVA   SBG  12/23/83; ENTERED USER MANUAL CARD
//  EXEC  MCPRG,MOD=GPVA
//FORT.SYSIN  DD *
@PROCESS SC(TGMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,TSNIO,WTOR,CPCOM,SGX,SGW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW,DOPEN,DREAD,DWRITE,DCLOSE)
      SUBROUTINE MAIN0
C **** CRT DISPLAY OF VAS-RETRIEVAL GUESS AT CURSOR POSITION
C ? DISPLAY VAS GUESS PROFILE AT CURSOR LOC OR AT SPECIFIED LAT/LON (CM
C ? KEYIN: GPVA <GSS> <SFC> <LAT> <LON> <MDNG>
C ?    POSITIONAL PARAMETERS
C ?        GSS - 'C'  'P' FOR CLIM OR PROF..DEFAULT TO MDNG
C ?        SFC - NO SURFACE ANALYSES IF SET TO NON ZERO INTEGER
C ?        LAT - LATITUDE*100
C ?        LON - LONGITUDE*100 (POS W,NEG E)
C ?        MDNG - MD FILE FOR GUESS,DEFAULT TO 'VASTEXT' ENTRY
C   SSEC/MCIDAS USERS MANUAL - CHAP12
      DIMENSION KOUT(10),IP(15),LP(6),VDAT(13)
      DIMENSION IHDR(112)
      COMMON/ATMOS/P(40),T(40),W(40)
      COMMON/DBUG/KBUG
      COMMON/DOC/IDOC(112)
      COMMON/GUESS/TGES(15),DGES(6)
      COMMON/NAV/VLAT,VLON,VZEN,SZEN,IL,IE,IRAS,IPIC,ITIME,JTIME,JDAY
      COMMON/SURF/IZ10,ITSFC,IDSFC,IPSTA,IEL,LSTA
      CHARACTER*12 CPP,CGES
      CHARACTER*8 MFILE
      DATA MFILE /'VASTEXT '/,LUN/20/,LEN/100/
      DATA MISS/Z80808080/
      DATA IP/1000,850,700,500,400,300,250,200,150,100,70,50,30,20,10/
      DATA LP/40,37,35,31,28,26/
      CGES=CPP(1,' ')
      NOSFC=IPP(2,0)
      KBUG=IKWP('BUG',1,0)
      ICLIM=0
      IF (CGES.EQ.'C')ICLIM=1
      IF (CGES.EQ.'P')ICLIM=2
      VLAT=.01*FLOAT(IPP(3,0))
      VLON=.01*FLOAT(IPP(4,0))
      IF (LUC(81).NE.0)GO TO 111
      CALL TSNIO(1,1,1,1,1,1,IDOC)
      GO TO 112
  111 ITERM=IKWP('TERM',1,0)
      IF(ITERM.EQ.0)ITERM=LUC(-20)
C     READ VASTEXT FOR MD INFO
      CALL DOPEN(MFILE,LUN,LEN)
      CALL DREAD(LUN,ITERM,IDOC)
      CALL DCLOSE(LUN)
  112 MDNG=IPP(5,0)
      IF(MDNG.EQ.0)MDNG=IDOC(38)
      IF (VLAT.NE.0.)GO TO 3
```

```
      IFRM=LUC(-1)
      IRAS=LUC(-11)
      IPIC=LUC(-12)
      CALL TVSAT(IFRM,IRAS,IPIC,IL,IE,JS,JDAY,JTIME)
C      PUT THE SS BACK INTO SSYYDDDD
      JDAY=JS 100000+JDAY
      IF (LUC(81).EQ.0)GO TO 115
      VDAT(1)=-1.
C     GET VAS NAVIGATION IF REQUIRED
      CALL VASDAT(IL,IE,VDAT)
    3 LAT=IROUND(VLAT 100.)
C     HRTOPO NEEDS LON + EAST
      LON=IROUND(-VLON 100.)
      GO TO 120
  115 CONTINUE
      CALL TSNIO(1,1,1,1,1,1,IHDR)
      CALL TSNIO(0,1,IL,IE,1,1,LAT)
      CALL TSNIO(0,2,IL,IE,1,1,LON)
      VLAT=.01 LAT
C     COMMON /NAV/ NEEDS VLON + WEST
      VLON=-.01 LON
  120 CONTINUE
      CALL HRTOPO(LAT,LON,IEL,ICH)
      IF (ICLIM.EQ.1)CALL SDEST(' CLIMATOLOGICAL GUESS',0)
      CALL GESPRO(ICLIM,NOSFC,MDNG)
      CALL SDEST('      LAT       LON       ELEV      PSTA      Z1000     TSFC
     SFC',0)
      KOUT(1)=7
      KOUT(2)=LAT
      KOUT(3)=-LON
      KOUT(4)=IEL
      KOUT(5)=IPSTA
      KOUT(6)=IZ10
      KOUT(7)=ITSFC
      KOUT(8)=IDSFC
      CALL OUTINT(KOUT)
      CALL SDEST('    LEVEL    PRESS      TEMP      DEWP      WMIX',0)
      DO 180 I=1,6
      KOUT(1)=5
      KOUT(2)=I
      KOUT(3)=IP(I)
      KOUT(4)=IROUND(TGES(I)*100.)
      KOUT(5)=IROUND(DGES(I)*100.)
      J=LP(I)
      KOUT(6)=IROUND(W(J)*1000.)
      CALL OUTINT(KOUT)
  180 CONTINUE
      DO 200 I=7,15
      KOUT(1)=3
      KOUT(2)=I
      KOUT(3)=IP(I)
      KOUT(4)=IROUND(TGES(I)*100.)
```

```
      CALL OUTINT(KOUT)
  200 CONTINUE
      RETURN
      END
```

```
//ESVA5600 JOB CLASS=A,MSGLEVEL=(0,0)
//  VLESVA   JMN  03/07/84: MEMBER UPDATED
//  VLESVA   SBG  12/23/83; ENTERED USER MANUAL CARD
// EXEC MCPRG,MOD=ESVA
//FORT.SYSIN DD
      SUBROUTINE MAIN0
C ?      EDIT SURFACE REPORT AT CURSOR LOCATION
C ?       KEYIN: ESVA  KEYWORDS
C ?          KEYWORDS:
C ?             'Z100' 1000MB HEIGHT VALUE
C ?             'TSL' TEMPERATURE AT SEA  LEVEL
C ?             'DD' DEWPOINT DEPRESSION
C ?             'TSFC' SURFACE TEMP TO BE CONVERTED TO TSL
C ?      IF NO RPT. EXISTS IN CURSOR, ONE IS ADDED; OTHERWISE VALUES
C ?      OF KEYWORDS ARE SUBSTITUTED. VALUE OF 1 ENTERS 'MISS' (EDITS).
C ?         N.B.  TSL AND DD MUST BE K 100, Z100 IS M
C    SSEC/MCIDAS USERS MANUAL - CHAP12
       DIMENSION MF(64),JOUT(25),IOUT(25)
       DIMENSION LIST(25),ISCL(25),IUN(25),LOCS(25)
       DIMENSION KOUT(10)
       INTEGER*4 CBUF(20)
       CHARACTER*8 MFILE
       REAL*8 DLIT
       COMMON /DOC/IDOC(112)
       COMMON /NAV/FLAT,FLON,ZENLOC,SZEN,IL,IE,IRAS,IPIC,IHMS,JT,JD
       COMMON /IDENT/IYMD,JHMS,NROWM,NSAT
       COMMON /ANALS/NOAN,LTOP
       COMMON /ORBIT/MODE
       COMMON /TIGHT/ITOL
       COMMON /ENTRY/INIT
       COMMON/ORIENT/YCOORD,XCOORD
       COMMON/DEBUG/KBUG
       DATA MFILE,LUN,LEN/'VASTEXT ',20,100/
       DATA IOUT /25*Z80808080/,JOUT/25*Z80808080/
       DATA MISS/Z80808080/,NCOLS/56/,ICOL/1/,NDAT/0/
       IWRM=LUC(-5)
       IFRM=LUC(-1)
       IRAS=LUC(-11)
       IPIC=LUC(-12)
       CALL TVSAT(IFRM,IRAS,IPIC,IL,IE,JS,JD,JT)
C      SAVE LOCATION FOR POSSIBLE ADDITION
       ALIN=FLOAT(IL)
       AELE=FLOAT(IE)
       IF (LUC(81).EQ.0)GO TO 110
C      READ VASTEXT CONTEXT FILE
       ITERM=LUC(-20)
       CALL DOPEN(DLIT(MFILE),LUN,LEN)
       CALL DREAD(LUN,ITERM,IDOC)
       CALL DCLOSE(LUN)
C      INITIALIZE NAVIGATION
       JD=JS*100000+JD
       CALL NVINIT(RETAIN,BETDOT,INAV,PTIME)
```

```
      GO TO 120
  110 CALL TSMIC(1,1,1,1,1,1,IDOC)
      INIT=1
      NODE=IDOC(7)
C     SET UP AREA FOR IMAGE SPACE
      CALL TVSAT(IFRM,005,335,LTOP,LELE,ISS,ID,IT)
      CALL TVSAT(IFRM,495,335,LBOT,LELE,ISS,ID,IT)
      IF (LTOP.LT.1)LTOP=1
      IF (LBOT.GT.NROWM)LBOT=NROWM
      NROWS=LBOT-LTOP+1
      NPTS=NROWS NCOLS
  120 CONTINUE
      MDNO=IDOC(36)
      MDR=IDOC(37)
    2 IF (MDOPEN(MDNO,2).NE.0)GO TO 900
C     READ ROW HEADER RECORD
      IOK=MDGET(MDNO,MDR,0,IOUT)
      IF (IOK.LT.0)GO TO 902
      MREC=IOUT(3)
C     READ TEST RECORD
      M=0
    3 M=M+1
      IF (M.GT.10)GO TO 904
      IOK=MDGET(MDNO,MDR,M,IOUT)
      IF (IOK.LT.0)GO TO 3
      KBUG=IKWP('BUG',1,0)
      IF (KBUG.NE.0)CALL SDEST(' OPERATING WITH ROW= ',MDR)
C     READ IN KEYS
      NKEYS=MDKEYS(MDNO,-1,LIST,ISCL,IUN,LCCS)
C     FIND DATA-ADDRESSES IN KEYLIST
      DO 11 N=1,NKEYS
      IF (LIST(N).EQ.LIT('LAT '))NLAT=N
      IF (LIST(N).EQ.LIT('LON '))NLON=N
      IF (LIST(N).EQ.LIT('HMS '))NTIM=N
      IF (LIST(N).EQ.LIT('Z100'))NZ=N
      IF (LIST(N).EQ.LIT('TSL '))NT=N
      IF (LIST(N).EQ.LIT('DD  '))ND=N
   11 CONTINUE
      KOUT(1)=5
      KOUT(2)=NLAT
      KOUT(3)=NLON
      KOUT(4)=NZ
      KOUT(5)=NT
      KOUT(6)=ND
      IF (KBUG.NE.0)CALL OUTINT(KOUT)
      IZ100=IKWP('Z100',1,0)
      ITMSL=IKWP('TSL',1,0)
C     ADD OPTION TO KEY IS SURFACE TEMP
      ITSFC=IKWP('TSFC',1,0)
      IDD=IKWP('DD',1,0)
      IF (IZ100.EQ.1)IZ100=MISS
      IF (ITMSL.EQ.1)ITMSL=MISS
```

```
      IF (IDD.EQ.1)IDD=MISS
      ITEM=0
      IF (IZ100.NE.0)ITEM=IZ100
      IF (ITMSL.NE.0)ITEM=ITMSL
      IF (IDD.NE.0)ITEM=IDD
      IF (ITSFC.NE.0)ITEM=ITSFC
      IF (ITEM.EQ.0)GO TO 80
      IF (ITEM.EQ.MISS)ITEM=999
      IF (KBUG.NE.0)CALL SDEST(' INPUT DATUM = ',ITEM)
      CALL GETFRM(IFRM,MF)
      MAG=MF(10)
      IF (MAG.LT.10)MAG=10
      CALL INITPL(IWRM,0)
      IF (LUC(81).EQ.0)GO TO 60
C
C     FOR VAS GET CURSOR DIMENSIONS IN LAT/LON
      IRINC=LUC(-9)
      IPINC=LUC(-10)
      IRS=LUC(-11)
      IPC=LUC(-12)
      IRAS=IRS-IRINC/2
      IPIC=IPC-IPINC/2
      CALL TVSAT(IFRM,IRAS,IPIC,IL,IE,JS,JD,JT)
      KOUT(2)=IL
      KOUT(3)=IE
      FLIN=FLOAT(IL)
      FELE=FLOAT(IE)
      CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON,1,INAV,BETAIN,BETDOT,0.)
C     CONVERT TO +W-E CONVENTION
      FLON=-FLON
      KOUT(4)=ILALO(FLAT)
      KOUT(5)=ILALO(FLON)
      ILAMAX=IROUND(FLAT*10000.)
      ILOMAX=IROUND(FLON*10000.)
      IRAS=IRS+IRINC/2
      IPIC=IPC+IPINC/2
      CALL TVSAT(IFRM,IRAS,IPIC,IL,IE,JS,JD,JT)
      FLIN=FLOAT(IL)
      FELE=FLOAT(IE)
      CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON,1,INAV,BETAIN,BETDOT,0.
C     CONVERT TO +W-E CONVENTION
      FLON=-FLON
      KOUT(6)=ILALO(FLAT)
      KOUT(7)=ILALO(FLON)
      ILAMIN=IROUND(FLAT*10000.)
      ILOMIN=IROUND(FLON*10000.)
      KOUT(1)=6
      IF (KBUG.NE.0)CALL OUTINT(KOUT)
C     NAV COMPLETE
      GO TO 68
C     PICK UP TOVS LOCATION
   60 CALL TSNIC(0,1,IL,IE,1,1,LAT)
```

```
      CALL TSNIO(0,2,IL,IE,1,1,LON)
      ALAT=LAT/100.
      ALON=LON/100.
      ILAMIN=LAT*100-2000
      ILAMAX=ILAMIN+4000
      ILOMIN=-LON*100-2000
      ILOMAX=ILOMIN+4000
      KOUT(1)=4
      KOUT(2)=ILAMIN
      KOUT(3)=ILAMAX
      KOUT(4)=ILOMIN
      KOUT(5)=ILOMAX
      IF (KBUG.NE.0)CALL OUTINT(KOUT)
   68 DO 70 M=1,MREC
      IOK=MDGET(MDNO,MDR,M,IOUT)
      KOUT(1)=3
      KOUT(2)=M
      KOUT(3)=IOUT(NLAT)
      KOUT(4)=IOUT(NLON)
      IF (KBUG.NE.0)CALL OUTINT(KOUT)
C     DELETE EVERYONE WITHIN CURSOR
      IF (IOUT(NLAT).LT.ILAMIN.OR.IOUT(NLAT).GT.ILAMAX)GO TO 70
      IF (IOUT(NLON).LT.ILOMIN.OR.IOUT(NLON).GT.ILOMAX)GO TO 70
C     THIS SOUNDING MUST GO
      FLAT=IOUT(NLAT)*.0001
      FLON=IOUT(NLON)*.0001
      IF (LUC(81).EQ.0)GO TO 69
C     LOCATE RASTER AND PIXEL OF SOUNDING
C     REVERT TO -W+E CONVENTION
      FLON=-FLON
      CALL SATEAR(FTIME,FLIN,FELE,FLAT,FLON,2,INAV,EETAIN,EETDOT,0.)
      IL=FLIN+0.5
      IE=FELE+0.5
      GO TO 690
   69 CONTINUE
C     USE TOVS NAVIGATION ROUTINE
      LATS=FLAT*100.
      LONGS=FLON*100.
      CALL SRCH(LATS,LONGS,IM,IL,IE,NPTS,NROWS)
      IF (KBUG.NE.0)CALL SDEST(' SRCH GIVES IM = ',IM)
      IF (IM.EQ.0)GO TO 70
  690 CALL SATTV(IFRM,IL,IE,IRAS,IPIC,JS,JD,JT)
      KOUT(1)=4
      KOUT(2)=IL
      KOUT(3)=IE
      KOUT(4)=ILALO(FLAT)
      KOUT(5)=ILALO(FLON)
      IF (KBUG.NE.0)CALL OUTINT(KOUT)
      IF (IZ100.NE.0)IOUT(NZ)=IZ100
      IF (ITMSL.NE.0)IOUT(NT)=ITMSL
      IF (IDD.NE.0)IOUT(ND)=IDD
      IOK=MDPUT(MDNO,MDR,M,IOUT)
```

```
      IF (IOK.NE.0)GO TO 906
      IRAX=IRAS+MAG
      IPIX=IPIC+MAG
      JHI=MAG*4/3
      JWI=1
      CALL PLTDIG(IRAX,IPIX,ITEM,JHI,JWI,ICOL)
      NDAT=1
   70 CONTINUE
      IF (NDAT.NE.0)GO TO 100
      IF (ITEM.EQ.999)GO TO 907
      GO TO 85
   80 CALL SDEST(' NO ACTION INDICATED AT THIS LOCATION',0)
      RETURN
   85 IF(LUC(81).NE.0)CALL SATEAR(PTIME,ALIN,AELE,ALAT,ALON,1,INAV,
      BETAIN,BETOCT,0.)
C     PREPARE TO ADD VALUE(S)
      IF (IZ100.EQ.0)IZ100=MISS
      IF (ITMSL.EQ.0)ITMSL=MISS
      IF (IDD.EQ.0)IDD=MISS
      IF (ITSFC.EQ.0)GO TO 300
C     CHECK REASONABLENESS OF TOPOGRAPHY
      MLAT=ALAT*100.
      MLON=ALON*100.
      CALL HRTOPO(MLAT,MLON,IEL,ICH)
      Z=IEL
      TSL=.01*FLOAT(ITSFC)+Z*0.0065
      ITMSL=TSL*100.
  300 CONTINUE
C   CHANGE SIGN OF LONGITUDE FOR MD FILE
      ALON=-ALON
      JOUT(1)=IOUT(1)
      JOUT(2)=IOUT(2)
      JOUT(3)=IOUT(3)+1
      JOUT(4)=0
      JOUT(NLAT)=IROUND(ALAT*10000.)
      JOUT(NLON)=IROUND(ALON*10000.)
      JOUT(NTIM)=JOUT(2)
      JOUT(NZ)=IZ100
      JOUT(NT)=ITMSL
      JOUT(ND)=IDD
      IOK=MDPUT(MDNO,MDR,JOUT(3),JOUT)
      IF (IOK.NE.0)GO TO 906
      IOK=MDPUT(MDNO,MDR,0,JOUT)
      IF (IOK.NE.0)GO TO 906
      IRAX=IRAS+MAG
      IPIX=IPIC+MAG
      JHI=MAG*4/3
      JWI=1
      CALL PLTDIG(IRAX,IPIX,ITEM,JHI,JWI,ICOL)
  100 CALL ENDPLT
      IF (IKWP('PACK',1,0).NE.1)RETURN
      CALL ENKODE(' (K5,7X,"COL,CMAX",4X,"ROW,",K2,6X)',OBUF,MDNO,MDR)
```

```
      CALL USCX('MDPACK  ',OBUF(1),OBUF(4),OBUF(7))
      RETURN
  900 CALL SDEST(' CANNOT OPEN MDFILE NO. ',MDNO)
      RETURN
  902 CALL SDEST(' CANNOT LOCATE ROW NO. ',MDR)
      RETURN
  904 CALL SDEST(' TOO MANY ERRORS IN READING ROW NO. ',MDR)
      RETURN
  906 CALL SDEST(' TROUBLE WRITING DATA RECORD NO ',M)
      RETURN
  907 CALL SDEST(' CANNOT LOCATE REPORT TO EDIT ',C)
      RETURN
      END
```

```
//VTPW5670 JOB CLASS=B,MSGLEVEL=(0,0)
//  VLVTPW    HMW   04/09/84: MEMBER UPDATED
//  VLVTPW    HMW   03/30/84: OBTAIN 'BEG' & 'END' FROM VASTEXT
//  VLVTPW    HMW   03/16/84: INSTALL MD-OUTPUT-FREQUENCY OPTION
//  VLVTPW    HMW   03/16/84: CORRECT GUESS-HANDLING
//  VLVTPW    GSW   03/06/84: CORRECT IMAGE FILL/BOXINESS.
//  VLVTPW    GSW   03/05/84: HANDLE 'VENETIAN BLIND' IMAGERY.
//  VLVTPW    WLS   03/02/84: ADDED 'STAB' ESTIMATE.
// EXEC MCPROG,MOD=VTPW
//FORT.SYSIN DD
      SUBROUTINE MAIN0
C ?   VAS TOTAL PRECIPITABLE WATER VAPOR RETRIEVAL (H.M.WOOLF)
C ?     KEYIN: VTPW <KEYWORDS>
C ?       KEYWORDS (DEFAULTS IN PARENTHESES):
C ?         ARA=NUMBER OF DIGITAL AREA FOR IMAGING (0)
C ?         BOX=N (SIZE IN FOV'S OF SQR BOX (11)) @EVEN-LC,ODD-SC@
C ?         SPC=IL IE (LINE AND ELEMENT SPACING OF RTVLS (BOX BOX))
C ?         BEG=IL IE(FIRST LINE AND ELEMENT TO DEFINE AREA (VASTEXT))
C ?         END=LL LE (LAST LINE AND ELEMENT TO DEFINE AREA (VASTEXT))
C ?         NTH=FREQUENCY WITH WHICH TO PUT RTVLS IN MD-FILE (2)
C ?         ... 1 = EVERY ONE, 2 = EVERY OTHER, 4 = EVERY FOURTH, ETC.
C ?         GSS=C (CLIM) G (GRID,DEFAULT)
C ?         SFC=1 (NO SURFACE ANALYSIS (0))
C ?         TCH=5 (TEMP.CHANNEL (6))
C ?         PLT=1 (PLOT RESULTS ON GRAPHICS (0))
C ?         CRT=1 (FOR CRT OUTPUT (0))
C ?         BUG=1 (FOR DEBUG DIAGNOSTICS (0))
C ?         TER=N (TERMINAL NUMBER (LOCAL))
C ?     THE FOLLOWING POINTERS MUST BE SET WITH 'SPVA':
C ?        MDNR,MDRR,NRET ... ALWAYS
C ?        ... IF MDRR .LT. 0, RESULTS ARE NOT OUTPUT TO MD FILE
C ?        MDNG,MDRG ... FOR GRID GUESS
C ?        MDNS,MDRS,NGFG,NGFS,ZGRID,TGRID,DGRID ... FOR SFC ANALYSIS
C ? NOTE *** MDNR MUST BE CREATED WITH SCHEMA 'VTWV', NOT 'VRET'
      DIMENSION LBUF(33),NSPIN(13),VDAT(13),VAS8(11,11)
      DIMENSION IRET(250),IAMES(8),IARET(660),IABUF(165),NFAIL(7)
      DIMENSION FARRAY(11,11),LFLAG(11,11),RATIO(11)
      INTEGER*2 IARRAY(11,11,60)
      CHARACTER*12 CGES,CKWP,CBLNK,CLETC,CLETG
      COMMON/ARENT/IDIR(64)
      COMMON/ATMOS/P(40),T(40),W(40)
      COMMON/AUTO/IBOX,ILU,ILL,IEL,IER,MAG
      COMMON/DANGLE/VLATS,VLONS
      COMMON/DEBUG/LBUG
      COMMON/DOC/IDOC(100)
      COMMON/GEE/GV(12),DV(12),EV(12)
      COMMON/GUESS/TGES(15),DGES(6)
      COMMON/LAST/LASLIN,LASELE,LELEV,ICHAR
      COMMON/MODE/IDET(13),ISPIN(13)
      COMMON/NAV/VLAT,VLON,VZEN,SZEN,IL,IE,IRAS,IPIC,ITIME,JTIME,JDAY
      COMMON/RADV/VRAD(13)
      COMMON/SIZE/NBXS
```

```
      COMMON/SURF/IZ10,ITSFC,IDSFC,IPSTA,IELEV,LSTA
      COMMON/TERMN/ITERM
      COMMON/BOX/KBOX,LINET,NOSFC
      DATA CBLNK/' '/,CLETC/'C'/,CLETG/'G'/
      DATA ICV/7/,ICW/8/
      DATA TOTC/347./
      DATA VMISG/999999./,MISG/Z80808080/
      DATA MAXARA/9999/,MAXWV/255/
      DATA NL/40/,LENV/18/
C     *****
C     FOLLOWING IS PROGRAM VERSION DATE ... KEEP CURRENT!!!
      IVER=84100
      MCINIT=0
      IF(LUC(-25).EQ.1) MCINIT=LUC(-23)
      CALL CALDAY(IVER,IVY,IVM,IVD,IVMO)
      CALL ENKODE('(132X,T1,"BEGIN * VTPW * VERSION OF ",
     * I2,1X,A4,I2," AT INIT ",I2/)',LBUF,IVD,IVMO,IVY,MCINIT)
C     *****
C   LOOK FOR DIGITAL AREA NUMBER
      NARA=IKWP('ARA',1,0)
      IF(NARA.GT.MAXARA) GO TO 900
C   CHECK FOR OVERRIDE OF TERMINAL NUMBER
      IDEF=LUC(-20)
      ITERM=IKWP('TER',1,IDEF)
C   READ 'VASTEXT' DOCUMENTATION RECORD & EXTRACT NEEDED INFO
      CALL VRTIC(IRET,0,0)
      JDAY=IDOC(1)
C ++++ N.B. - 'JDAY' IS 'SSYYDDD'
      JTIME=IDOC(2)
      MDNG=IDOC(38)
      MDNP=IDOC(40)
      MDRR=IDOC(41)
C   CHECK FOR 'NO-OUTPUT-MEDIUM'
      IF(NARA.EQ.0.AND.MDRR.LT.0) GO TO 920
      LRC=IDOC(100)
      CGES=CKWP('GSS',1,' ')
      IGES=0
      IF (CGES.EQ.CBLNK) GO TO 100
      IF (CGES.EQ.CLETG) GO TO 100
      IF (CGES.EQ.CLETC) IGES=1
  100 CONTINUE
      IFILL=0
      IDEF=0
      NOSFC=IKWP('SFC',1,IDEF)
      IDEF=11
      NBXS=IKWP('BOX',1,IDEF)
      IF(MOD(NBXS,2).EQ.0) IFILL=1
      IF(MOD(NBXS,2).EQ.0) NBXS=NBXS-1
      NBXS=MIN0(NBXS,11)
      IBOX=1
      IF(NBXS.EQ.1) IBOX=0
      IDEF=NBXS
```

```
      INCRL=IKWP('SPC',1,IDEF)
      INCRE=IKWP('SPC',2,IDEF)
C
      LLINE=IKWP('END',1,IDOC(55))
      LELEM=IKWP('END',2,IDOC(56))
      ILINE=IKWP('BEG',1,IDOC(57))
      IELEM=IKWP('BEG',2,IDOC(58))
C
      LBUG=IKWP('BUG',1,0)
      LCRT=IKWP('CRT',1,0)
      IPLT=IKWP('PLT',1,0)
      KTCH=IKWP('TCH',1,6)
      NLP1=NL+1
      IFRM=LUC(-1)
      INRAS=LUC(-11)
      INPIC=LUC(-12)
      IF(MDRR.LT.0) GO TO 120
      NTH=IKWP('NTH',1,2)
      IRET(1)=JDAY
      IRET(2)=JTIME
      IRET(3)=LRC
      IOK=MDPUT(MDNR,MDRR,0,IRET)
      IF(IOK.NE.0) GO TO 880
  120 IF(ILINE*IELEM.EQ.0) GO TO 140
      IF(LUC(16).EQ.0) GO TO 180
      CALL GETFRM(IFRM,IDIR)
      JSAT=IDIR(1)
      JDATE=IDIR(2)
      GO TO 160
  140 CALL TVSAT(IFRM,INRAS,INPIC,ILINE,IELEM,JSAT,JDATE,JTIME)
C - NEED 'OLDSTYLE' JDAY FOR NAVIGATION WITHIN VASDAT
  160 JDAY=JSAT 100000+JDATE
  180 IF((LLINE LELEM).NE.0) GO TO 200
      LLINE=ILINE
      LELEM=IELEM
  200 CONTINUE
      VDAT(1)=-1.
      ILINES=ILINE
      IELEMS=IELEM
C  ** VASDAT CHANGES ARGUMENTS 'ILINES' AND 'IELEMS'
      CALL VASDAT(ILINES,IELEMS,VDAT)
      VDAT(1)=VMISG
      JSAT=IDIR(3)
      ISAT=ISATNV(JSAT)
      JDATE=IDIR(4)
      JDAY=JSAT 100000+JDATE
      JTIME=IDIR(5)
      ILRES=IDIR(12)
      IERES=IDIR(13)
C     SET DEFAULT TO SQUARE SAMPLES..IE EQUATE RESOLUTIONS
      INC=ILRES/IERES
      INCIL=ILRES INCRL
```

```
      IF(INCIL.EQ.0) INCIL=1
      INCIE=IERES*INCRF*INC
      IF(INCIE.EQ.0) INCIE=1
      MAG2=MAG/2
      MAG=MAX0(MAG,6)
      IF(NARA.EQ.0) GO TO 220
C
C   SET UP AREA AND ASSOCIATED PARAMETERS
      INCLA=ILRES
      INCEA=IERES*INC
      NBXH=NBXS/2
      ILINA=ILINE-NBXH*INCLA
      IELEA=IELEM-NBXH*INCEA
      LLINA=LLINE+NBXH*INCLA
      LELEA=LELEM+NBXH*INCEA
C
C  EMPIRICAL ADJUSTMENT!
      ILINA=ILINA-1
      LLINA=LLINA-1
C
      MLIN=(LLINA-ILINA)/INCLA+1
      MELE=(LELEA-IELEA)/INCEA+1
      IF(MELE.GT.660) GO TO 940
      CALL ARASIZ(NARA,MLIN,MELE)
      CALL ENAREA(NARA,JSAT,JDATE,JTIME,ILINA,IELEA,INCLA,INCEA,IAMES)
      CALL OPNA(NARA)
      KLINES=NBXS
      KELEMS=NBXS
      KBOXES=MELE/KELEMS
      IF(MOD(MELE,KELEMS).NE.0) KBOXES=KBOXES+1
      IAREC=0
      IF(LCRT.EQ.0) GO TO 220
      CALL ENKODE('("AREA ",I4," OPENED WITH NL =",I4,", NE =",I4)',
     * LBUF,NARA,MLIN,MELE)
      CALL VTG(LBUF)
  220 CONTINUE
C
      NDONE=0
      LINET=-1
      DO 820 IL=ILINE,LLINE,INCIL
      LINET=LINET+1
      IF(LCRT.EQ.0) GO TO 240
      CALL ENKODE('("BEGIN LINE ",I4)',LBUF,IL)
      CALL VTG(LBUF)
  240 LASLIN=-1
      KBOX=0
C
      DO 740 IE=IELEM,LELEM,INCIE
      KBOX=KBOX+1
      LASELE=-1
      IGFLG=1
      ITFLG=1
```

```
      DO 640 KLIN=1,KLINES
      DO 620 KELE=1,KELEMS
      IUPET=MAXWV
      DO 260 I=4,LENV
  260 IRET(I)=MISG
      DO 280 I=1,13
      VDAT(I)=VMISG
  280 NSPIN(I)=0
      ILINES=IL
      IELES=IE
      CALL VASDAT(ILINES,IELES,VDAT)
      IFAIL=1
      IF(ABS(VLATS).GE.90.) GO TO 600
      IF(ABS(VZEN).GT.60.) GO TO 600
      IF(LELEV.EQ.999999) GO TO 600
      IF(VDAT(1).LT.0.) GO TO 600
      IMF=1
      MSAM=0
      DO 320 I=1,12
      IF(VDAT(I).LT.180.OR.VDAT(I).GT.330.) VDAT(I)=VMISG
      IF(VDAT(I).EQ.VMISG) GO TO 320
      IF(IMF.EQ.0) GO TO 300
      MSAM=MSAM+1
      IMF=0
  300 CONTINUE
      URET=VMISG
      IF(VDAT(ICV).EQ.VMISG) GO TO 600
      IF(VDAT(ICW).EQ.VMISG) GO TO 600
C     SAVE REPRESENTATIVE SPIN BLOCK (ASSUME SPIN BUDGET INVARIANT)
      NSPIN(I)=ISPIN(I)
  320 CONTINUE
      IELEV=LELEV
  340 CONTINUE
      IFAIL=2
      IF(MSAM.EQ.0) GO TO 600
      IF(IGFLG.EQ.0) GO TO 480
      CALL GESPRC(IGES,NOSFC,MDNG)
      IFAIL=3
      IF(TGES(1).LE.0.) GO TO 600
      IGFLG=0
      PSTA=IPSTA
      TSTA=.01 FLOAT(ITSFC)
      TDSTA=0.01 FLOAT(IDSFC)
      E=SATVAP(TDSTA)
      WSTA=(622.*E)/(PSTA-E)
      LS=NL
      DO 360 J=1,20
      I=NLP1-J
      DP=P(I)-PSTA
      IF(DP.GT.0.) GO TO 360
      IS=I+1
      IS=MIN0(IS,NL)
```

```
      GO TO 380
  360 CONTINUE
  380 NLS=MIN0(IS,LS)
      DTS=TSTA-T(NLS)
      ILO=37
      IF(IGES.EQ.1) ILO=26
      IF(ILO.GE.NLS) ILO=NLS-3
      DTDF=DTS/(P(NLS)-P(ILO))
      DO 440 I=1,40
      IF(I.GE.20) GO TO 400
      W(I)=.02
      GO TO 440
  400 CONTINUE
      IF(I.GE.26) GO TO 420
      E=SATVAP(T(I))
      W(I)=0.2*(622*E)/(P(I)-E)
      GO TO 440
  420 W(I)=W(I)*WSTA/W(NLS)
      T(I)=T(I)+DTDF*(P(I)-P(ILO))
      IF(I.LE.NLS) GO TO 440
      T(I)=TSTA
      W(I)=WSTA
  440 CONTINUE
  460 CONTINUE
      IF(LBUG.EQ.0) GO TO 480
      CALL ENCODE('(132X,T1,5X,"PRESSURE   ", 15F7.1/)',LBUF,P(26))
      CALL ENCODE('(5X,"GUESS TEMP", 15F7.1/)',LBUF,T(26))
      CALL ENCODE('(5X,"GUESS WVMR", 15F7.3/)',LBUF,W(26))
      CALL ENCODE('(1X,"Z1000 =",I8,5X,"TSFC =",I8,5X,"TDSFC =",I8/)',
     * LBUF,IZ10,ITSFC,IDSFC)
  480 CONTINUE
      LAT=IROUND(VLATS*100.)
      LON=IROUND(VLONS*100.)
      IVZ=IROUND(VZEN*100.)
      ISZ=IROUND(SZEN*100.)
      IF(LBUG.EQ.0.AND.LCRT.EQ.0) GO TO 500
      CALL ENCODE('(4X,"LINE",4X,"ELEM",5X,"LAT",5X,"LON",4X,"LZEN",
     * 4X,"SZEN")',LBUF)
      IF(LCRT.NE.0) CALL VTQ(LBUF)
      IF(LBUG.NE.0) CALL LTQ(LBUF)
      CALL ENCODE('(132X,T1,6I8)',LBUF,IL,IE,LAT,LON,IVZ,ISZ)
      IF(LBUG.NE.0) CALL LTQ(LBUF)
      IF(LCRT.EQ.0) GO TO 500
      CALL VTQ(LBUF)
      CALL SDEST('  SAMPLE = ',MSAM)
  500 CONTINUE
C    DATA ACQUISITION COMPLETE
      CALL VASTPW(VDAT,TOTO,TSFS,URET,ITFLG,ISAT,KTCH,NLS,STAB)
      IFAIL=7
      IF(URET.EQ.VMISG) GO TO 600
      IFAIL=0
      KFLAG=2
```

```
C    OUTPUT ONLY EVERY 'NTH' RETRIEVAL TO MD-FILE
        NDONE=NDONE+1
        MDOUT=0
        IF(MOD(NDONE,NTH).EQ.1) MDOUT=1
        IF(NTH.EQ.1) MDOUT=1
C     MAXIMUM PERMISSIBLE VALUE OF 'URET' IS 12 CM
        IURET=IROUND(URET*20.)
C
        IF(MDRR.LT.0) GO TO 540
        IF(MDOUT.EQ.0) GO TO 540
        K=10
        DO 520 I=KTCH,8
        IF(KTCH.EQ.5.AND.I.EQ.6) GO TO 520
        K=K+1
        IF(VDAT(I).EQ.VMISG) GO TO 520
        IRET(K)=IROUND(VDAT(I)*100.)
  520   CONTINUE
        IRET(14)=IROUND(TSFS*100.)
  540   IF(LBUG.EQ.0.AND.LCRT.EQ.0) GO TO 560
        CALL ENKODE('(132X,T1," *** T-P-W = ",F6.3)',LBUF,URET)
        IF(LCRT.NE.0) CALL VTO(LBUF)
        IF(LBUG.NE.0) CALL LTO(LBUF)
  560   CONTINUE
        IF(IPLT.EQ.0) GO TO 580
        CALL SATTV(IFRM,ILINES,IELES,IRAS,IPIC,JSAT,JDAY,JTIME)
        JDAY=JSAT*100000+JDAY
        IUR=IROUND(URET*10.)
        KOLOR=KFLAG
        CALL VASDIG(IRAS+MAG2,IPIC+MAG2,IUR,MAG,1,KOLOR)
  580   CONTINUE
        IF(MDRR.LT.0) GO TO 600
        IF(MDOUT.EQ.0) GO TO 600
        IRET(4)=2-KFLAG
        IRET(5)=IROUND(VLATS*10000.)
        IRET(6)=IROUND(VLONS*10000.)
        IRET(7)=IROUND(VZEN*10000.)
        IRET(8)=ILINES
        IRET(9)=IELES
        IRET(10)=MSAM
        IRET(15)=IROUND(URET*1000.)
        IRET(16)=IROUND(STAB*1000.)
        LRO=LRO+1
        IOK=MDPUT(MDNR,MDRR,LRO,IRET)
        IF(IOK.NE.0) GO TO 860
        IRET(3)=LRO
        IOK=MDPUT(MDNR,MDRR,0,IRET)
        IF(IOK.NE.0) GO TO 880
  600   CONTINUE
        IF(IFAIL.NE.0) NFAIL(IFAIL)=NFAIL(IFAIL)+1
        IF(NARA.EQ.0) GO TO 620
        VAS8(KELE,KLIN)=VDAT(8)
        FARRAY(KELE,KLIN)=URET
```

```
C
      LFLAG(KELE,KLIN)= IFAIL
      IARRAY(KELE,KLIN,KBOX)=IURET
C
  620 CONTINUE
  640 CONTINUE
      IF(IFILL.NE.0) GO TO 720
      SRATIO=0.
      XMIN=VMISG
      DO 700 IIE=1,KELEMS
      RATIO(IIE)=0.
      S=0.
      DO 680 IIL=1,KLINES
      IF(LFLAG(IIE,IIL).EQ.7) GO TO 680
      X=VAS8(IIE,IIL)
      Y=FARRAY(IIE,IIL)
      IF(X.EQ.VMISG.OR.Y.EQ.VMISG) GO TO 660
      X=X-TSTA+3.
      IF(X.LE.0.) GO TO 660
      XMIN=AMIN1(XMIN,X)
      RATIO(IIE)=Y*X+RATIO(IIE)
      S=S+1.
      SRATIO=RATIO(IIE)/S
  660 CONTINUE
      IF(SRATIO.EQ.0.) GO TO 680
      IF(LFLAG(IIE,IIL).EQ.7) GO TO 680
      Y=FARRAY(IIE,IIL)
      IF(Y.NE.VMISG) GO TO 680
      X=VAS8(IIE,IIL)
      IF(X.EQ.VMISG) GO TO 680
      X=X-TSTA+3.
      IF(X.LE.0.) GO TO 680
      IF(X.LE.XMIN) GO TO 680
      Y=SRATIO/X
      IARRAY(IIE,IIL,KBOX)=IROUND(Y*20.)
  680 CONTINUE
  700 CONTINUE
  720 CONTINUE
  740 CONTINUE
      IF(NARA.EQ.0) GO TO 800
C
      DO 780 J=1,KLINES
      IA=0
      DO 760 K=1,KBOXES
      DO 760 I=1,KELEMS
      IA=IA+1
      IARET(IA)=IARRAY(I,J,K)
  760 CONTINUE
      CALL PACK(MELE,IARET,IABUF)
      CALL WRITA(NARA,IAREC,IABUF)
      IAREC=IAREC+1
  780 CONTINUE
```

```
C
  800 CALL EDEST('FINISHED LINE ',IL)
  820 CONTINUE
      IF(NORR.LT.0) GO TO 840
      IDOC(100)=LRC
      CALL VRTIO(IRET,0,1)
  840 IF(NARA.NE.0) CALL CLOSAC(NARA)
      CALL EDEST('*** ALL DONE *** ',0)
      CALL ENKODE('("FAILURE SUMMARY ",7I5)',LBUF,NFAIL)
      CALL VTG(LBUF)
      RETURN
  860 CALL EDEST('TROUBLE WRITING COLUMN(RECORD) ',LRC)
      GO TO 960
  880 CALL EDEST('TROUBLE WRITING HEADER FOR ROW ',NORR)
      GO TO 960
  900 CALL EDEST('INVALID AREA NUMBER ',NARA)
      GO TO 960
  920 CALL EDEST('NO OUTPUT MEDIUM (AREA OR MD) SPECIFIED ***',0)
      GO TO 960
  940 CALL EDEST('IMAGE TOO WIDE - MAXELE IS 660, ESIZ = ',MELE)
  960 CALL ABORT(0)
      RETURN
      END
```

References

MacMillin, L.M., 1978: An improved technique for obtaining clear radiances from cloud contaminated radiances. Mon. Wea. Rev., 11, 1590-1597.

Smith, W. L., 1983: The retrieval of atmospheric profiles from VAS geostationary radiance observations. J. Atmos. Sci., 40, 2025-2035.

Smith, W.L., and H.M. Woolf, 1984: Improved vertical soundings from an amalgamation of polar and geostationary radiance observations. Preprints, AMS Conference on Satellite Meteorology/Remote Sensing and Applications, Clearwater Beach, Florida, 25-29 June 1984. Published by the American Meteorological Society, Boston, Massachusetts.