UW-Madison.
SSEC Publication No.84.02.N1.

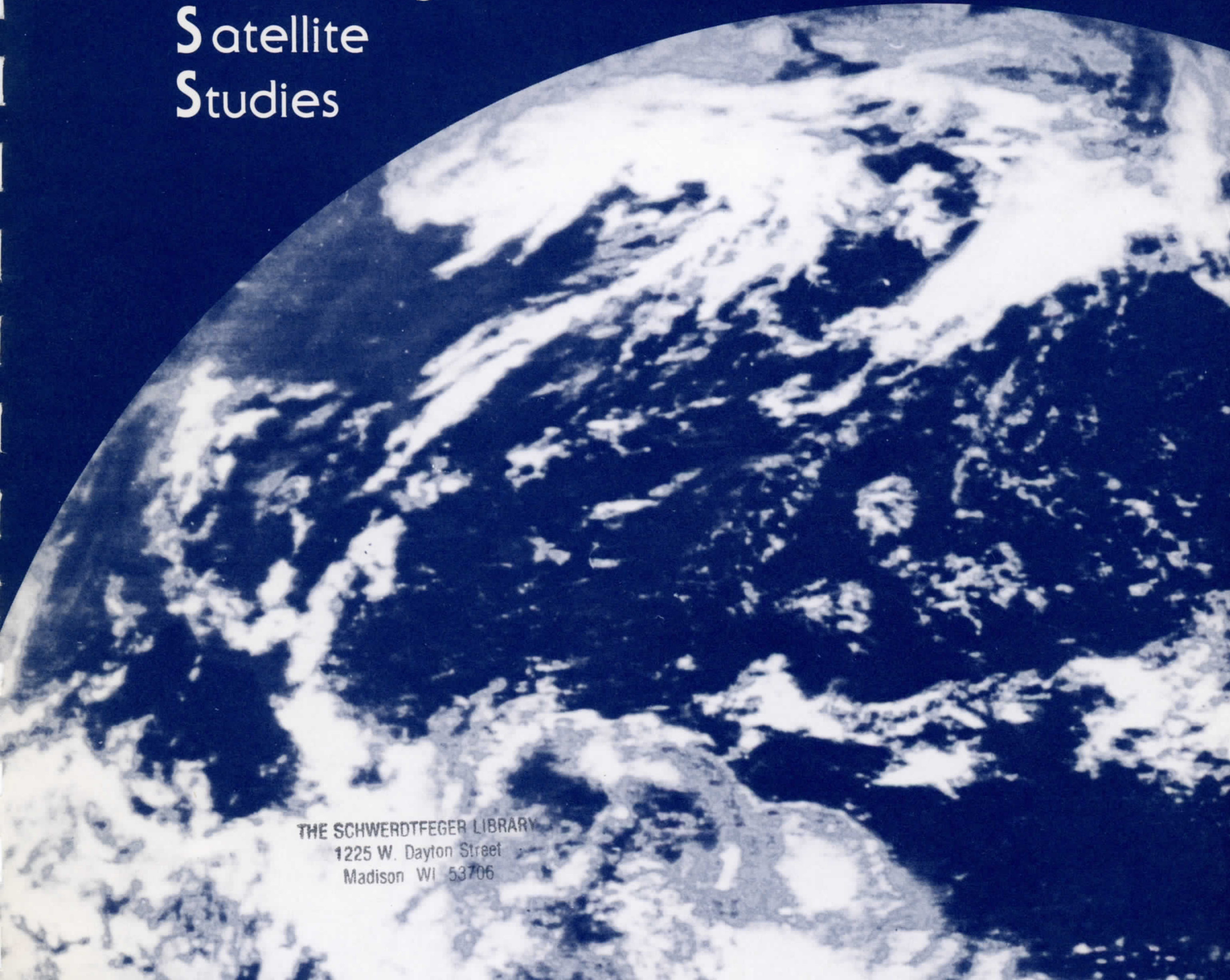
Space Science and Engineering Ce
University of Wisconsin-Madison

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

Space Science and Engineering Center
University of Wisconsin
1225 West Dayton Street
Madison, Wisconsin 53706

February 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. As the software is being developed in a research laboratory, it is constantly undergoing change, in attempts to upgrade and improve it. The programs described herein, with the exception of VTPX, are versions from February 8, 1984. The version of VTPX contained within the text is valid for February 15, 1984. In addition, the flowcharts (level I) contained within each program description represent the software of October 19, 1983 and therefore may show some inconsistencies with the corresponding software descriptions and listings. Finally, flow charts for SRAD, SRET, and VTPX will be forthcoming in the next edition.

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 McIDAS (Man-computer Interactive Data Access System) 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 four 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 those programs deemed optional for VAS retrieval usage.

The final part of the reference contains four appendices. Appendix 1 describes constructs which exist in one type of flowchart contained within this manual. Appendices 2 and 3 contain lists of all program subroutines and functions, including a description of what each subroutine and function does. Finally, Appendix 4 is comprised of VAS software reproductions.

CHAPTER 1

General Retrieval Outline

In this chapter, a 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 noted by a superscript. Since there are a total of 30 commands, there will be 30 corresponding superscripts located at the proper locations within the text. Only minor modifications in the retrieval scheme are required to process a single retrieval. For further details on the retrieval process, see VLHOWVAS and(or) 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 load the digital radiance data for the area into a specified image on the television frame.[2] This will cause the band 8 (window) image corresponding to the sounder area indicated in the DF keyin to be displayed on the TV screen. In addition, it is also possible to plot a satellite-projection map over the image using system program IC.[3] After the radiance data has been loaded and displayed, the sounder area corresponding to the time the user wishes to process 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. As a quick check of the data quality, one can use VDVA, which can be entered to plot 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.)

Up to this point, we have dealt only with the raw VAS radiance data. Retrieval processing using an iterative retrieval scheme 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. One should check to see if these files exist using system programs MDU and IGU;[6] otherwise, they can be created using the same 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 MD file (schema: RSVC; for 1000 mb heights, sea level temperatures and 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) and surface 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 subsequently is used 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 given guess MD file via GSVA.[13] In other words, values from all grids (i.e., at different levels) for a given gridpoint are stored at a certain row, 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 station dewpoint depressions for each station and places the results in the designated surface MD file.[14] Then, SRVA takes this data, calculates a grid of one of the parameters, and places the results in a grid location within a designated 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, one should first clear the graphics screen using system program EG, and then go back and plot the surface data with PLVA. 18 Then, bad 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

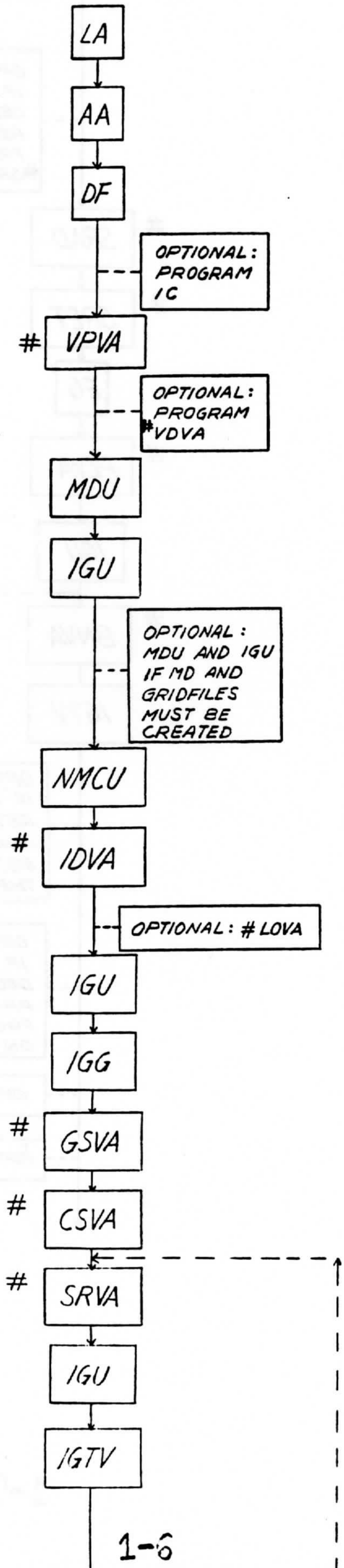
With the completion of the surface analyses, it is now possible to do the actual VAS retrievals. First, SRAD processes (collects, filters and averages) the radiance data for the sounder area pointed at with VPVA. 21 Then, SRET does the actual iterative retrievals, making use of the processed radiance data, as well as both the surface data (optional) and upper air guess data. 22 Note the example of the VASTEXT file as it stands after the area of retrievals has been processed. After the retrievals have been generated, EG should be executed again to clear the graphics screen, after which 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 wherever the user desires the results of BNVA to be placed. 24 Usually, the retrieval parameter grids are stored in a different gridfile than either the first guess or surface gridfiles. Of course, the user should check beforehand to make sure the gridfile exists, and create it if necessary. Then, BNVA 25 is run, with the results being stored in the "scratch" gridfile. 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 26 grids. Bad analyses can be corrected in the same manner as the surface data analyses (EG, 27 PLVA, MDU and XRVA). If some deletions are done, one should repeat the steps from BNVA (step 25) onward to assess how the 28 deletions affect the gridded contours.

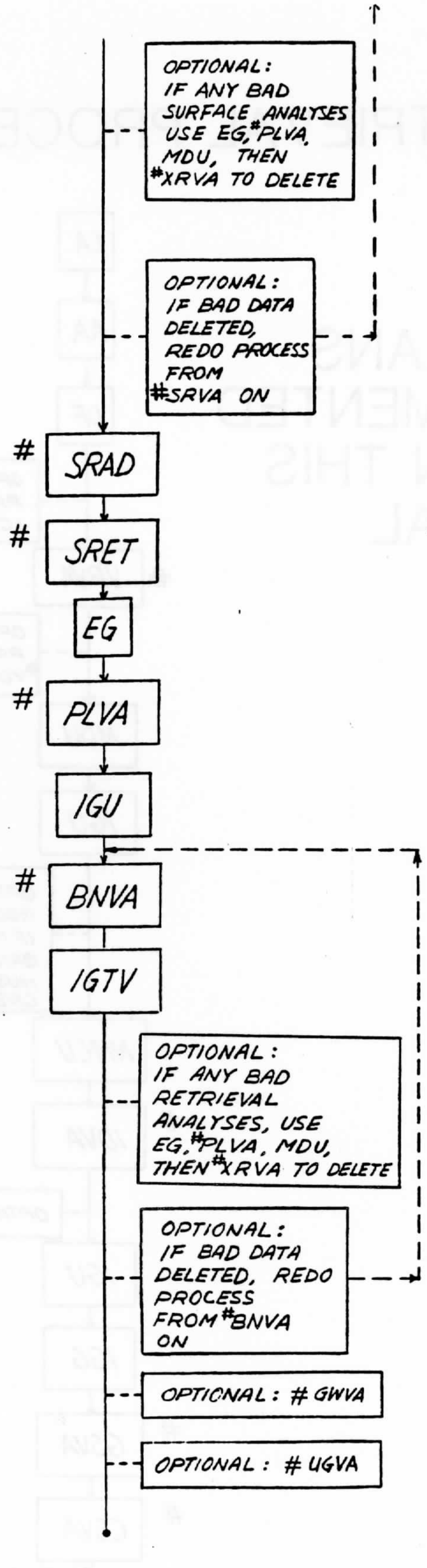
Finally, VAS gradient winds can be produced using GWVA, 29 and if retrievals from a different time period must also be processed, a more up to date (closer in time) first guess can be generated using the just-completed retrievals via program UGVA, 30 and the entire procedure begins anew.

On the pages following the flowchart, a sample of McIDAS keyins for the processing of a sounder 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, #PLVA, MDU, THEN #XRVA TO DELETE
OPTIONAL: IF BAD DATA DELETED, REDO PROCESS FROM #SRVA ON
SRAD
SRET
EG
PLVA
IGU
BNVA
IGTV
OPTIONAL: IF ANY BAD RETRIEVAL ANALYSES, USE EG, #PLVA, MDU, THEN #XRVA TO DELETE
OPTIONAL: IF BAD DATA DELETED, REDO PROCESS FROM #BNVA ON
OPTIONAL: #GWVA
OPTIONAL: #UGVA

Summary of Files Used

A.) sounder area (data at 1248 GMT, 01NOV 1983): 4181

B.) surface data MD file (schema: RSVC; containing Z100, TSL and DD): 4186

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

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

E.) upper air guess gridfile: 4180

F.) surface grid file (containing grids of Z100, TSL and DD): 4181

G.) scratch gridfile for temporary storage of retrieval grids: 4179

Command Sequence

1.) Locate, and move radiance data:

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

then:

AA 1305 4181 ASIZE=ALL

2.) Load digital sounder area into given image frame:

DF 4181 1 EC 35 90 4

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

IC

4.) Point to correct sounder area:

VPVA 4181

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 4185 VRET 1 0 0 83305 "VAS RET. 1248 GMT 01 NOV83

IGU MAKE 4181 "EDITED SFC DATA 12GMT 01NOV83

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

NMCU 0 4180 "LFM GSS FROM 00GMT 01 NOV 83

9.) Initialize VASTEXT file and row 1 of retrieval MD file:

IDVA 4185 1

10.) Check VASTEXT file contents:

LOVA

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

IGU SET 4180

12.) List grids in gridfile 4180:

IGG LIST Note: grid 27 starts sequence of 12-hour forecast grids.

13.) Reformat guess from guess gridfile 4180 into guess MD file 4184, for the 12-hour forecast sequence, starting at grid 27:

GSVA 4180 27 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) 4186:

CSVA 4186 Note: hour of surface data will be 12 GMT (truncation of satellite observation time), and thus, row 12 in MD file 4186 will be used.

15.) Create one grid each of Z100, TSL and DD (data taken from surface MD file 4186) and store in grid positions 1-3 in surface gridfile 4181:

```
SRVA Z100 4181
SRVA TSL 4181
SRVA DD 4181
```

16.) Examine results; first set gridfile to 4181:

```
IGU SET 4181
```

17.) Contour data in gridfile 4181:

```
IGTV 1 20 SAT
IGTV 2 3 SAT
IGTV 3 3 SAT
```

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

```
EG
```

then:

PLVA Z100 MDNR=4186 MDRR=12 for Z100 (with MDNR, MDRR indicating MD file and row numbers for surface data)

then:

PLVA TSL MDNR=4186 MDRR=12 for TSL, and

PLVA DD MDNR=4186 MDRR=12 for DD

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

```
MDU SET 4186
```

then:

```
XRVA Z100
```

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

21.) Process radiance data for area of retrievals:

SRAD GO END=3513 6997 PLT=1 Note: 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 letter "I" by itself); starting location is cursor position.

22.) Process retrievals for the sounder area:

SRET PLT=1

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
83305	124800	16	16	51171	34114	1	29	4181

MDNS	MDRS	MDNG	MDRG	MDNR	MDRR
4186	12	4184	0	4185	1

NGFG	NGFS	ZGRID	TGRID	DGRID
4180	4181	1	2	3

NO.RETRIEVALS=84
CURRENT RETRIEVAL OPTIONS..

TYP	GSS	SPC	SIZ	SFC	ENDL	ENDE	BEGL	BEGE	TER	PLT
0	0	10	5	0	3049	10001	1793	4985	30	1

Note: The above information pertains to the processing of a sounder area sensed in the large detector mode of VAS. For the small detector mode, variables X-RES and Y-RES would always be equal to 8, while variables SPC and SIZ would normally be 20 and 10, respectively.

23.) Plot retrieval data:

always be equal to 8, while variables SFC and SIZ would normally be 20 and 10, respectively.

23.) Plot retrieval data:

EG

then:

PLVA Z 500 MDNR=4185 MDRR=1

24.) Set gridfile before BNVA:

IGU SET 4179

25.) Generate grids of retrieval parameters and store in gridfile 4179:

BNVA Z 500

26.) Select the appropriate grid (#52, 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 52 30 SAT (contour interval of 30 meters)

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

EG

PLVA Z 500 MDNR=4185 MDRR=1

MDU SET 4185

XRVA Z 500

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

29.) Generate VAS gradient winds (optional):

GWVA 52 500

30.) Update upper air first guess (i.e., grids 27-48 in gridfile

4180) if more retrievals are to be made at a different time, and then begin again at step 1 to process the next sounder area:

UGVA 27 4180 Note: the updated grids will be added onto the end of gridfile 4180.

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 system level or assembly language subroutines in most cases, which is sufficient 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.

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

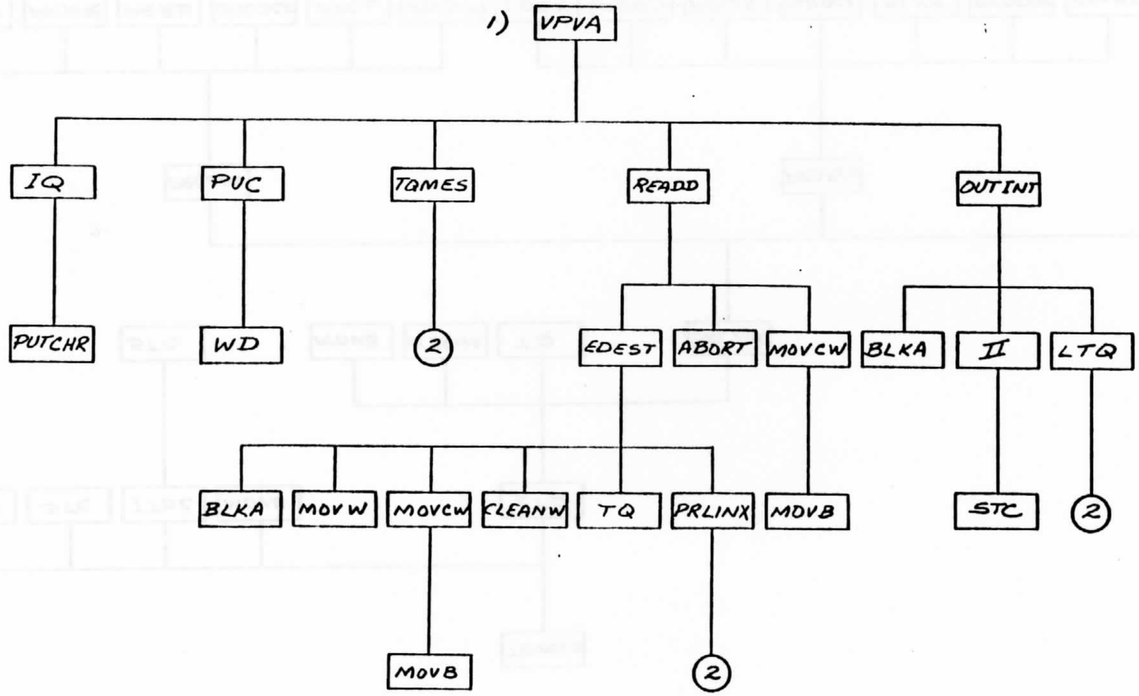
16. BNVA

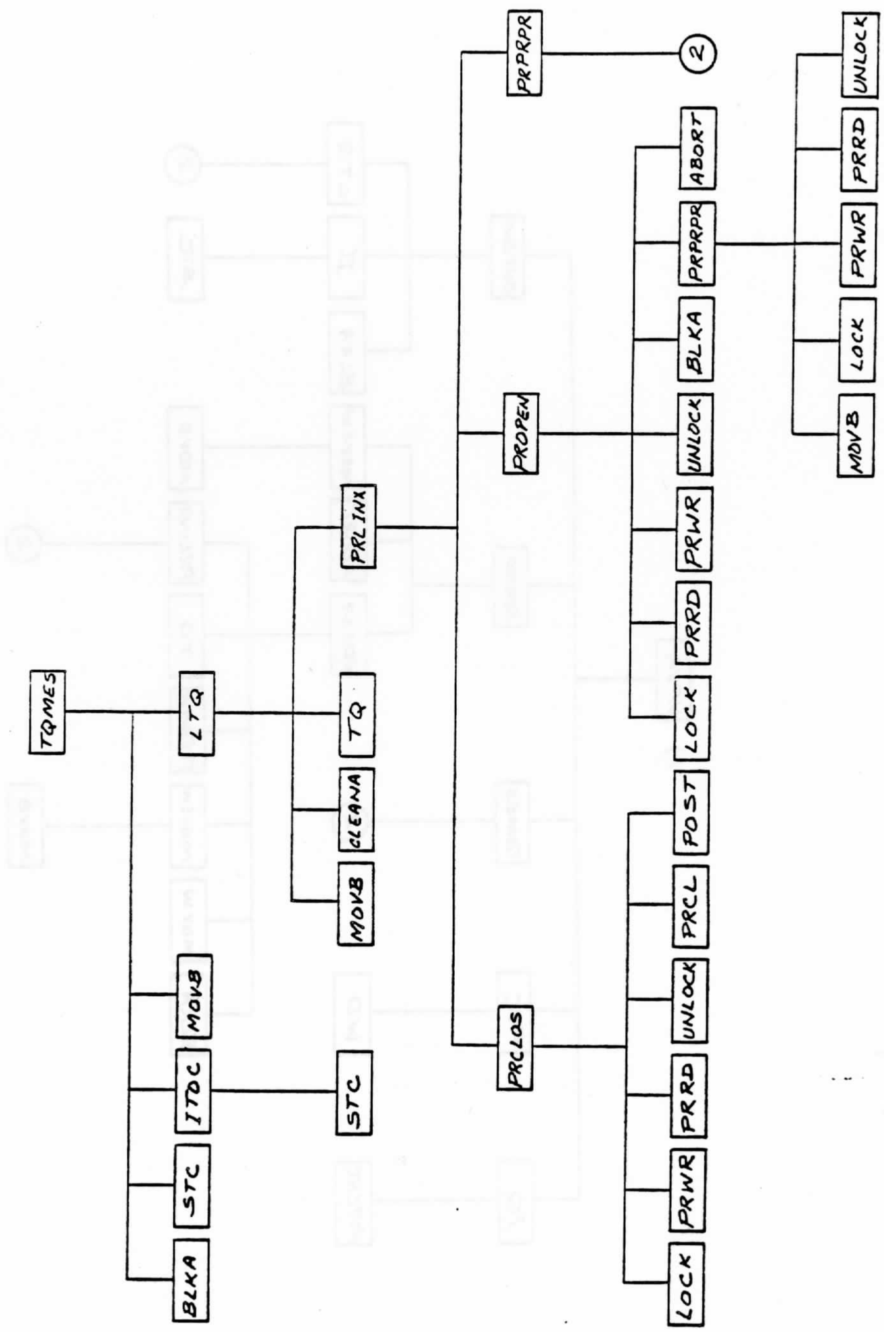
17. GWVA

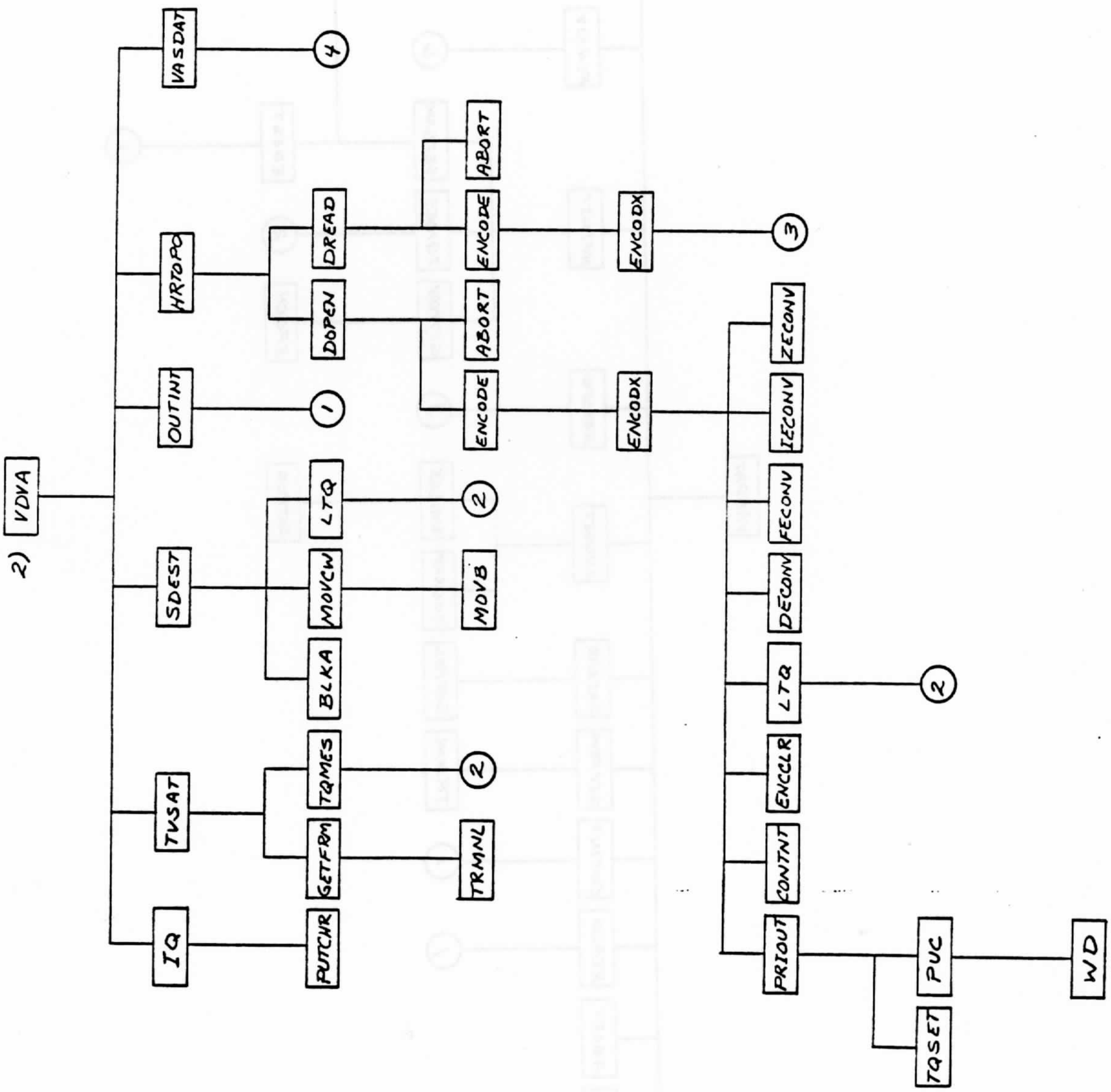
18. ESVA

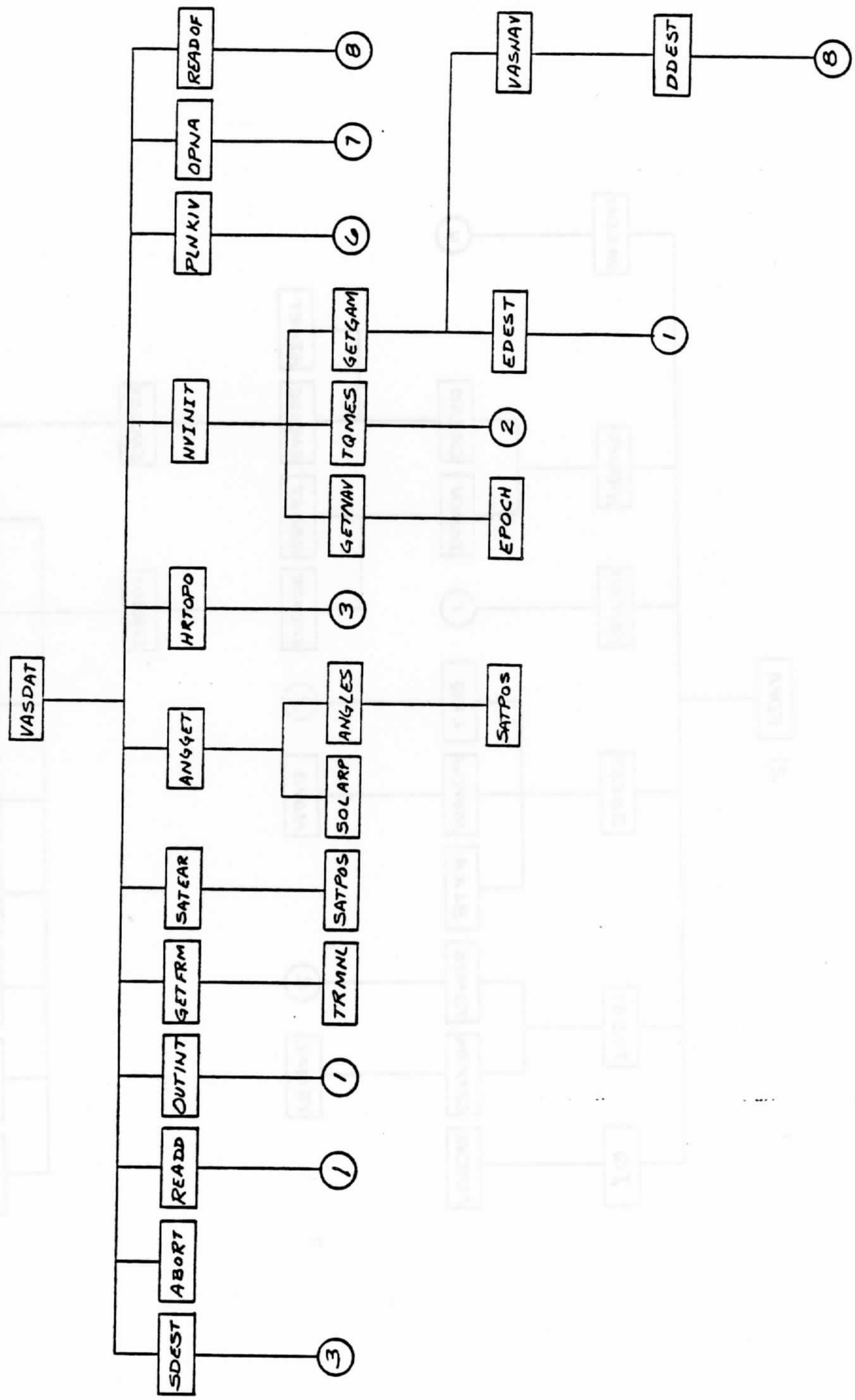
19. VTPX

1)

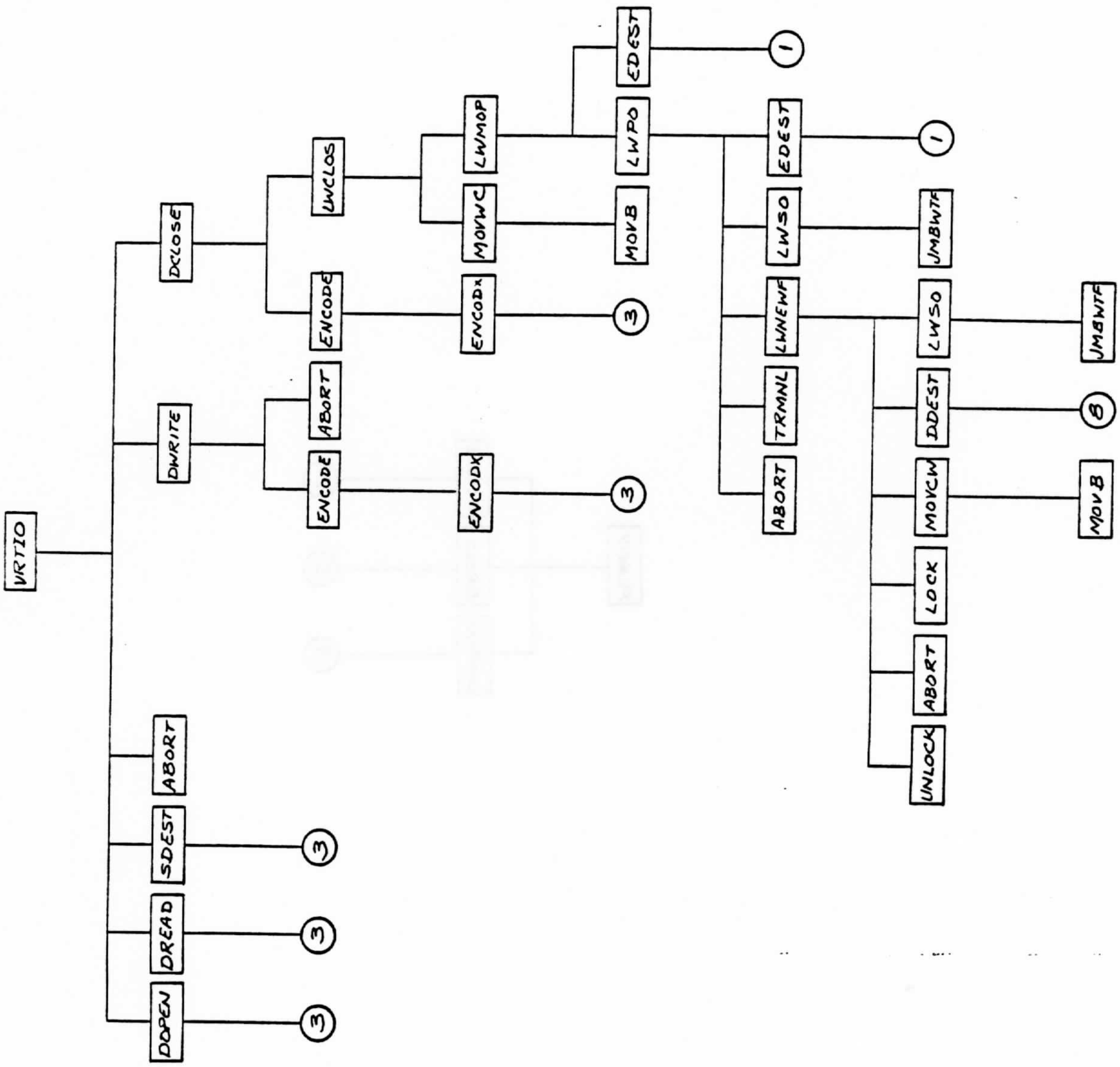

VPVA
IQ
PUC
TQMES
READD
OUTINT
PUTCHR
WD
2
EDEST
ABORT
MOVCW
BLKA
II
LTQ
BLKA
MOVW
MOVCW
CLEANW
TQ
PRLINX
MOVB
STC
2
MOVB
2

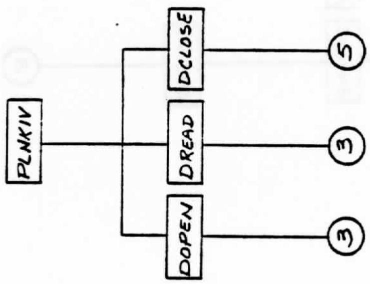

TQMES
BLKA
STC
ITDC
MOVB
LTQ
STC
MOVB
CLEANA
TQ
PRLINX
PRCLOS
LOCK
PRWR
PRRD
UNLOCK
PRCL
POST
PROPEN
LOCK
PRRD
PRWR
UNLOCK
BLKA
PRPRPR
ABORT
MOVB
LOCK
PRWR
PRRD
UNLOCK
PRPRPR
2

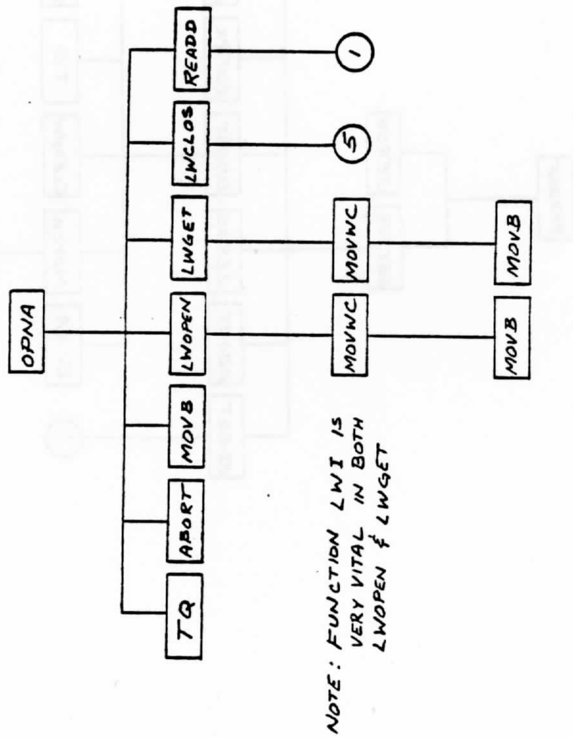

2)
VDYA
IQ
TVSAT
SDEST
OUTINT
HRTOPO
VASDAT
PUTCHR
GETFRM
TQMES
TRMNL
BLKA
MOVCW
LTQ
MOVB
DOPEN
DREAD
ENCODE
ABORT
ENCODX
PRIOUT
CONTNT
ENCCLR
DECONV
FECONV
IECONV
ZECONV
TQSET
PUC
WD

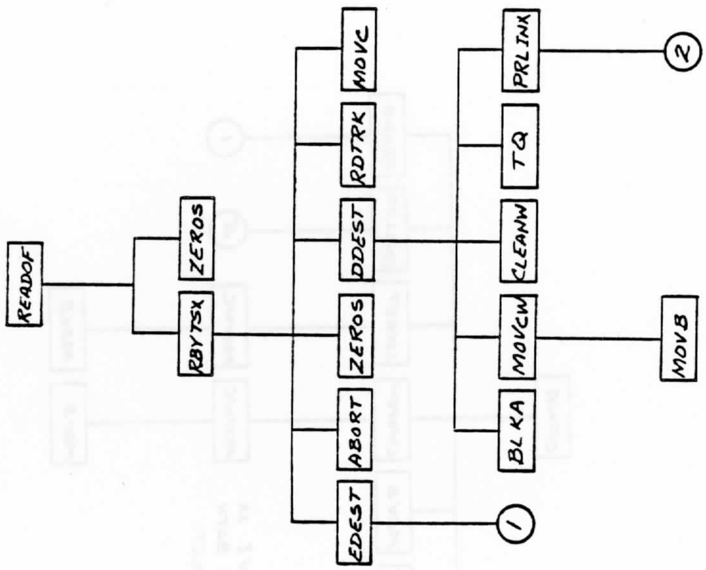


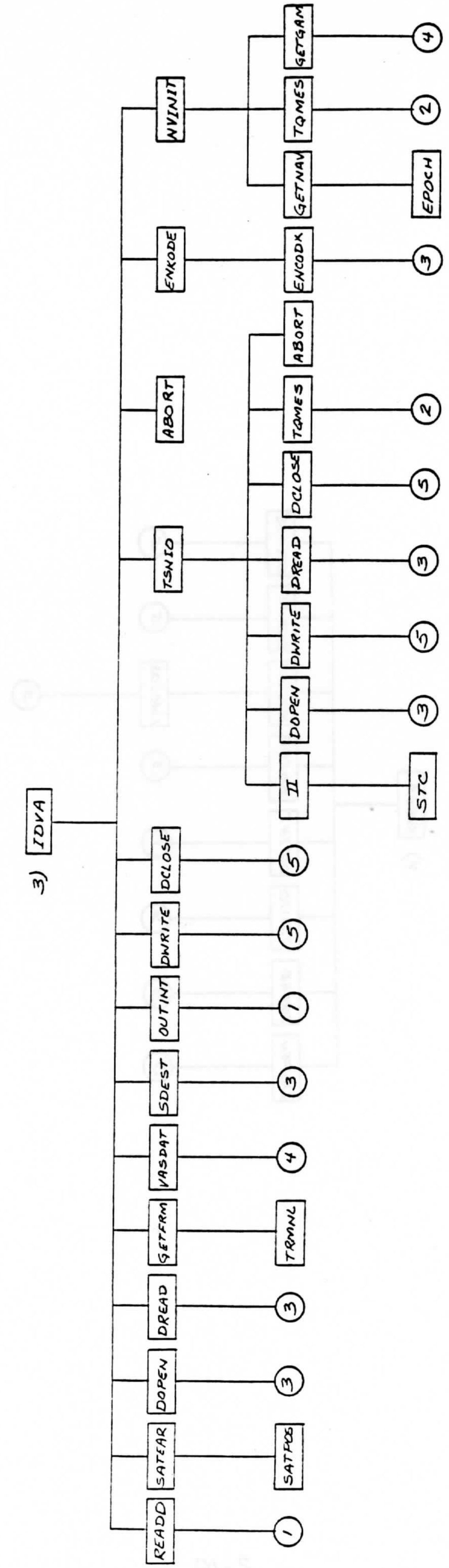
VASDAT
SDEST
3
ABORT
READD
1
OUTINT
1
GETFRM
TRMNL
SATEAR
SATPOS
ANGGET
SOLARP
ANGLES
SATPOS
HRTOPO
3
NVINIT
GETNAV
EPOCH
TQMES
2
GETGAM
EDEST
1
VASNAV
DDEST
8
PLNKIV
6
OPNJA
7
READOF
8


VRTIO
DOPEN
DREAD
SDEST
ABORT
DWRITE
ENCODE
ABORT
ENCODX
DCLOSE
ENCODE
ENCODx
LWCLOS
MOVWC
LWMOP
MOVB
LWPO
EDEST
EDEST
LWSO
JMBWTF
LWNEWF
TRMNL
ABORT
LWSO
JMBWTF
DDEST
MOVCW
MOVB
LOCK
ABORT
UNLOCK
3
3
3
3
3
1
1
8

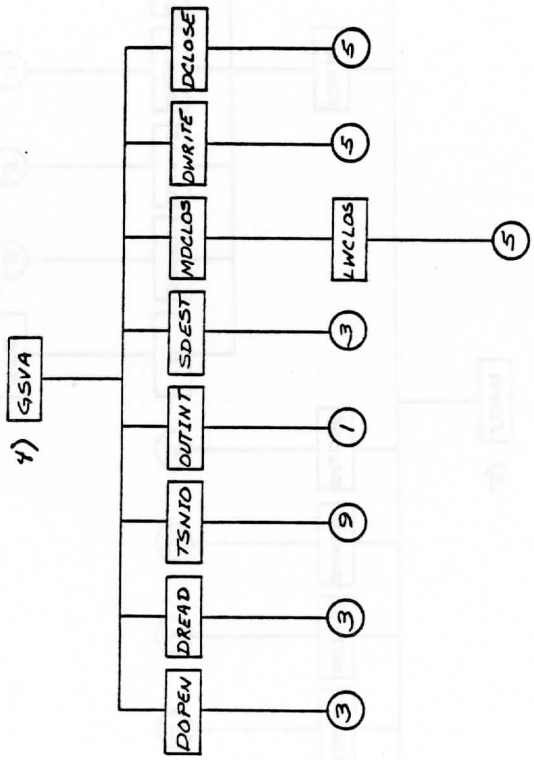

PLNKIV
DOPEN
DREAD
DCLOSE
3
3
5


OPNA
TQ
ABORT
MOVB
LWOPEN
LWGET
LWCLOS
READD
5
1
MOVWC
MOVWC
MOVB
MOVB
NOTE: FUNCTION LWI IS VERY VITAL IN BOTH LWOPEN & LWGET

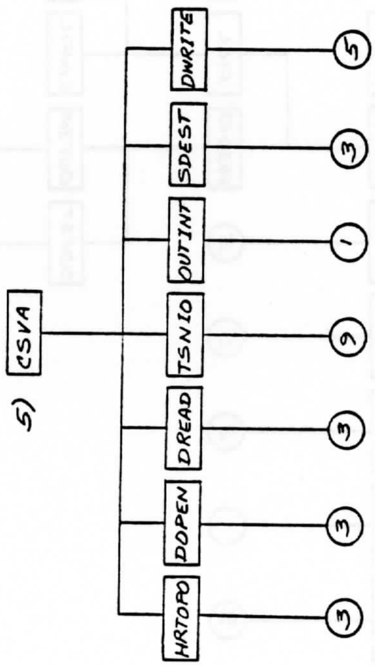

READOF
RBYTSK
ZEROS
EDEST
ABORT
ZEROS
DDEST
RDTRK
MOVC
BLKA
MOVCW
CLEANW
TQ
PRLINK
MOVB
1
2

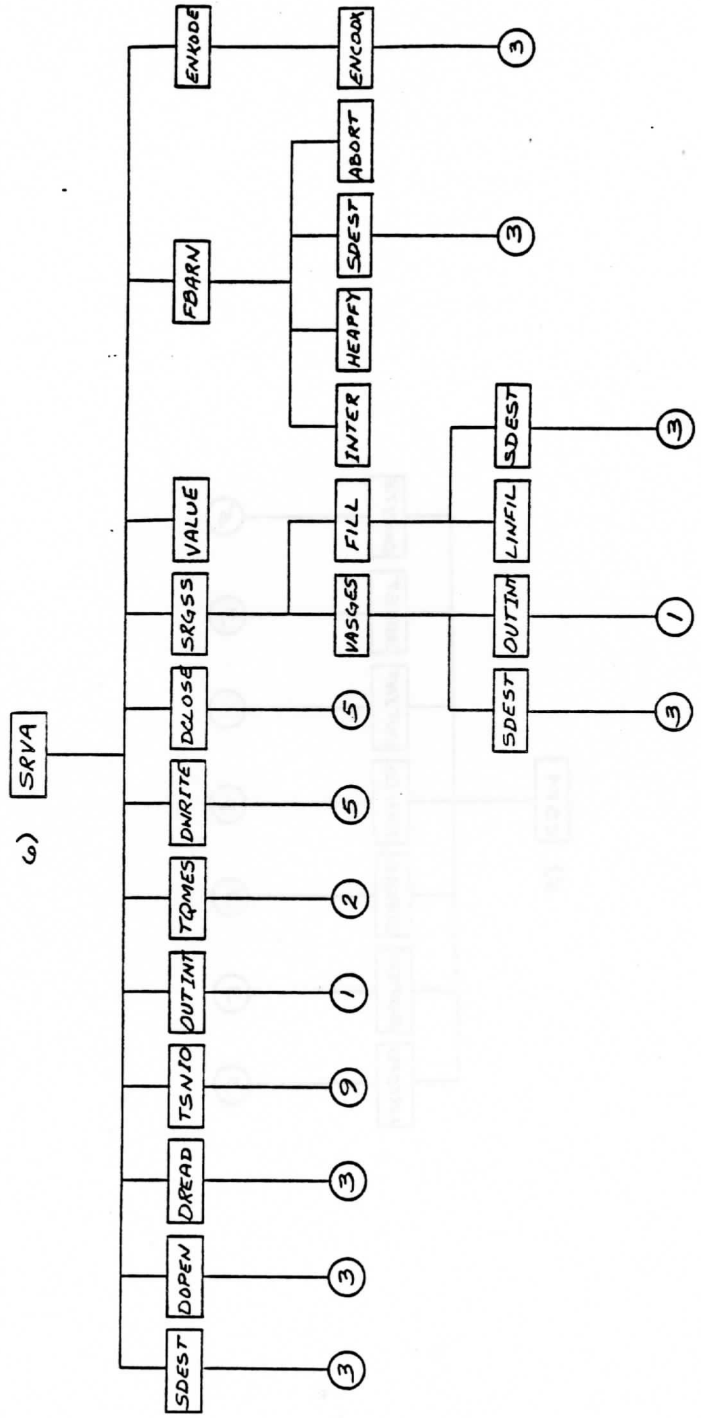


3) IDVA
READD
SATEAR
SATPOS
DOPEN
DREAD
GETFRM
TRMNL
VASDAT
SDEST
OUTINT
DWRITE
DCLOSE
TSNIO
II
STC
DOPEN
DWRITE
DREAD
DCLOSE
TQMES
ABORT
ABORT
ENKODE
ENCODX
NVINIT
GETNAV
EPOCH
TQMES
GETGAM
1
3
3
4
3
1
5
5
3
5
3
5
2
3
2
4

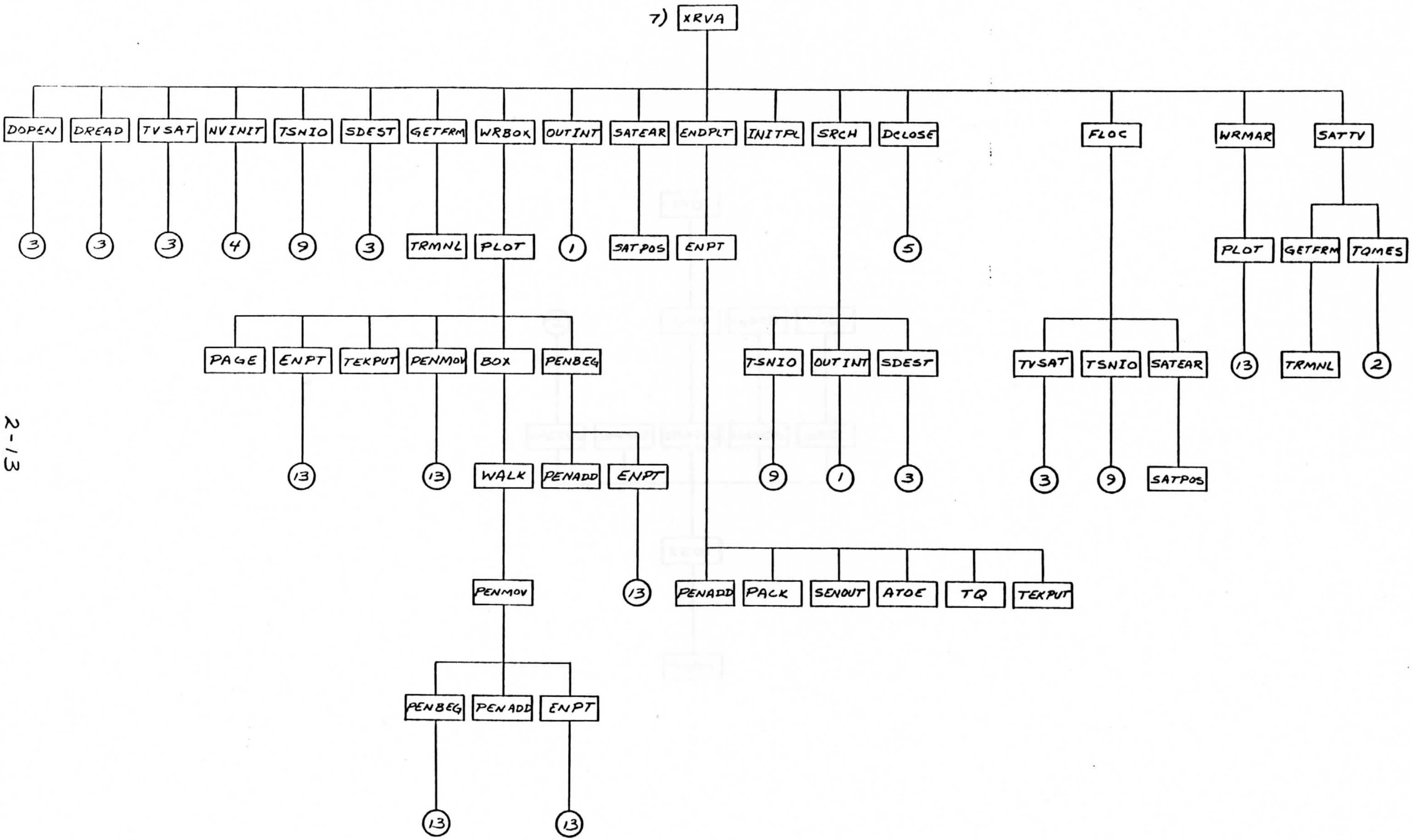

4)
GSVA
DOPEN
DREAD
TSNIO
OUTINT
SDEST
MDCLOS
LWCLOS
DWRITE
DCLOSE
3
3
9
1
3
5
5
5

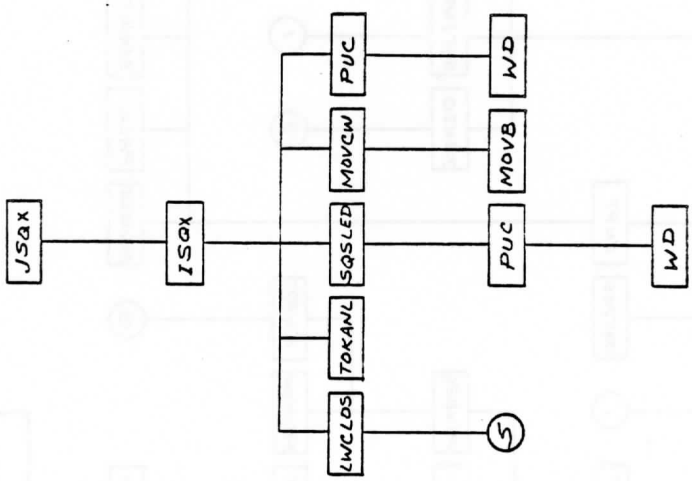
5)



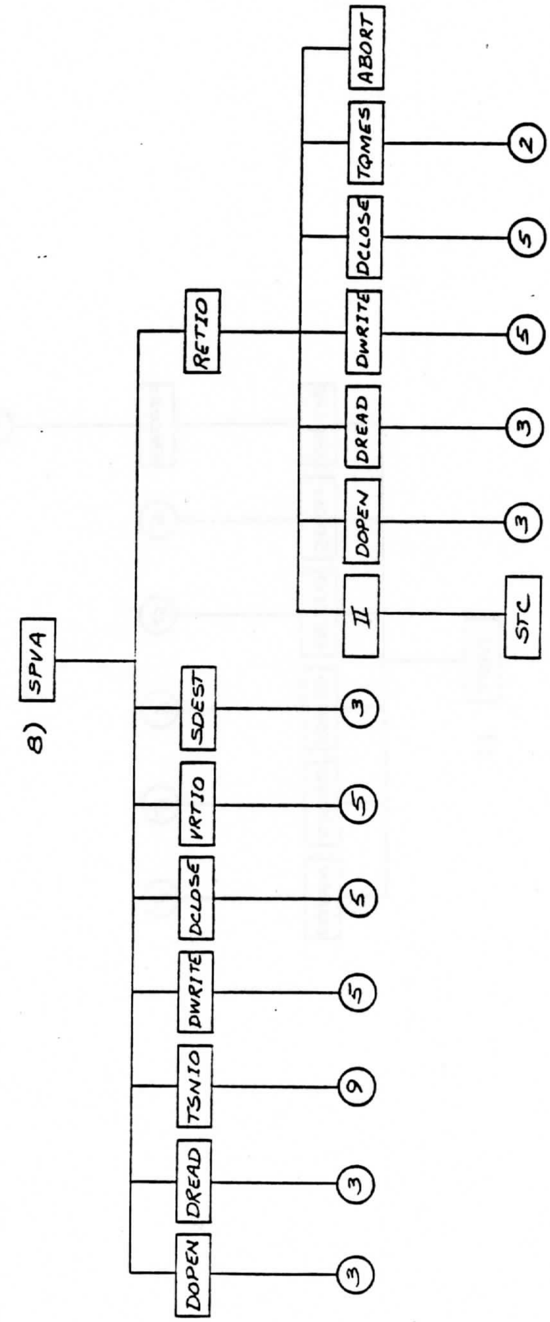

a)
SRVA
SDEST
DOPEN
DREAD
TSNIO
OUTINT
TQMES
DWRITE
DCLOSE
SRGSS
VALUE
FBARN
ENKODE
VASGES
FILL
INTER
HEAPFY
SDEST
ABORT
ENCODE
SDEST
OUTINT
LINFIL
SDEST
3
3
3
9
1
2
5
5
3
1
3
3
3

7)

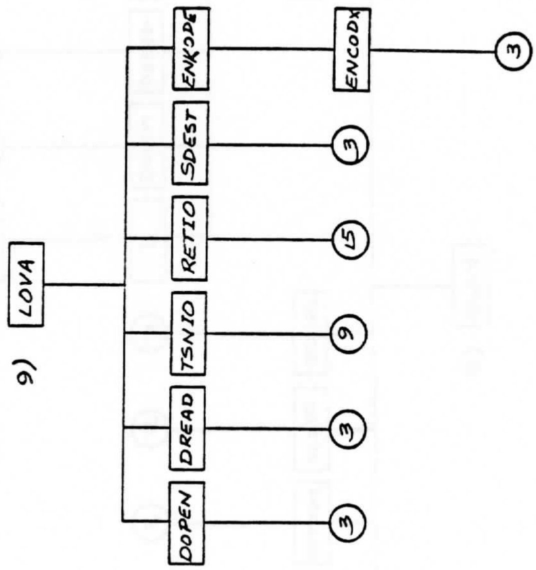

XRVA
DOPEN
DREAD
TVSAT
NVINIT
TSNIO
SDEST
GETFRM
WRBOX
OUTINT
SATEAR
ENDPLT
INITPL
SRCH
DCLOSE
FLOC
WRMAR
SATTV
TRMNL
PLOT
SATPOS
ENPT
GETFRM
TQMES
PAGE
ENPT
TEKPUT
PENMOV
BOX
PENBEG
TSNIO
OUTINT
SDEST
TVSAT
TSNIO
SATEAR
TRMNL
WALK
PENADD
ENPT
SATPOS
PENMOV
PENADD
PACK
SENOUT
ATOE
TQ
TEKPUT
PENBEG
PENADD
ENPT


JSQX
ISQX
LWCLOS
TOKANL
SQSLED
MOVCW
PUC
5
MOVB
WD
PUC
WD

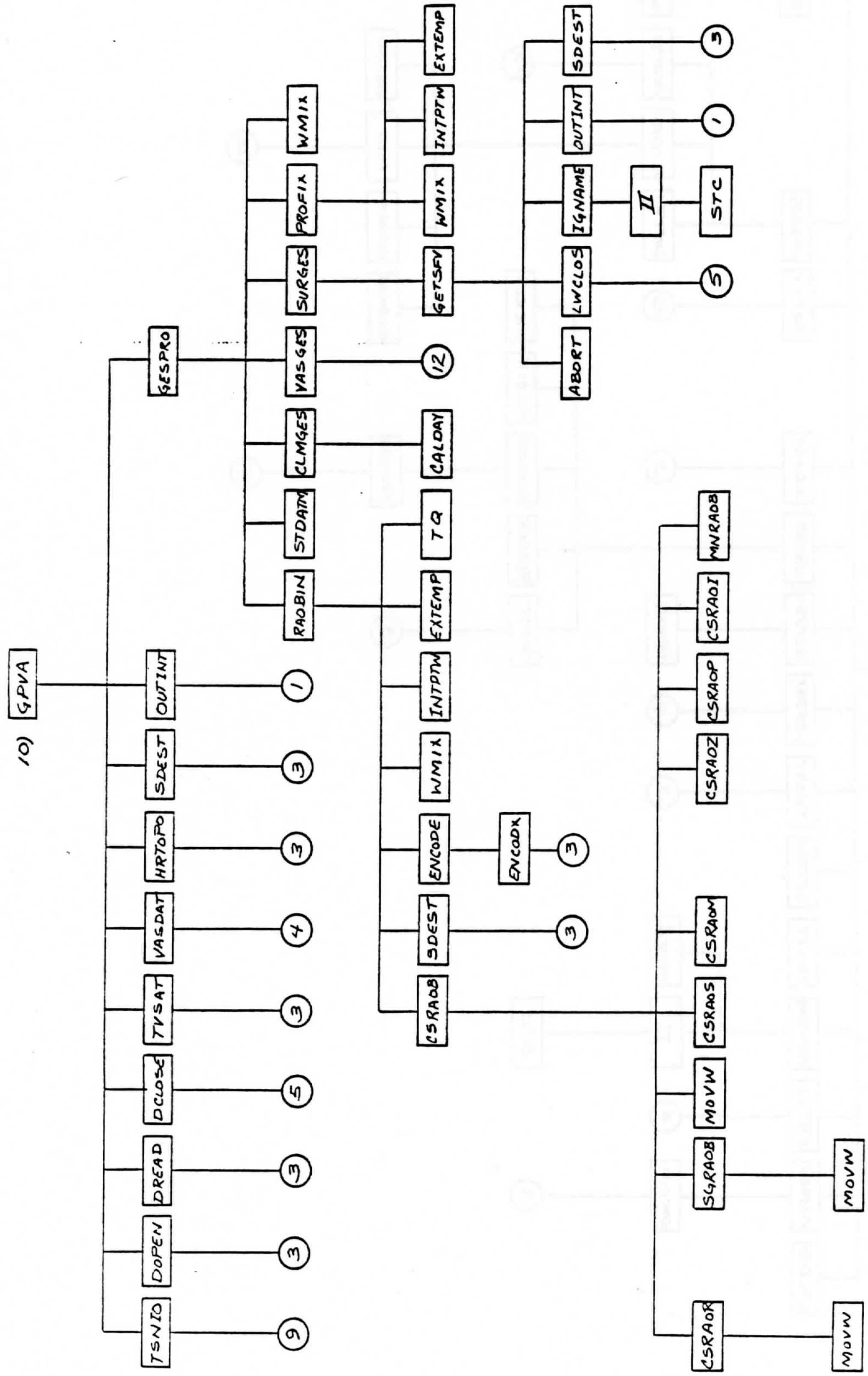
8)


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

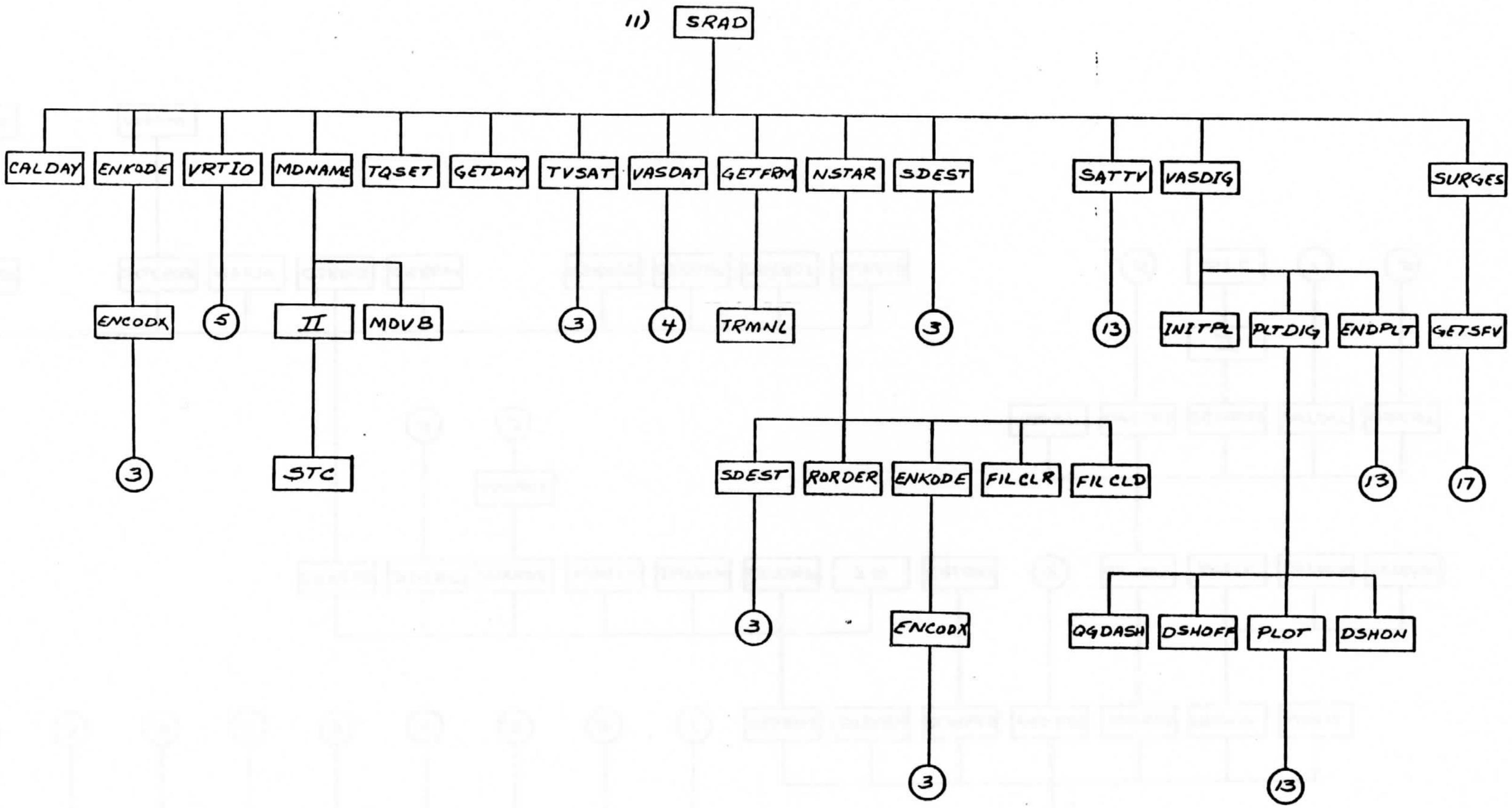
9)



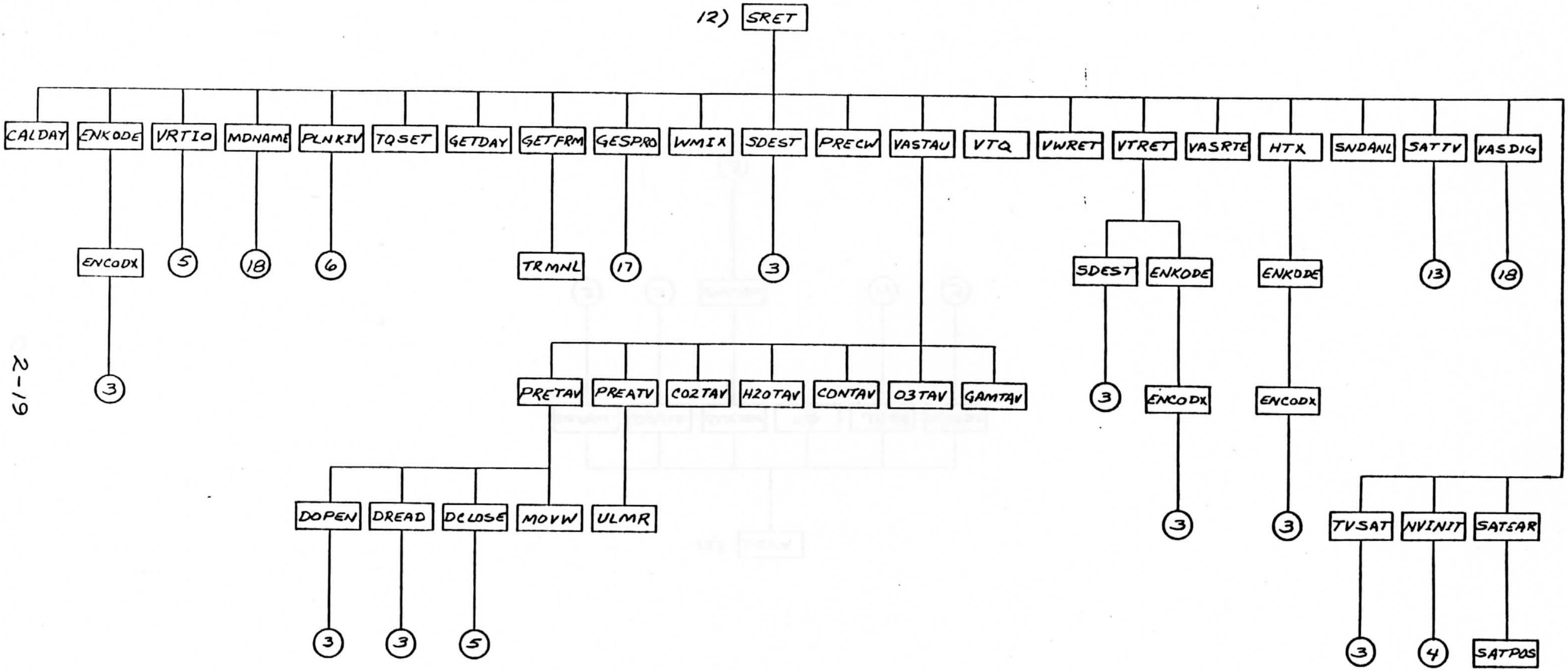
10)

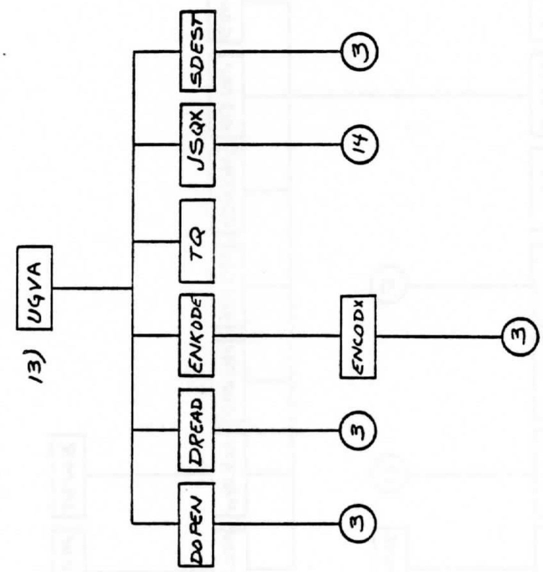

GPVA
TSNIO
DOPEN
DREAD
DCLOSE
TVSAT
VASDAT
HRTOPO
SDEST
OUTINT
GESPRO
RAOBIN
STDATM
CLMGES
VASGES
SURGES
PROFIX
WMIX
EXTEMP
INTPTM
TQ
CALDAY
GETSFV
ENCODE
ENCODX
CSRAOB
CSRAOR
SGRAOB
MOVW
CSRAOS
CSRAOM
CSRAOZ
CSRAOP
CSRAOI
MNRAOB
ABORT
LWCLOS
IGNAME
II
STC

11)

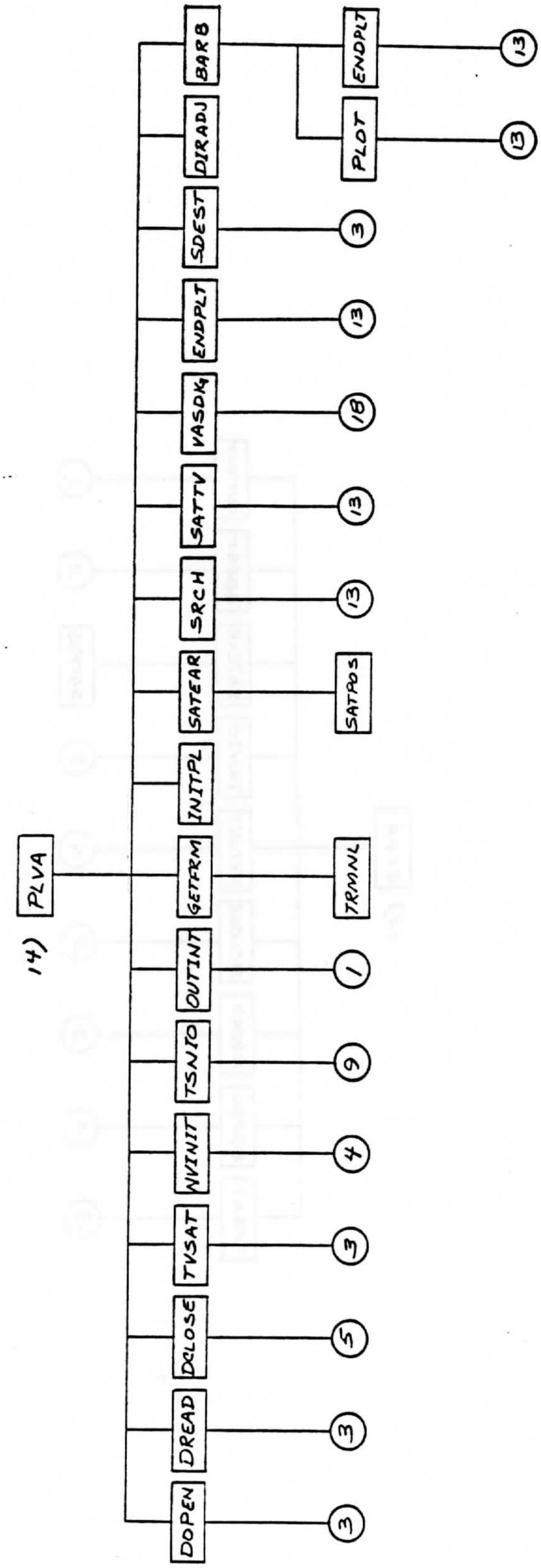

SRAD
CALDAY
ENKODE
VRTIO
MDNAME
TQSET
GETDAY
TVSAT
VASDAT
GETFRM
NSTAR
SDEST
SATTV
VASDIG
SURGES
ENCODX
5
II
MDVB
3
4
TRMNL
3
13
INITPL
PLTDIG
ENDPLT
GETSFV
3
STC
SDEST
RORDER
ENKODE
FILCLR
FILCLD
13
17
3
ENCODX
QGDASH
DSHOFF
PLOT
DSHON
3
13

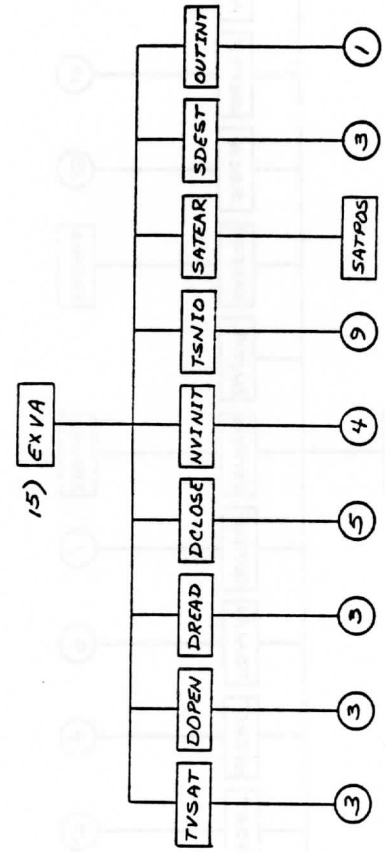
12)


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

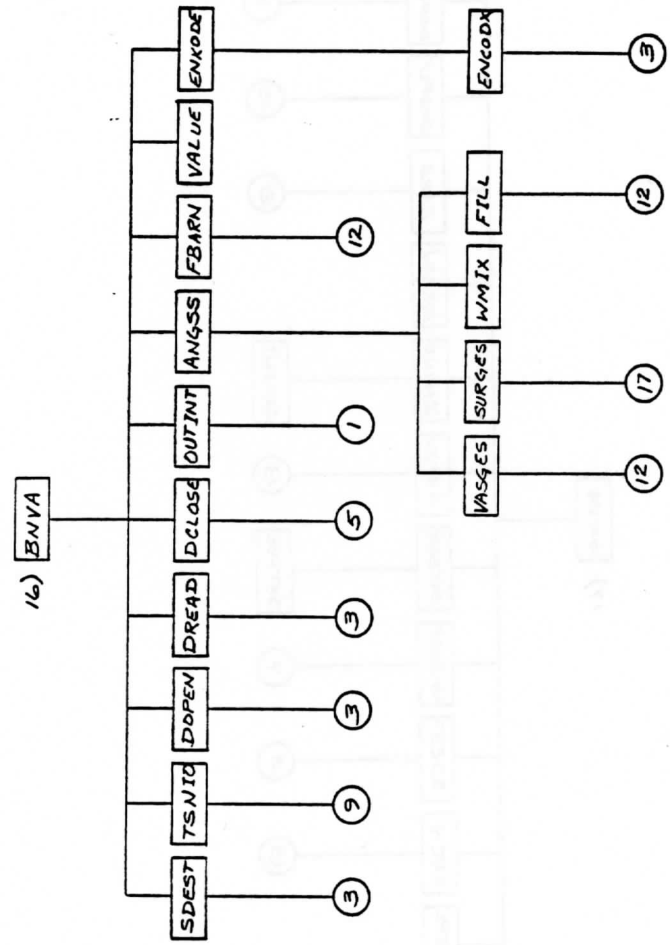

13)
UGVA
DOPEN
DREAD
ENKODE
TQ
JSQX
SDEST
ENCODX
3
3
3
14
3
3

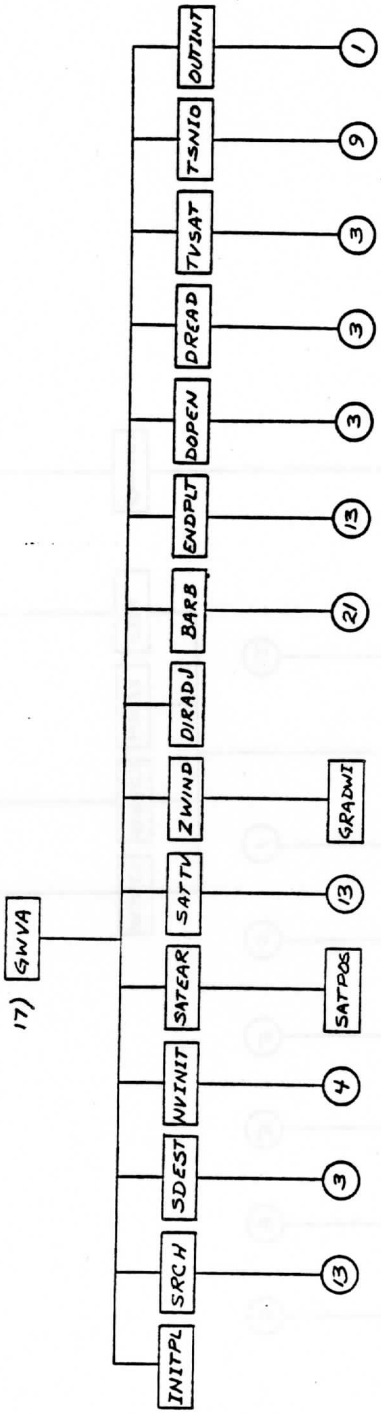
14)

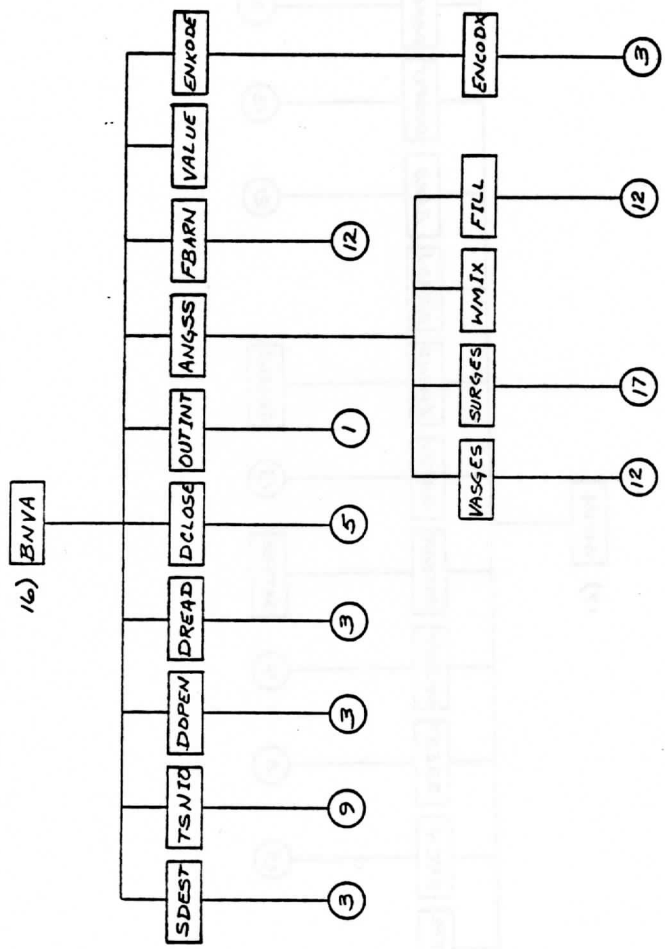

PLVA
DOPEN
DREAD
DCLOSE
TVSAT
NVINIT
TSNIO
OUTINT
GETFRM
TRMNL
INITPL
SATEAR
SATPOS
SRCH
SATTV
VASDK4
ENDPLT
SDEST
DIRADJ
BARB
PLOT
ENDPLT
3
3
5
3
4
9
1
13
13
18
13
3
13
13


15)
EXVA
TVSAT
DOPEN
DREAD
DCLOSE
NVINIT
TSNIO
SATEAR
SDEST
OUTINT
3
3
3
5
4
9
SATPOS
3
1

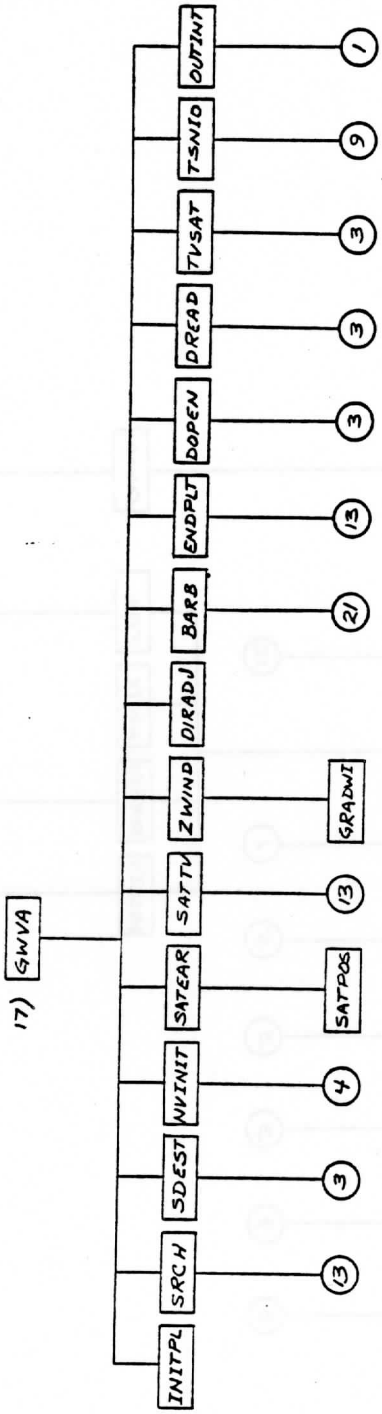
16)


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


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

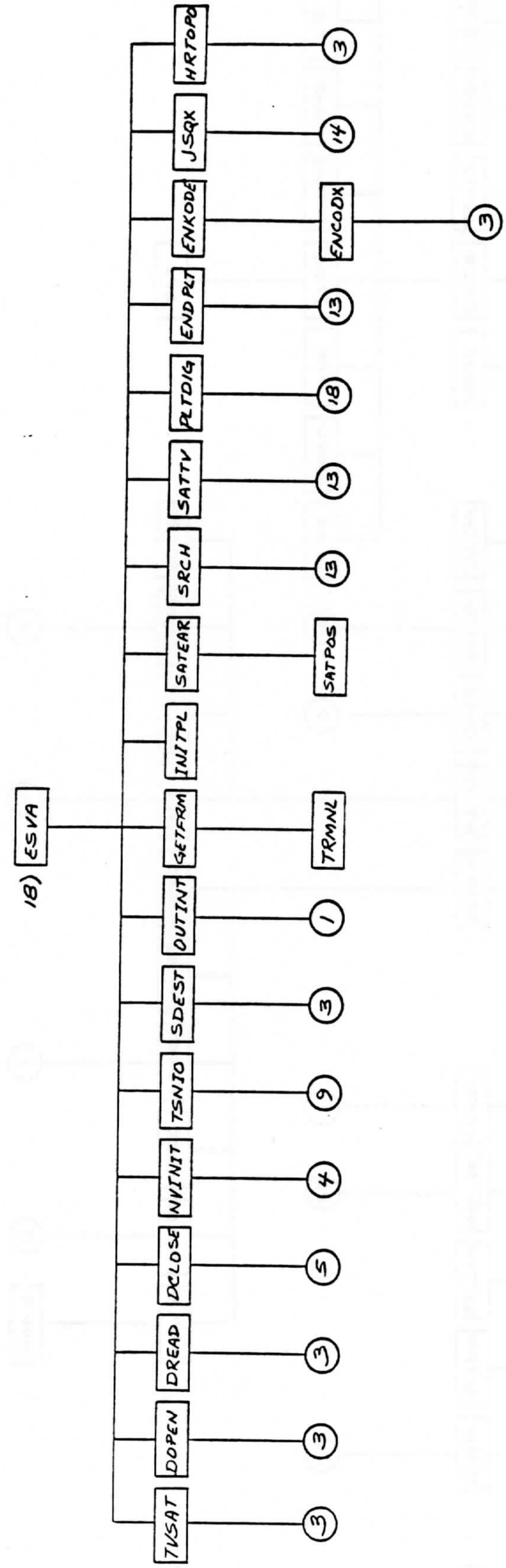

16)
BNVA
SDEST
TSNIO
DOPEN
DREAD
DCLOSE
OUTINT
ANGSS
FBARN
VALUE
ENKODE
VASGES
SURGES
WMIX
FILL
ENCODX
3
9
3
3
5
1
12
12
17
12
3

17)

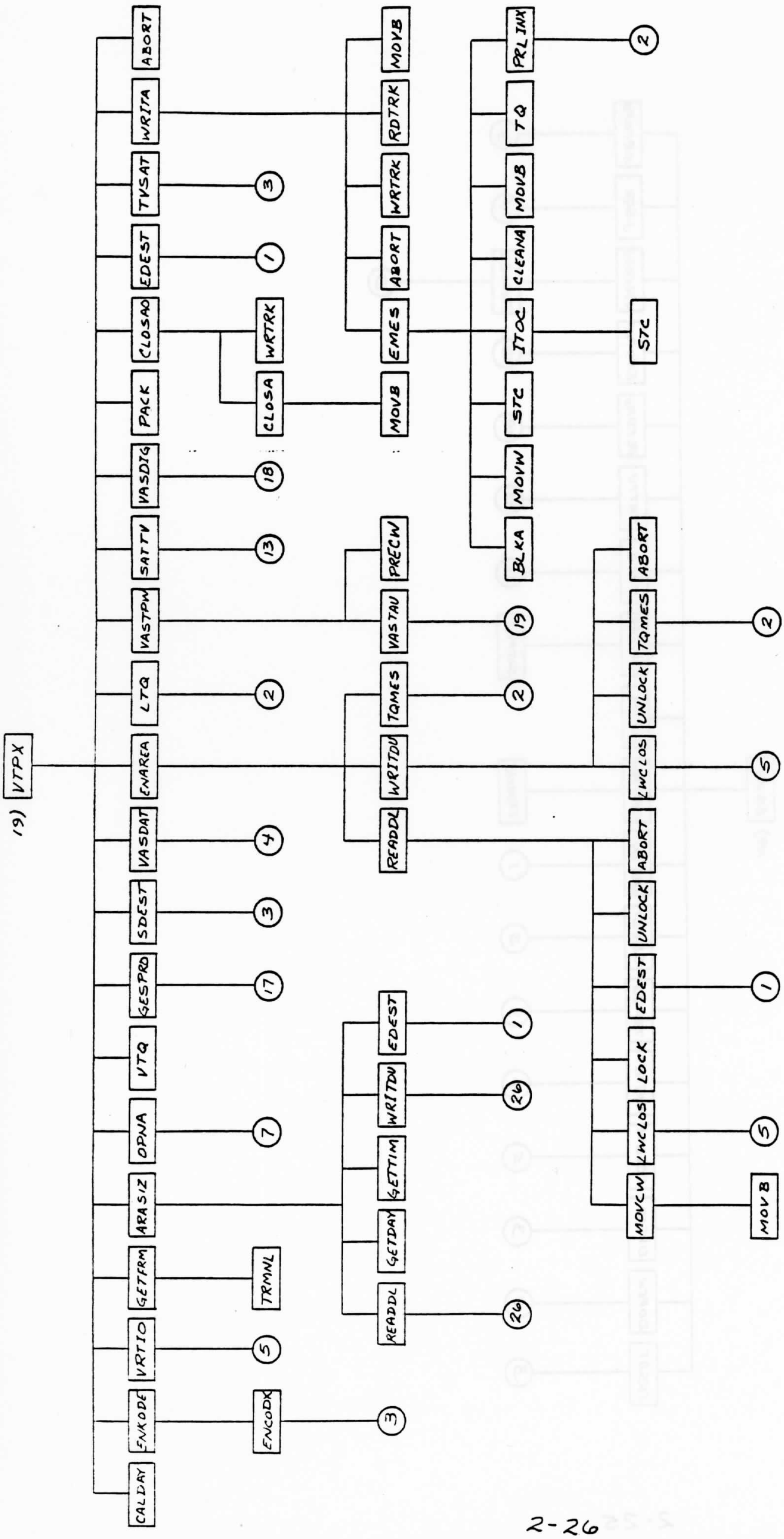


GWVA
INITPL
SRCH
13
SDEST
3
NVINIT
4
SATEAR
SATPOS
SATTV
13
ZWIND
GRADWI
DIRADJ
BARB
21
ENDPLT
13
DOPEN
3
DREAD
3
TVSAT
3
TSNIO
9
OUTINT
1

18)


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

19)


VTPX
CALDAY
ENKODE
ENCODX
3
VRTIO
5
GETFRM
TRMNL
ARASIZ
READDL
26
GETDAY
GETTIM
WRITDU
26
EDEST
1
OPNA
7
VTQ
GESPRD
17
SDEST
3
VASDAT
4
ENAREA
READDL
ABORT
UNLOCK
EDEST
1
LOCK
LWCLOS
5
MOVCW
MOVB
WRITDU
LWCLOS
5
UNLOCK
TQMES
2
ABORT
TQMES
2
LTQ
2
VASTPW
VASTAU
19
PRECW
SATTV
13
VASDIG
18
PACK
CLOSAO
CLOSA
MOVB
WRTRK
EDEST
1
TVSAT
3
WRITA
EMES
BLKA
MOVW
STC
ITOC
STC
CLEANA
MOVB
TQ
PRLINX
2
ABORT
WRTRK
RDTRK
MOVB
ABORT

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. Some of the flowchart constructs used are described in Appendix I. 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 IV.

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

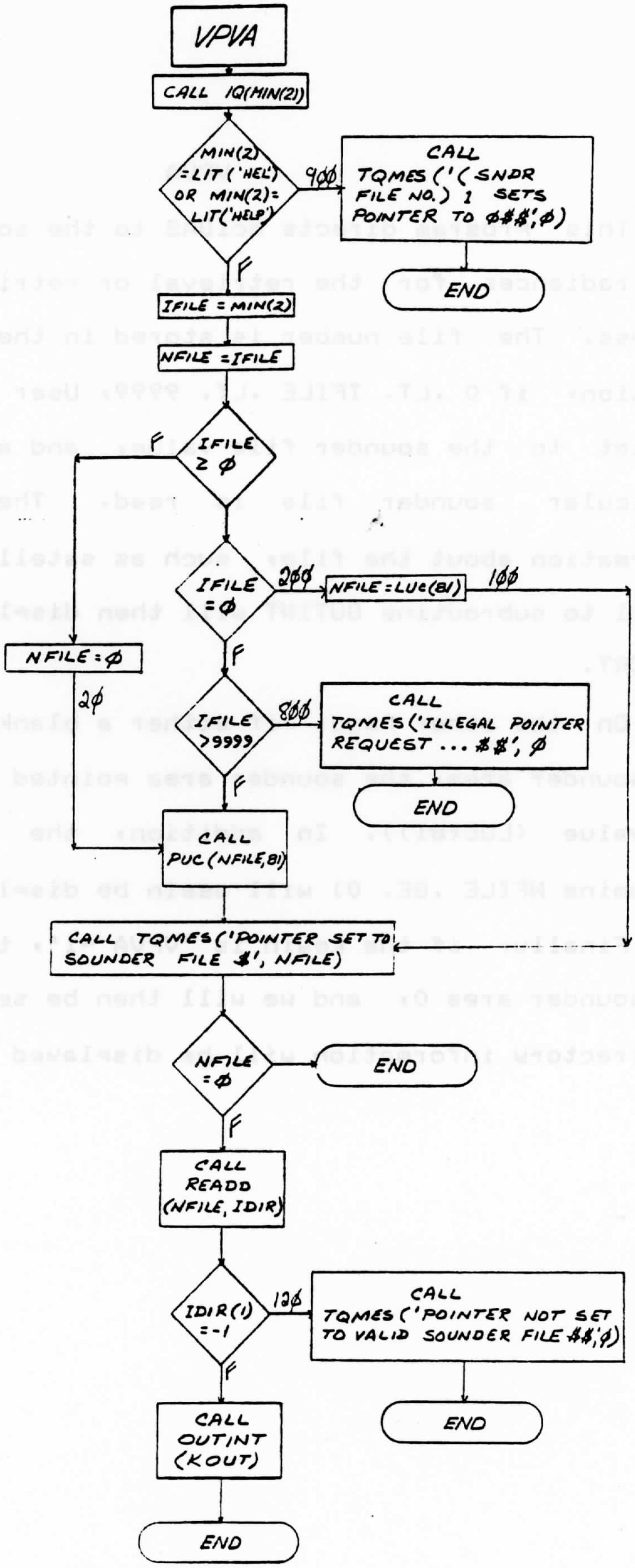
"I". This list should correspond exactly to the subroutines included in the program's modular flowchart in Chapter 2.

VPVA

This program directs McIDAS to the sounder area containing VAS radiances for the retrieval or retrievals one wishes to process. The file number is stored in the variable IFILE. In addition, if 0 .LT. IFILE .LT. 9999, User Common (UC) element 81 is set to the sounder file 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 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 (LUC(81)). In addition, the directory information (assuming NFILE .GE. 0) will again be displayed.

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


VPVA
CALL IQ(MIN(2))
MIN(2) =LIT('HEL') OR MIN(2)= LIT('HELP')
900
CALL TQMES('(SNDR FILE NO.)] SETS POINTER TO Ø$$',Ø)
END
F
IFILE = MIN(2)
NFILE = IFILE
IFILE ≥ Ø
F
NFILE = Ø
2Ø
IFILE = Ø
2ØØ
NFILE = LUC(81)
1ØØ
F
IFILE >9999
8ØØ
CALL TQMES('ILLEGAL POINTER REQUEST ... $$', Ø
END
F
CALL PUC(NFILE,81)
CALL TQMES('POINTER SET TO SOUNDER FILE $', NFILE)
NFILE = Ø
END
F
CALL READD (NFILE, IDIR)
IDIR(1) = -1
12Ø
CALL TQMES('POINTER NOT SET TO VALID SOUNDER FILE $$',Ø)
END
F
CALL OUTINT (KOUT)
END

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 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.

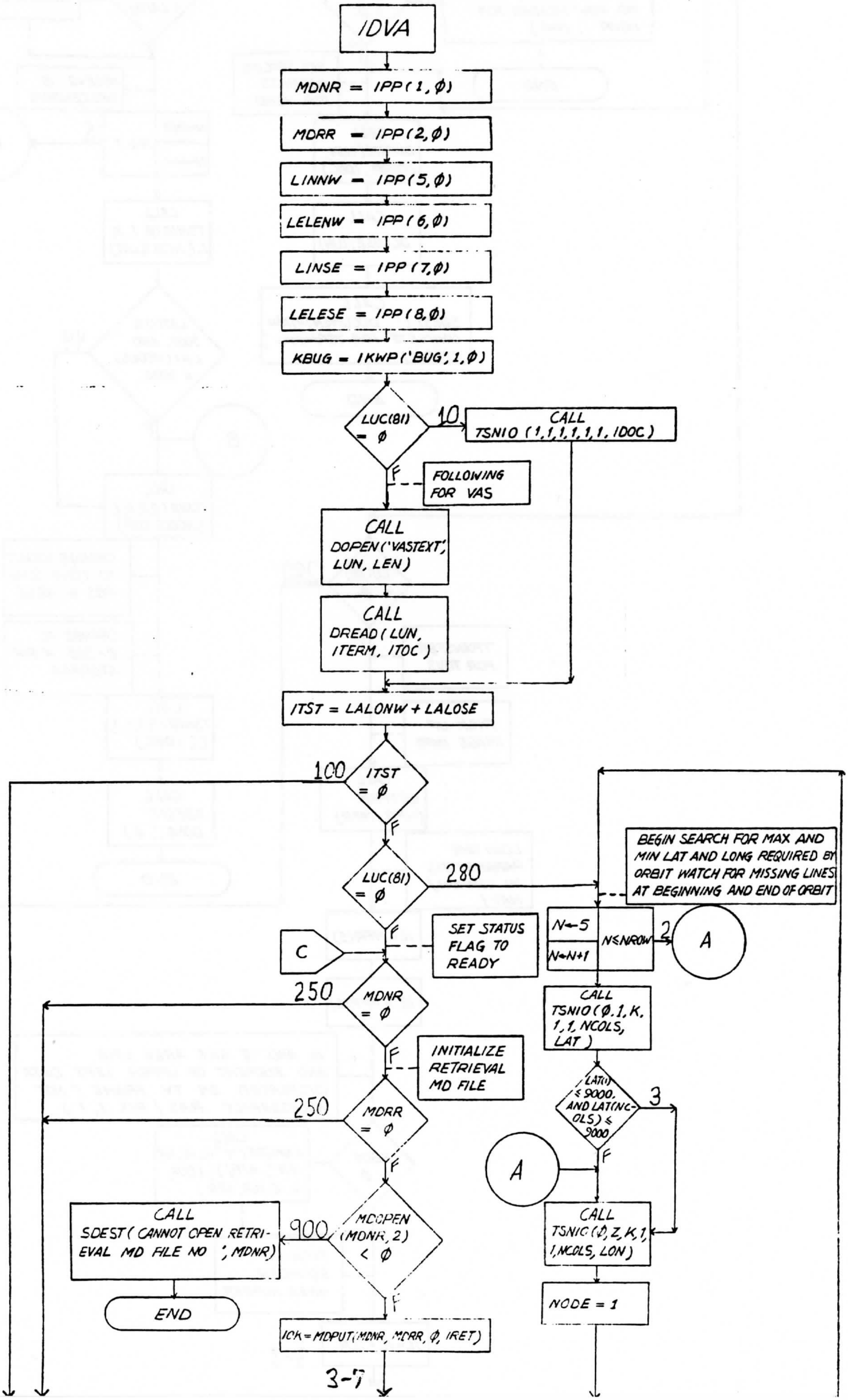
In the first mode case, variables MDNR, MDRR and KBUG are read in via positional parameters in code lines 35-39. Then, after the VASTEXT file is opened and read, control passes to statement 20 and then immediately to statement 100 (note that LALONW=0 for this mode, since it was not keyed in). At this 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 2 consecutive lines of data must be found before control will transfer to statement 130. 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 (code lines 161-168). These last two steps may be somewhat redundant, in light of the above call to GETFRM, but they are done to make sure the data is satisfactory. 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, then locate lower right line and element coordinates of the image). This is done in the code lines from approximately lines 181 to 211. 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 in code lines 212-215, 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 initialize the VASTEXT file. The row header is initialized by putting array IRET into the header via function MDPUT in code line 260, 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 array IDOC, which is then written, via subroutine DWRITE, into the VASTEXT file itself.

For the second "auto" mode case, IDVA proceeds identically to the first mode up to code line 48. Then, since both LALONW and

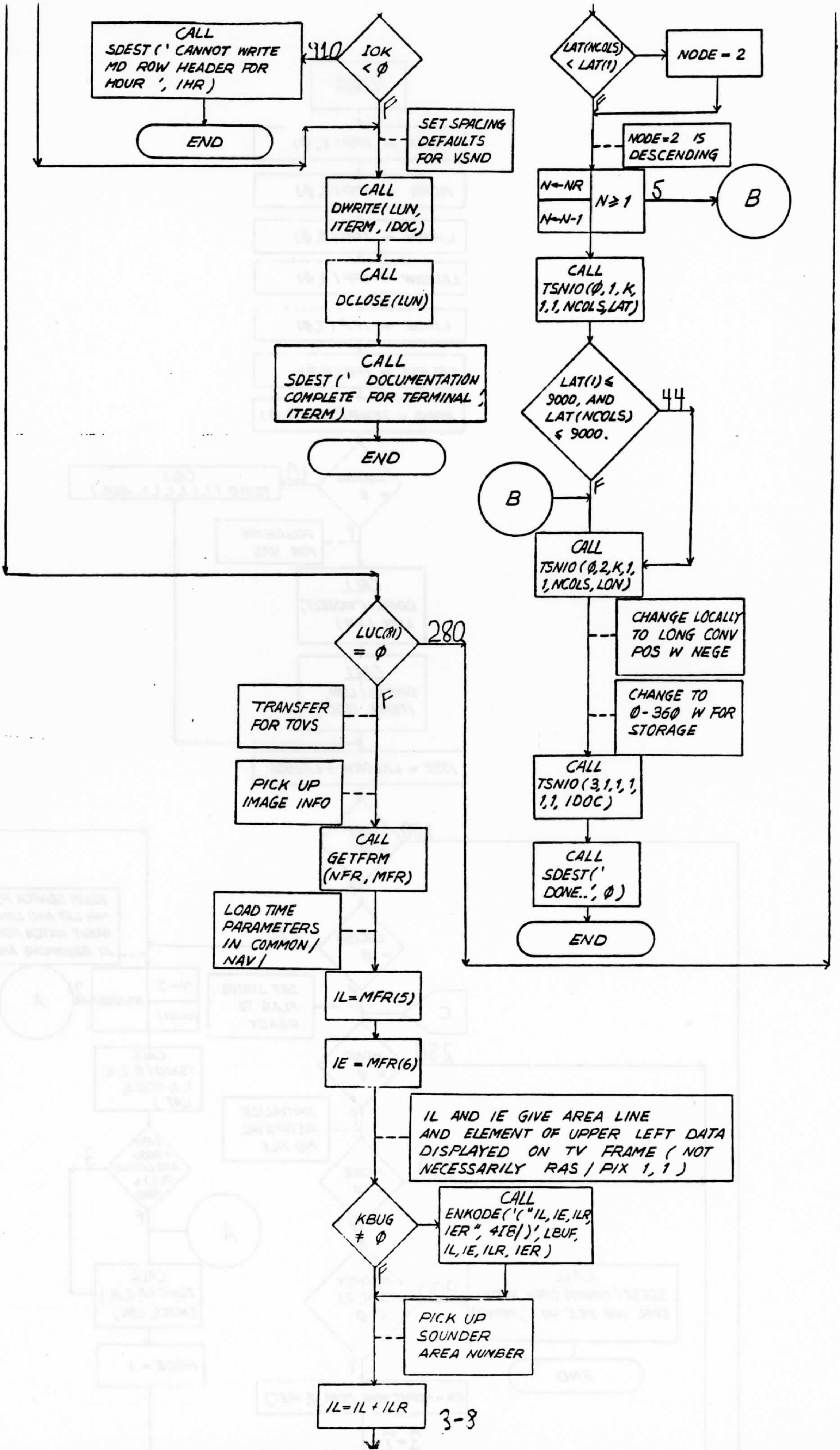
LALOSE have been keyed in, IDOC(25) and (26) are filled, variables LAN and LOW are calculated, and information pertaining to the sounder area is accessed via subroutine READD in code line 62. Following this, after the navigation has been initialized, four calls to subroutine SATEAR, in conjunction with other calculations, determine the line and element coordinates of the area to be processed (top line (IL), initial element (IE), lower 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.

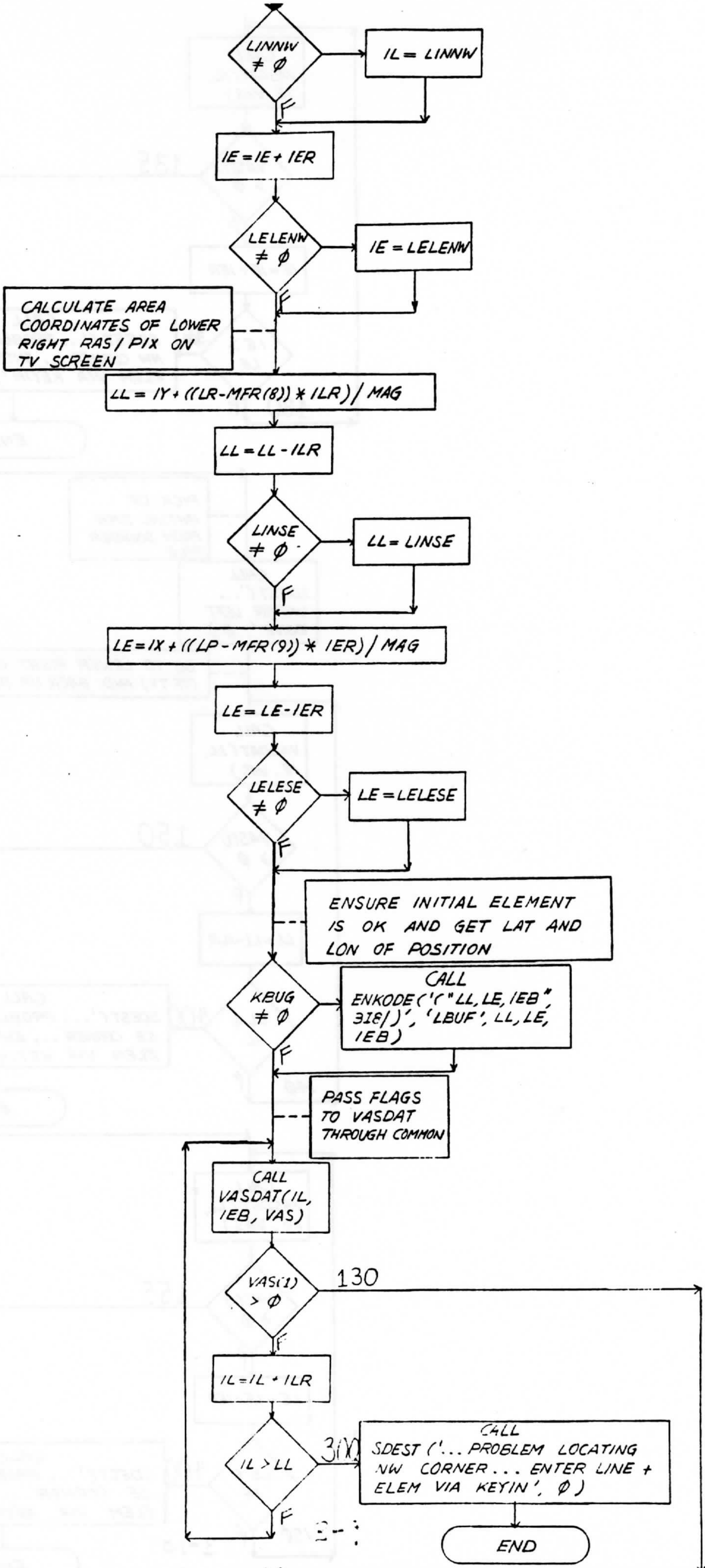
After control transfers to statement 115 (code line 124), IDVA functions like the previously-discussed non-'auto' mode up to code line 168. Namely, the upper bounds of the VAS data is located, giving the NW line coordinate, and then the western boundary is determined, giving the NW element coordinate.

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 code line 212 again, and it proceeds identically to the first mode from this point to the end.

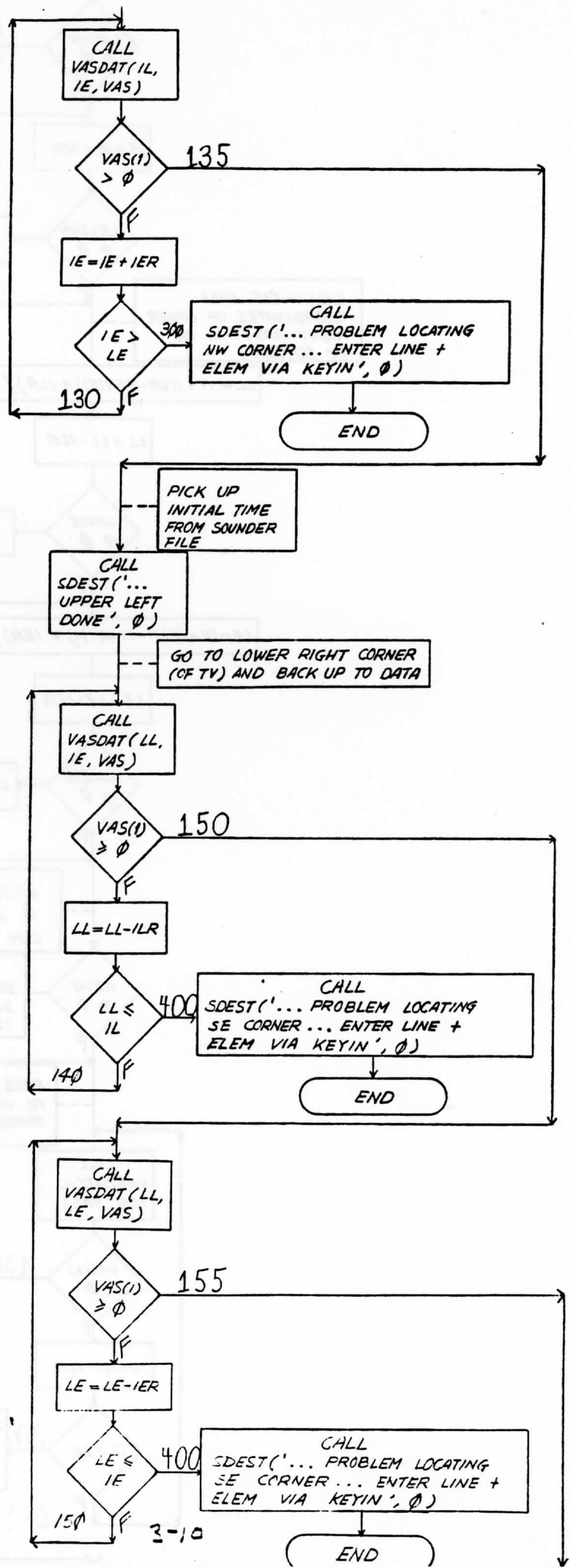


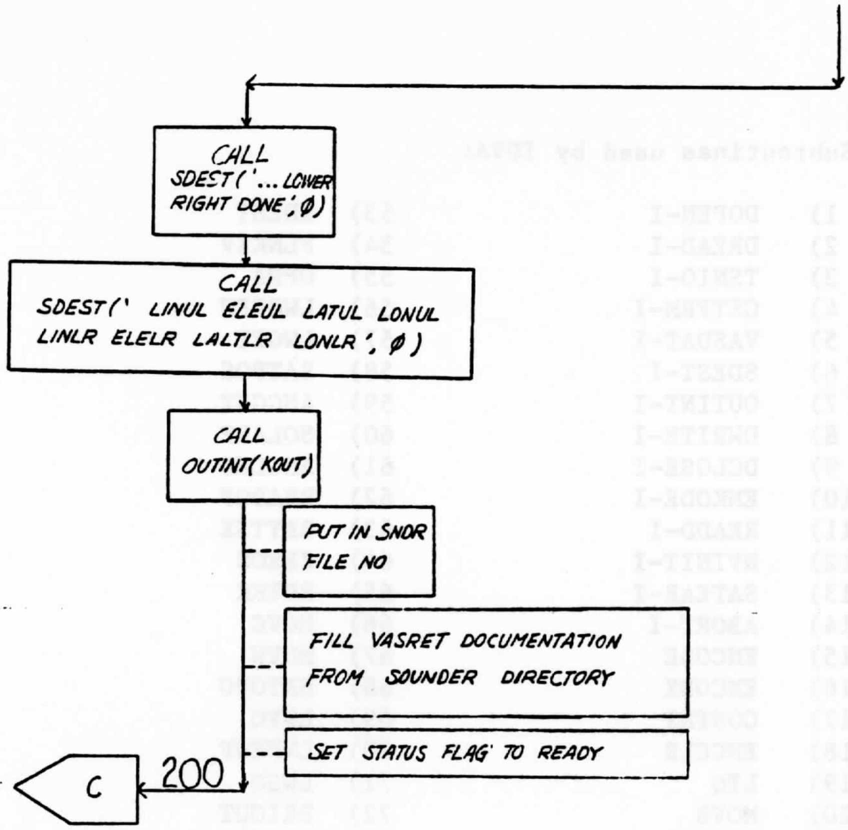
IDVA
MDNR = IPP(1, Ø)
MDRR = IPP(2, Ø)
LINNW = IPP(5, Ø)
LELENW = IPP(6, Ø)
LINSE = IPP(7, Ø)
LELESE = IPP(8, Ø)
KBUG = IKWP('BUG', 1, Ø)
LUC(81) = Ø
10
CALL TSNIO (1,1,1,1,1,1, IDOC)
F
FOLLOWING FOR VAS
CALL DOPEN('VASTEXT', LUN, LEN)
CALL DREAD(LUN, ITERM, ITOC)
ITST = LALONW + LALOSE
100
ITST = Ø
F
LUC(81) = Ø
280
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
F
C
SET STATUS FLAG TO READY
250
MDNR = Ø
F
INITIALIZE RETRIEVAL MD FILE
250
MDRR = Ø
F
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
CALL SDEST('CANNOT OPEN RETRIEVAL MD FILE NO ', MDNR)
900
MDOPEN (MDNR, 2) < Ø
F
END
ICK = MDPUT(MDNR, MDRR, Ø, IRET)


CALL SDEST (' CANNOT WRITE MD ROW HEADER FOR HOUR ', IHR)
910
IOK < Ø
F
END
SET SPACING DEFAULTS FOR VSND
CALL DWRITE (LUN, ITERM, IDOC)
CALL DCLOSE (LUN)
CALL SDEST (' DOCUMENTATION COMPLETE FOR TERMINAL ', ITERM)
END
LUC(91) = 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)
KBUG ≠ Ø
CALL ENKODE ('("IL, IE, ILR, IER", 4I8/)', LBUF, IL, IE, ILR, IER)
F
PICK UP SOUNDER AREA NUMBER
IL = IL + ILR
LAT(NCOLS) < LAT(1)
NODE = 2
F
NODE = 2 IS DESCENDING
N←NR
N←N-1
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 NEGE
CHANGE TO Ø-36Ø W FOR STORAGE
CALL TSNIO (3, 1, 1, 1, 1, 1, IDOC)
CALL SDEST (' DONE..', Ø)
END



LINNW ≠ Ø
IL = LINNW
F
IE = IE + IER
LELENW ≠ Ø
IE = LELENW
F
CALCULATE AREA COORDINATES OF LOWER RIGHT RAS / PIX ON TV SCREEN
LL = IY + ((LR-MFR(8)) * ILR) / MAG
LL = LL - ILR
LINSE ≠ Ø
LL = LINSE
F
LE = IX + ((LP - MFR(9)) * IER) / MAG
LE = LE - IER
LELESE ≠ Ø
LE = LELESE
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
PASS FLAGS TO VASDAT THROUGH COMMON
CALL VASDAT (IL, IEB, VAS)
VAS(1) > Ø
130
F
IL = IL + ILR
IL > LL
CALL SDEST ('... PROBLEM LOCATING NW CORNER ... ENTER LINE + ELEM VIA KEYIN', Ø)
F
END


CALL VASDAT(IL, IE, VAS)
VAS(1) > Ø
135
F
IE=IE+IER
IE > LE
3ØØ
CALL SDEST('... PROBLEM LOCATING NW CORNER ... ENTER LINE + ELEM VIA KEYIN', Ø)
END
130
F
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, IE, VAS)
VAS(1) ≥ Ø
150
F
LL=LL-ILR
LL ≤ IL
400
CALL SDEST('... PROBLEM LOCATING SE CORNER ... ENTER LINE + ELEM VIA KEYIN', Ø)
END
14Ø
F
CALL VASDAT(LL, LE, VAS)
VAS(1) ≥ Ø
155
F
LE=LE-IER
LE ≤ IE
400
CALL SDEST('... PROBLEM LOCATING SE CORNER ... ENTER LINE + ELEM VIA KEYIN', Ø)
END
15Ø
F


CALL SDEST('...LOWER RIGHT DONE',Ø)
CALL SDEST(' LINUL ELEUL LATUL LONUL LINLR ELELR LALTLR LONLR',Ø)
CALL OUTINT(KOUT)
PUT IN SNDR FILE NO
FILL VASRET DOCUMENTATION FROM SOUNDER DIRECTORY
SET STATUS FLAG TO READY
200
C

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) ENCCLR
19) LTQ
20) MOVB
21) CLEANA
22) TQ
23) PRLINX
24) PRCLOS
25) PROPEN
26) PRPRPR
27) LOCK
28) PRWR
29) PRRD
30) UNLOCK
31) PRCL
32) POST
33) BLKA
34) DECONV
35) FECONV
36) IECONV
37) ZECONV
38) II
39) STC
40) LWCLOS
41) MOVWC
42) LWMOP
43) TQMES
44) ITOC
45) TRMNL
46) MOVCW
47) GETNAV
48) EPOCH
49) GETGAM
50) VASNAV
51) DDEST
52) CLEANW
53) EDEST
54) PLNKIV
55) OPNA
56) LWOPEN
57) LWGET
58) SATPOS
59) ANGGET
60) SOLARP
61) ANGLES
62) READOF
63) RBYTSX
64) ZEROS
65) RDTRK
66) MOVC
67) MOVW
68) HRTOPO
69) LWPO
70) LWNEWF
71) LWSO
72) PRIOUT
73) TQSET
74) PUC
75) WD
76) JMBWTF

LOVA

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

First, the VASTEXT file is opened and read, returning such retrieval processing parameters as sounder area number (NSND), retrieval MD file number, number of retrievals processed (NRET), etc. through subroutine DREAD and array IDOC (see code line 33). 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 the rest 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 parenthses to the immediate right of each variable indicate(s) which program(s) affect that particular variable. Furthermore, the element of array IDOC that 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 to change several of the VASTEXT variables.

1.) YYDDD (IDVA-derived from IDOC(1))- year and date of VAS data

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

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

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

5.) LLNW (IDVA-IDOC(25))- latitude and longitude of NW corner of image

6.) LLSE (IDVA-IDOC(26))- latitude and longitude of SE corner of image

7.) STAT (IDVA-IDOC(28))- status flag

8.) NSAT (IDVA-derived from IDOC(1))-satellite number (29 for IR, 28 for visible-- tells what type of image is being displayed)

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

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

11.) MDRS (CSVA-IDOC((37))- row of surface data retrieval MD file that pertinent surface data is actually 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

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

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

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 (dewpoint depressions at station locations)

21.) NO. RETRIEVALS (SRAD-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...boxes are usually 5X5)

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

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

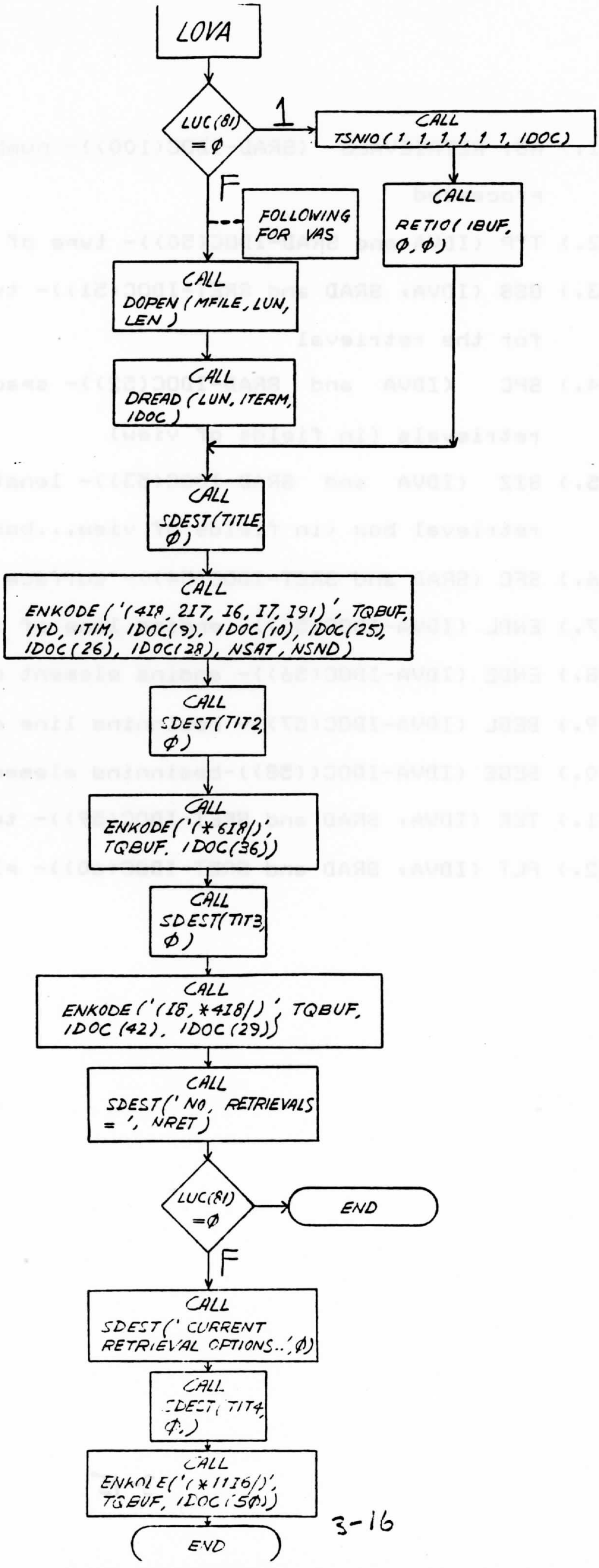
28.) ENDE (IDVA-IDOC(56))- ending element of retrieval area

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)
F
FOLLOWING FOR VAS
CALL RETIO(IBUF, Ø, Ø)
CALL DOPEN(MFILE, LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
CALL SDEST(TITLE, Ø)
CALL ENKODE('(4I8, 2I7, I6, I7, I91)', 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) ENCCLR
12) LTQ
13) DECONV
14) FECONV
15) IECONV
16) ZECONV
17) MOVB
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) BLKA
31) II
32) STC
33) LWCLOS
34) MOVWC
35) LWMOP
36) TQMES
37) ITOC
38) MOVCW
39) LWPO
40) EDEST
41) MOVW
42) CLEANW
43) LWNEWF
44) LWSO
45) PRIOUT
46) TQSET
47) PUC
48) WD
49) JMBWTF

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 set (or changed) 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 (if a change is desired) in the VASTEXT file in code lines 33-36. 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. Following this, the same is done for the surface data gridfile and its three associated grids (NGFS, ZGRID, TGRID and DGRID).

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 value for the number of retrievals is checked. If there are no changes to be made in this keyword (NRET= -1), SPVA is done. Otherwise, the retrieval MD file row header is accessed, which contains the old value of NRET. Then, the respective elements of the arrays IRET and IDOC are updated with the new keyed-in value of NRET, after which the updated row header is written back into row MDRR of the retrieval MD file, and the updated IDOC array is written back into the VASTEXT file. Both writes are accomplished by subroutine VRTIO in code line 63. 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 guess MD file

6.) MDNR- MD file for the VAS retrievals

7.) MDRR- row of retrieval MD file

8.) NGFG- gridfile for the first guess

9.) NGFS- gridfile for the surface data

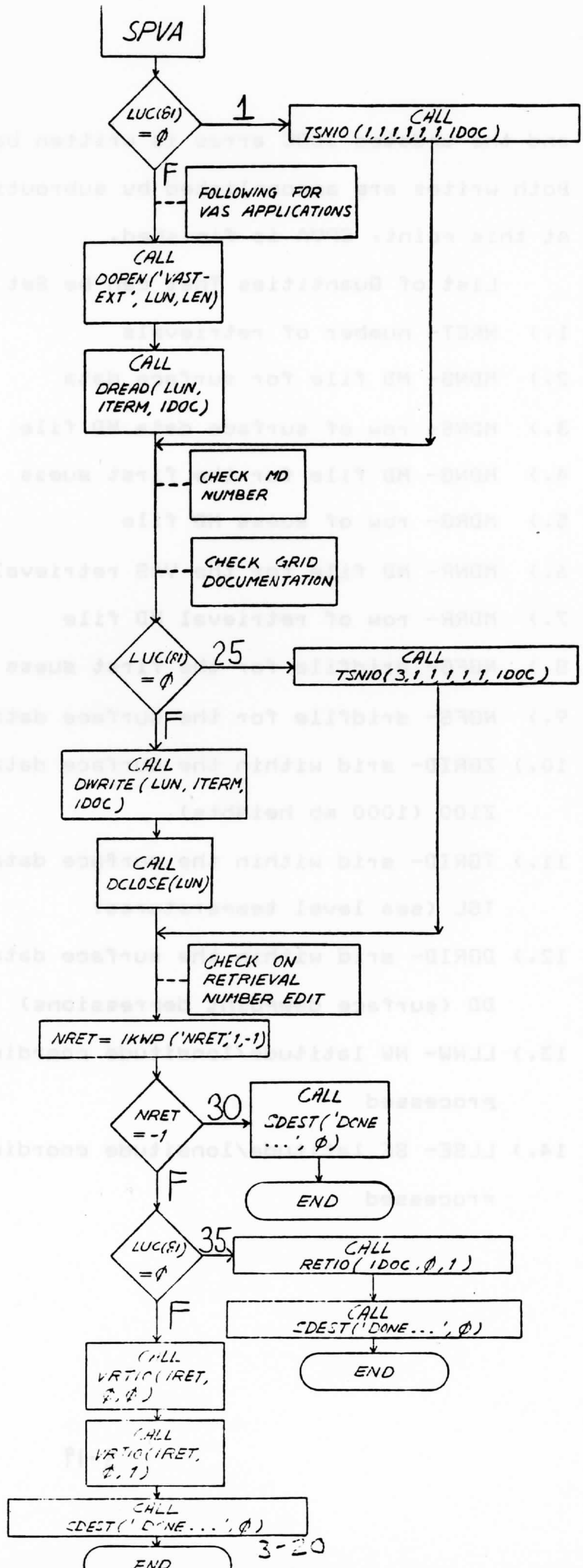
10.) ZGRID- grid within the surface data gridfile that contains Z100 (1000 mb heights)

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 image area being processed

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


SPVA
LUC(61) = Ø
1
CALL TSNIO (1,1,1,1,1,1,IDOC)
F
FOLLOWING FOR VAS APPLICATIONS
CALL DOPEN ('VAST-EXT', LUN, LEN)
CALL DREAD (LUN, ITERM, IDOC)
CHECK MD NUMBER
CHECK GRID DOCUMENTATION
LUC(61) = Ø
25
CALL TSNIO(3,1,1,1,1,1,IDOC)
F
CALL DWRITE (LUN, ITERM, IDOC)
CALL DCLOSE(LUN)
CHECK ON RETRIEVAL NUMBER EDIT
NRET= IKWP ('NRET',1,-1)
NRET = -1
30
CALL SDEST ('DONE ...', Ø)
END
F
LUC(61) = Ø
35
CALL RETIO (IDOC, Ø, 1)
CALL SDEST ('DONE ...', Ø)
END
F
CALL VRTIO (IRET, Ø, Ø)
CALL VRTIO (IRET, Ø, 1)
CALL SDEST ('DONE ...', Ø)
END

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) ENCCLR
14) LTQ
15) DECONV
16) FECONV
17) IECONV
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) II
34) STC
35) LWCLOS
36) MOVWC
37) LWMOP
38) ITOC
39) MOVCW
40) TQMES
41) LWPO
42) EDEST
43) MOVW
44) CLEANW
45) LWNEWF
46) LWSO
47) PRIOUT
48) TQSET
49) PUC
50) WD
51) JMBWTF

GSVA

This program prepares the upper air guess MD file by reformatting the first guess grids in a particular grid file into guess vectors as an MD file. The latitude and longitude serve as the row and column indicators for the MD file. The VASTEXT file is then 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 as it stands at the beginning of GSVA's execution. 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 grid boundaries (VASTEXT defaults), grid 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 (MDNG). This step is important because it allows GSVA to establish a relationship between grid scaling and MD scaling, which is used in code line 153. 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 we want it stored as degrees Kelvin * 1000 in the guess MD file, then the relationship between the two scalings will, in part, accomplish the task.

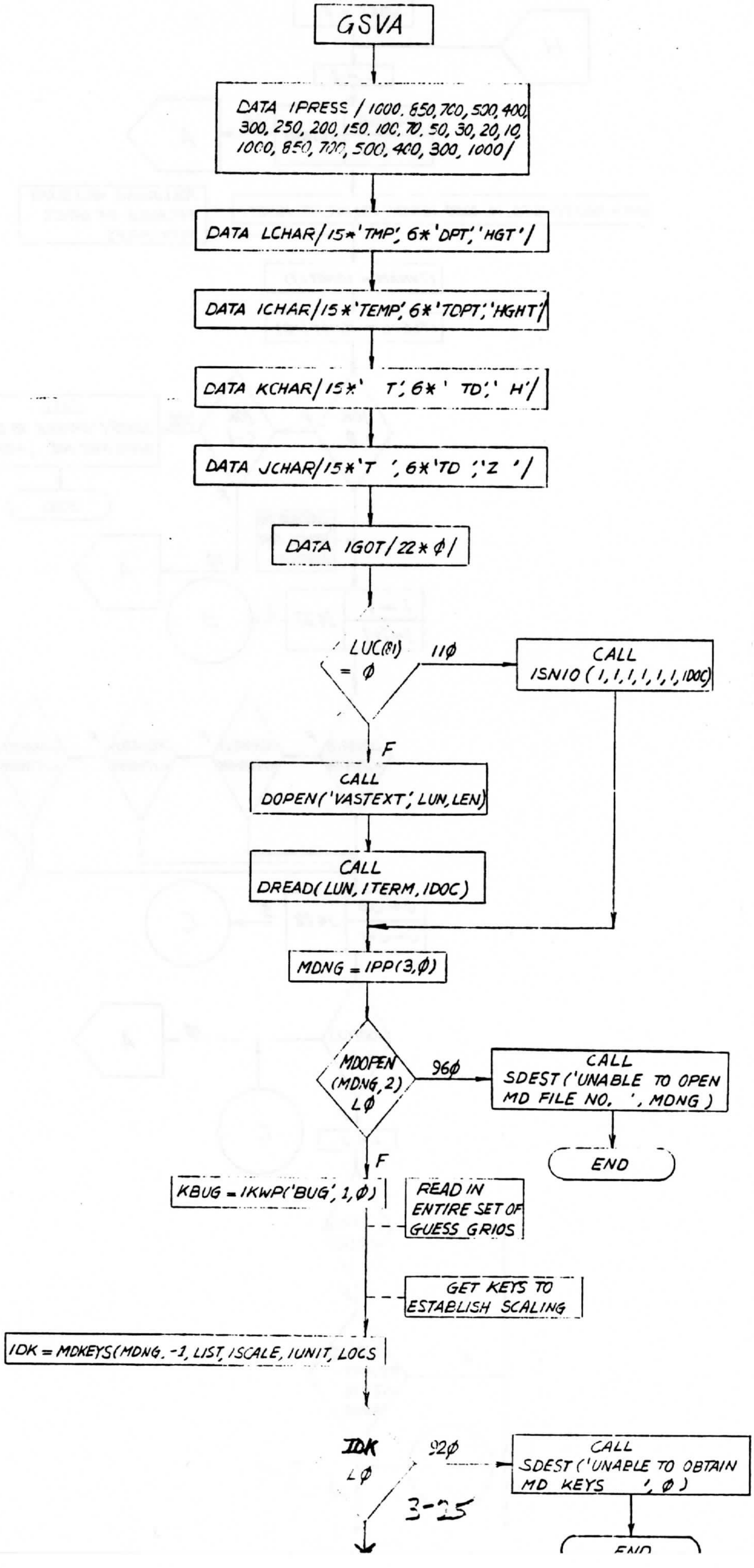
Following the reading of the keys, the guess grids are read from guess gridfile NGFG in DO-loop 10 via function IGGET, one at a time, and ordered for the guess MD file. As many as 22 grids can be accessed from the guess gridfile. Among these 22 grids are

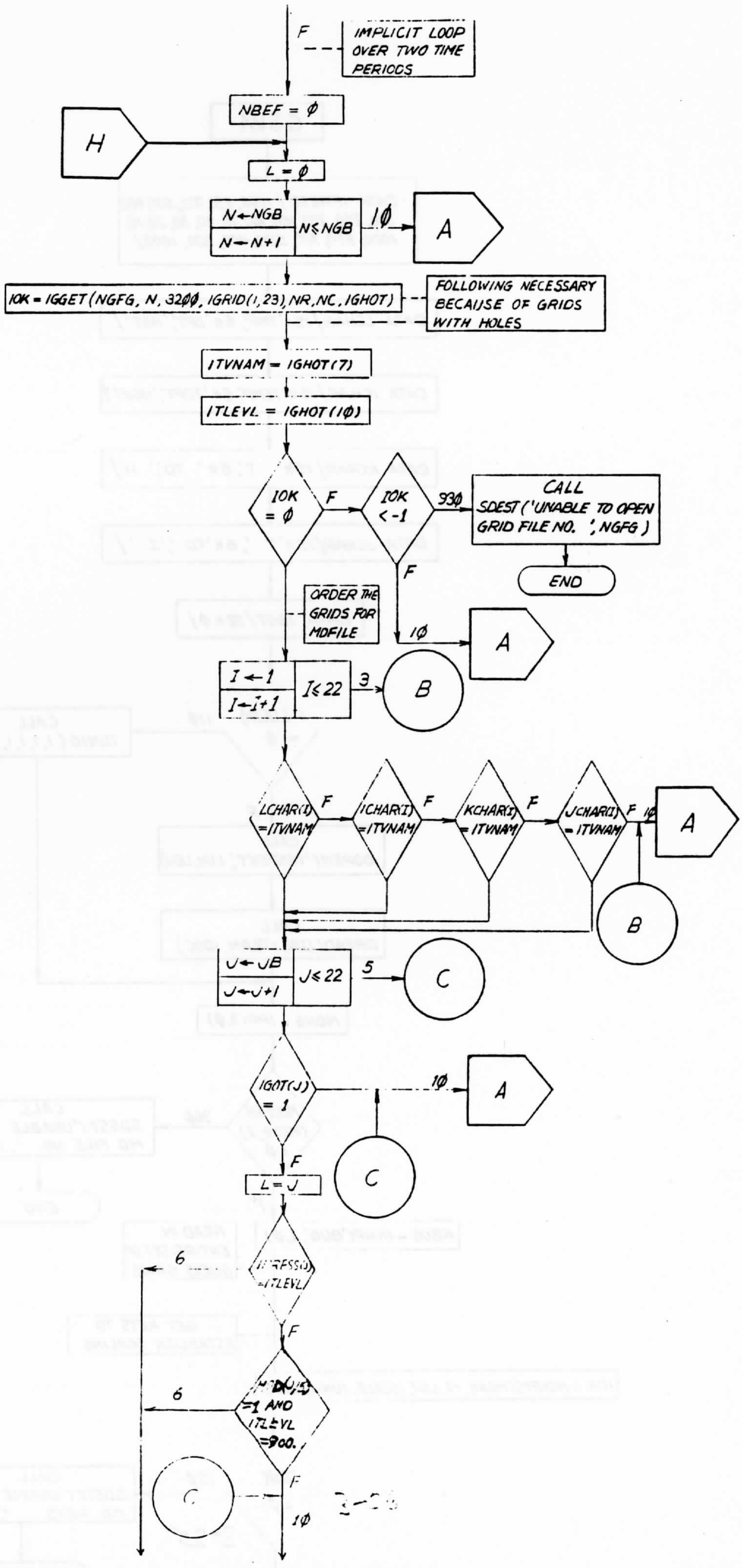
15 grids of temperature from 1000-300 mb, 6 grids of dewpoint temperature from 1000-300 mb, and a grid of 1000 mb heights. The ordered grids are placed in the array IGRID (code line 137), with each grid occupying one column of this array. Upon completion of this task, if the debug option is on (KBUG .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 (code line 160).

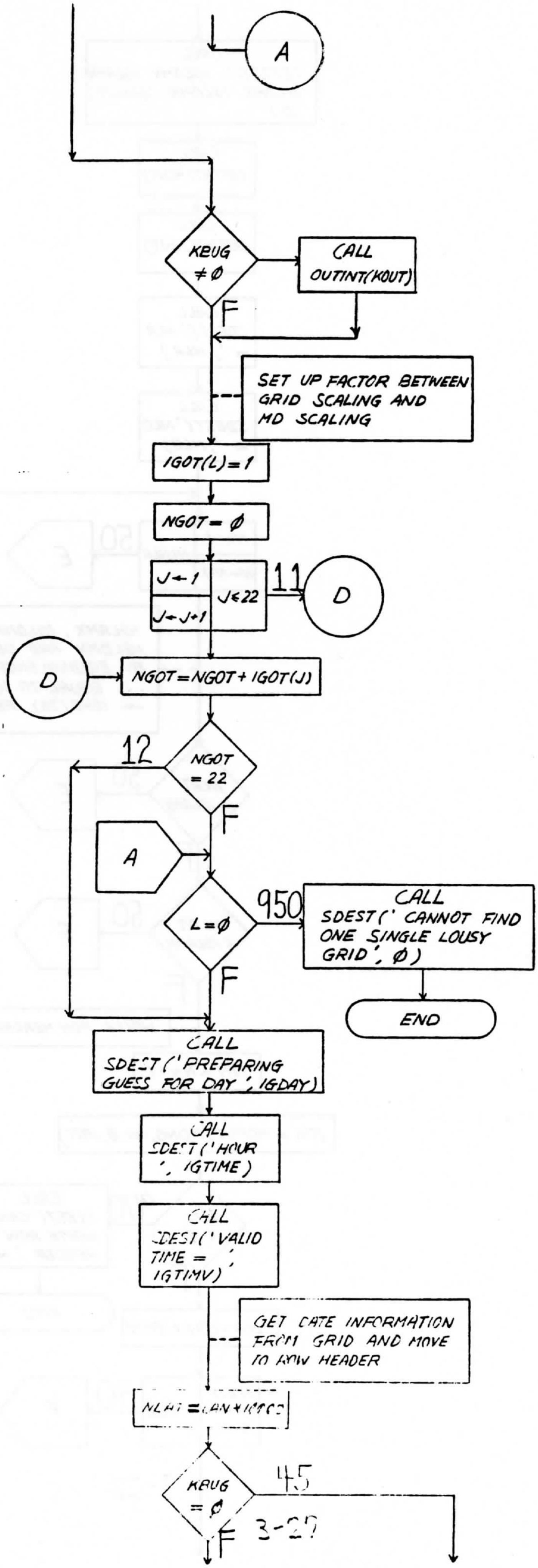
After all the guess grids have been accessed from the guess gridfile, three messages are displayed via the calls in code lines 164-166. Then, the bounds set previously by IDVA and accessed from the VASTEXT file (or explicitly keyed in) are used to generate latitude/longitude extents (variables LAN, LAS, LOW and LOE) for the MD file. These extents are limited (obviously) to the extents of the guess grids (see DO-loops 40 and 50). DO-loops 40 and 50 make up the vector of 25 pieces of guess information (day, hour (of model initialization), latitude, 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 in code line 268.

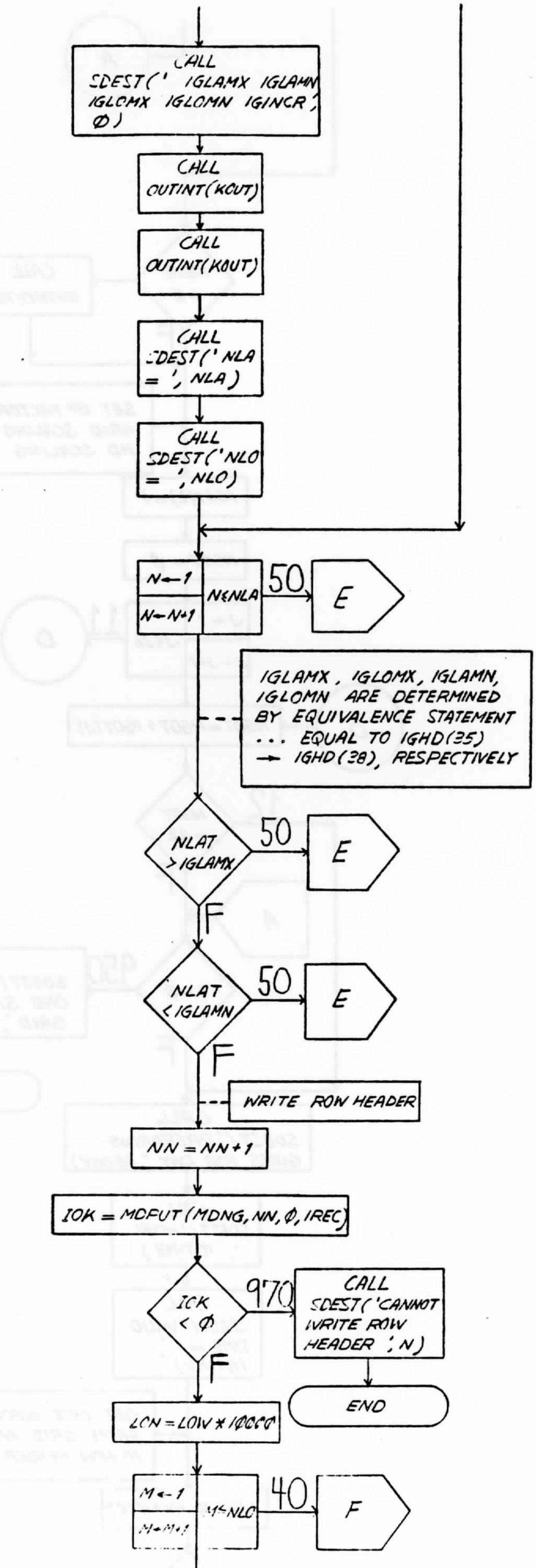
The data is reformatted row by row. In other words, starting in the upper left corner of the guess area, data is loaded in increments toward the right, and when that row is finished, data is loaded from the far left in the second row. In addition, if one needs a first guess which lies between two available guess grid times, a time interpolation option updates the latter 22 grid values of T,TD and Z100 to the correct time by using the earlier grid values and an interpolation factor (DT). (See code lines 272-278; note especially the return to statement 100 to access the earlier 22 grids of T, TD and Z100.)

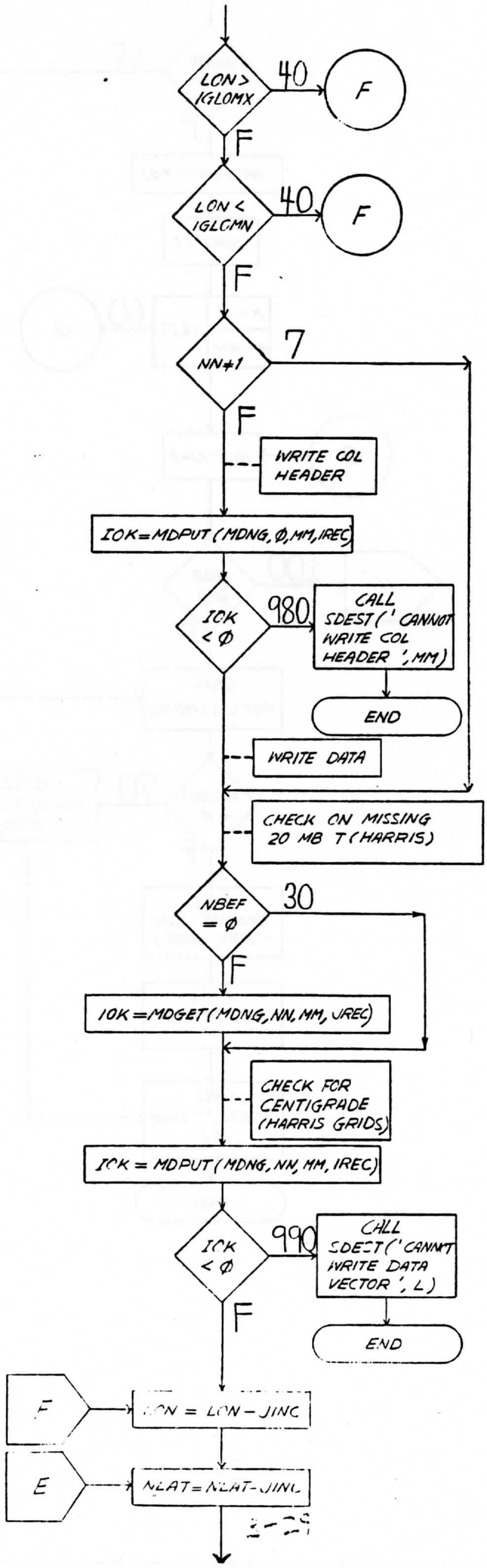
Finally, the guess MD file is closed, and the VASTEXT file is updated with the day, a quantity combining hour and valid time, guess MD file and row numbers, and the guess gridfile number, via subroutines DWRITE and DCLOSE. A subsequent execution of program LOVA will show that both the guess gridfile 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 ISNIO (1,1,1,1,1,1,IDOC)
F
CALL DOPEN('VASTEXT', LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
MDNG = IPP(3,Ø)
MDOPEN (MDNG,2) LØ
96Ø
CALL SDEST ('UNABLE TO OPEN MD FILE NO. ', MDNG)
END
F
KBUG = IKWP('BUG', 1, Ø)
READ IN ENTIRE SET OF GUESS GRIDS
GET KEYS TO ESTABLISH SCALING
IDK = MDKEYS(MDNG, -1, LIST, ISCALE, IUNIT, LOCS
IDK LØ
92Ø
CALL SDEST ('UNABLE TO OBTAIN MD KEYS ', Ø)
END

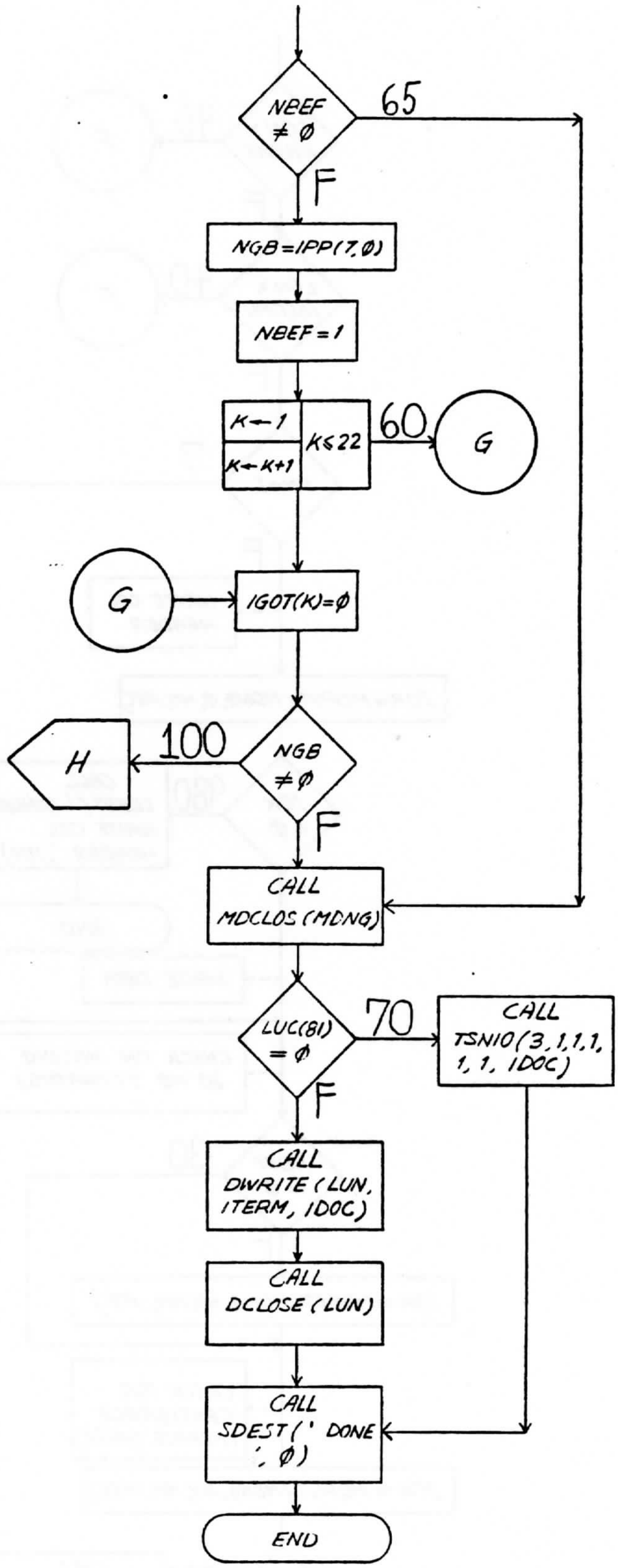

F
IMPLICIT LOOP OVER TWO TIME PERIODS
NBEF = Ø
H
L = Ø
N ← NGB
N ← N+1
N ≤ NGB
1Ø
A
IOK = IGGET(NGFG, N, 32ØØ, IGRID(1,23), NR, NC, IGHOT)
FOLLOWING NECESSARY BECAUSE OF GRIDS WITH HOLES
ITVNAM = IGHOT(7)
ITLEVL = IGHOT(1Ø)
IOK = Ø
F
IOK < -1
93Ø
CALL SDEST('UNABLE TO OPEN GRID FILE NO. ', NGFG)
END
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
B
J ← JB
J ← J+1
J ≤ 22
5
C
IGOT(J) = 1
1Ø
A
C
F
L = J
6
=ITLEVL
F
6
=1 AND
ITLEVL
=900.
C
F
1Ø


A
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
SET UP FACTOR BETWEEN GRID SCALING AND MD SCALING
IGOT(L)=1
NGOT= Ø
J←1
J≤22
J←J+1
11
D
D
NGOT=NGOT+IGOT(J)
12
NGOT = 22
F
A
L=Ø
950
CALL SDEST(' CANNOT FIND ONE SINGLE LOUSY GRID', Ø)
END
F
CALL SDEST('PREPARING GUESS FOR DAY ', IGDAY)
CALL SDEST('HOUR ', IGTIME)
CALL SDEST('VALID TIME = ', IGTIMV)
GET DATE INFORMATION FROM GRID AND MOVE TO NEW HEADER
NLAT =
KBUG = Ø
45
F


CALL SDEST(' IGLAMX IGLAMN IGLOMX IGLOMN IGINCR', Ø)
CALL OUTINT(KOUT)
CALL OUTINT(KOUT)
CALL SDEST(' NLA = ', NLA)
CALL SDEST(' NLO = ', NLO)
N←1
N←N+1
N≤NLA
50
E
IGLAMX, IGLOMX, IGLAMN, IGLOMN ARE DETERMINED BY EQUIVALENCE STATEMENT ... EQUAL TO IGHD(35) → IGHD(38), RESPECTIVELY
NLAT > IGLAMX
50
E
F
NLAT < IGLAMN
50
E
F
WRITE ROW HEADER
NN = NN+1
IOK = MDPUT(MDNG, NN, Ø, IREC)
IOK < Ø
970
CALL SDEST('CANNOT WRITE ROW HEADER', N)
END
F
LON = LOW * 10000
M←1
M←M+1
M≤NLO
40
F



LON > IGLOMX
40
F
F
LON < IGLOMN
40
F
F
NN≠1
7
F
WRITE COL HEADER
IOK=MDPUT(MDNG,Ø,MM,IREC)
IOK < Ø
980
CALL SDEST('CANNOT WRITE COL HEADER ',MM)
END
WRITE DATA
CHECK ON MISSING 20 MB T(HARRIS)
NBEF = Ø
30
F
IOK=MDGET(MDNG,NN,MM,JREC)
CHECK FOR CENTIGRADE (HARRIS GRIDS)
IOK = MDPUT(MDNG,NN,MM,IREC)
IOK < Ø
990
CALL SDEST('CANNOT WRITE DATA VECTOR ',L)
END
F
F
LON = LON - JINC
E
NLAT=NLAT-JINC


NBEF ≠ Ø
65
F
NGB=IPP(7,Ø)
NBEF=1
K←1
K←K+1
K≤22
60
G
G
IGOT(K)=Ø
100
H
NGB ≠ Ø
F
CALL MDCLOS(MDNG)
LUC(81) = Ø
70
CALL TSNIO(3,1,1,1,1,1, IDOC)
F
CALL DWRITE(LUN, ITERM, IDOC)
CALL DCLOSE(LUN)
CALL SDEST(' DONE ', Ø)
END

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) ENCCLR
14) DECONV
15) FECONV
16) IECONV
17) ZECONV
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) II
34) STC
35) LWCLOS
36) MOVWC
37) LWMOP
38) MOVCW
39) ITOC
40) LWPO
41) EDEST
42) MOVW
43) CLEANW
44) LWNEWF
45) LWSO
46) PRIOUT
47) TQSET
48) PUC
49) WD
50) JMBWTF

CSVA

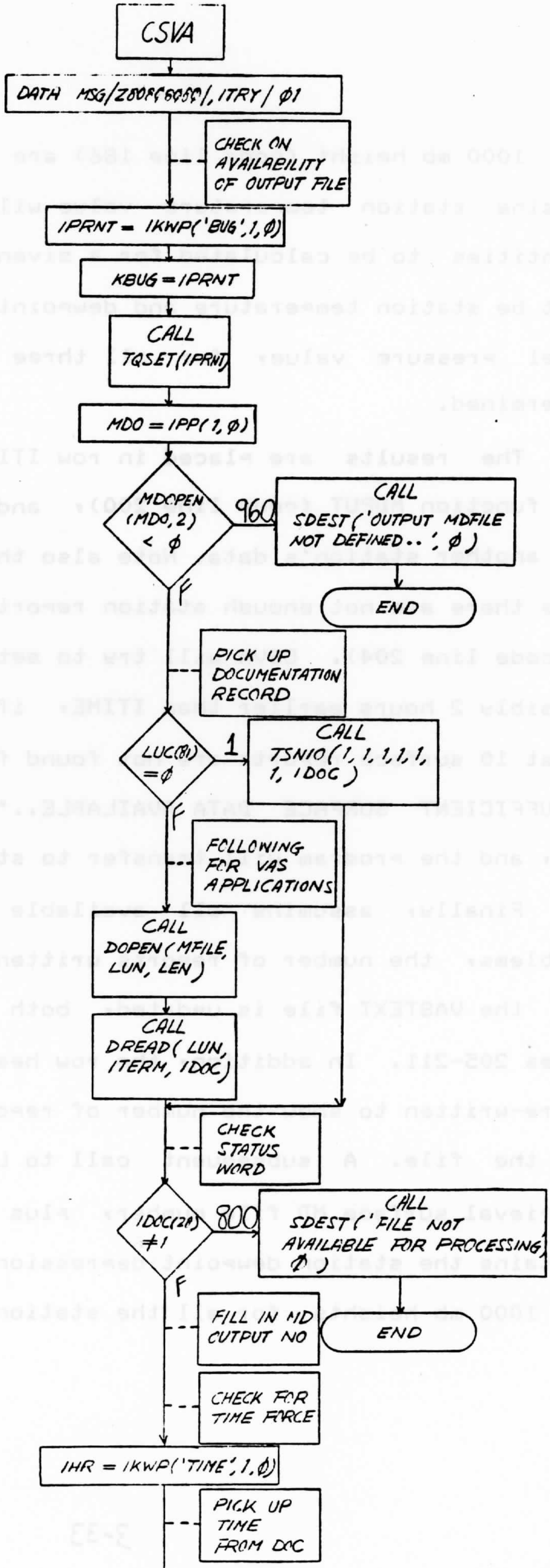
CSVA takes surface data from a given MD file, calculates surface dewpoint depression, sea level temperature and 1000 mb height for each station location, and places the resulting data into surface MD file MDNS.

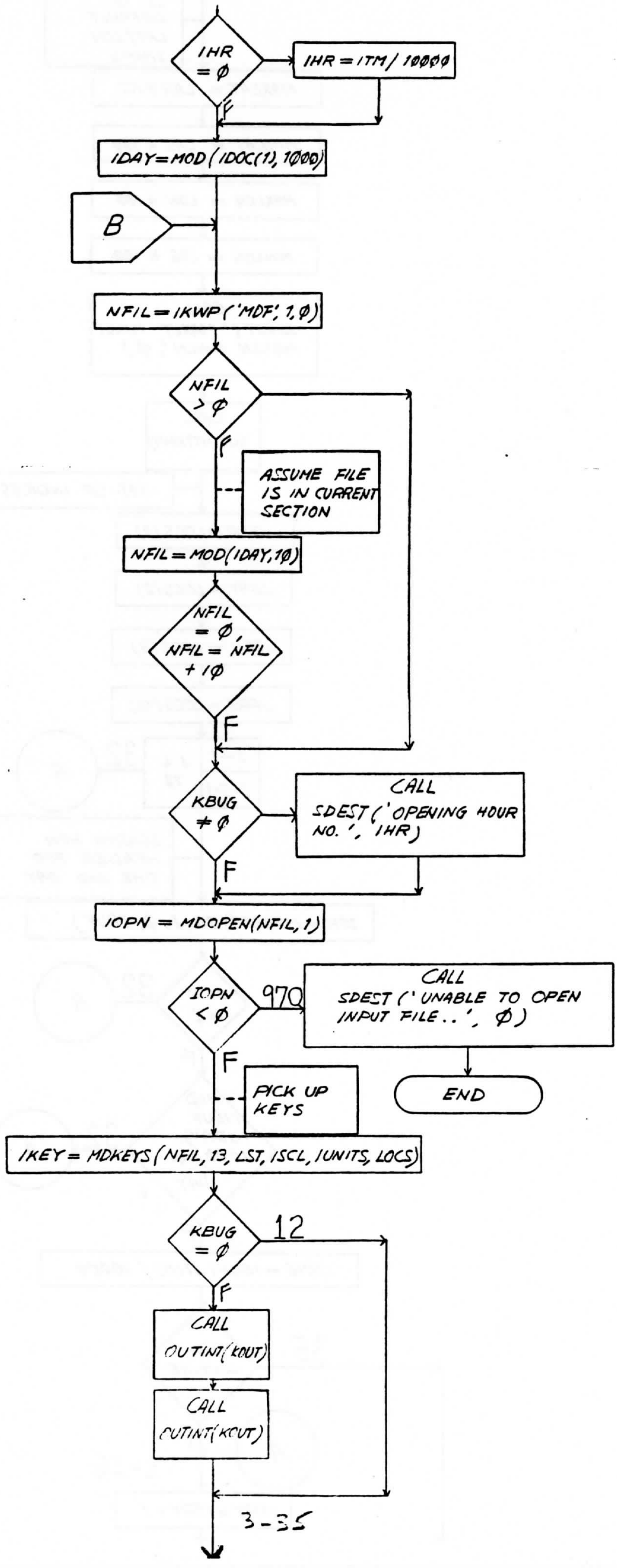
After the output MD file is opened for read/write and the existing documentation (VASTEXT file) accessed, the status word is checked (set to 1 by IDVA). Then, the MD file from which the conventional surface data is to be taken for the calculations is determined (NFIL), using the date of the radiance observations as stored in the VASTEXT file (IDOC(1)), unless the date and(or) time are explicitly keyed in. Note also that the conventional surface data MD file can be explicitly keyed in (code line 58). After the input MD file has been determined, it is opened for reading, the keys for file NFIL are read via function MDKEYS, and default latitude and longitude limits are set up via IDOC(25) and (26) to use in limitting the extent from which the raw observations can be taken (unless the limits are keyed in using 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 output MD file row ITIME is written. Then, the program begins gathering station data in implicit DO-loop 40 from input MD file NFIL, row ISV using function MDGET (code line 154), making sure the station is within the latitude-longitude bounds set earlier.

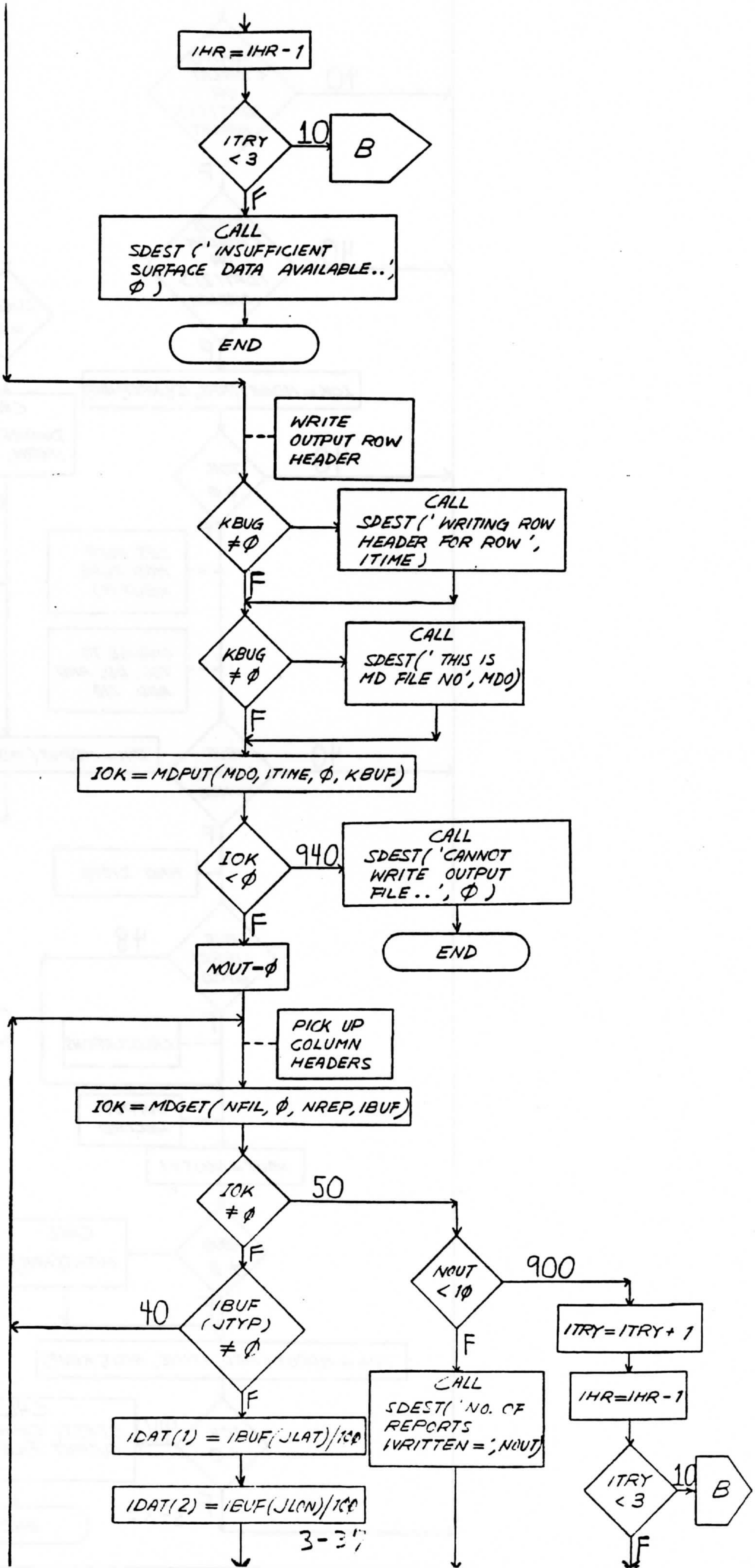
For each station whose data is read, the station dewpoint depression (code line 169), sea level temperature (code line 179) and 1000 mb height (code line 186) are calculated. Note that a missing station temperature value will cause none of these quantities to be calculated for a given report, and that 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.

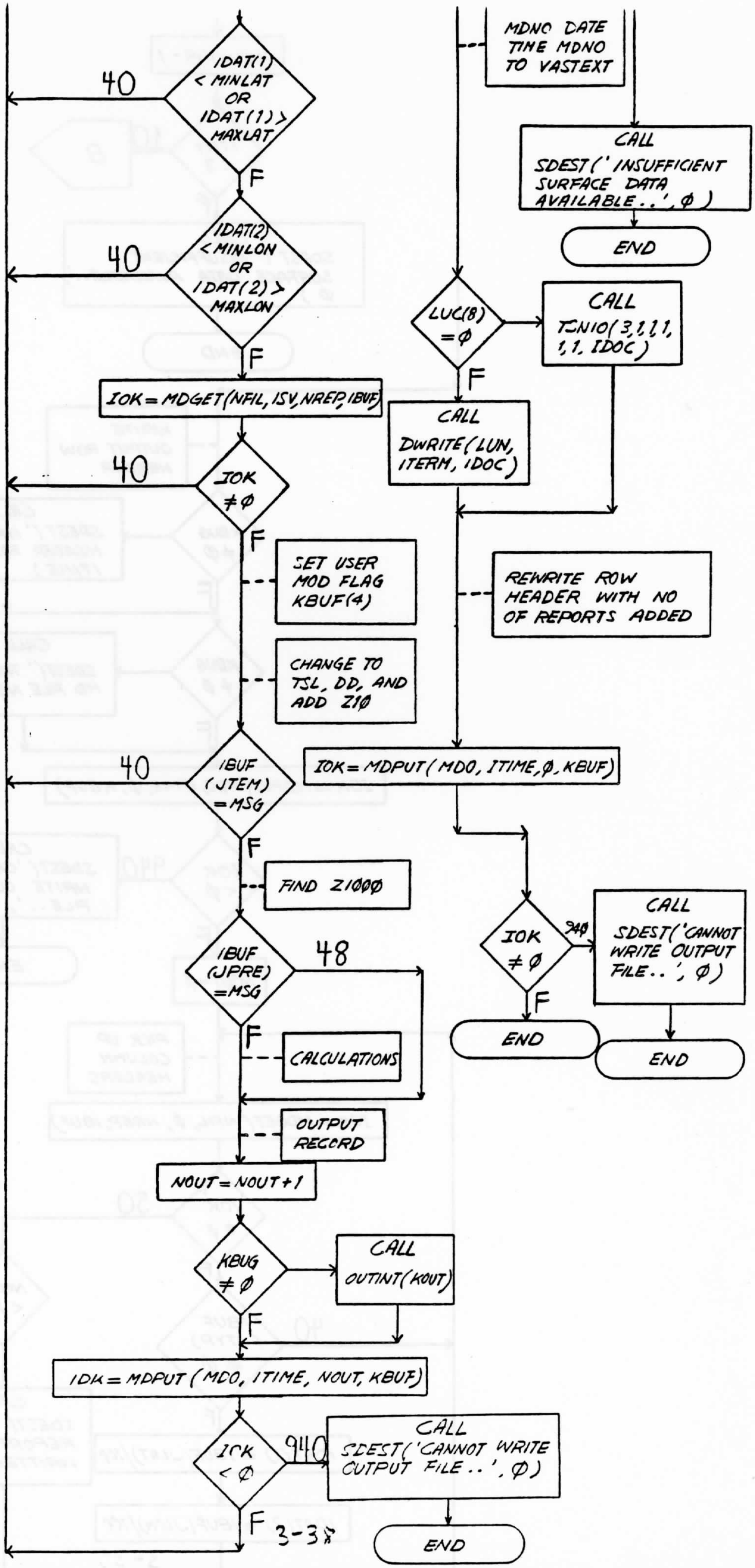
The results are placed in row ITIME of output MD file MDO via function MDPUT (code line 200), and the program goes back to get another station's data. Note also that provisions are made in case there are not enough station reports accessed (variable NOUT in code line 204). CSVA will try to get surface data for 1, or possibly 2 hours earlier than ITIME, if necessary. Then, if at least 10 surface reports are not found for either earlier time, " 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 is outputted to the CRT, and the VASTEXT file is updated, both events occurring in code lines 205-211. In addition, the row header of the output MD file is re-written to show the number of reports written into that row of the file. A subsequent call to LOVA will now show the retrieval surface MD file number, plus the row of the file that contains the station dewpoint depressions, sea level temperatures and 1000 mb heights for all the stations accessed.


CSVA
DATA MSG/Z80F(6050)/, ITRY/Ø1
CHECK ON AVAILABILITY OF OUTPUT FILE
IPRNT = IKWP('BUG', 1, Ø)
KBUG = IPRNT
CALL TQSET(IPRNT)
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
CALL TSNIO(1, 1, 1, 1, 1, IDOC)
F
FOLLOWING FOR VAS APPLICATIONS
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)
F
END
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
NFIL=IKWP ('MDF', 1, Ø)
NFIL > Ø
F
ASSUME FILE IS IN CURRENT SECTION
NFIL = MOD(IDAY, 1Ø)
NFIL = Ø, NFIL = NFIL + 1Ø
F
KBUG ≠ Ø
CALL SDEST ('OPENING HOUR NO. ', IHR)
F
IOPN = MDOPEN(NFIL, 1)
IOPN < Ø
970
CALL SDEST (' UNABLE TO OPEN INPUT FILE.. ', Ø)
END
F
PICK UP KEYS
IKEY = MDKEYS (NFIL, 13, LST, ISCL, IUNITS, LOCS)
KBUG = Ø
12
F
CALL OUTINT(KOUT)
CALL OUTINT(KOUT)


IHR=IHR-1
ITRY < 3
10
B
F
CALL SDEST ('INSUFFICIENT SURFACE DATA AVAILABLE..', Ø)
END
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
PICK UP COLUMN HEADERS
IOK = MDGET(NFIL, Ø, NREP, IBUF)
IOK ≠ Ø
50
F
NOUT < 1Ø
900
ITRY=ITRY + 1
IHR=IHR-1
ITRY < 3
10
B
F
40
IBUF (JTYP) ≠ Ø
F
CALL SDEST('NO. OF REPORTS WRITTEN =', NOUT)
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
IOK = MDGET(NFIL, ISV, NREP, IBUF)
IOK ≠ Ø
40
F
SET USER MOD FLAG KBUF(4)
CHANGE TO TSL, DD, AND ADD Z1Ø
IBUF (JTEM) = MSG
40
F
FIND Z1ØØØ
IBUF (JPRE) = MSG
48
F
CALCULATIONS
OUTPUT RECORD
NOUT = NOUT + 1
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
IOK = MDPUT (MDO, ITIME, NOUT, KBUF)
IOK < Ø
940
CALL SDEST('CANNOT WRITE OUTPUT FILE..', Ø)
END
F
MDNO DATE TIME MDNO TO VASTEXT
CALL SDEST(' INSUFFICIENT SURFACE DATA AVAILABLE..', Ø)
END
LUC(8) = 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) ENCCLR
13) LTQ
14) DECONV
15) FECONV
16) IECONV
17) ZECONV
18) MOVB
19) CLEANA
20) TQ
21) PRLINX
22) PRCLOS
23) PROPEN
24) PRPRPR
25) LOCK
26) PRWR
27) PRRD
28) UNLOCK
29) PRCL
30) POST
31) BLKA
32) II
33) STC
34) DCLOSE
35) LWCLOS
36) MOVWC
37) LWMOP
38) ITOC
39) MOVCW
40) LWPO
41) EDEST
42) MOVW
43) CLEANW
44) LWNEWF
45) LWSO
46) PRIOUT
47) TQSET
48) PUC
49) WD
50) JMBWTF

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, one will run SRVA three times to generate one grid each of sea level temperature, station 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 in code lines 98-104. 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 by reading the row header for row MDRS (or keyed in explicitly), and then printed on the CRT via subroutine SDEST. The data contained in these reports will give the 1000 mb height, sea level temperature or dewpoint depression data that is to be converted into gridded form.

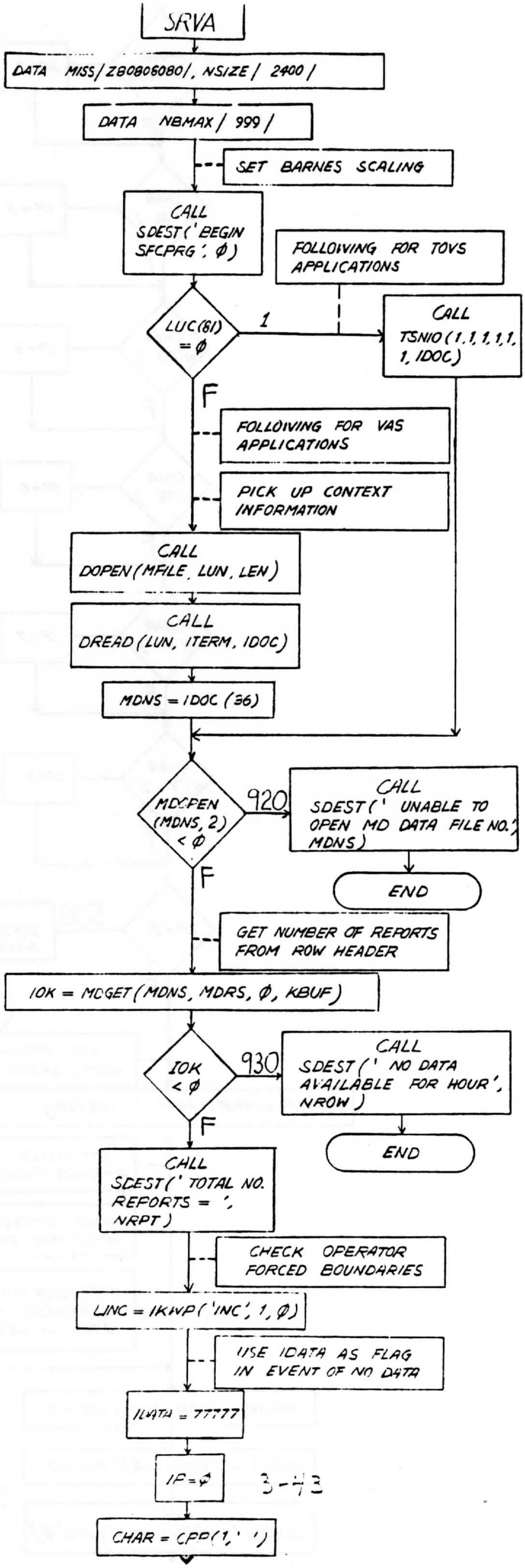
After the operator forced boundaries (set up in IDVA or keyed in) that will determine the area of the gridded data are determined, the character to be gridded is set by function CPP (code line 136), and the analysis field variable arrays IDL and ITG are initialized to 0. Note also the different types of variables that can be gridded in code lines 137-142, and the gross error check array that is set up in code line 145. This array is used later for both the gross error check of the

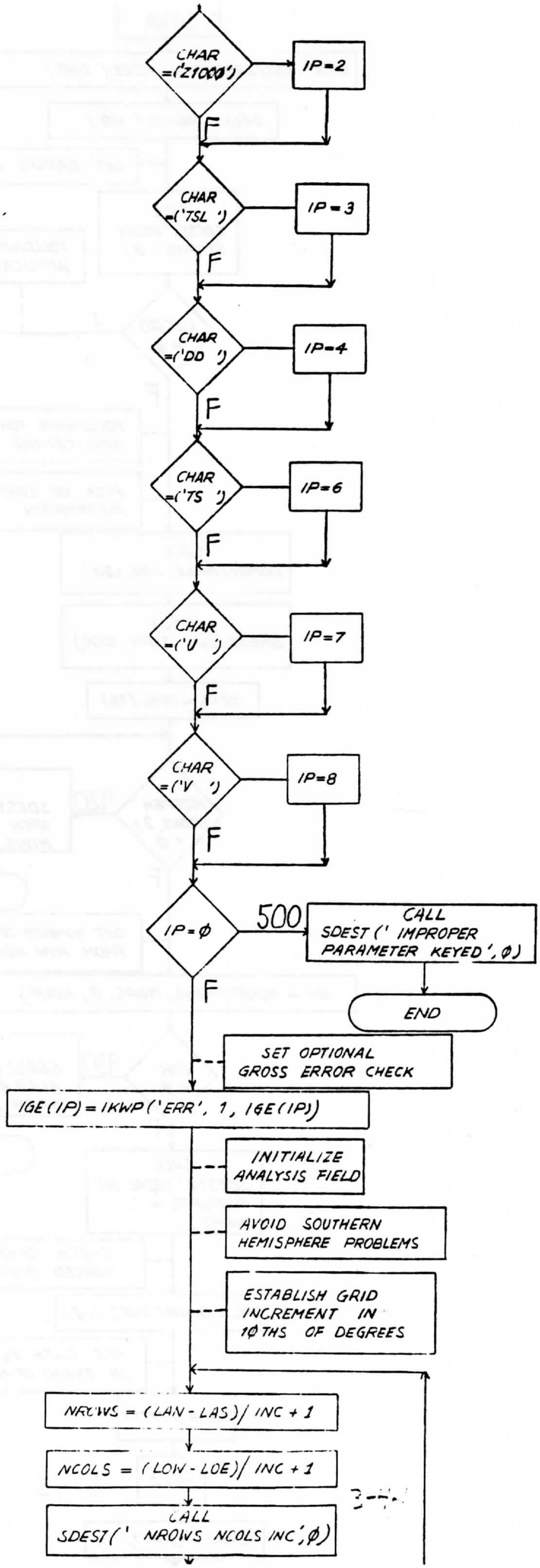
"VASGSS" (guess MD file MDNG) guess field and the edit option quality control check. Then, the grid increment (in tenths of degrees) is established. Keep in mind that the increment can be either keyed in or determined within the program. Note that if the grid increment is NOT keyed in, it will be calculated such that the total number of gridpoints is less than or equal to 2400.

At this point, it is possible to acquire the information to be used in the surface analyses in 3 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, one can get the guess data via subroutine SRGSS (the "VASGSS" route (code line 194)) and then the actual surface data via function MDGET (code line 225). On the other hand, guess information can also be accessed from a guess gridfile (NGRF (code line 205)) before the real observed data is read through the same function MDGET. Note the NODAT option in code line 212. If it is in effect, only the guess data will be stored in surface gridfile NGRFS in code line 344. In addition, note the preparation of variable FCK just before SRVA enters DO- loop 148. The third method of accessing the information for the analyses involves skipping the guess options (no keyin for keyword parameter GSS, or variable CGS = ' ') and simply reads only the actual surface MD file data. This third option would be used in the event there is good data coverage.

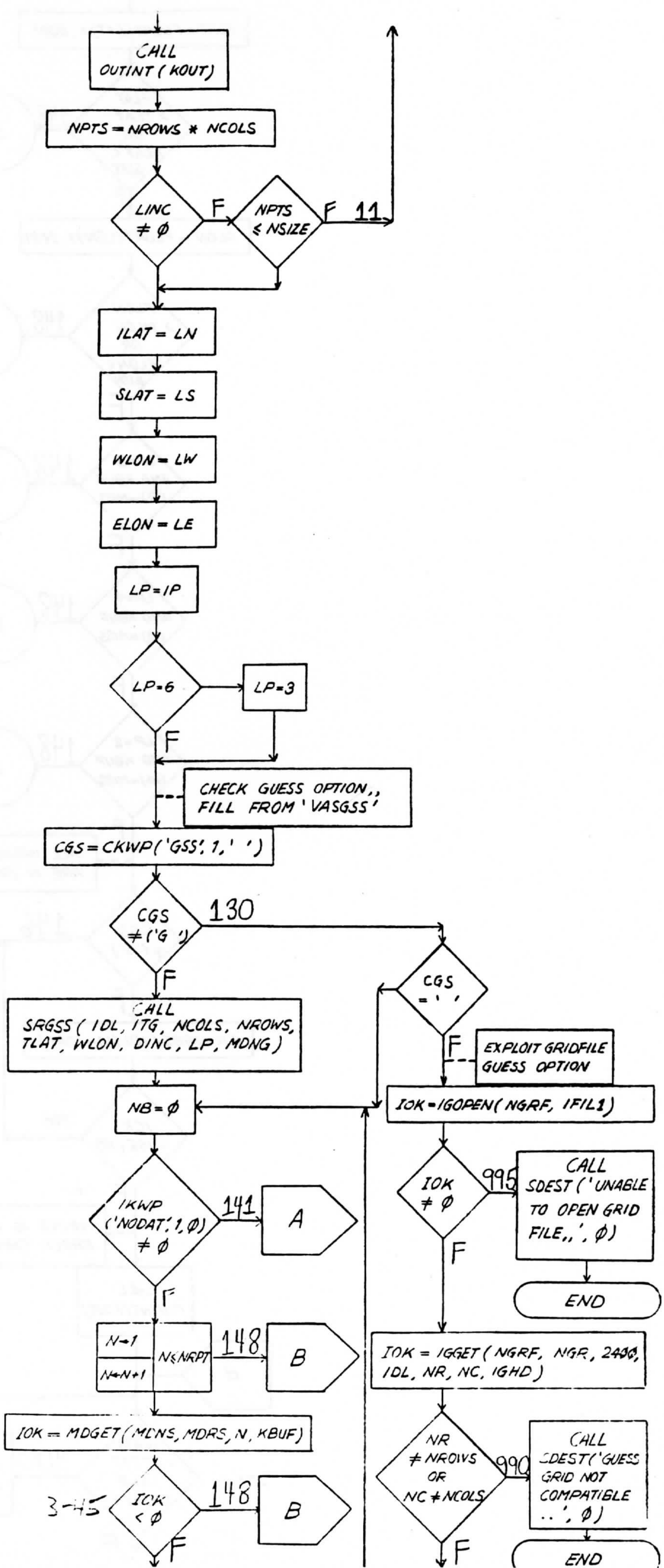
Following the accessing of the guess data, DO-loop 148 reads all the actual surface data for either the 1000 mb height, sea level temperature or dewpoint depression (depending upon which variable is being gridded) from the surface MD file (MDNS), performs a test (only for the "VASGSS" guess option) to screen out very poor surface data (code lines 245-254), and fills Barnes arrays DAS, RW and CL for use in the forthcoming Barnes analysis. Then, a Barnes analysis is set up and executed by calling subroutine FBARN. The Barnes analysis results in a uniform latitude-longitude grid of values. At this point, 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 FDAT and a value derived from the final grid (VAL) is .GE. than a certain standard deviation (the minimum of the calculated standard deviation and one set earlier in SRVA in array SDL), the actual surface data report is replaced with variable MISS in surface MD file MDNS, and the program returns to statement 110 to read in the surface data again and repeat the Barnes analysis.

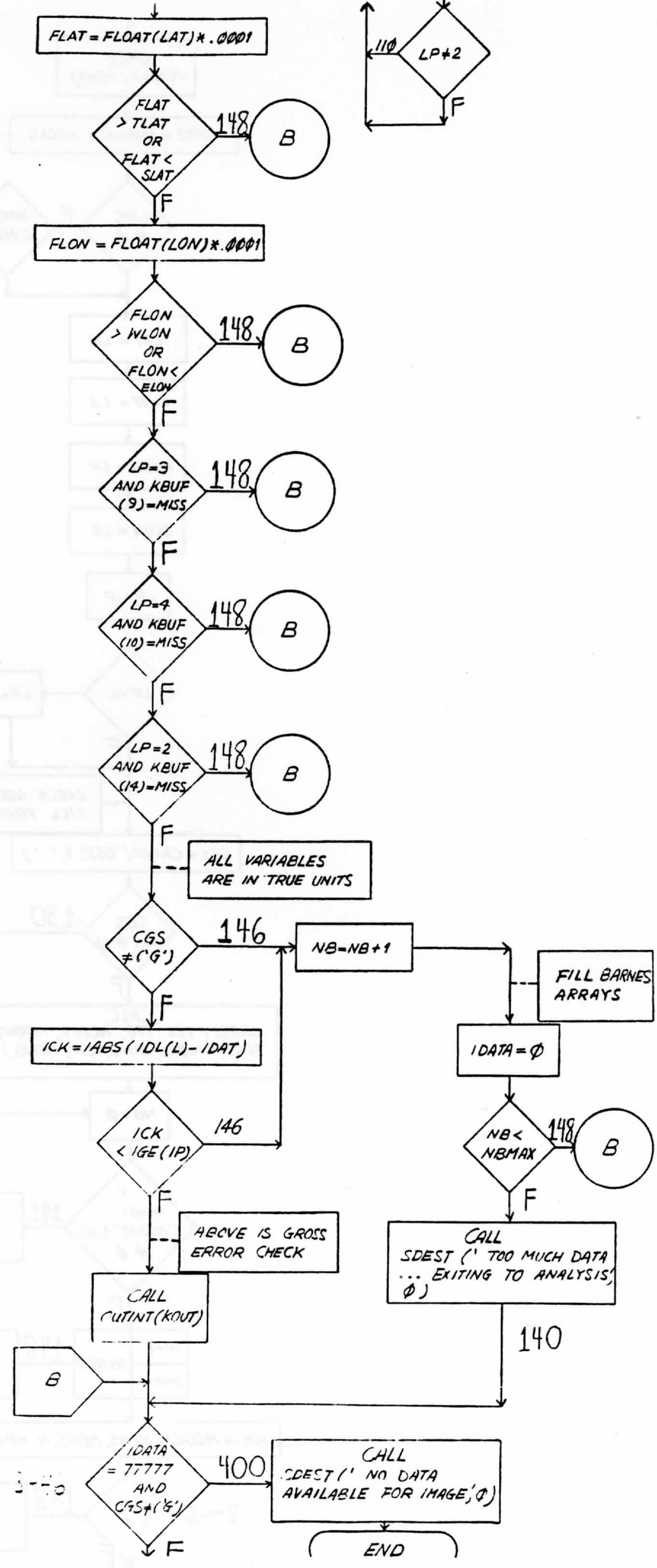
Finally, assuming the data editing passes, an output gridfile (NGRFS) is opened, and the sea level temperature, dewpoint depression or 1000 mb height gridpoint values are placed within it by function IGPUT in code line 344. In addition, array IDOC is updated to show both the surface gridfile number (NGRFS) and the individual grid number (ISTAT). This information is then subsequently written into the VASTEXT file via subroutine DWRITE.

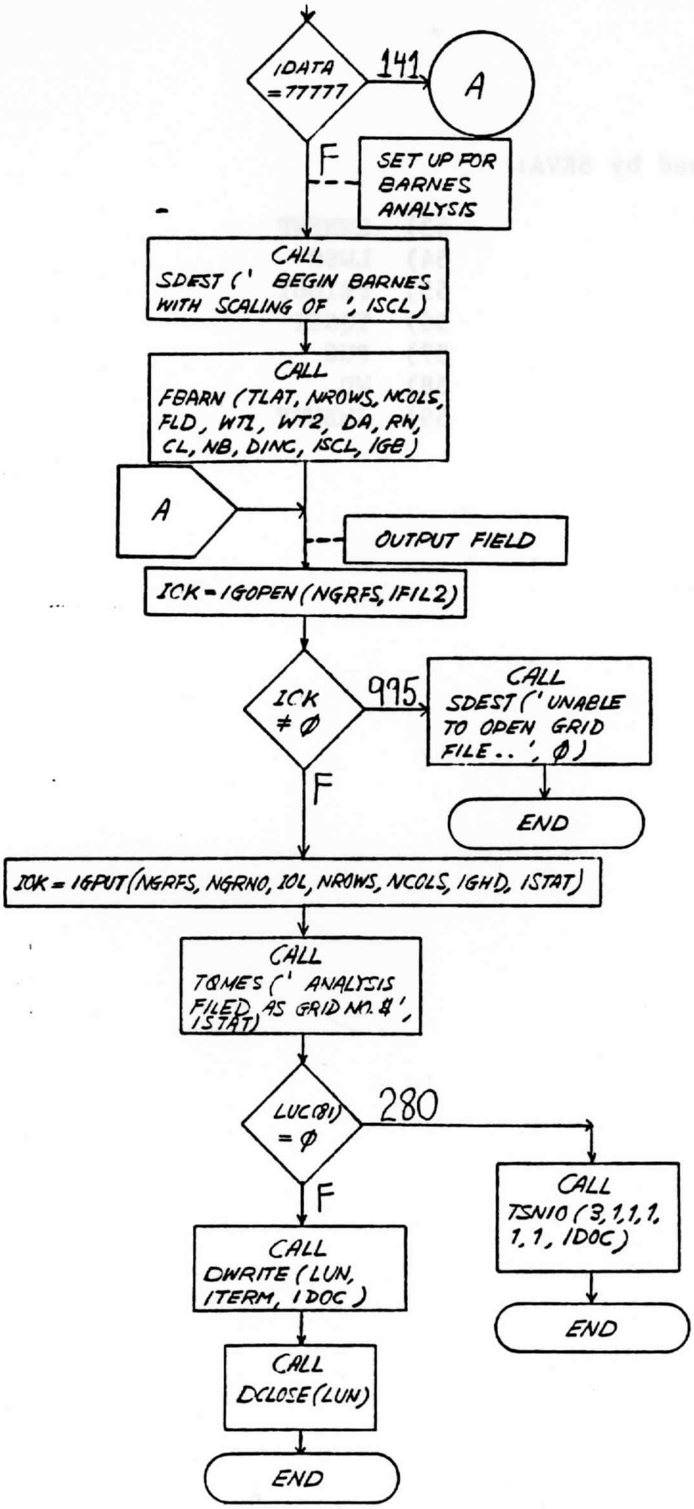

SRVA
DATA MISS/Z80808080/, NSIZE/ 2400/
DATA NBMAX/ 999/
SET BARNES SCALING
CALL SDEST('BEGIN SFCPRG', Ø)
FOLLOWING FOR TOVS APPLICATIONS
LUC(61) = Ø
1
CALL TSNIO(1,1,1,1,1, IDOC)
F
FOLLOWING FOR VAS APPLICATIONS
PICK UP CONTEXT INFORMATION
CALL DOPEN(MFILE, LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
MDNS = IDOC(36)
MDOPEN (MDNS, 2) < Ø
920
CALL SDEST(' UNABLE TO OPEN MD DATA FILE NO.', MDNS)
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', NROW)
END
F
CALL SDEST(' TOTAL NO. REPORTS = ', NRPT)
CHECK OPERATOR FORCED BOUNDARIES
LINC = IKWP('INC', 1, Ø)
USE IDATA AS FLAG IN EVENT OF NO DATA
IDATA = 77777
IP = Ø
CHAR = CPP(1,' ')



CHAR =('Z1000')
IP=2
F
CHAR =('TSL ')
IP=3
F
CHAR =('DD ')
IP=4
F
CHAR =('TS ')
IP=6
F
CHAR =('U ')
IP=7
F
CHAR =('V ')
IP=8
F
IP=∅
500
CALL SDEST(' IMPROPER PARAMETER KEYED',∅)
END
F
SET OPTIONAL GROSS ERROR CHECK
IGE(IP)=IKWP('ERR', 1, IGE(IP))
INITIALIZE ANALYSIS FIELD
AVOID SOUTHERN HEMISPHERE PROBLEMS
ESTABLISH GRID INCREMENT IN 1∅THS OF DEGREES
NROWS = (LAN-LAS)/INC + 1
NCOLS = (LON-LOE)/INC + 1
CALL SDEST(' NROWS NCOLS INC',∅)


CALL OUTINT (KOUT)
NPTS = NROWS * NCOLS
LINC ≠ Ø
F
NPTS ≤ NSIZE
F
11
ILAT = LN
SLAT = LS
WLON = LW
ELON = LE
LP = IP
LP = 6
LP = 3
F
CHECK GUESS OPTION,, FILL FROM 'VASGSS'
CGS = CKWP('GSS', 1, ' ')
CGS ≠ ('G ')
130
F
CALL SRGSS (IDL, ITG, NCOLS, NROWS, TLAT, WLON, DINC, LP, MDNG)
CGS = ' '
F
EXPLOIT GRIDFILE GUESS OPTION
IOK = IGOPEN(NGRF, IFIL1)
NB = Ø
IKWP ('NODAT', 1, Ø) ≠ Ø
141
A
IOK ≠ Ø
995
CALL SDEST ('UNABLE TO OPEN GRID FILE,,', Ø)
F
END
F
N←1
N←N+1
N≤NRPT
148
B
IOK = IGGET (NGRF, NGR, 2400, IDL, NR, NC, IGHD)
IOK = MDGET (MDNS, MDRS, N, KBUF)
NR ≠ NROWS OR NC ≠ NCOLS
990
CALL SDEST ('GUESS GRID NOT COMPATIBLE ..', Ø)
IOK < Ø
148
B
F
F
END


FLAT = FLOAT(LAT)*.0001
FLAT > TLAT OR FLAT < SLAT
148
B
F
FLON = FLOAT(LON)*.0001
FLON > WLON OR FLON < ELON
148
B
F
LP=3 AND KBUF (9)=MISS
148
B
F
LP=4 AND KBUF (10)=MISS
148
B
F
LP=2 AND KBUF (14)=MISS
148
B
F
ALL VARIABLES ARE IN TRUE UNITS
CGS ≠ ('G')
146
F
NB=NB+1
FILL BARNES ARRAYS
IDATA=0
NB < NBMAX
148
B
F
CALL SDEST (' TOO MUCH DATA ... EXITING TO ANALYSIS', 0)
140
ICK=IABS(IDL(L)-IDAT)
ICK < IGE(IP)
146
F
ABOVE IS GROSS ERROR CHECK
CALL CUTINT(KOUT)
B
IDATA = 77777 AND CGS≠('G')
400
CALL SDEST(' NO DATA AVAILABLE FOR IMAGE', 0)
END
F
110
LP≠2
F
3-46


IDATA =77777
141
A
F
SET UP FOR BARNES ANALYSIS
CALL SDEST (' BEGIN BARNES WITH SCALING OF ', ISCL)
CALL FBARN (TLAT, NROWS, NCOLS, FLD, WT1, WT2, DA, RN, CL, NB, DINC, ISCL, IGB)
A
OUTPUT FIELD
ICK = IGOPEN (NGRFS, IFIL2)
ICK ≠ Ø
995
CALL SDEST (' UNABLE TO OPEN GRID FILE.. ', Ø)
END
F
IOK = IGPUT (NGRFS, NGRNO, IOL, NROWS, NCOLS, IGHD, ISTAT)
CALL TQMES (' ANALYSIS FILED AS GRID NO. #', ISTAT)
LUC(81) = Ø
280
CALL TSNIO (3, 1, 1, 1, 1, 1, IDOC)
END
F
CALL DWRITE (LUN, ITERM, IDOC)
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) ENCCLR
34) DECONV
35) FECONV
36) IECONV
37) ZECONV
38) II
39) STC
40) LWCLOS
41) MOVWC
42) LWMOP
43) ITOC
44) VASGES
45) FILL
46) LINFIL
47) INTER
48) HEAPFY
49) LWPO
50) EDEST
51) MOVW
52) CLEANW
53) LWNEWF
54) LWSO
55) PRIOUT
56) TQSET
57) PUC
58) WD
59) JMBWTF

XRVA

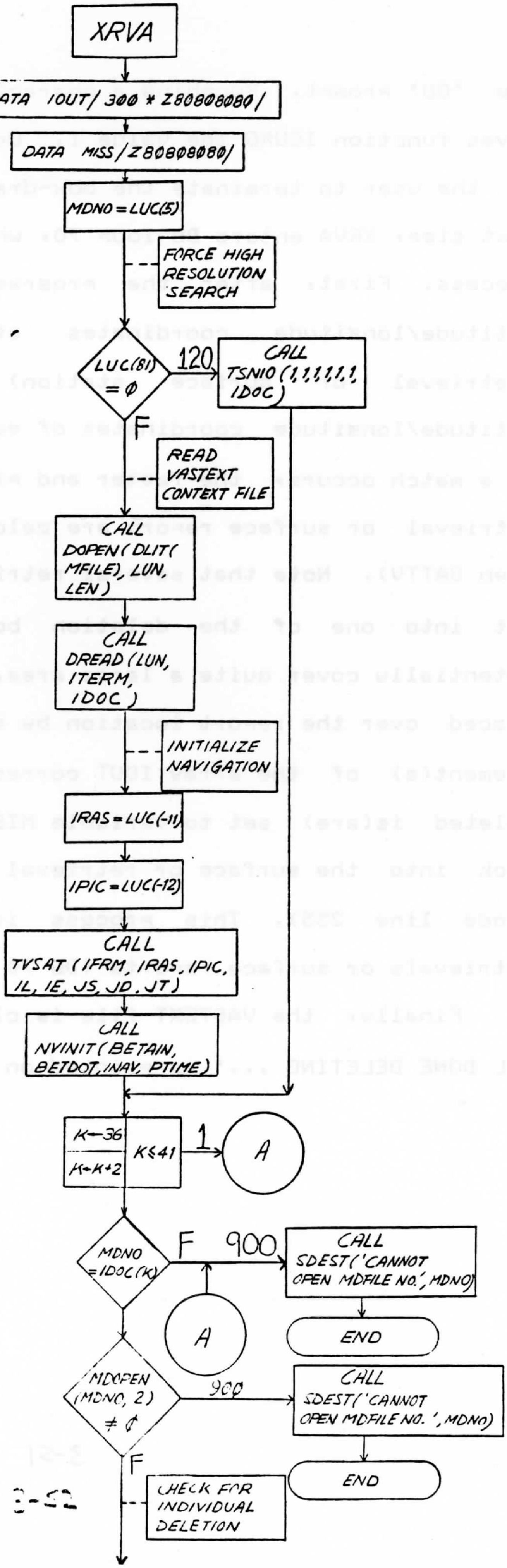
XRVA deletes values within either VAS or TOVS surface or retrieval MD files. If one is working on a surface data MD file (MDNS), the user can delete an entire report or reports contained within the cursor ,or he/she can delete only a particular parameter value for single or multiple reports within the cursor. The parameters can be either Z100 (1000 mb height), TSL (sea level temperature) or DD (station dewpoint depression). On the other hand, if he/she is working on a retrieval MD file, an entire retrieval can be deleted if keyword parameter NRET is keyed in by itself,or one or more retrievals contained within the cursor can be deleted if NRET is not keyed in. 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 that one must do an 'MDU SET' before the execution of XRVA to assure that the computer will be operating on the correct surface or retrieval MD file.

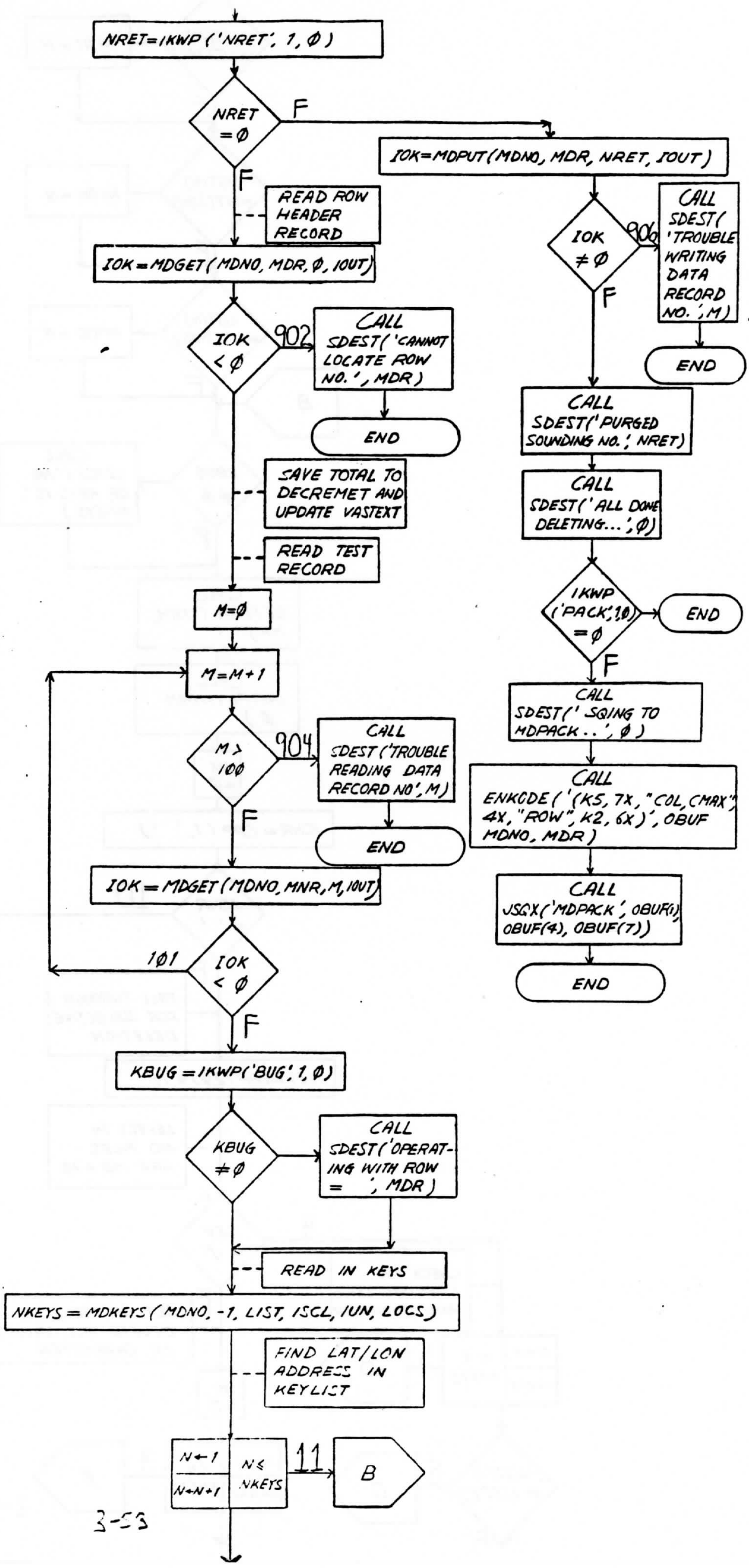
After the VASTEXT file is opened and read, and the cursor line/element position (IL,IE) is determined via function TVSAT, the navigation is initialized, and the current MD file listed in User Common (MDNG) is opened for read/write, provided that it matches either IDOC(36) or (40). Note that the MD file listed in User Common after the 'MDU SET' should not = IDOC(38), because IDOC(38) is the upper air guess MD file, which cannot be operated on by program XRVA. Therefore, even though IDOC(38) will satisfy DO-loop 1, it will cause problems later on. As a further note, IDOC(36) and (40) are the surface and retrieval MD file numbers, respectively, from the VASTEXT file. Then, XRVA checks to see if an entire individual retrieval should be deleted in code line 74. If this is not the case (NRET .EQ. 0), 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. Following this, the keys for that MD file are read in code line 97, and then XRVA checks to see what parameter should be deleted (variable ICHR) and what level (or all levels) it should be deleted from. If no character is specified, control passes directly to statement 10, and the process to delete all reports at all levels under the cursor begins.

Now, after the prompt "GO" is displayed on the CRT by function ICURG, the user should place the cursor over the report to be edited, and then push the spacebar, which will result in a box for the upcoming deletions being traced out around the outside dimensions of the cursor by subroutine WRBOX. Then, the latitude/longitude coordinates for the four corners of the 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 to be edited and push the spacebar again. As few as 1, or as many as 40 boxes can be traced out on the video screen (exact number stored in variable NX).

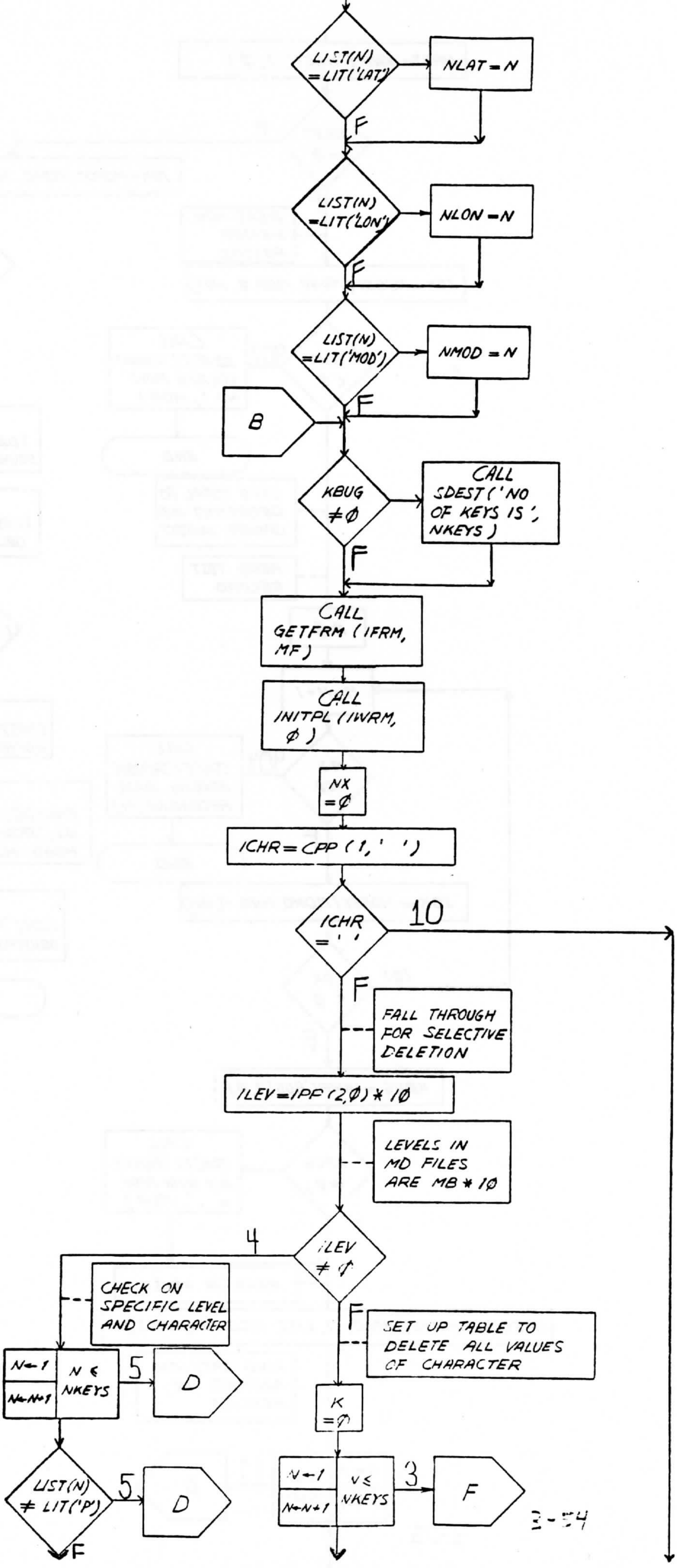
The deletion stage of the program begins either when NX=40 or something other than the terminal's spacebar is punched after the "GO" prompt. Punching a character other than the spacebar gives function ICURG the value 1. Usually, "R END" will be typed by the user to terminate the box-drawing stage of XRVA. In the next step, XRVA enters DO-loop 70, which carries out the deletion process. First, after the program enters DO-loop 40, the latitude/longitude coordinates of a given MD file entry (retrieval or surface station) are compared with the latitude/longitude coordinates of each of the NX plotted boxes. If a match occurs, the raster and pixel (TV coordinates) of the retrieval or surface report are calculated (subroutines SATEAR, then SATTV). Note that several retrievals or surface reports may fit into one of the deletion boxes, since the boxes can potentially cover quite a large area. In addition, a red "X" is placed over the report location by subroutine WRMAR. Then, the element(s) of the array IOUT corresponding to the data to be deleted is(are) set to variable MISS, after which IOUT is put back into the surface or retrieval MD file via function MDPUT (code line 255). This process is undertaken for all MREC retrievals or surface reports (DO 70 M=1,MREC).

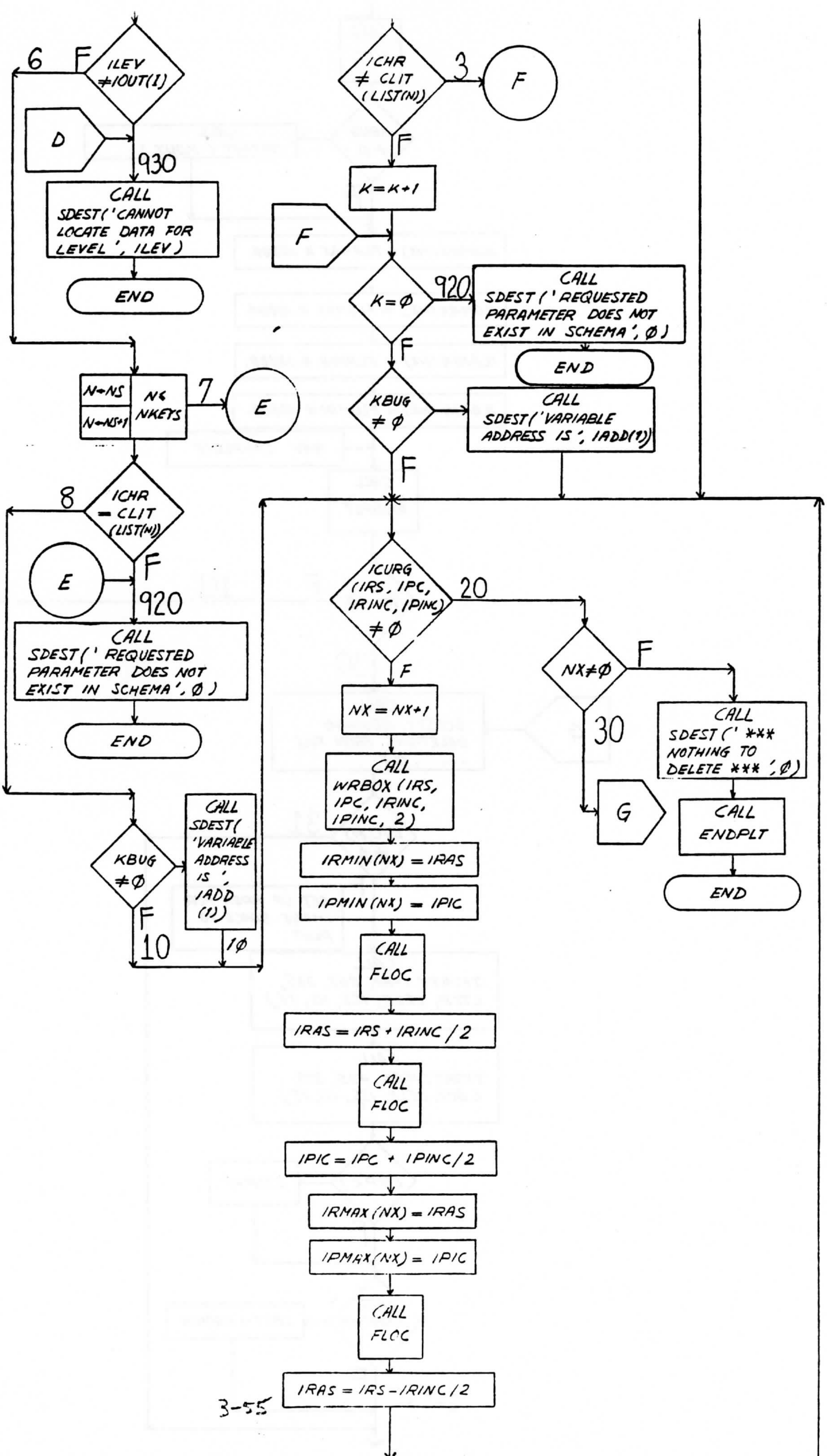
Finally, the VASTEXT file is closed, and a message saying " ALL DONE DELETING ..." is printed on the CRT.

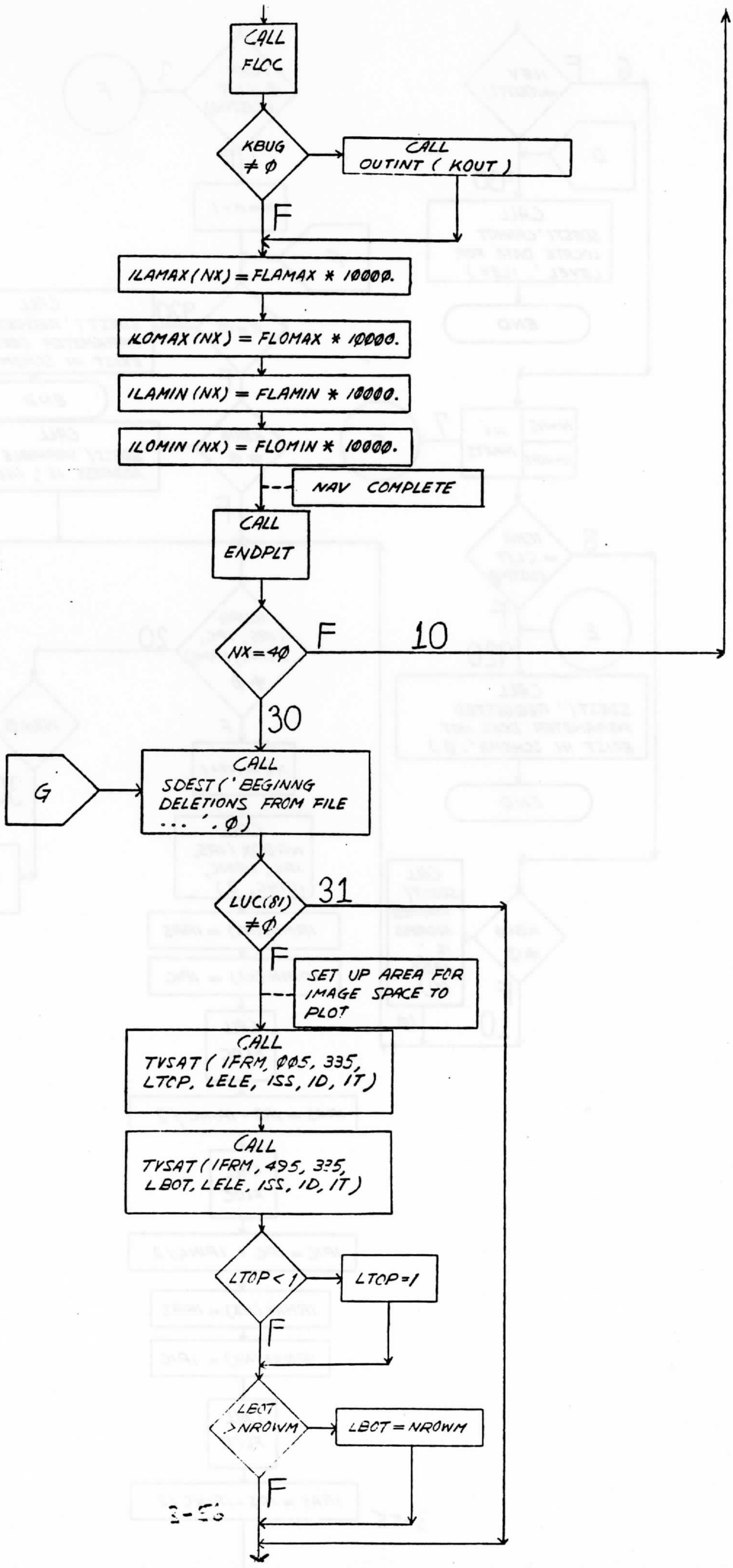

XRVA
DATA IOUT/ 300 * Z80808080/
DATA MISS/Z80808080/
MDNO = LUC(5)
FORCE HIGH RESOLUTION SEARCH
LUC(81) = Ø
120
CALL TSNIO (1,1,1,1,1,1, IDOC)
F
READ VASTEXT CONTEXT FILE
CALL DOPEN (DLIT(MFILE), LUN, LEN)
CALL DREAD (LUN, ITERM, IDOC)
INITIALIZE NAVIGATION
IRAS = LUC(-11)
IPIC = LUC(-12)
CALL TVSAT (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
A
MDOPEN (MDNO, 2) ≠ Ø
900
CALL SDEST('CANNOT OPEN MDFILE NO.', MDNO)
END
F
CHECK FOR INDIVIDUAL DELETION

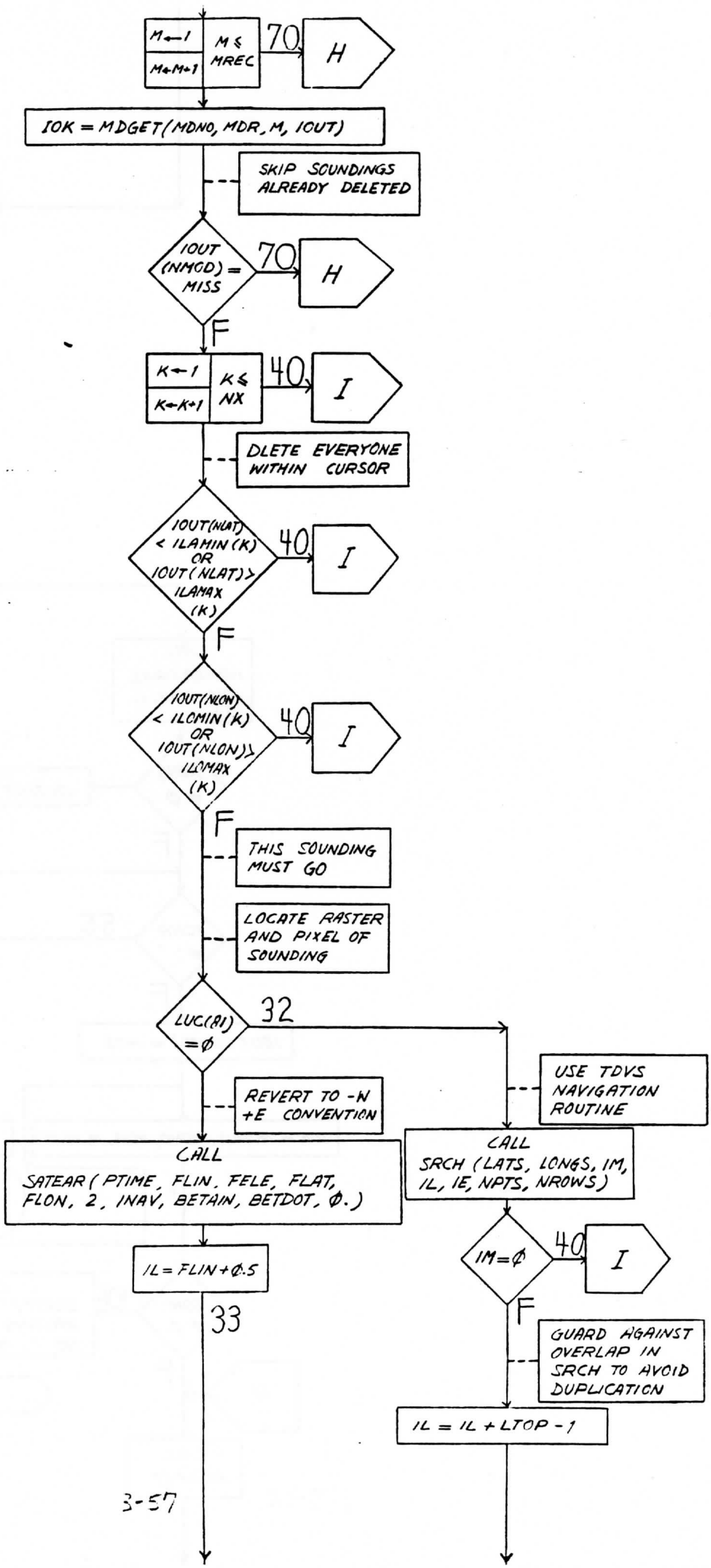


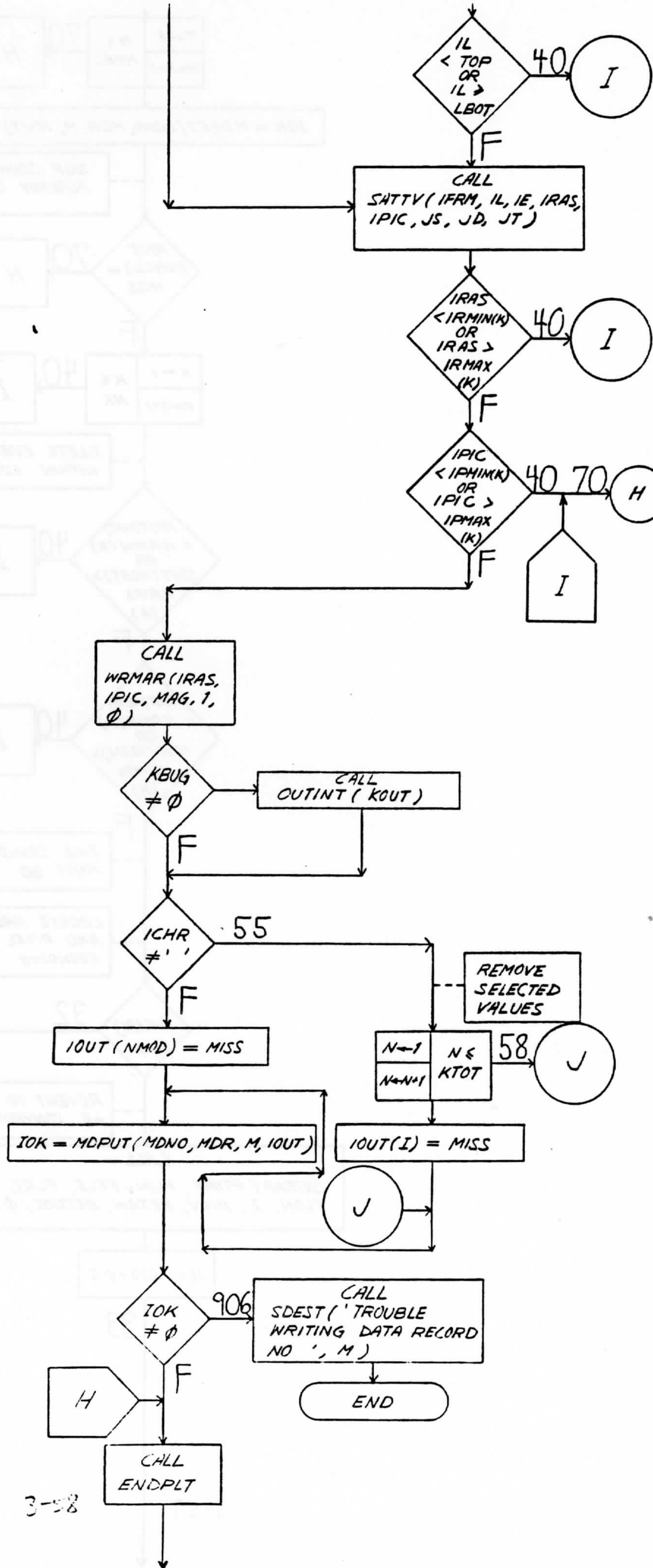
NRET=IKWP('NRET', 1, Ø)
NRET = Ø
F
F
IOK=MDPUT(MDNO, MDR, NRET, IOUT)
READ ROW HEADER RECORD
IOK=MDGET(MDNO, MDR, Ø, IOUT)
IOK ≠ Ø
906
CALL SDEST('TROUBLE WRITING DATA RECORD NO.', M)
F
END
IOK < Ø
902
CALL SDEST('CANNOT LOCATE ROW NO.', MDR)
END
CALL SDEST('PURGED SOUNDING NO.', NRET)
SAVE TOTAL TO DECREMET AND UPDATE VASTEXT
CALL SDEST('ALL DONE DELETING...', Ø)
READ TEST RECORD
IKWP ('PACK', 1, Ø) = Ø
END
M=Ø
F
M=M+1
CALL SDEST(' SQING TO MDPACK..', Ø)
M > 100
904
CALL SDEST('TROUBLE READING DATA RECORD NO', M)
CALL ENKODE('(K5, 7X, "COL, CMAX", 4X, "ROW", K2, 6X)', OBUF MDNO, MDR)
F
END
IOK=MDGET(MDNO, MNR, M, IOUT)
CALL JSQX('MDPACK', OBUF(1), OBUF(4), OBUF(7))
101
IOK < Ø
END
F
KBUG=IKWP('BUG', 1, Ø)
KBUG ≠ Ø
CALL SDEST('OPERATING WITH ROW = ', MDR)
READ IN KEYS
NKEYS=MDKEYS(MDNO, -1, LIST, ISCL, IUN, LOCS)
FIND LAT/LON ADDRESS IN KEYLIST
N←1
N←N+1
N ≤ NKEYS
11
B

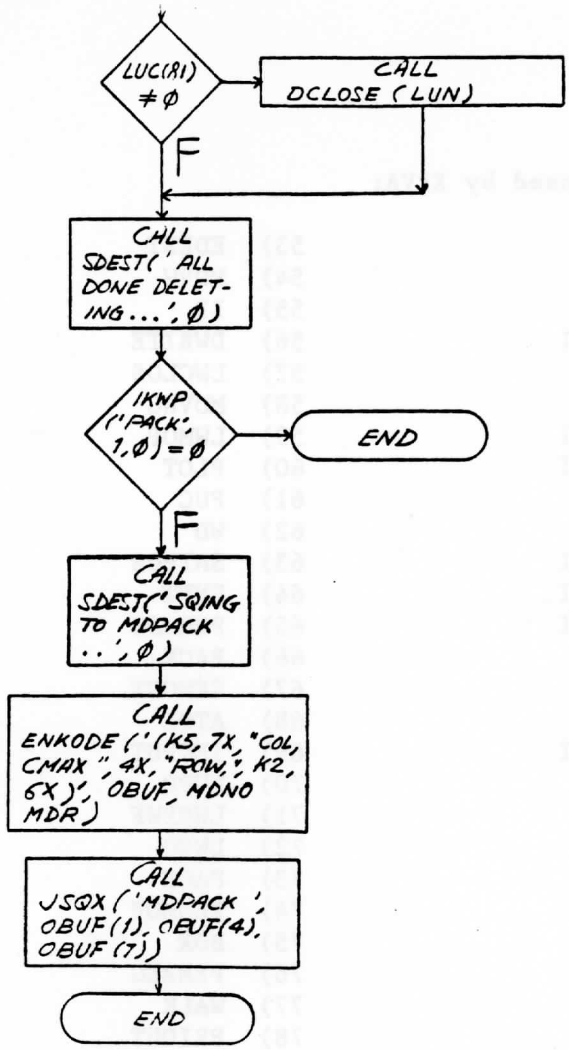

LIST(N) =LIT('LAT')
NLAT = N
F
LIST(N) =LIT('LON')
NLON = N
F
LIST(N) =LIT('MOD')
NMOD = N
B
F
KBUG ≠ Ø
CALL SDEST('NO OF KEYS IS', NKEYS)
F
CALL GETFRM (IFRM, MF)
CALL INITPL (IWRM, Ø)
NX =Ø
ICHR = CPP (1, ' ')
ICHR = ' '
10
F
FALL THROUGH FOR SELECTIVE DELETION
ILEV = IPF (2,Ø) * 1Ø
LEVELS IN MD FILES ARE MB * 1Ø
ILEV ≠ Ø
4
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
1Ø
ICHR ≠ CLIT (LIST(N))
3
F
F
K=K+1
F
K=Ø
920
CALL SDEST('REQUESTED PARAMETER DOES NOT EXIST IN SCHEMA', Ø)
F
END
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
NX≠Ø
F
30
G
CALL SDEST(' *** NOTHING TO DELETE ***', Ø)
CALL ENDPLT
END


CALL FLOC
KBUG ≠ Ø
CALL OUTINT (KOUT)
F
ILAMAX (NX) = FLAMAX * 10000.
ILOMAX (NX) = FLOMAX * 10000.
ILAMIN (NX) = FLAMIN * 10000.
ILOMIN (NX) = FLOMIN * 10000.
NAV COMPLETE
CALL ENDPLT
NX=4Ø
F
10
30
G
CALL SDEST ('BEGINNG DELETIONS FROM FILE ...', Ø)
LUC(81) ≠ Ø
31
F
SET UP AREA FOR IMAGE SPACE TO PLOT
CALL TVSAT (IFRM, ØØ5, 335, LTOP, LELE, ISS, ID, IT)
CALL TVSAT (IFRM, 495, 335, LBOT, LELE, ISS, ID, IT)
LTOP < 1
LTOP=1
F
LBOT >NROWM
LBOT = NROWM
F


M←1
M←M+1
M ≤ MREC
70
H
IOK = MDGET(MDNO, MDR, M, IOUT)
SKIP SOUNDINGS ALREADY DELETED
IOUT (NMOD) = MISS
70
H
F
K←1
K←K+1
K ≤ NX
40
I
DLETE 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
CALL SATEAR (PTIME, FLIN, FELE, FLAT, FLON, 2, INAV, BETAIN, BETDOT, Ø.)
IL = FLIN + Ø.5
33
USE TDVS NAVIGATION ROUTINE
CALL SRCH (LATS, LONGS, IM, IL, IE, NPTS, NROWS)
IM = Ø
40
I
F
GUARD AGAINST OVERLAP IN SRCH TO AVOID DUPLICATION
IL = IL + LTOP - 1


IL < TOP OR IL > LBOT
40
I
F
CALL SATTV(IFRM, IL, IE, IRAS, IPIC, JS, JD, JT)
IRAS < IRMIN(K) OR IRAS > IRMAX (K)
40
I
F
IPIC < IPMIN(K) OR IPIC > IPMAX (K)
40
70
H
I
F
CALL WRMAR(IRAS, IPIC, MAG, 1, Ø)
KBUG ≠ Ø
CALL OUTINT (KOUT)
F
ICHR ≠' '
55
REMOVE SELECTED VALUES
F
IOUT (NMOD) = MISS
N←1
N←N+1
N ≤ KTOT
58
J
IOK = MDPUT(MDNO, MDR, M, IOUT)
IOUT(I) = MISS
J
IOK ≠ Ø
906
CALL SDEST('TROUBLE WRITING DATA RECORD NO ', M)
F
END
H
CALL ENDPLT


LUC(RI) ≠ Ø
CALL DCLOSE (LUN)
F
CALL SDEST(' ALL DONE DELETING...', Ø)
IKWP ('PACK', 1, Ø) = Ø
END
F
CALL SDEST(' SQING TO MDPACK ..', Ø)
CALL ENKODE ('(K5, 7X, "COL, CMAX", 4X, "ROW," K2, 6X)', OBUF, MDNO MDR)
CALL USQX ('MDPACK', OBUF(1), OBUF(4), OBUF(7))
END

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) ENCCLR
23) LTQ
24) DECONV
25) FECONV
26) IECONV
27) ZECONV
28) MOVB
29) CLEANA
30) TO
31) PRLINX
32) PRCLOS
33) PROPEN
34) PRPRPR
35) LOCK
36) PRWR
37) PRRD
38) UNLOCK
39) PRCL
40) POST
41) BLKA
42) TRMNL
43) TQMES
44) STC
45) ITOC
46) GETNAV
47) EPOCH
48) GETGAM
49) VASNAV
50) DDEST
51) MOVCW
52) CLEANW
53) EDEST
54) MOVW
55) II
56) DWRITE
57) LWCLOS
58) MOVWC
59) LWMOP
60) PLOT
61) PUC
62) WD
63) SATPOS
64) ENPT
65) PENADD
66) PACK
67) SENOUT
68) ATOE
69) TEKPUT
70) LWPO
71) LWNEWF
72) LWSO
73) PAGE
74) PENMOV
75) BOX
76) PENBEG
77) WALK
78) PRIOUT
79) TQSET
80) JMBWTF

SRAD

This program prepares VAS brightness temperatures (TBBs) for SRET, the VAS retrieval program. It should be executed with the positional parameter "GO" and suitable keywords. (The option of running SRAD with the positional parameter "CLE" should be used only if the user wishes to update IDOC(50-60) in the VASTEXT file with the values seen in code lines 87-89. It normally will not be used.) In addition, SRAD can be keyed in to process either an entire sounder area or an individual retrieval. Note that my discussion of SRAD will concern itself primarily with the non-"AUTO" option; any deviations from this format will be duly noted. Note also that I will occasionally interchange radiance and TBB. The two, of course, can be related through Planck's Equation.

After the VASTEXT file and the row header of the retrieval MD file have been read, the succeeding lines of code (after "100 CONTINUE") define the area of retrievals to be processed (determined from IDOC (25) and (26)). Then, several other parameters, such as retrieval type (CTYP), surface data option (NOSFC), retrieval line and element spacing (in fields of view (FOVs)/soundings), retrieval size (in FOVs/soundings, ranging from 3X3 to 11X11), etc. are set approximately from code lines 103-131. Of particular importance is the keyword parameter defining the retrieval type. Note also the debug option (KBUG). If it is in effect (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). In addition, note the plot option (code line 142) and the setting of variables MDNG and MDNR in code lines 149-150.

At this point, the cursor TV coordinates (raster, pixel) are accessed from User Common via function LUC, and 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) should be keyed in using keyword parameter "END". The initial positions (NW corner of image) normally will not be keyed in and will thus be determined from the cursor position via subroutine TVSAT in code line 171. The cursor should normally be positioned near the NW corner of the image before SRAD is run. Note also the call to VRTIO in line 168. This call will result in an update to the VASTEXT file, based on the changes made to this point in the array IDOC.

In addition, if the "AUTO" option is on (IKWP('AUTO',1,0)

gleaned from the VASTEXT file. This option is used to speed up the data processing, and must also be used in program IDVA if it is used here.

Returning to the non-"AUTO" mode again, if the last line and element positions have not been keyed in (LLINE,LELEM = 0), SRAD sets the SE and NW line and element positions equal to each other, which will result in only one retrieval being performed at the cursor location. At this point, subroutine VASDAT accesses navigation data, including information such as image line and element resolution. Then, after the line and element spacings between each retrieval have been determined (INCIL and INCIE, respectively), subroutine GETFRM returns frame information (including the image magnification), and SRAD begins the actual

TBB processing. The processing is done line by line (DO LOOP 2400), moving successively down each line retrieval by retrieval (see statement 230: IEP=IEP+INCIE). When the retrieval element variable (IEP) exceeds the last retrieval element for a given line (LELEM), the row header is rewritten, the VASTEXT file is updated, and the next image line is processed (unless LASRET=MAXRET, in which case SRAD terminates). After this point, the code up to line 231 primarily initializes several of the arrays and variables to be used in the subsequent radiance/TBB processing.

At code line 235, SRAD enters an implicit DO-loop which runs up to line 281, within which VAS brightness temperature data is accessed for the initial retrieval box, which is usually 5X5 FOVs in size. In addition, if the first field of view's TBB data is being accessed, some tests are performed to make sure VASDAT is accessing data from a reasonable location. For instance, if the latitude of the FOV is north or south of the previously-set sounder area boundaries (LN,LS), the program will move to the next 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 increment to the right and tries again), or longitude less than the eastern boundary (in which case SRAD behaves as though IEP has exceeded the last retrieval element for a given line). In addition, if the satellite zenith angle (VZEN) is greater than 60 degrees, SRAD will move one-half retrieval increment to the right and attempt to access another set of retrieval radiances (am assuming LBEG = 0 here). If there are no problems, the TBB information will be collected for each band and sounding location in the retrieval box and stored in the array RADS (N,I) in DO-loop 400. DO-loop 400 is also used to update the TBB array VDAT if any gross TBB errors exist, and it stores the spin budget for each band in the array NSPIN as well. The implicit DO-loop will be exited when variable IBOX is set to 3 in VASDAT, which will occur when all the radiance information for that retrieval box has been accessed.

Following the implicit DO-loop, the average surface elevation, latitude and longitude for the retrieval box are stored in the variables SELEV, SLAT and SLON, respectively. Then, a call to subroutine SURGES returns surface information, including variable ITSFC, which is then used to calculate the retrieval surface temperature (stored in variable TSTA). In addition, after TMN is determined, the total number of soundings with radiance data for a given band (k) in the retrieval box, as well as the sum of the TBBs for all soundings in the retrieval box for a given band, are stored in the arrays NSAM (or IUSE) and AVG, respectively. At this point, a quality control check involving the band 8 brightness temperature data checks to see that 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 increment to the right and try again. Then, DO LOOP 590 will set IUSE(k)=0 for any band "k" other than band 8 that does not have at least 3 TBB observations within the retrieval box.

Up to this point (code line 321), SRAD has run the same for all five retrieval options (F (forced), N (N-star), C (clear), O (overcast) or default (B, or no "TYP" keyin). Now, however, different options will require separate program discussions. Therefore, from here on, I will discuss SRAD in terms of a given retrieval option.

If the retrieval type is N-star (option N), SRAD skips to statement 770, and clear column radiances are inferred mathematically in each band for the retrieval box from the mixture of clear, partly cloudy and cloudy radiances by subroutine NSTAR (MacMillin, 1978). Failure in NSTAR results in a new retrieval attempt one-half retrieval increment to the right of the failed attempt. Otherwise, if subroutine NSTAR is successful, SRAD enters DO-loop 830, where the expected error of the brightness temperature observations for each band (array EX) are computed. After this DO-loop, the surface skin temperature (TS) is calculated. Then, the output buffer (IRET...note equivalence to IRETD in code line 54) containing, among other things, the final brightness temperature and expected error of the brightness temperature observations for each band, is filled, and the results are written into column LASRET of the retrieval MD file by subroutine VRTIO. Then, if the plot option is on, the band 8 TBB is plotted on the video screen at the retrieval location in KOLOR=3 (normally yellow) via subroutine VASDIG. For any successful attempt, SRAD moves one retrieval increment to the right to process another set of TBBs, unless either of three things happens. First, if the just-completed retrieval equals the maximum number of retrievals allowed (MAXRET), the retrieval MD file row header is written, the VASTEXT file is updated, and SRAD terminates. Second, if the process of incrementing to the right after a successful retrieval exceeds the rightmost extent within the sounder area for which retrievals can be processed (LELEM), the row header is written, the VASTEXT file is updated, and SRAD moves to the next line of TBB data to be processed. Finally, if the previous retrieval was the last one possible in the sounder area, SRAD concludes in a similar fashion to the above MAXRET case.

Returning to code line 321, if the retrieval type is forced, clear, overcast or default (options F,C,O or B/no "TYP" keyin, respectively), the maximum brightness temperature and sounding/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 sounding number(s) which show(s) the maximum TBB for these bands. At this point, if the maximum band 8 TBB is less than a threshold temperature TMN, which was set previously in code lines 304-305, or if JSAV or JSAV6 = 0, SRAD moves one-half retrieval increment to the right and attempts a new retrieval.

If the threshold test passes, and 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 FOVs in a given retrieval box. In addition, the arrays NSAM and AVG are re-calculated with the clearest radiance information. DO-loop 720 tests for possible deletions of a band measurement due to cloud contamination for a given sounding/FOV, while DO-loop 700 (nested within 720) updates the NSAM and AVG arrays. Following this, the average retrieval brightness temperature for each band is calculated in DO LOOP 760 and placed in the array VDAT. Then, if the retrieval type is F, C or O, the expected error of the brightness temperature observations for each band is computed, as well as the surface skin temperature. 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.

The default type retrieval path differs from the F,C or O retrievals in only one way. However, it is an important difference. Returning to code line 394, if there are five or more sufficiently clear band 8 FOVs/soundings in the retrieval box, the retrieval process proceeds identically to the F,C or O paths. However, if there are less than five sufficiently clear band 8 FOVs, the retrieval path becomes N-star, and clear column radiances will be inferred via subroutine NSTAR. Then, from code line 400 on, SRAD proceeds identically to the previously-discussed N-star path. In other words, the default retrieval path is identical to the forced, clear or overcast paths, so long as there are at least 5 sufficiently clear band 8 FOVs within the retrieval box, but default retrievals will become N-star retrievals if there are less than 5 sufficiently clear band 8 FOVs. A summary of the VAS retrieval method is contained in Smith (1983).

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) ENCCLR
18) LTQ
19) DECONV
20) FECONV
21) IECONV
22) ZECONV
23) MOVB
24) CLEANA
25) TQ
26) PRLINX
27) PRCLOS
28) PROPEN
29) PRPRPR
30) LOCK
31) PRWR
32) PRRD
33) UNLOCK
34) PRCL
35) POST
36) BLKA
37) ABORT
38) DOPEN
39) ENCODE
40) DREAD
41) DCLOSE
42) LWCLOS
43) MOVWC
44) LWMOP
45) LWPO
46) EDEST
47) MOVW
48) MOVCW
49) CLEANW
50) II
51) STC
52) DWRITE
53) TRMNL
54) TQMES
55) ITOC
56) NVINIT
57) GETNAV
58) EPOCH
59) GETGAM
60) VASNAV
61) DDEST
62) READD
63) OUTINT
64) PLNKIV
65) OPNA
66) LWOPEN
67) LWGET
68) SATEAR
69) SATPOS
70) ANGGET
71) SOLARP
72) ANGLES
73) READOF
74) RBYTSX
75) ZEROS
76) RDTRK
77) MOVC
78) HRTOPO
79) IGNAME
80) RORDER
81) FILCLR
82) FILCLD
83) INITPL
84) QGDASH
85) DSHOFF
86) PLOT
87) DSHON
88) ENDPLT
89) ENPT
90) PENADD
91) PACK
92) SENOUT
93) ATOE
94) TEKPUT
95) LWNEWF
96) LWSO
97) PAGE
98) PENMOV
99) BOX
100) PENBEG
101) WALK
102) PRIOUT
103) TQSET
104) PUC
105) WD
106) JMBWTF
107) GETSFV

SRET uses the VAS brightness temperature (TBB) measurements prepared by SRAD and retrieves vertical profiles of temperature and moisture.

Initially, after the VASTEXT file and retrieval MD file row header have been read, important parameters such as the type of guess (CGES), surface analysis option (NOSFC), retrieval type (forced, default (no kevin), clear, overcast or N-star), etc. are determined. 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 plot option (IPLT), enhancement option (IENH), and first and last retrievals to be processed (NB,LASRET). Note that if the plot option is on (IPLT .NE. 0), the cursor location in lines and elements is determined, the navigation is initialized, and frame information is accessed. This is done in code lines 122-132.

Then, the main retrieval DO-loop (2400) is entered. This loops 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 processed brightness temperatures for each band. DO-loop 700 stores these values in the arrays EX and VDAT, respectively. In addition, array VRAD stores computed Planck radiances for each band. Now, the guess temperature and moisture profiles for the retrieval are accessed via subroutine GESPRO in code line 179. If the surface temperature (TSTA) is more than 15 degrees warmer than variable TCK (approximately equal to the skin temperature), IRETD(3) is set to 7777, array IRET is put back into the retrieval MD file, and SRET moves to process the next retrieval. Otherwise, if this test passes, the first pressure level below the surface is calculated (variable IS...maximum value = 40), and the lower portion of the guess temperature profile is adjusted to take into account the surface temperature. Then, after some guess temperature and dewpoint data is prepared for the retrieval MD file in the arrays TGS and DGS, the surface mixing ratio (WSTA) is calculated in code line 231. Code lines 241-258 set a cloud flag (IFLG), based on some tests involving band 12 TBB, skin temperature (TS), etc. The tests are used to determine if a cloud contamination problem exists. Concurrently, the tests also determine if the surface skin temperature calculated by SRAD will be believed, as well as whether the temperature enhancement will be done during the subsequent retrieval process. The first check, which is only done if IUSE(12) =1, compares the band 12 TBB value, which can be contaminated by excessive reflected sunlight, with the skin temperature. If large enough differences are detected, IFLG will be henceforth equal to 99, and the retrieval will be classified as overcast. The next check compares the skin temperature to a threshold variable TMN. Again, if a sufficiently large difference exists, IFLG remains equal to 99. If both tests pass, IFLG is set to 0, and the retrieval will be classified as clear. Now, for cases where IFLG = 99, the pressure level immediately below the first level colder than the surface skin temperature (LS) is calculated. Then, statement 980 calculates variable NLS, which defines the lower boundary level for calculating radiance via the radiative transfer equation (RTE) in the following relative humidity (RH) calculations (DO-loop 1100). Note that for cases where IFLG = 99, NLS will be the minimum of the first level below the surface (IS) and the just calculated variable LS, while for cases where IFLG = 0, NLS will be the minimum of the first level below the surface and the number 40.

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 is partially filled, resulting in values for the top 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 results are stored in the array IHUM. In addition, the final RH value returned for each band layer will be between (or equal to) .10 and 1 (10-100%). Next, after the operations in code lines 323-338 are executed (note especially code line 323), DO-loop 1300 is entered, in which the original guess mixing ratio profile for the lower 15 atmospheric levels (300 to 1000 mb) is updated by linear interpolation, using the band 7, 9, 10 and surface RH values just calculated. After this loop has been completed, the transmittances for bands 7-10 are determined, along with the total precipitable water vapor for the updated mixing ratio profile.

At this point, SRET begins the actual temperature retrieval process. First, the number of possible iterations (ITLIM) is set to 3 in the usual case, but it will be 10 for a climatology first guess. Then, note that if IFLG .NE. 0 and the retrieval will be using a climatology first guess (code line 385), IRETD(3) is set to 7777, array IRET is put back into the retrieval MD file, and SRET moves on to the next retrieval. If this is not the case, DO-loop 1380 sets the values of variables KC1 and KC2, depending on whether a given band 1-6 (CO2 bands) is to be used for the forthcoming temperature retrieval (note the value of element I or J of array IUSE). If 5 or more of the CO2 bands are not being used, SRET will again transfer to statement 2200.

Otherwise, assuming a sufficient number of CO2 bands are being used, SRET enters DO-loop 1520 (code line 400). In this loop, the transmittance for a given CO2 band is determined (through all 40 retrieval levels), after which the lowest retrieval level to be used for radiative transfer calculations (IA) is computed. This computation is only done once for each retrieval. After this, subroutine VASRTE calculates 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 can be seen in code lines 447-448. The final step in loop 1520 involves the calculation of the difference between the satellite- measured brightness temperature and the brightness temperature calculated from the guess profile (array DELT). This information will eventually be used when evaluating the iterative form of the radiative transfer equation used to retrieve the temperature profile (Smith, 1983).

At this point, several data quality checks are performed (only for the first temperature iteration) to further determine if cloud contamination is a problem. The tests involve variables determined from the above-mentioned TBB differences (AB5, AB6, etc.). Note that these tests will only be performed if IUSE(5) and (6) are .NE. 0. If AB6 .GE. AB5 and AB4, band 6 will no longer be used (IUSE(6)=0). In addition, if the absolute value of variable CCK is .GT. 2.5, and RHCK .GT. 0.95, a cloud problem is indicated, and IFLG is reset to 1. Then, if the "cloud flag" IFLG still has a non-zero value, DO-loop 1560 checks suspected cloudy retrievals and deletes those bands more than 0.25 degrees Celsius cooler than the guess temperature profile value of TBB. If the band 4 brightness temperature data is deleted during this procedure (KUSE(4) .EQ. 0), the same original SRAD data read in via VRTIO in code line 147 is placed back in column NN of the retrieval MD file, and the program returns to attempt the next retrieval. Assuming the band 4 data is acceptable, SRET skips to statement 1670 (remember, IPASS .EQ. 0 at this point).

Now, since SRET is on the initial pass, the program enters DO-loop 1720, whose purpose is to retrieve the actual temperature profile. Note that code line 554 has the same form as equation (3) in Smith (1983) and does the actual temperature calculations. The weighting function value for each CO2 band at a given pressure level "I" is calculated in line 548. Following the temperature retrieval, if the surface pressure is greater than 850 mb, the lower levels of the temperature profile 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 is set equal to TSTA, after which the mixing ratio, saturated mixing ratio, and dewpoint for each pressure level between 300 and 1000 mb is calculated and stored in the arrays W, S and TD, respectively in DO-loop 1860. Finally, a check is performed to see if the temperature profile satisfies the convergence criterion. The actual test occurs at code line 607. If the profile does satisfy the convergence criterion after the first temperature iteration, which is possible in unusual cases, the program skips to statement 1940. Otherwise, if the convergence criterion is not satisfied, SRET returns to statement 1390. Note that ITER is now equal to 1. Within DO-loop 1520 this second time, only radiances for the CO2 bands being used, as well as weighting functions and DELT values, are recomputed. No transmittance recalculations are done. Then, since ITER .GT. 0, the program skips to statement 1580 and then to 1670, after which a second 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. Then, the same steps as above are done again. The temperature iterations 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). At this point, after the last iteration has been performed, SRET skips to statement 1940 (whether the convergence criterion is satisfied or not), and level LS is redefined with the new retrieved temperature profile in code lines 617-627. Following this, transmittances for bands 7-10 are recalculated, 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 are determined once again. Then, IPASS is set equal to 1, and SRET once again returns to statement 1390. This concludes the first iterative pass of SRET.

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 weighting functions and DELT values are calculated again, and SRET skips to statement 1580. At this point, if the absolute value of the DELT value for any band is .GT. 2.5, the retrieval is aborted. Assuming this problem does not occur, a check is performed in DO-loop 1665 to see whether the difference between the observed and calculated brightness temperature for each CO2 band is within the expected noise of the brightness temperature observation for that band.

If all the bands pass this second convergence test, SRET moves to statement 1951, no temperature profile enhancement is done, and the post - retrieval process (calculating stability indices, heights, preparing the output array for insertion into the retrieval MD file, etc.) begins.

On the other hand, if any band fails the test, and IFLG .NE. 0, SRET moves to statement 1940, the moisture enhancement, precipitable water and dewpoint calculations are done one final time, and control transfers to statement 1951.

Finally, if there is a convergence failure and IFLG .EQ. 0, a temperature profile enhancement is attempted via subroutine VTRET. If the enhancement works (IOK set to +1 by VTRET), the lower portion of the temperature profile is again adjusted, and the mixing ratio, saturated mixing ratio and dewpoint arrays are all recalculated. IPASS is then incremented to 2, and the program moves once again to statement 1390. Then, DO-loop 1520 is executed again (minus the transmittance recalculations), after which the convergence test between DELT and the expected error of the brightness temperature observations (EX) is repeated (keep in mind the test at code line 518). Success in the test this second time or any subsequent time causes a transfer to 1951, while failure causes a second temperature enhancement attempt.

Success in the second temperature enhancement results in a repeat of the previously-mentioned low level adjustments and dewpoint calculations, and then yet another return to 1390 (IPASS=3 now). After the transmittance and DELT calculations have been done again, the convergence test in DO-loop 1665 is repeated. Failure causes a third and final temperature enhancement. Success in the third enhancement causes a transfer of control to statement 1730, and then another return to 1390 and subsequent transfer to 1940, while a failure at this late point causes ITLIM to be raised again, and a possible several additional temperature iterations to be performed. In any case, control will eventually pass to statement 1940 (the user should satisfy him/herself of this fact), and then to 1951.

Returning to the point where a second temperature enhancement attempt is made, if a failure in this attempt occurs, SRET will simply perform a third enhancement (IPASS set to 3 as well). If this third enhancement succeeds, the low level adjustments, mixing ratio, saturated mixing ratio and dewpoint calculations will be done again, control will transfer to 1390, and then eventually to 1940 again. But, if the enhancement fails yet another time, ITLIM is raised by 3, and more temperature iterations can be performed. Again, control will eventually pass to statement 1940 and then to 1951.

On the other hand, a failure in the first temperature profile enhancement (IOK = -1, IFLG still = -1) causes IPASS to be incremented to 2, and a subsequent transfer of control back to statement 1668, where another temperature enhancement is attempted. In situations when IPASS is incremented to 4 after the third failure of VTRET, ITLIM is raised by 3, and SRET returns to statement 1670, recalculates the temperature profile, readjusts the lower portion of the profile, recomputes mixing ratio, saturation mixing ratio and dewpoint, and checks to see if the convergence criterion after statement 1885 has been satisfied. If the criterion has been satisfied, control transfers to statement 1940. Otherwise, since ITLIM has been increased, 2 or more additional temperature iterations can now be performed (unless ABS(DELT(KC)) .GT. 2.5 for any band), until either the convergence criterion is satisfied, or ITER becomes .GT. ITLIM. In either case, control finally passes to statement 1940, and one last moisture enhancement is done, after which control passes to statement 1951.

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 is greater than GTST. If this check is passed, subroutine HTX calculates heights from the retrieval 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, etc. Next, the heights computed via subroutine HTX are read into 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, variable NOGO is set to 0, meaning a successful retrieval was performed, and SRET is ready to prepare an output array for insertion into the retrieval MD file.

The final stage of the retrieval process consists mainly of inserting the retrieval data into a buffer array IRETD, which is then inserted into column NN of the retrieval MD file via subroutine VRTIO. Some of the retrieval quantities placed in IRETD include total precipitable water, 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. placed into IRETD complete the output array.

Lastly, NRET is updated and, if the plot option is in effect, the total totals index will be plotted at the retrieval location in one of three colors, which are usually blue-green, yellow or magenta, corresponding to clear (KOLOR=2), N-star (partly cloudy, KOLOR=3) or overcast (KOLOR=1) retrievals, respectively. After this, the final data is put into the retrieval MD file by VRTIO, and SRET moves on to the next retrieval.

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) ENCCLR
28) LTQ
29) DECONV
30) FECONV
31) IECONV
32) ZECONV
33) MOVB
34) CLEANA
35) TQ
36) PRLINX
37) PRCLOS
38) PROPEN
39) PRPRPR
40) LOCK
41) PRWR
42) PRRD
43) UNLOCK
44) PRCL
45) POST
46) BLKA
47) ABORT
48) DOPEN
49) ENCODE
50) DREAD
51) DCLOSE
52) LWCLOS
53) MOVWC
54) LWMOP
55) LWPO
56) EDEST
57) MOVW
58) MOVCW
59) CLEANW
60) II
61) STC
62) DWRITE
63) TRMNL
64) STDATM
65) RAOBIN
66) CSRAOB
67) MNRAOB
68) TQMES
69) ITOC
70) GETNAV
71) EPOCH
72) GETGAM
73) VASNAV
74) DDEST
75) SGRAOB
76) CSRAOS
77) CSRAOM
78) CSRAOZ
79) CSRAOP
80) CSRAOR
81) CSRAOI
82) INTPTW
83) EXTEMP
84) CLMGES
85) VASGES
86) OUTINT
87) SURGES
88) GETSFV
89) IGNAME
90) PROFIX
91) PRETAV
92) PREATV
93) ULMR
94) CO2TAV
95) H2OTAV
96) CONTAV
97) O3TAV
98) GAMTAV
99) INITPL
100) OGDASH
101) DSHOFF
102) PLOT
103) DSHON
104) ENDPLT
105) ENPT
106) PENADD
107) PACK
108) SENOUT
109) ATOE
110) TEKPUT
111) LWNEWF
112) LWSO
113) PAGE
114) PENMOV
115) BOX
116) PENBEG
117) WALK
118) PRIOUT
119) SATPOS
120) PUC
121) WD
122) JMBWTF

PLVA

PLVA is used to plot either VAS retrieval or surface data on the video screen over the band 8 satellite image. Two different types of data can be displayed: a certain parameter or differences between two values of the same parameter at a given level, or differences between retrieval and first guess for a certain parameter at a given level.

Initially, data such as debug option, the VASTEXT file, etc. are read in. Then, the satellite coordinates (line, element) of the cursor are determined via subroutine TVSAT (code line 60), the navigation is initialized, and the coordinates of the image/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, E and W extent of the data to be plotted. Note that these latitude/longitude boundaries can also be included in the program keyin. 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 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 MD file MDNO 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 from code lines 120-129 to make sure data exists in MDNO. Note also that the maximum possible number of data entries is stored in variable MMAX in code line 123. (The array MDHD was filled via the previous call to function MDINFO in code line 108.) Following this, the keys for that particular MD file are read, which becomes vital when the actual data is read from the file.

Next, the first and second positional parameters are read in via the statements CCHR=CPP(1,'Z '), ILEV=IPP(2,0) * 10 and CLEV=CPP(2,' '). The information loaded into these three variables will determine both what type of data is to be plotted (Z, T, TDIF, brightness temperature, etc.), as well as how the data should be plotted (either values or differences at a level). The default values are Z and 500, meaning a 500 mb height plot will be outputted to the video screen if PLVA is keyed in by itself. In addition, variable IL2 stores the second level in case a thickness plot is desired (code line 158). Then, the data accessed previously by function MDKEYS is used in DO-loop 18 to find the location within the MD file output array IOUT of several variables, including latitude and longitude of a particular report. Next, the number of reports to be read (not necessarily plotted) is determined and placed in variable MREC.

The next major step involves the reading and plotting of the actual data, approximately from code lines 179-302. This process encompasses the vast majority of the remainder of PLVA. Initially, after the first report has been read in code line 186, there is an option to skip a given report if it has been previously rejected (IOUT(NMOD) .NE. 0), or plot all reports (regardless of previous rejection), in code line 189. Then, if the level (ILEV) has been filled, and the plot consists of a non-difference variable (T, Z, etc.), the elements of the array IOUT storing the correct level and character to be plotted are calculated in DO-loops 45 and 47. Then, assuming the retrieval is within the previously-defined plotting area, its line/element, and then raster/pixel (IRAS,IPIC) coordinates are determined in succession. Finally, the color the data will be plotted in, and the actual data plot itself, are carried out after code line 267. The subroutine which does the actual plotting is VASDIG.

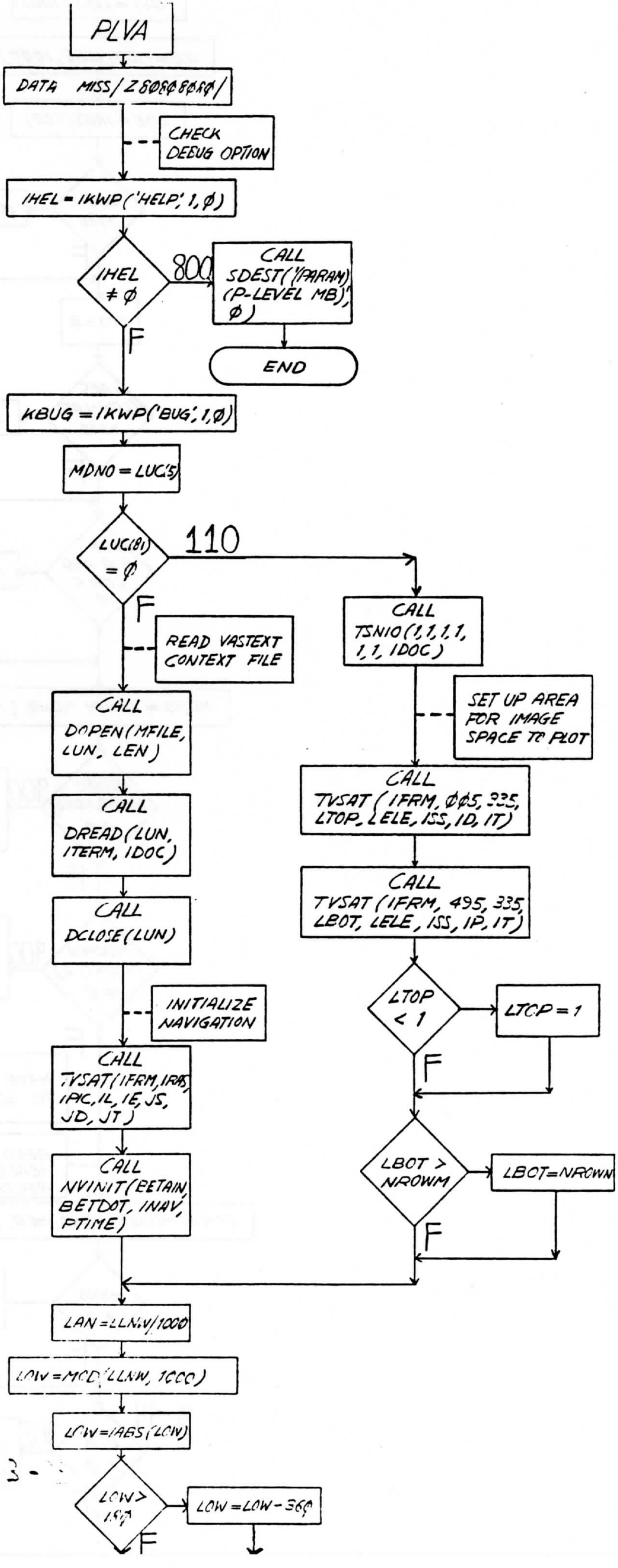
On the other hand, if we are plotting RH or brightness temperature values (no level set), the element of the array IOUT which stores the variable to be plotted is determined in DO-loop 43, after which the raster/pixel coordinates are again calculated. Then, the value is plotted, and PLVA moves to the next report.

If differences between brightness temperature values for the retrieval and guess are to be plotted (again no level set), the procedure is the same as the above paragraph, except a difference quantity is calculated in code line 291 and plotted, rather than a single value.

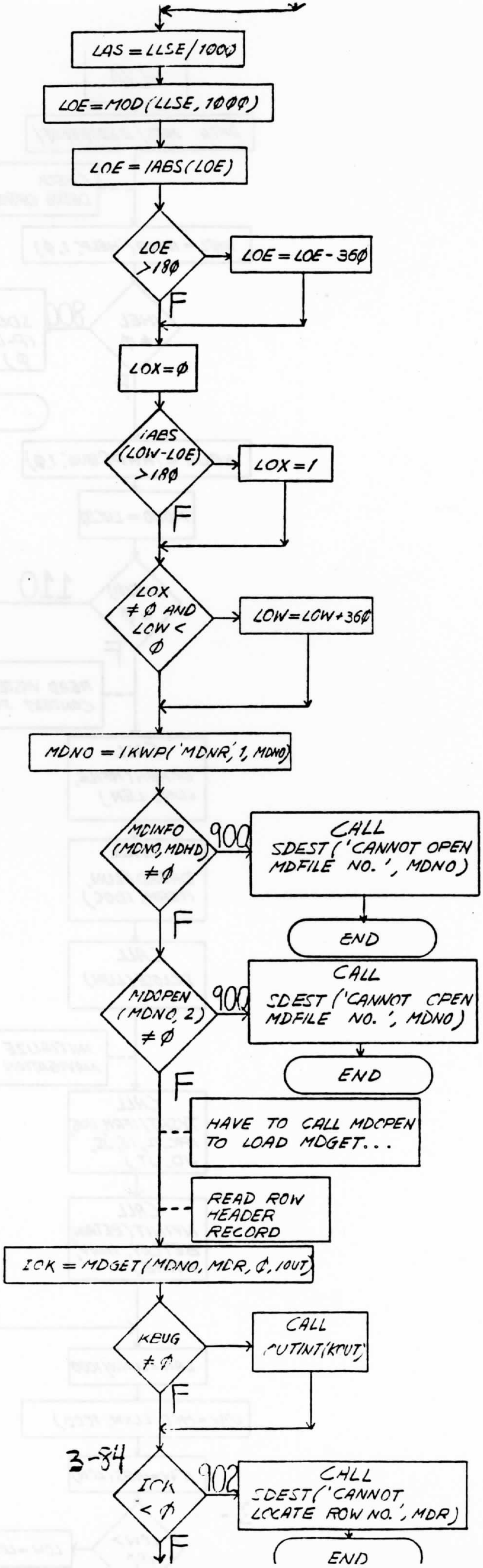
If thicknesses are to be plotted, variable IL2 will be given a value in code line 158. Then, the elements within IOUT containing the first (lower) level and character (Z) are determined in DO-loops 45 and 47, and the same process is repeated for the second level and character (Z again, of course) in DO-loops 52 and 54. Following this, the retrieval raster/pixel coordinates are calculated, and the thickness to be plotted (variable IDAT in code line 272) is displayed on the video screen again via subroutine VASDIG.

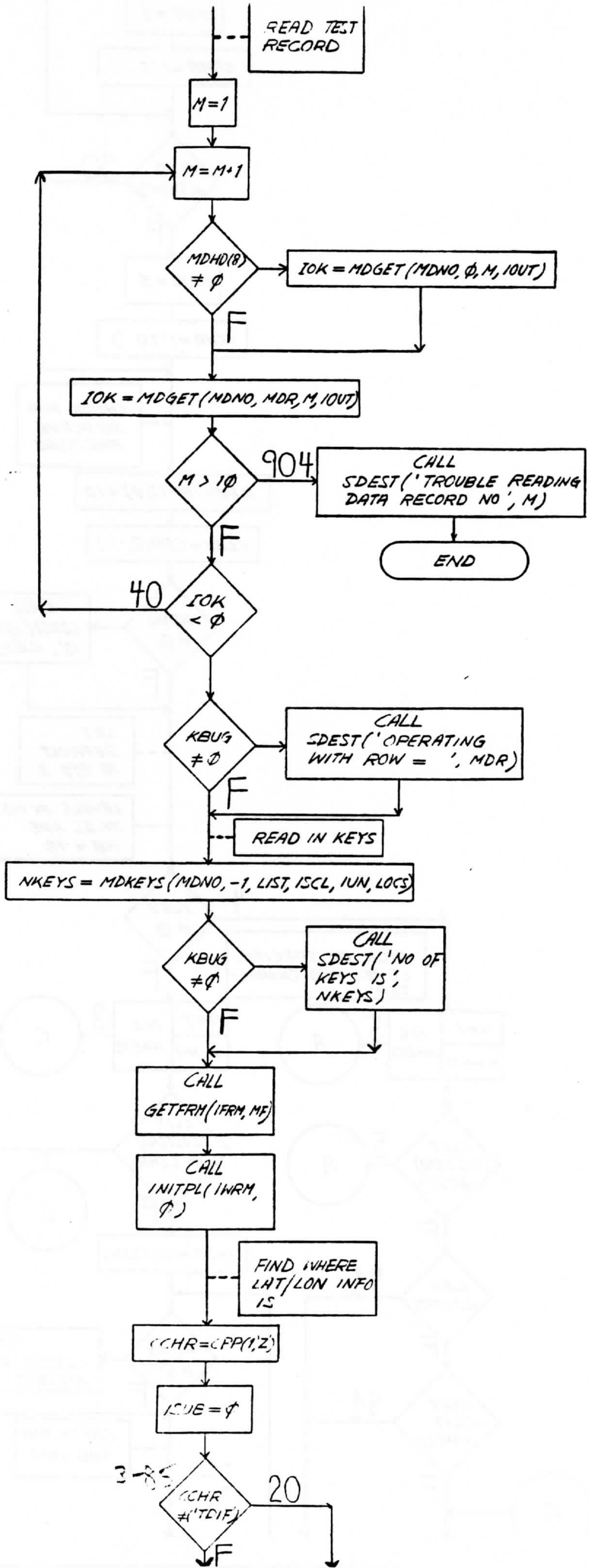
Finally, if difference plots between temperature or dewpoint for retrieval minus first guess are desired, the level at which the differences will be plotted is determined, the raster/pixel coordinates are again calculated, and the difference quantity is formed in code line 291. Then, the difference quantity is plotted by VASDIG. Note that it is also possible to plot winds at a given level with PLVA.

Essentially, the main point to remember here is that one has to worry only about the parameter itself (not the level) when plotting data dealing with RH or brightness temperature, whereas one must worry about both parameter AND level when plotting Z, T, TD, TDIF, etc. Usually, the nature of the variable will enable one to deduce whether a level is involved.

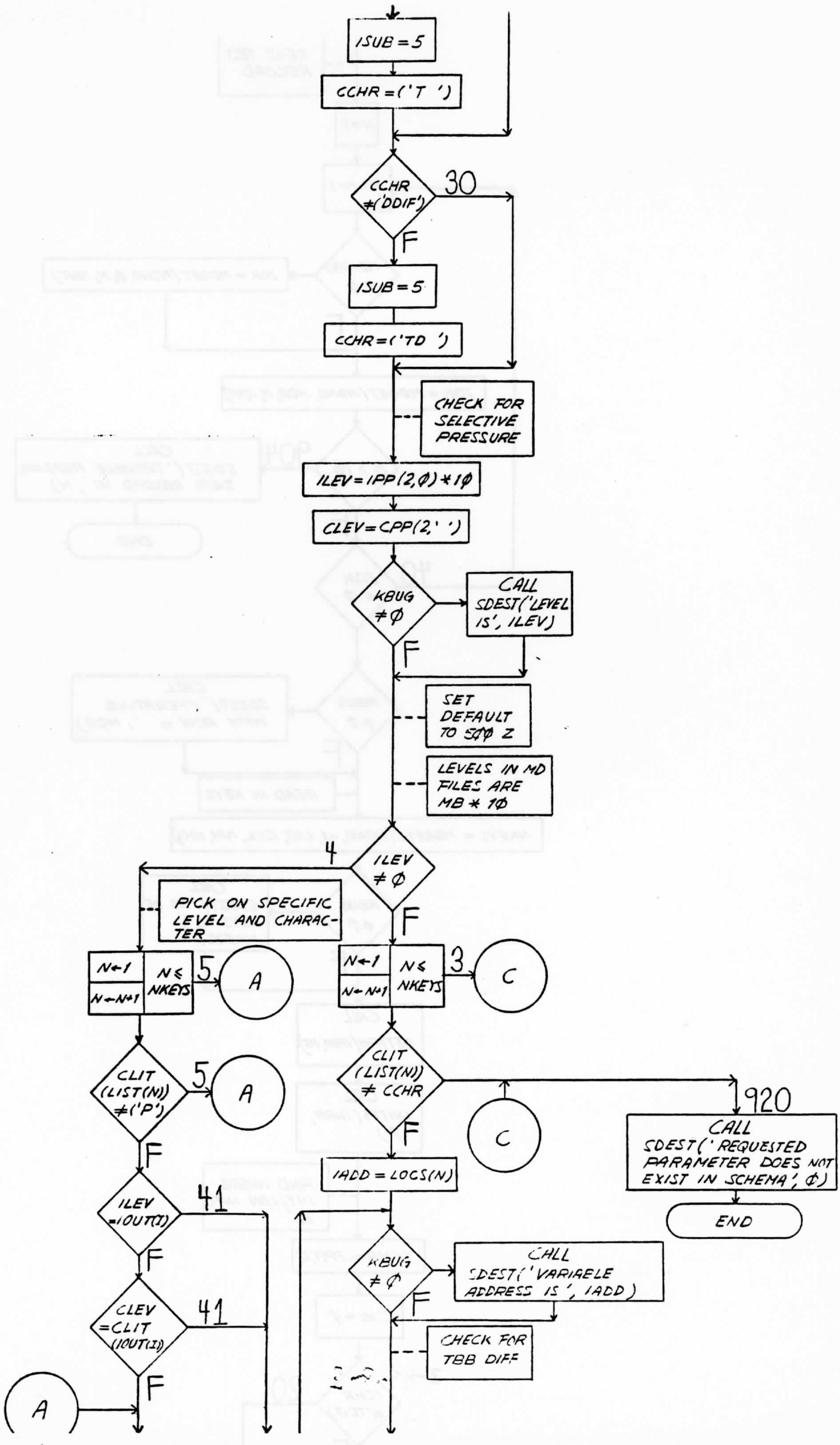


PLVA
DATA MISS/Z8080808Ø/
CHECK DEBUG OPTION
IHEL = IKWP('HELP', 1, Ø)
IHEL ≠ Ø
800
CALL SDEST('/PARAM) (P-LEVEL MB)', Ø)
END
F
KBUG = IKWP('BUG', 1, Ø)
MDNO = LUC(5)
LUC(81) = Ø
110
F
READ VASTEXT CONTEXT FILE
CALL DOPEN(MFILE, LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
CALL DCLOSE(LUN)
INITIALIZE NAVIGATION
CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)
CALL NVINIT(BETAIN, BETDOT, INAV, PTIME)
CALL TSNIO(1, 1, 1, 1, 1, 1, IDOC)
SET UP AREA FOR IMAGE SPACE TO PLOT
CALL TVSAT(IFRM, ØØ5, 335, LTOP, LELE, ISS, ID, IT)
CALL TVSAT(IFRM, 495, 335, LBOT, LELE, ISS, IP, IT)
LTOP < 1
LTOP = 1
F
LBOT > NROWM
LBOT = NROWM
F
LAN = LLNW/1000
LOW = MOD(LLNW, 1000)
LOW = IABS(LOW)
LOW > 180
LOW = LOW - 360
F

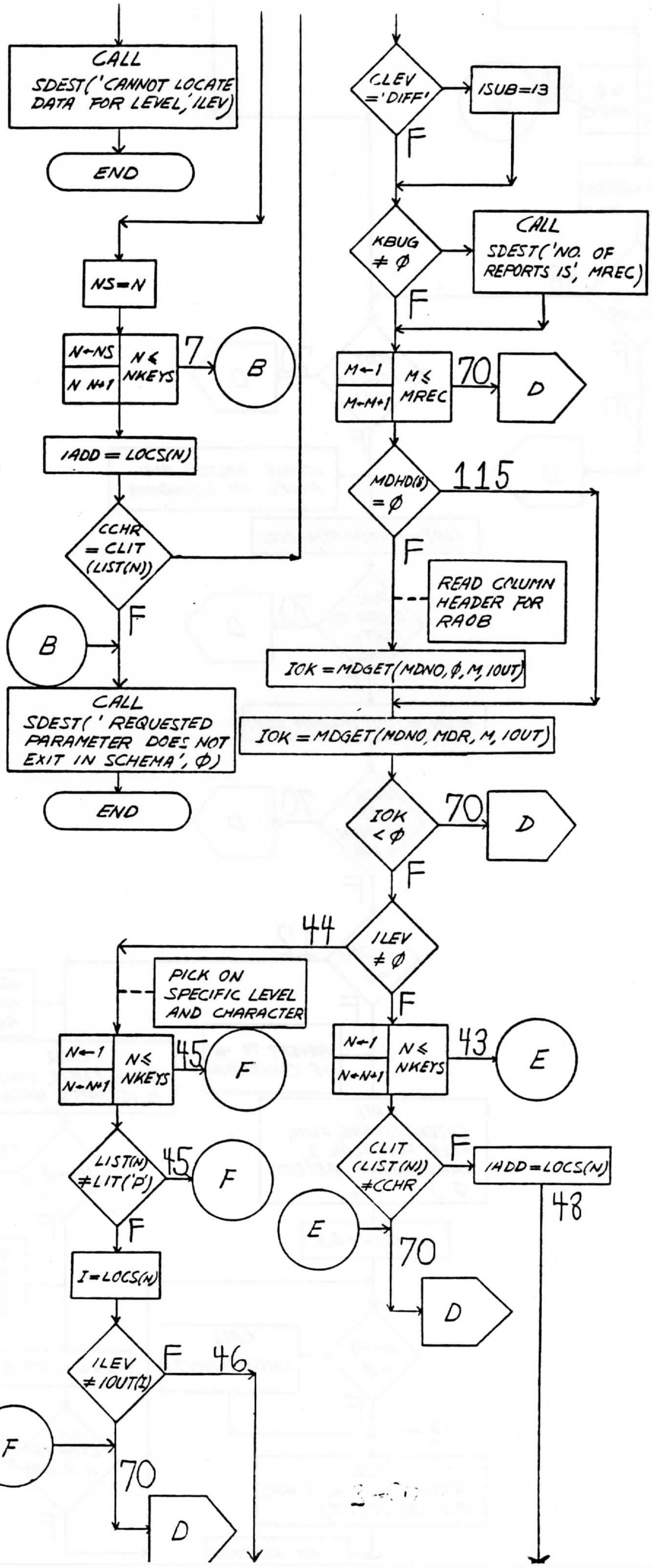

LAS = LLSE/1000
LOE = MOD(LLSE, 1000)
LOE = IABS(LOE)
LOE > 180
LOE = LOE - 360
F
LOX = 0
IABS (LOW-LOE) > 180
LOX = 1
F
LOX ≠ 0 AND LOW < 0
LOW = LOW + 360
MDNO = IKWP('MDNR', 1, MDNO)
MDINFO (MDNO, MDHD) ≠ 0
900
CALL SDEST('CANNOT OPEN MDFILE NO.', MDNO)
END
F
MDOPEN (MDNO, 2) ≠ 0
900
CALL SDEST ('CANNOT OPEN MDFILE NO.', MDNO)
END
F
HAVE TO CALL MDOPEN TO LOAD MDGET...
READ ROW HEADER RECORD
IOK = MDGET(MDNO, MDR, 0, IOUT)
KBUG ≠ 0
CALL OUTINT(KOUT)
F
IOK < 0
902
CALL SDEST('CANNOT LOCATE ROW NO.', MDR)
END
F

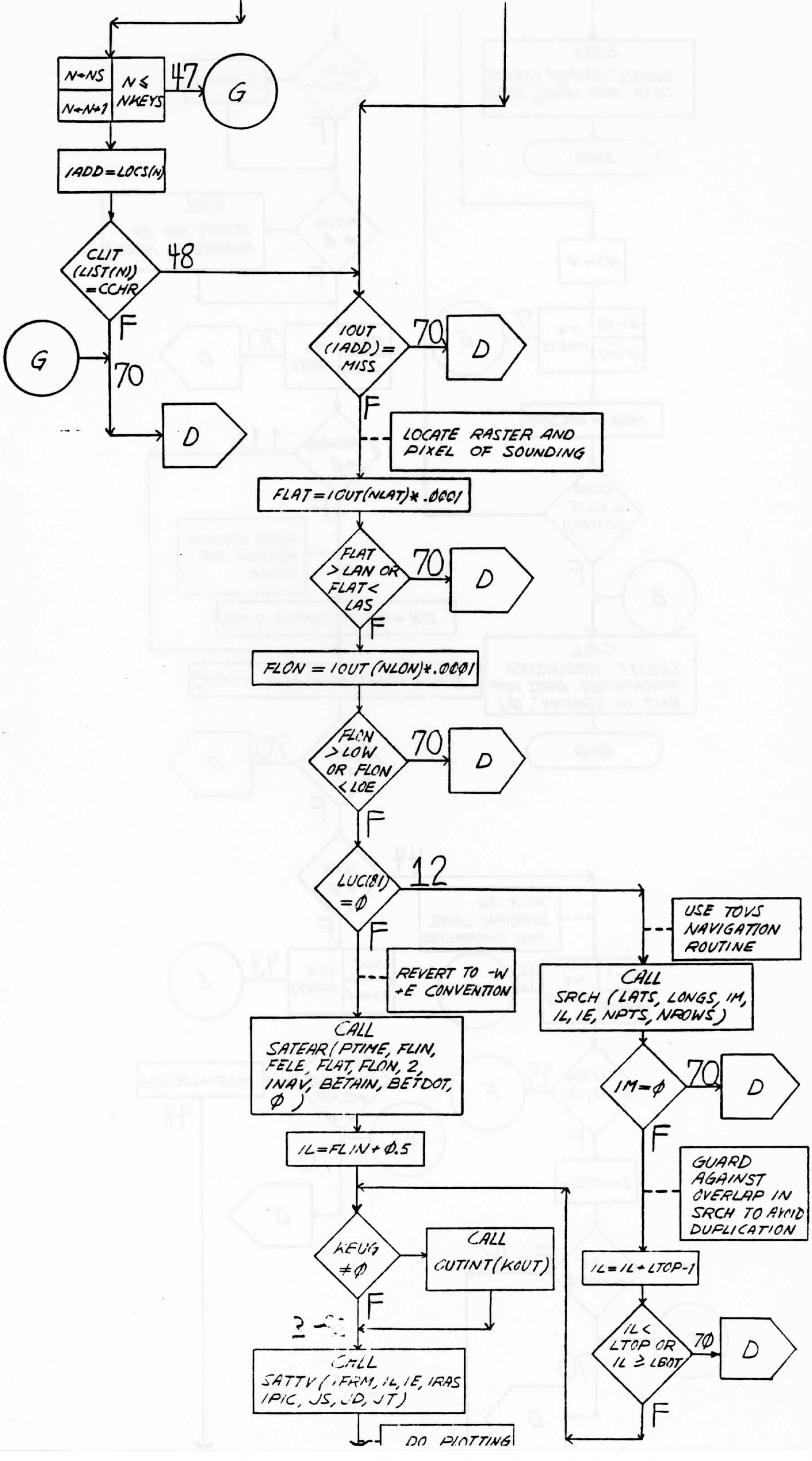


READ TEST RECORD
M=1
M=M+1
MDHD(8) ≠ Ø
IOK = MDGET(MDNO, Ø, M, IOUT)
F
IOK = MDGET(MDNO, MDR, M, IOUT)
M > 1Ø
904
CALL SDEST('TROUBLE READING DATA RECORD NO', M)
END
F
40
IOK < Ø
KBUG ≠ Ø
CALL SDEST('OPERATING WITH ROW = ', MDR)
F
READ IN KEYS
NKEYS = MDKEYS(MDNO, -1, LIST, ISCL, IUN, LOCS)
KBUG ≠ Ø
CALL SDEST('NO OF KEYS IS', NKEYS)
F
CALL GETFRM(IFRM, MF)
CALL INITPL(IWRM, Ø)
FIND WHERE LAT/LON INFO IS
CCHR=CPP(1,'Z')
ISUE = Ø
CCHR ≠('TDIF)
20
F

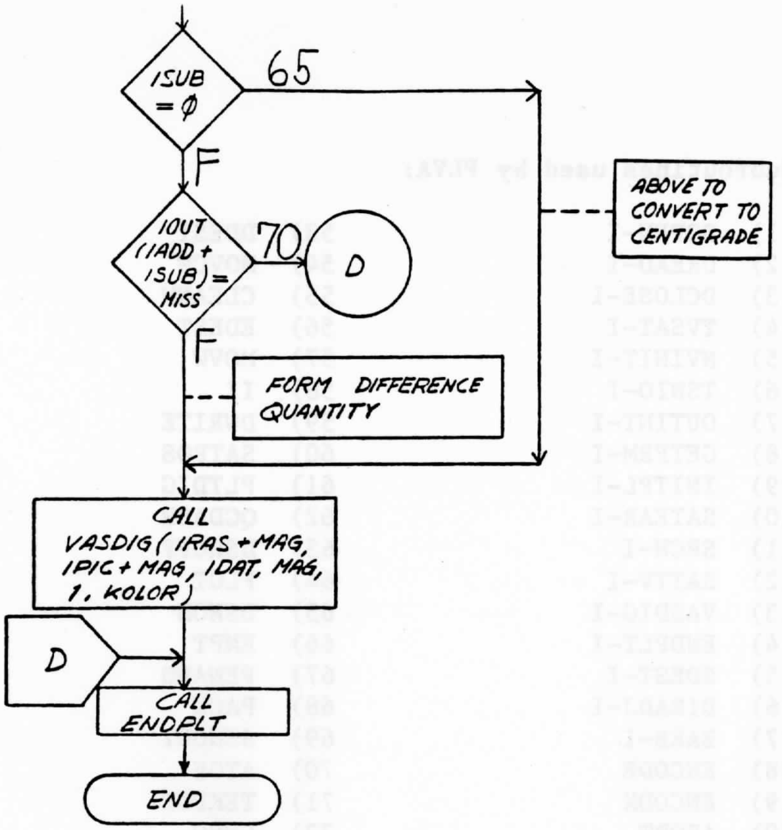


ISUB = 5
CCHR = ('T ')
CCHR ≠('DDIF')
30
F
ISUB = 5
CCHR = ('TD ')
CHECK FOR SELECTIVE PRESSURE
ILEV = IPP(2,Ø) * 1Ø
CLEV = CPP(2,' ')
KBUG ≠ Ø
CALL SDEST('LEVEL IS', ILEV)
F
SET DEFAULT TO 5ØØ Z
LEVELS IN MD FILES ARE MB * 1Ø
ILEV ≠ Ø
4
F
PICK ON SPECIFIC LEVEL AND CHARACTER
N←1
N←N+1
N ≤ NKEYS
5
A
N←1
N←N+1
N ≤ NKEYS
3
C
CLIT (LIST(N)) ≠('P')
5
A
F
CLIT (LIST(N)) ≠ CCHR
920
CALL SDEST('REQUESTED PARAMETER DOES NOT EXIST IN SCHEMA', Ø)
END
C
F
IADD = LOCS(N)
ILEV =IOUT(I)
41
F
KBUG ≠ Ø
CALL SDEST('VARIABLE ADDRESS IS', IADD)
F
CLEV =CLIT (IOUT(I))
41
CHECK FOR TBB DIFF
F
A


CALL
SDEST('CANNOT LOCATE DATA FOR LEVEL,' ILEV)
END
NS=N
N←NS
N N+1
N≤ NKEYS
7
B
IADD = LOCS(N)
CCHR = CLIT (LIST(N))
F
B
CALL
SDEST(' REQUESTED PARAMETER DOES NOT EXIT IN SCHEMA', Ø)
END
CLEV ='DIFF'
ISUB=13
F
KBUG ≠ Ø
CALL
SDEST('NO. OF REPORTS IS', MREC)
F
M←1
M←M+1
M≤ MREC
70
D
MDHD(8) = Ø
115
F
READ COLUMN HEADER FOR RAOB
IOK = MDGET(MDNO, Ø, M, IOUT)
IOK = MDGET(MDNO, MDR, M, IOUT)
IOK < Ø
70
D
F
ILEV ≠ Ø
44
PICK ON SPECIFIC LEVEL AND CHARACTER
F
N←1
N←N+1
N≤ NKEYS
45
F
N←1
N←N+1
N≤ NKEYS
43
E
LIST(N) ≠LIT('P')
45
F
F
I=LOCS(N)
CLIT (LIST(N)) ≠CCHR
F
IADD=LOCS(N)
48
E
70
D
ILEV ≠ IOUT(I)
F
46
F
70
D



N←NS
N←N+1
N ≤ NKEYS
47
G
IADD=LOCS(N)
CLIT (LIST(N)) =CCHR
48
F
G
70
D
IOUT (IADD)= MISS
70
D
F
LOCATE RASTER AND PIXEL OF SOUNDING
FLAT=IOUT(NLAT)*.0001
FLAT >LAN OR FLAT< LAS
70
D
F
FLON = IOUT(NLON)*.0001
FLON >LOW OR FLON <LOE
70
D
F
LUC(81) =0
12
F
REVERT TO -W +E CONVENTION
USE TOVS NAVIGATION ROUTINE
CALL SRCH (LATS, LONGS, IM, IL, IE, NPTS, NROWS)
CALL SATEAR(PTIME, FLIN, FELE, FLAT, FLON, 2, INAV, BETAIN, BETDOT, 0)
IL=FLIN+0.5
IM=0
70
D
F
GUARD AGAINST OVERLAP IN SRCH TO AVOID DUPLICATION
KEUG ≠0
CALL CUTINT(KOUT)
F
IL=IL+LTOP-1
IL< LTOP OR IL ≥ LBOT
70
D
F
CALL SATTV(IFRM, IL, IE, IRAS IPIC, JS, JD, JT)
DO PLOTTING


ISUB = ∅
65
F
IOUT (IADD + ISUB) = MISS
70
D
F
FORM DIFFERENCE QUANTITY
ABOVE TO CONVERT TO CENTIGRADE
CALL VASDIG(IRAS + MAG, IPIC + MAG, IDAT, MAG, 1, KOLOR)
D
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) VASDIG-I
14) ENDPLT-I
15) SDEST-I
16) DIRADJ-I
17) BARB-I
18) ENCODE
19) ENCODX
20) ABORT
21) CONTNT
22) ENCCLR
23) LTQ
24) DECONV
25) FECONV
26) IECONV
27) ZECONV
28) MOVB
29) CLEANA
30) TQ
31) PRLINX
32) PRCLOS
33) PROPEN
34) PRPRPR
35) LOCK
36) PRWR
37) PRRD
38) UNLOCK
39) PRCL
40) POST
41) BLKA
42) LWCLOS
43) MOVWC
44) LWMOP
45) TRMNL
46) TQMES
47) STC
48) ITOC
49) GETNAV
50) EPOCH
51) GETGAM
52) VASNAV
53) DDEST
54) MOVCW
55) CLEANW
56) EDEST
57) MOVW
58) II
59) DWRITE
60) SATPOS
61) PLTDIG
62) QGDASH
63) DSHOFF
64) PLOT
65) DSHON
66) ENPT
67) PENADD
68) PACK
69) SENOUT
70) ATOE
71) TEKPUT
72) LWPO
73) LWNEWF
74) LWSO
75) PAGE
76) PENMOV
77) BOX
78) PENBEG
79) WALK
80) PRIOUT
81) TOSET
82) PUC
83) WD
84) JMBWTF

BNVA

The purpose of BNVA is to prepare grid files of retrieval parameters. It is also possible to generate grids of parameters in other MD files as well. Quantities that can be gridded with BNVA include differences between retrieval and guess for temperature(T), dewpoint(TD), brightness temperature(TBB), etc. 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. As an aside, the gridfile you wish the results of BNVA to be stored in must be set before the program is run to make sure the final grid is stored in the desired gridfile. (This is assuming you are NOT using the guess grid option; otherwise, the results will be stored automatically in gridfile NGFG).

Initially, things such as the debug option (code line 98) are set, and the VASTEXT file is opened and read. At this point (statement 112), several other variables are determined, such as the MD file and row, the upper air guess grid file (NGFG) and the upper air guess MD file (MDNG). All four of these parameters have VASTEXT defaults if no keyword parameter has been designated. Now, after the parameter to be gridded and the level (ILEV) have been determined, the MD file is opened for read/write, the row header is read, a test to assure that data exists in the MD file is performed ("READ TEST RECORD"), and a call to function MDKEYS (code line 140) reads the keys contained within the schema for that particular MD file. Now, the positions within the key list for the number of retrievals or reports (variable NCA), latitude, longitude, etc. are determined in DO-loop 2. Variable NCA is important because it indicates which element of the array KBUF (a buffer array containing some particular MD file report) contains the number of retrievals/reports whose data is to be gridded. This value is stored in variable NRPT.

The following lines of code set variables such as ISUB (.NE. 0 implies some type of difference to be gridded). In addition, if one keys in "DIFF" for the second positional parameter (generate grid of TBB differences between retrieval and first guess), ISUB will be set to 13. Keyword parameter "LEV2" is also checked in code line 177 to see if a thickness calculation and resultant grid generation is desired. Next, variable NGG (code line 183) stores the number of the guess grid to be used in the event a guess grid option is desired. This is the first of two guess options in BNVA. The other guess option, which comes later in the program, involves accessing data from guess MD file MDNG rather than from the guess gridfile. 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 the guess could lead to a poor analysis within the retrieval/report-poor regions.

From code line 183 on, I will discuss BNVA separately for each of the three program options: guess grid, guess MD file or no guess.

First, in the case of the guess grid option, BNVA moves to statement 14 ("CONTINUE"). Then, after the analysis field has been initialized to zero, the guess grid is accessed from the guess gridfile (NGFG) by using function IGGET. The user must make sure the guess grid being accessed exists and is no larger than 2400 words; otherwise, IOK will be less than 0 (program code line 244) and BNVA will terminate. Following this, the number of gridpoints is calculated, the guess grid data is transferred to array FLD, the latitude/longitude dimensions of the grid are determined (LAN, LAS, LOW and LOE), 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, if keyword parameter NODAT is keyed in .NE. 0, the guess grid just accessed is stored in gridfile NGRF by function IGPUT, and BNVA terminates.

Normally, however, the program will enter an implicit DO-loop, whose purpose is to read data from MD file MDNO, and then prepare the input data for the subsequent grid generation. The loop runs from code line 297 to code line 420. If the correct data cannot be located (either in terms of locating the correct parameter or data/pressure level), or is missing, or if it is found but lies outside the grid boundaries, the loop increments and moves to the next data report in the MD file. Otherwise, assuming data is accessed from the MD file with no-problems, the final data to be gridded (either a value, difference between retrieval and guess, thickness, etc.) and the latitude/longitude coordinates of the data, are stored in the arrays DA, STLAT and STLON, respectively. Now, U and V wind components (if desired) are calculated, and Barnes arrays DAS, RW and CL are filled. Then, if there are less than 5000 data values accessed so far, and NN .LE. NRPT and NMAX, BNVA goes to statement 15 and looks for the next report in row MDR of the MD file.

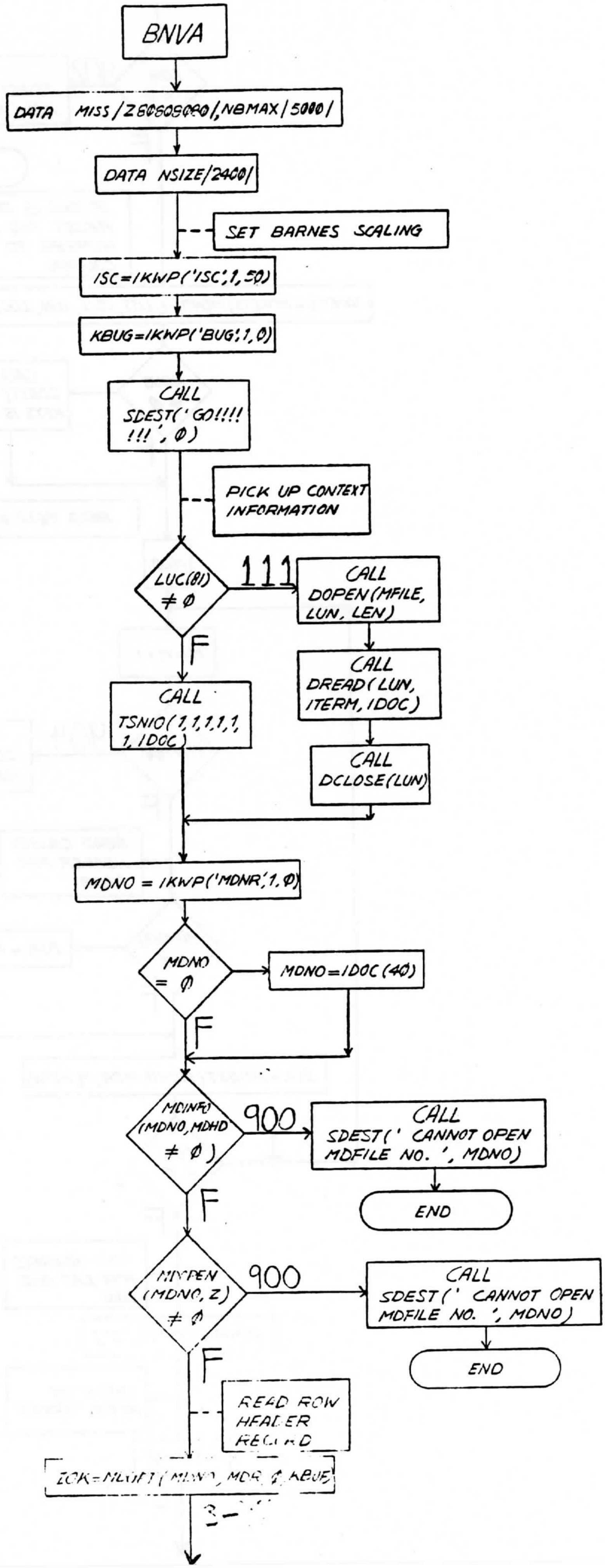
After all the MD data has been read and differences (if needed) calculated, a standard deviation (ISD) is determined from both the actual data to be gridded and previously-set "ERR" values (see code lines 290-294). Then, a Barnes analysis (subroutine FBARN) is called to produce a uniform set of gridpoint values based on the data to be gridded and the guess grid. The Barnes analysis makes only 1 pass if a guess grid or guess MD file option is used, while 2 passes are made for the no guess option. The final data is then stored in array IDL. Following this, if the edit option is not in operation (IKWP ('EDIT',1,0) .LE. 0), the grid is written into gridfile NGFG in the next empty slot after 0. In addition, if the pass weights are not to be written into the gridfile, BNVA is finished. However, if the weights for the one pass of the Barnes analysis are to be written into the gridfile, one more call to function IGPUT in code line 502 accomplishes this task, after which BNVA terminates.

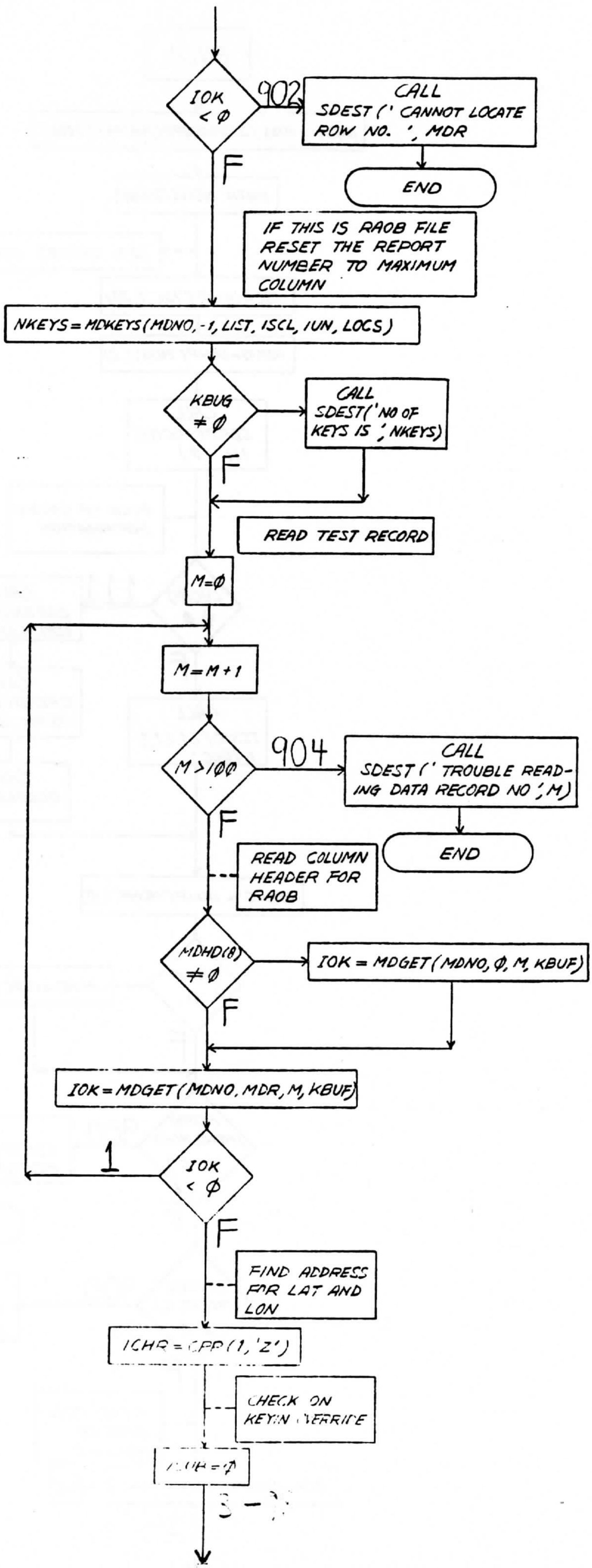
On the other hand, if the edit option is in operation, a value determined from the final grid is compared to the original data value from before the Barnes analysis (variable DAT), and if the difference is .GE. variable ERR, the element of the array KBUF which indicates rejection (NMOD) is set to a non-zero value. Then, the MD file is updated via function MDPUT to reflect this change. Note that the grid will not be written to an output gridfile if the edit option is on.

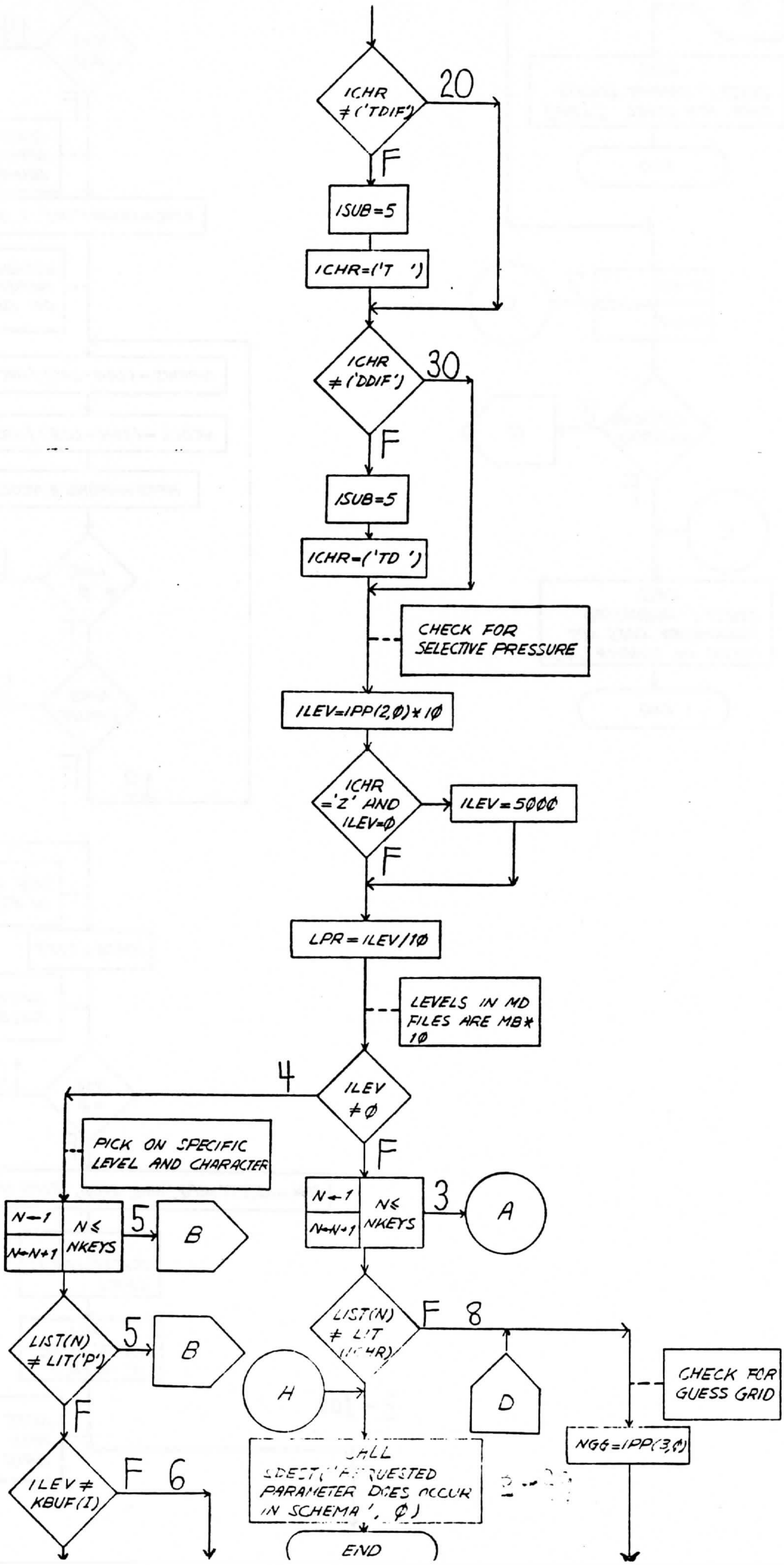
Returning to program code line 184, I will now discuss briefly the guess MD file option of BNVA. First, the boundaries of the grid (LN, LS, LE and LW) are set by keyed-in values of "LAT" and(or) "LON", or by VASTEXT default values if no keyin exists. Then, the grid increment (in tenths of degrees) is determined. If the grid increment has been keyed in (LINC .NE. 0), the number of gridpoints is arbitrary (can be .GT. 2400). However, if the grid increment has not been keyed in, the number of gridpoints must be .LE. 2400. Now, the analysis field is initialized to 0, and the guess data is read from the guess MD file (MDNG), with the surface option taken into account, via subroutine ANGSS. Then, after keyword parameter NODAT has been checked, 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, storing the final data (values, differences or thicknesses) and their latitude/longitude positions again in the arrays DA, STLAT and STLON, respectively. Next, assuming the edit option is in effect, a gross error check is performed comparing the value to be gridded (value, difference or thickness) with a value from the guess MD file. (Note that this test could cause problems in the event differences were being gridded and an MD file first guess was being used. In these cases, it would be best to turn off the edit option by setting keyword parameter "EDIT"

moves to the next MD file report (i.e., the implicit DO-loop increments). On the other hand, if the test does pass, the Barnes arrays are filled, 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, the standard deviation of the data to be gridded is calculated, and the Barnes analysis is performed. Then, from code line 439 to the end, BNVA with the guess MD file option functions similarly to the guess grid option, except the destination gridfile for the non-edit option is LUC(6) instead of NGFG.

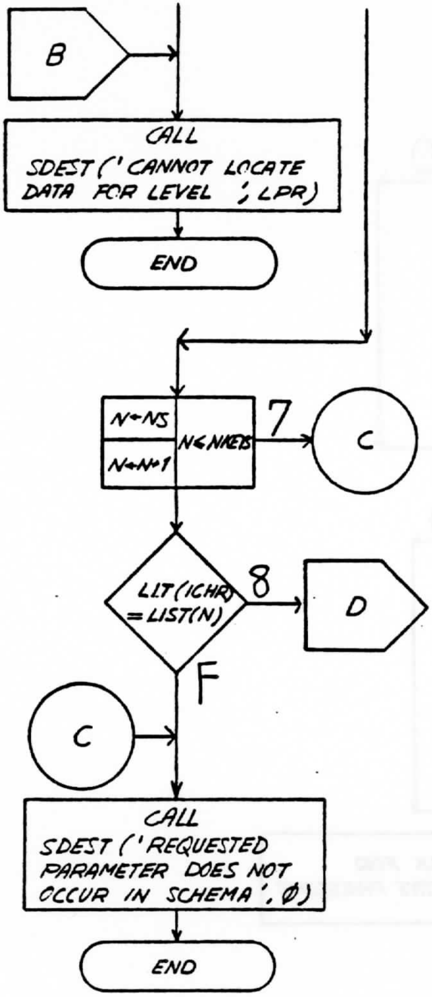
Lastly, the no guess option of BNVA will be described. This version should be used if the retrieval data coverage is fairly complete over the grid area. Once again, we return to program code line 184. From here, BNVA progresses with the no guess option precisely as the guess MD file option down through the initialization of the analysis field. Then, after passing to statement 105 (note that NGG=0), NODAT is checked, and the program passes directly into the implicit DO-loop. After the loop has finished accessing and processing the MD data, depending upon which type of quantity is to be gridded, the Barnes analysis is performed by subroutine FBARN. Then, from program code line 439 to the end, BNVA processes the final gridded data in a similar fashion to the guess MD file option, with the exception that 2 weight arrays can be stored with this option (as opposed to 1), due to the fact the Barnes analysis makes 2 passes for the no guess option.

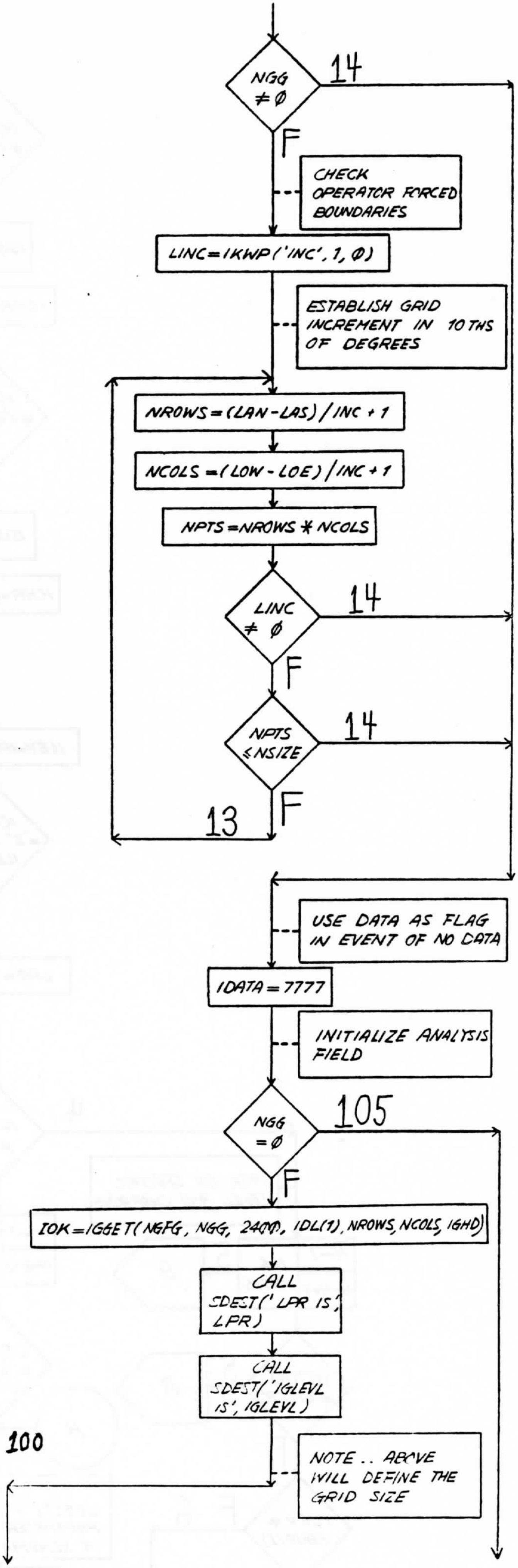

BNVA
DATA MISS/Z80808080/,NBMAX/5000/
DATA NSIZE/2400/
SET BARNES SCALING
ISC=IKWP('ISC',1,50)
KBUG=IKWP('BUG',1,0)
CALL SDEST(' GO!!!! !!!', 0)
PICK UP CONTEXT INFORMATION
LUC(81) ≠ 0
111
F
CALL DOPEN(MFILE, LUN, LEN)
CALL DREAD(LUN, ITERM, IDOC)
CALL DCLOSE(LUN)
CALL TSNIO(1,1,1,1,1, 1, IDOC)
MDNO = IKWP('MDNR',1,0)
MDNO = 0
MDNO=IDOC(40)
F
MDINFO (MDNO,MDHD ≠ 0)
900
CALL SDEST(' CANNOT OPEN MDFILE NO. ', MDNO)
END
F
MIOPEN (MDNO, Z) ≠ 0
900
CALL SDEST(' CANNOT OPEN MDFILE NO. ', MDNO)
END
F
READ ROW HEADER RECORD

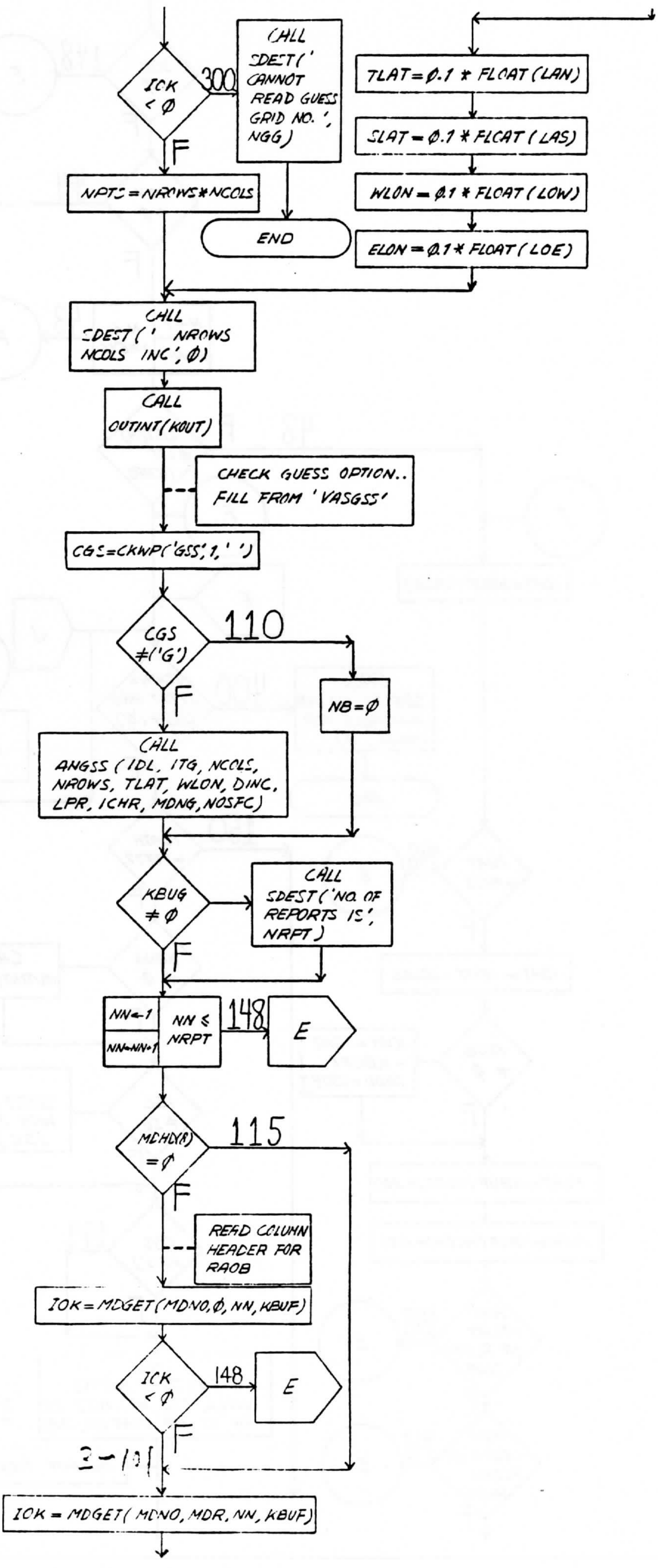

IOK < Ø
902
CALL SDEST(' CANNOT LOCATE ROW NO. ', MDR
END
F
IF THIS IS RAOB FILE RESET THE REPORT NUMBER TO MAXIMUM COLUMN
NKEYS = MDKEYS(MDNO, -1, LIST, ISCL, IUN, LOCS)
KBUG ≠ Ø
CALL SDEST('NO OF KEYS IS ', NKEYS)
F
READ TEST RECORD
M=Ø
M=M+1
M > 1ØØ
904
CALL SDEST(' TROUBLE READING DATA RECORD NO ', M)
END
F
READ COLUMN HEADER FOR RAOB
MDHD(8) ≠ Ø
IOK = MDGET(MDNO, Ø, M, KBUF)
F
IOK = MDGET(MDNO, MDR, M, KBUF)
IOK < Ø
1
F
FIND ADDRESS FOR LAT AND LON
ICHR = CPP(1,'Z')
CHECK ON KEYIN OVERRIDE

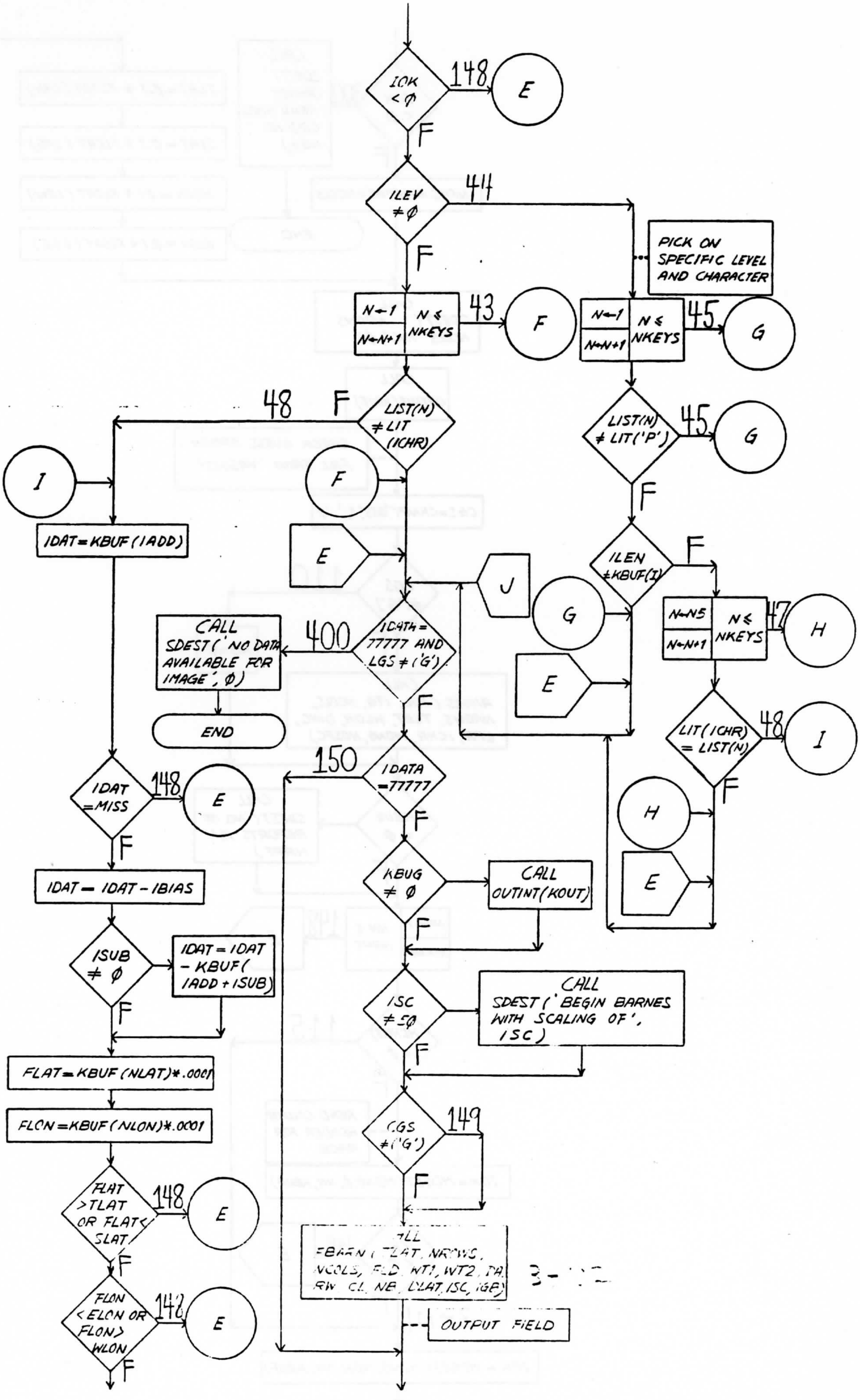


ICHR ≠('TDIF')
20
F
ISUB=5
ICHR=('T ')
ICHR ≠('DDIF')
30
F
ISUB=5
ICHR=('TD ')
CHECK FOR SELECTIVE PRESSURE
ILEV=IPP(2,Ø)*1Ø
ICHR ='Z' AND ILEV=Ø
ILEV=5ØØØ
F
LPR=ILEV/1Ø
LEVELS IN MD FILES ARE MB* 1Ø
4
ILEV ≠Ø
F
PICK ON SPECIFIC LEVEL AND CHARACTER
N←1
N←N+1
N≤ NKEYS
5
B
3
A
LIST(N) ≠ LIT('P')
5
B
F
LIST(N) ≠ LIT
F 8
H
D
CHECK FOR GUESS GRID
NGG=IPP(3,Ø)
ILEV≠ KBUF(I)
F 6
CALL
PARAMETER DOES OCCUR
IN SCHEMA ', Ø)
END

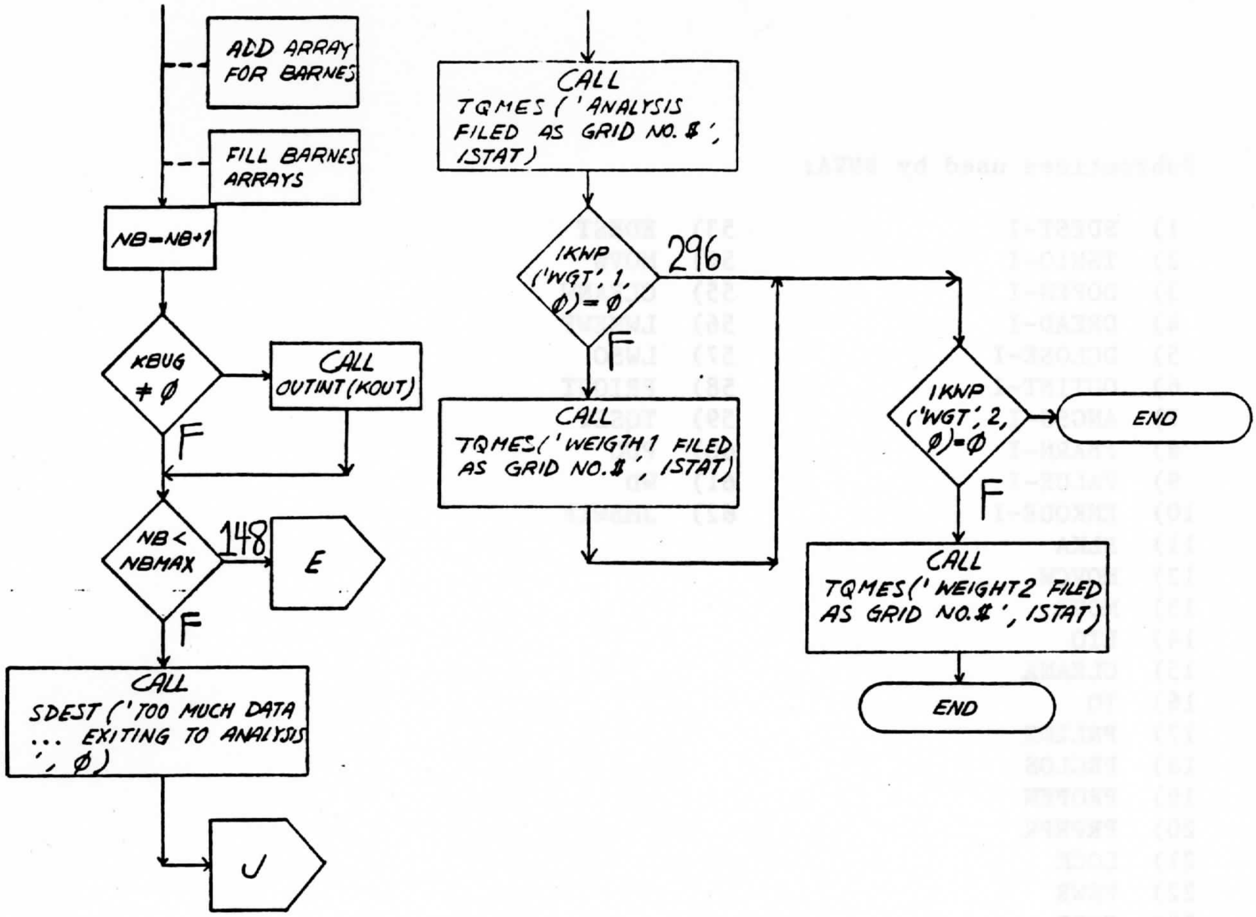

B
CALL SDEST ('CANNOT LOCATE DATA FOR LEVEL ';LPR)
END
N←NS
N←N+1
N≤NKEYS
7
C
LIT(ICHR) =LIST(N)
8
D
F
C
CALL SDEST ('REQUESTED PARAMETER DOES NOT OCCUR IN SCHEMA', Ø)
END


NGG ≠Ø
14
F
CHECK OPERATOR FORCED BOUNDARIES
LINC=IKWP('INC',1,Ø)
ESTABLISH GRID INCREMENT IN 10THS OF DEGREES
NROWS=(LAN-LAS)/INC+1
NCOLS=(LOW-LOE)/INC+1
NPTS=NROWS*NCOLS
LINC ≠Ø
14
F
NPTS ≤NSIZE
14
F
13
USE DATA AS FLAG IN EVENT OF NO DATA
IDATA=7777
INITIALIZE ANALYSIS FIELD
NGG =Ø
105
F
IOK=IGGET(NGFG, NGG, 2400, IDL(1), NROWS, NCOLS, IGHD)
CALL SDEST('LPR IS', LPR)
CALL SDEST('IGLEVL IS', IGLEVL)
NOTE.. ABOVE WILL DEFINE THE GRID SIZE


ICK < Ø
300
CALL SDEST(' CANNOT READ GUESS GRID NO. ', NGG)
END
F
NPTS = NROWS*NCOLS
TLAT = Ø.1 * FLOAT(LAN)
SLAT = Ø.1 * FLOAT(LAS)
WLON = Ø.1 * FLOAT(LOW)
ELON = Ø.1 * FLOAT(LOE)
CALL SDEST(' NROWS NCOLS INC', Ø)
CALL OUTINT(KOUT)
CHECK GUESS OPTION.. FILL FROM 'VASGSS'
CGS=CKWP('GSS',1,' ')
CGS ≠('G')
110
NB=Ø
F
CALL ANGSS(IDL, ITG, NCOLS, NROWS, TLAT, WLON, DINC, LPR, ICHR, MDNG, NOSFC)
KBUG ≠ Ø
CALL SDEST('NO. OF REPORTS IS', NRPT)
F
NN←1
NN←NN+1
NN ≤ NRPT
148
E
MDHDY(R) = Ø
115
F
READ COLUMN HEADER FOR RAOB
IOK = MDGET(MDNO,Ø,NN,KBUF)
ICK < Ø
148
E
F
3-101
IOK = MDGET(MDNO, MDR, NN, KBUF)



IOK < Ø
148
E
F
ILEV ≠ Ø
44
PICK ON SPECIFIC LEVEL AND CHARACTER
N←1 N←N+1 N ≤ NKEYS
43
F
N←1 N←N+1 N ≤ NKEYS
45
G
48
LIST(N) ≠ LIT (ICHR)
LIST(N) ≠ LIT('P')
45
G
I
IDAT=KBUF(IADD)
F
E
J
ILEN ≠KBUF(I)
G
N←N5 N←N+1 N ≤ NKEYS
47
H
CALL SDEST('NO DATA AVAILABLE FOR IMAGE', Ø)
400
IDATA= 77777 AND LGS ≠ ('G')
E
END
LIT(ICHR) = LIST(N)
48
I
150
IDATA =77777
IDAT =MISS
148
E
H
E
KBUG ≠ Ø
CALL OUTINT(KOUT)
IDAT = IDAT - IBIAS
ISUB ≠ Ø
IDAT = IDAT - KBUF(IADD + ISUB)
ISC ≠ 50
CALL SDEST('BEGIN BARNES WITH SCALING OF', ISC)
FLAT=KBUF(NLAT)*.0001
FLON=KBUF(NLON)*.0001
LGS ≠ ('G')
149
FLAT >TLAT OR FLAT< SLAT
148
E
CALL FBARN(TLAT, NROWS, NCOLS, FLD, WT1, WT2, PA, RW, CL, NB, DLAT, ISC, IGB)
FLON <ELON OR FLON> WLON
148
E
OUTPUT FIELD


ADD ARRAY FOR BARNES
FILL BARNES ARRAYS
NB=NB+1
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
NB < NBMAX
148
E
F
CALL SDEST ('TOO MUCH DATA ... EXITING TO ANALYSIS ', Ø)
J
CALL TQMES ('ANALYSIS FILED AS GRID NO.$', ISTAT)
IKNP ('WGT', 1, Ø) = Ø
296
F
CALL TQMES('WEIGTH1 FILED AS GRID NO.$', ISTAT)
IKNP ('WGT', 2, Ø)=Ø
END
F
CALL TQMES('WEIGHT2 FILED AS GRID NO.$', ISTAT)
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) ENCCLR
34) DECONV
35) FECONV
36) IECONV
37) ZECONV
38) DWRITE
39) LWCLOS
40) MOVWC
41) LWMOP
42) ITOC
43) VASGES
44) SURGES
45) GETSFV
46) IGNAME
47) WMIX
48) FILL
49) LINFIL
50) INTER
51) HEAPFY
52) LWPO
53) EDEST
54) MOVW
55) CLEANW
56) LWNEWF
57) LWSO
58) PRIOUT
59) TQSET
60) PUC
61) WD
62) 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 are still quite important to the retrieval process on the whole. The format used in describing the programs in this chapter is identical to that used in Chapter 3.

GWVA

Program GWVA is used to produce and (optional) plot wind vectors from the previously-calculated height grids generated by program BNVA.

The first things done in GWVA are to initialize the plot routines using subroutine INITPL, set the plot option (IPLT), and read the VASTEXT file. Assuming the plot option is on (IPLT=0), the line and element coordinate of the cursor are determined via subroutine TVSAT, and the navigation file is opened via subroutine NVINIT. Then, variables MDNO and MDR are set to the retrieval MD file and row numbers, respectively, listed in the VASTEXT file (IDOC (40) and (41)). After the retrieval MD file is opened for read/write, the wind type to be calculated is determined. 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 (code line 122, default=1), grid file (NGRF) and grid number are set, after which the height grid from which the winds are to be produced is accessed via function IGGET. Note that the gridfile must be set before GWVA is executed using system program IGU to ensure that the proper height grid is accessed. 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 (LEVEL), or GWVA will terminate. Then, a check is performed (DO LOOP 14) to see if the keyed in level is a mandatory level.

Next, the grid boundaries (LAN, LOW, LAS and LOE) and grid increment are determined, as well as the maximum and minimum latitude and longitude grid values for which winds can be computed (LATMAX, LONMAX, LATMIN and LONMIN).

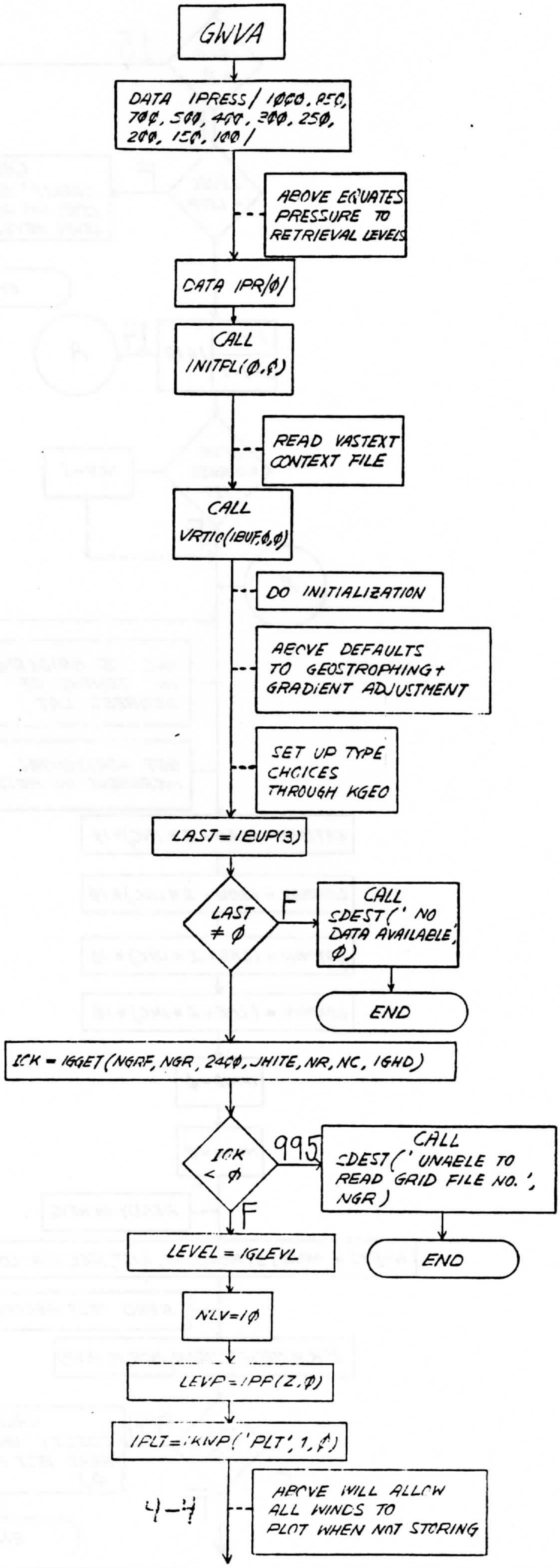
At this point, the keys for the retrieval MD file are read via function MDKEYS. After a test is performed to make sure data exists in the retrieval MD file, during which the row header of the retrieval file corresponding to row MDR is read, program GWVA enters DO-loop 11. In this loop, addresses within the keylist corresponding to number of retrievals, latitude and longitude are determined. These addresses will indicate which elements within array IBUF contain the total number of retrievals for row MDR, as well as the latitude and longitude of a given retrieval. This information is then used to assign a value to variable LAST.

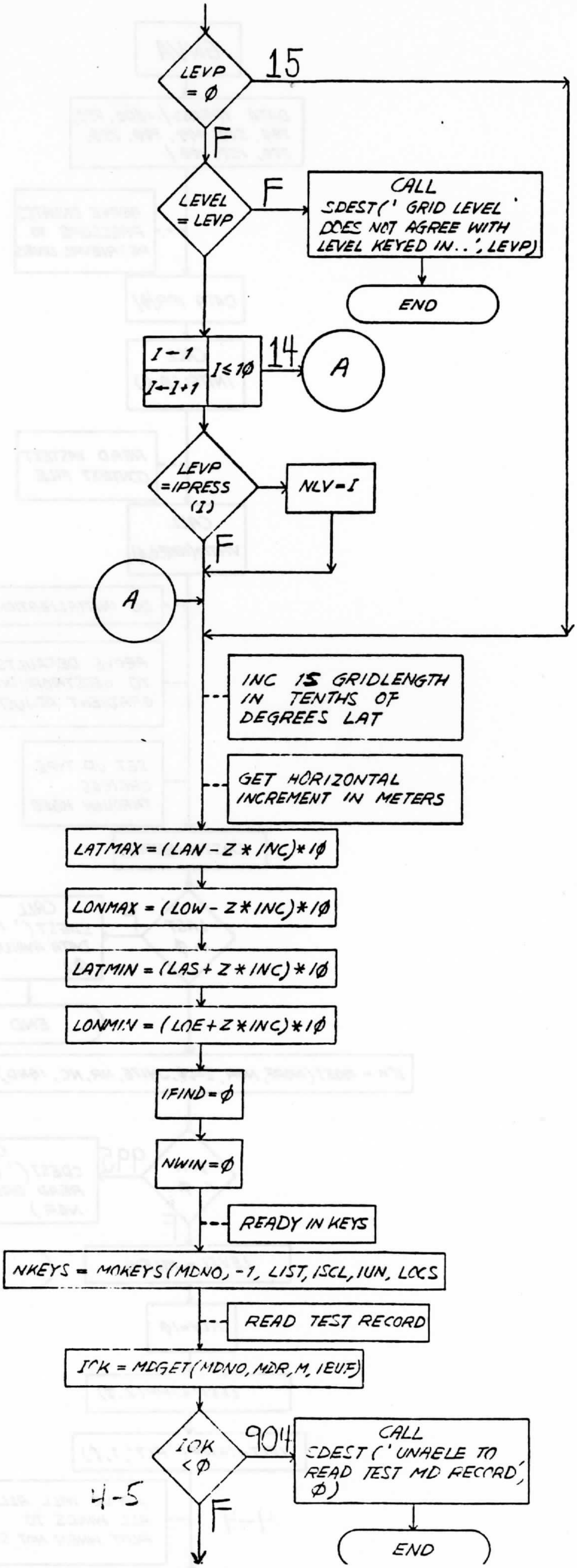
Assuming LAST has a value .NE. 0, variables DELT and NGR are filled. These variables should be keyed in if one wants to produce either isallobaric or ageostrophic winds, since both of these winds require changes of wind over a period of time or, in other words, another earlier height grid. If the earlier grid is desired for one of these wind options, it is read in code line 199 via IGGET.

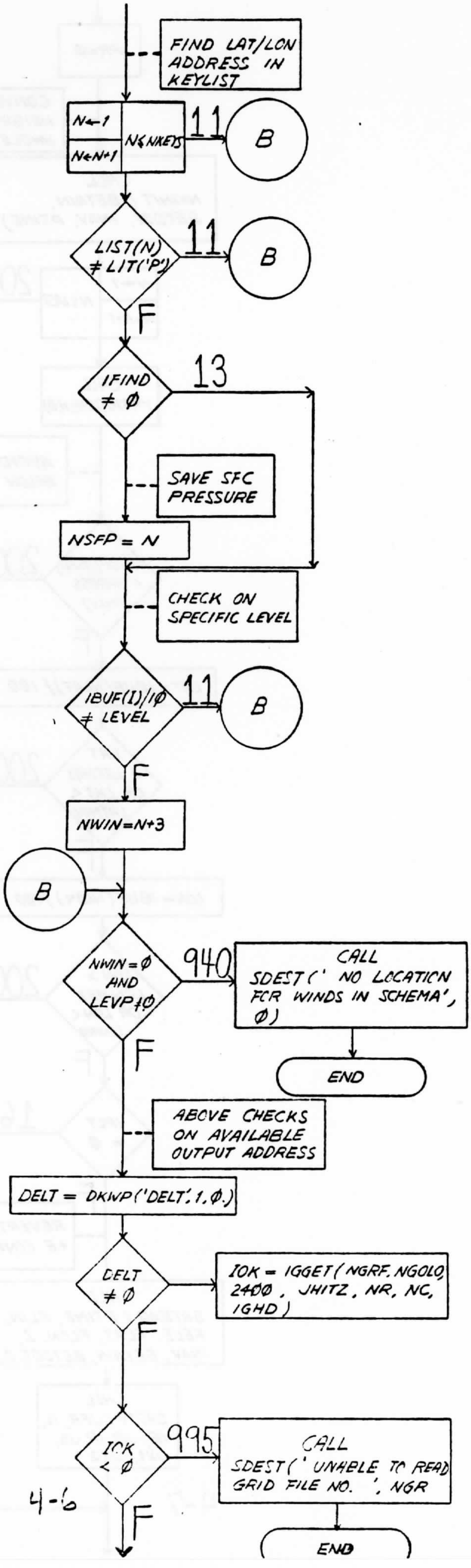
Finally, we enter DO LOOP 200 at code line 215. The purpose of this large loop is to calculate the desired winds at each retrieval location within the grid boundaries, and plot the results if the plot option is in effect. After the retrieval report is read via function MDGET, DO-loop 110 finds the location within array IBUF which stores retrieval surface pressure, in addition to computing the value of variable NWIN, which is used later to determine into which element of IBUF the final wind direction should be placed. Note that if NWIN is not able to be calculated, but a level (LEVP) was keyed in, no wind will be calculated at this retrieval location. In addition, the winds will not be calculated if: the retrieval location is outside the previously determined wind calculation boundaries within the grid (LATMAX, LONMAX, etc.), or if the pressure level for the wind is below the retrieval surface. Further discussion of GWVA beyond code line 241 will be divided into two categories: plot option and no plot option.

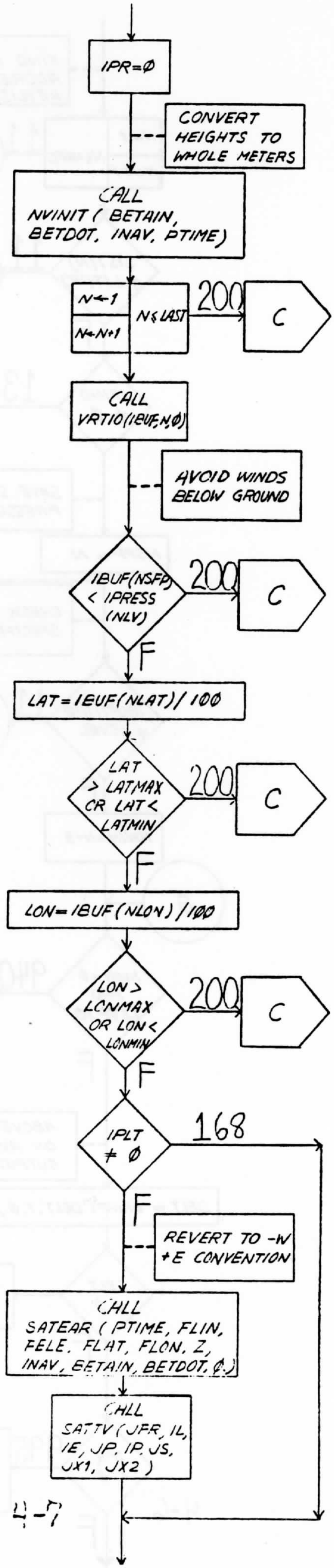
First, if the plot option is in effect (IPLT=0), GWVA calculates TV coordinates (raster, pixel) JP and IP, using subroutines SATEAR and SATTV, and then derives the desired wind at grid location (I,J) by calling subroutine ZWIND. Then, the wind barb itself is plotted by subroutine BARB after the direction has been adjusted by DIRADJ. Finally, if a level was keyed in (LEVP .NE. 0), the wind information is put into the retrieval MD file via function MDPUT in code line 285, and the program loops back to process another retrieval. On the other hand, if no level was keyed in, none of the wind results will be put into the retrieval MD file.

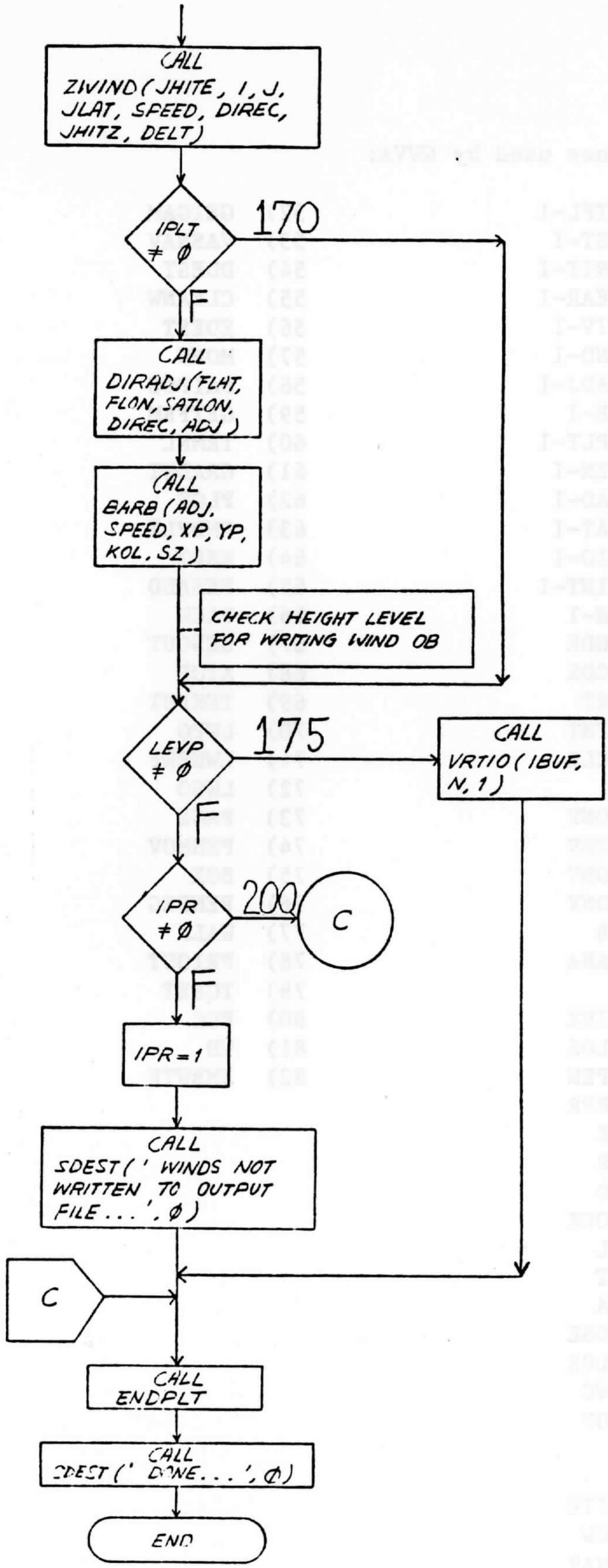
Second, if the plot option is not in effect, GWVA immediately calculates the wind at the grid location. Then, after skipping to statement 170 (code line 275), the same process of checking the value of variable LEVP described above takes place.


GWVA
DATA IPRESS/1000, 850, 700, 500, 400, 300, 250, 200, 150, 100/
ABOVE EQUATES PRESSURE TO RETRIEVAL LEVELS
DATA IPR/0/
CALL INITFL(0,0)
READ VASTEXT CONTEXT FILE
CALL VRTIO(IBUF,0,0)
DO INITIALIZATION
ABOVE DEFAULTS TO GEOSTROPHING + GRADIENT ADJUSTMENT
SET UP TYPE CHOICES THROUGH KGEO
LAST = IBUF(3)
LAST ≠ 0
F
CALL SDEST(' NO DATA AVAILABLE', 0)
END
ICK = IGGET(NGRF, NGR, 2400, JHITE, NR, NC, IGHD)
ICK < 0
995
CALL SDEST(' UNABLE TO READ GRID FILE NO. ', NGR)
F
END
LEVEL = IGLEVL
NLV = 10
LEVP = IPF(Z,0)
IPLT = IKWP('PLT', 1, 0)
ABOVE WILL ALLOW ALL WINDS TO PLOT WHEN NOT STORING


LEVP = Ø
15
F
LEVEL = LEVP
F
CALL SDEST(' GRID LEVEL DOES NOT AGREE WITH LEVEL KEYED IN..', LEVP)
END
I←1
I←I+1
I≤1Ø
14
A
LEVP =IPRESS (I)
NLV=I
F
A
INC IS GRIDLENGTH IN TENTHS OF DEGREES LAT
GET HORIZONTAL INCREMENT IN METERS
LATMAX=(LAN-Z*INC)*1Ø
LONMAX=(LOW-Z*INC)*1Ø
LATMIN=(LAS+Z*INC)*1Ø
LONMIN=(LOE+Z*INC)*1Ø
IFIND=Ø
NWIN=Ø
READY IN KEYS
NKEYS = MOKEYS(MDNO, -1, LIST, ISCL, IUN, LOCS
READ TEST RECORD
IOK = MDGET(MDNO, MDR, M, IEUF)
IOK <Ø
904
CALL SDEST(' UNABLE TO READ TEST MD RECORD', Ø)
END
F


FIND LAT/LON ADDRESS IN KEYLIST
N←1
N≤NKEYS
N←N+1
11
B
LIST(N) ≠ LIT('P')
11
B
F
IFIND ≠ Ø
13
SAVE SFC PRESSURE
NSFP = N
CHECK ON SPECIFIC LEVEL
IBUF(I)/1Ø ≠ LEVEL
11
B
F
NWIN=N+3
B
NWIN = Ø AND LEVP≠Ø
940
CALL SDEST(' NO LOCATION FOR WINDS IN SCHEMA', Ø)
END
F
ABOVE CHECKS ON AVAILABLE OUTPUT ADDRESS
DELT = DKWP('DELT', 1, Ø.)
DELT ≠ Ø
IOK = IGGET(NGRF, NGOLO, 2400, JHITZ, NR, NC, IGHD)
F
IOK < Ø
995
CALL SDEST(' UNABLE TO READ GRID FILE NO. ', NGR
F
END


IPR=Ø
CONVERT HEIGHTS TO WHOLE METERS
CALL NVINIT (BETAIN, BETDOT, INAV, PTIME)
N←1
N←N+1
N≤LAST
200
C
CALL VRTIO(IBUF,N,Ø)
AVOID WINDS BELOW GROUND
IBUF(NSFP) < IPRESS (NLV)
200
C
F
LAT=IBUF(NLAT)/100
LAT > LATMAX OR LAT < LATMIN
200
C
F
LON=IBUF(NLON)/100
LON > LONMAX OR LON < LONMIN
200
C
F
IPLT ≠ Ø
168
F
REVERT TO -W +E CONVENTION
CALL SATEAR (PTIME, FLIN, FELE, FLAT, FLON, Z, INAV, BETAIN, BETDOT, Ø.)
CALL SATTV (JFR, IL, IE, JP, IP, JS, JX1, JX2)


CALL
ZIVIND (JHITE, I, J, JLAT, SPEED, DIREC, JHITZ, DELT)
IPLT ≠ Ø
170
F
CALL
DIRADJ (FLAT, FLON, SATLON, DIREC, ADJ)
CALL
BARB (ADJ, SPEED, XP, YP, KOL, SZ)
CHECK HEIGHT LEVEL FOR WRITING WIND OB
LEVP ≠ Ø
175
CALL
VRTIO (IBUF, N, 1)
F
IPR ≠ Ø
200
C
F
IPR = 1
CALL
SDEST (' WINDS NOT WRITTEN TO OUTPUT FILE...', Ø)
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) ENCCLR
21) LTQ
22) DECONV
23) FECONV
24) IECONV
25) ZECONV
26) MOVB
27) CLEANA
28) TQ
29) PRLINX
30) PRCLOS
31) PROPEN
32) PRPRPR
33) LOCK
34) PRWR
35) PRRD
36) UNLOCK
37) PRCL
38) POST
39) BLKA
40) DCLOSE
41) LWCLOS
42) MOVWC
43) LWMOP
44) II
45) STC
46) DWRITE
47) MOVCW
48) GETNAV
49) EPOCH
50) TQMES
51) ITOC
52) GETGAM
53) VASNAV
54) DDEST
55) CLEANW
56) EDEST
57) MOVW
58) SATPOS
59) GETFRM
60) TRMNL
61) GRADWI
62) PLOT
63) ENDPLT
64) ENPT
65) PENADD
66) PACK
67) SENOUT
68) ATOE
69) TEKPUT
70) LWPO
71) LWNEWF
72) LWSO
73) PAGE
74) PENMOV
75) BOX
76) PENBEG
77) WALK
78) PRIOUT
79) TQSET
80) PUC
81) WD
82) JMBWTF

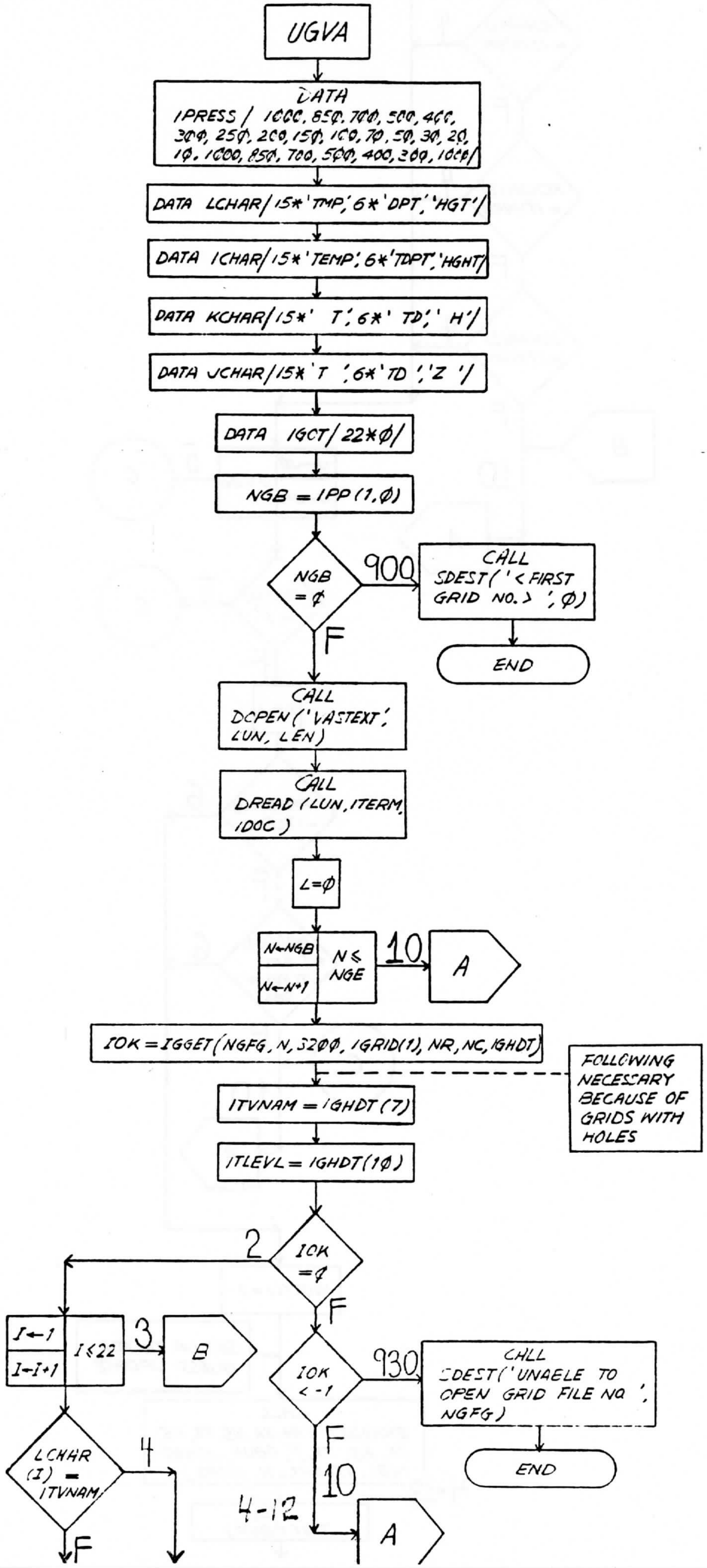
UGVA

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 LFM first guess. The LFM first guess would be used to process the 1400 GMT retrievals. Then, UGVA would allow the 1600 GMT retrievals to be processed with an updated version of the LFM first guess (updated by the 1400 GMT retrievals), rather than the original 1200 GMT first guess.

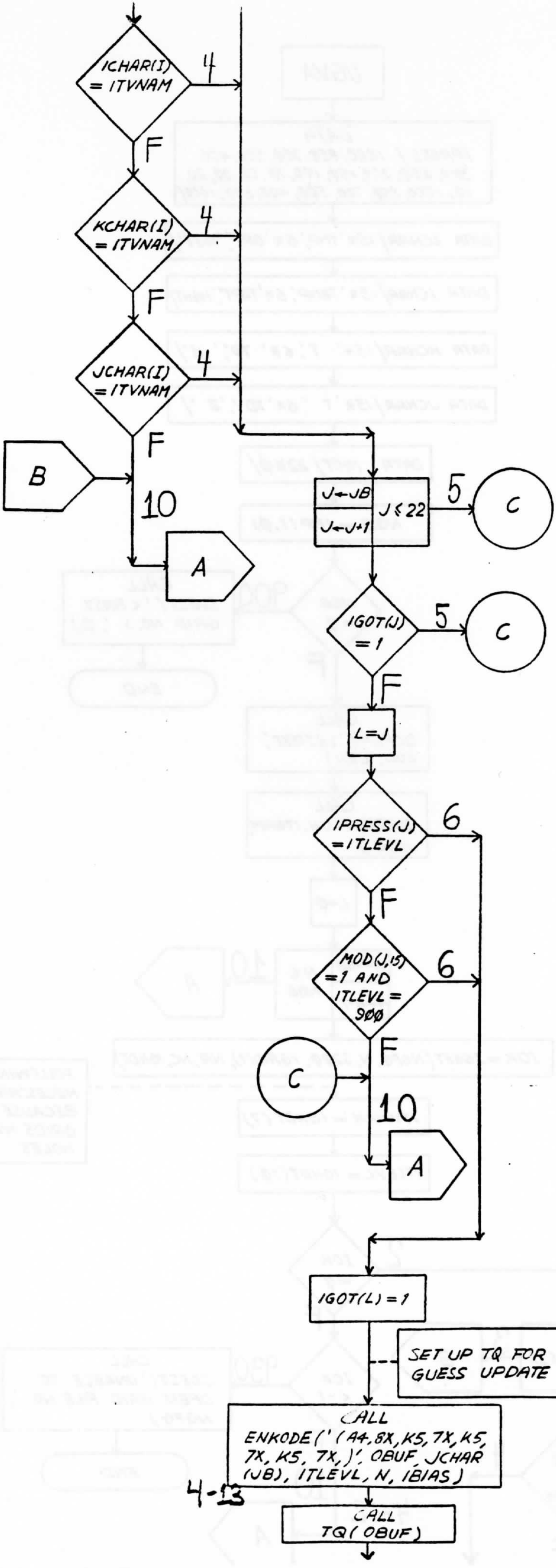
After the first grid to be updated, the VASTEXT file and the guess gridfile to be updated have been determined, UGVA enters DO LOOP 10, which encompasses the vast majority of the remainder of the program.

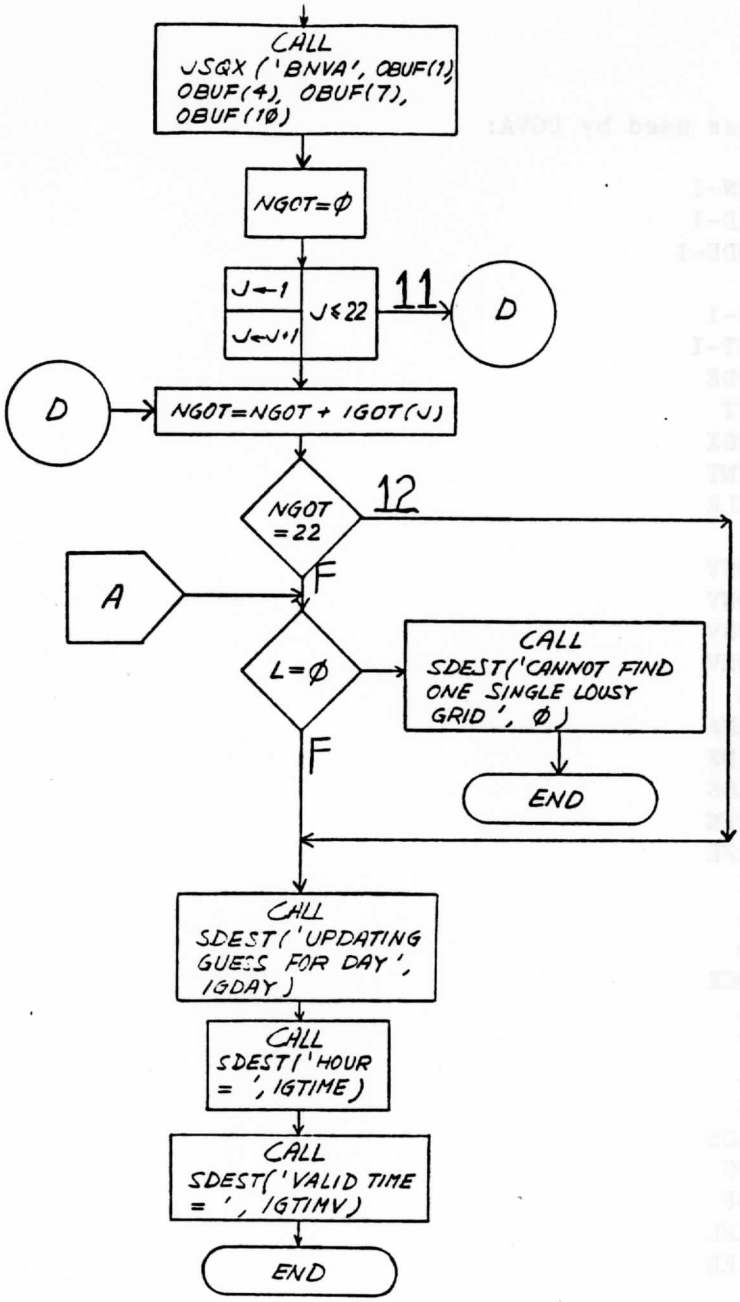
First, a given grid N is read from the guess gridfile via function IGGET (code line 89). UGVA terminates if there is no such grid file. In addition, the program will move to read the next grid if grid N does not exist or is too big to be read. Assuming the grid is read with no problems, the gridded variable's name (ITVNAM) is found in one of the four character lists (LCHAR, ICHAR, KCHAR and JCHAR), and the element of the array IPRESS that equals the pressure level of the grid is determined. UGVA will move to read the next grid if either the grid name or pressure level cannot be found. Assuming that the name and pressure level of the grid are found, the grid header is stored in array IGHD, and element L of the array IGOT is set to 1 to indicate that particular grid has now been accessed and updated. 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). This grid will serve as the guess grid in the subsequent execution of program BNVA.

Next, in one of the most important steps of UGVA, a call to subroutine JSQX results in the running 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, DO LOOP 10 is exited. As many as 22 new updated guess grids (22 calls to BNVA) can be created: 15 grids of temperature, 6 grids of dewpoint and 1 grid of 1000 mb height.



UGVA
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 IGCT/22*Ø/
NGB = IPP(1,Ø)
NGB = Ø
900
CALL SDEST('< FIRST GRID NO. >', Ø)
END
F
CALL DOPEN('VASTEXT', LUN, LEN)
CALL DREAD (LUN, ITERM, IDOC)
L=Ø
N←NGB
N←N+1
N ≤ NGE
10
A
IOK = IGGET(NGFG, N, 32ØØ, IGRID(1), NR, NC, IGHDT)
FOLLOWING NECESSARY BECAUSE OF GRIDS WITH HOLES
ITVNAM = IGHDT(7)
ITLEVL = IGHDT(1Ø)
2
IOK = Ø
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
4-12
A
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
F
L=J
IPRESS(J) = ITLEVL
6
F
MOD(J,15) = 1 AND ITLEVL = 900
6
F
C
10
A
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)
CALL TQ(OBUF)


CALL
JSQX ('BNVA', OBUF(1), OBUF(4), OBUF(7), OBUF(10)
NGOT=Ø
J←1
J≤22
J←J+1
11
D
D
NGOT=NGOT + IGOT(J)
NGOT =22
12
F
A
L=Ø
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) ENCCLR
12) LTQ
13) DECONV
14) FECONV
15) IECONV
16) ZECONV
17) MOVB
18) CLEANA
19) PRLINX
20) PRCLOS
21) PROPEN
22) PRPRPR
23) LOCK
24) PRWR
25) PRRD
26) UNLOCK
27) PRCL
28) POST
29) BLKA
30) ISQX
31) LWCLOS
32) MOVWC
33) LWMOP
34) TOKANL
35) SQSLED
36) PUC
37) WD
38) MOVCW
39) LWPO
40) EDEST
41) MOVW
42) CLEANW
43) LWNEWF
44) LWSO
45) PRIOUT
46) TQSET
47) JMBWTF

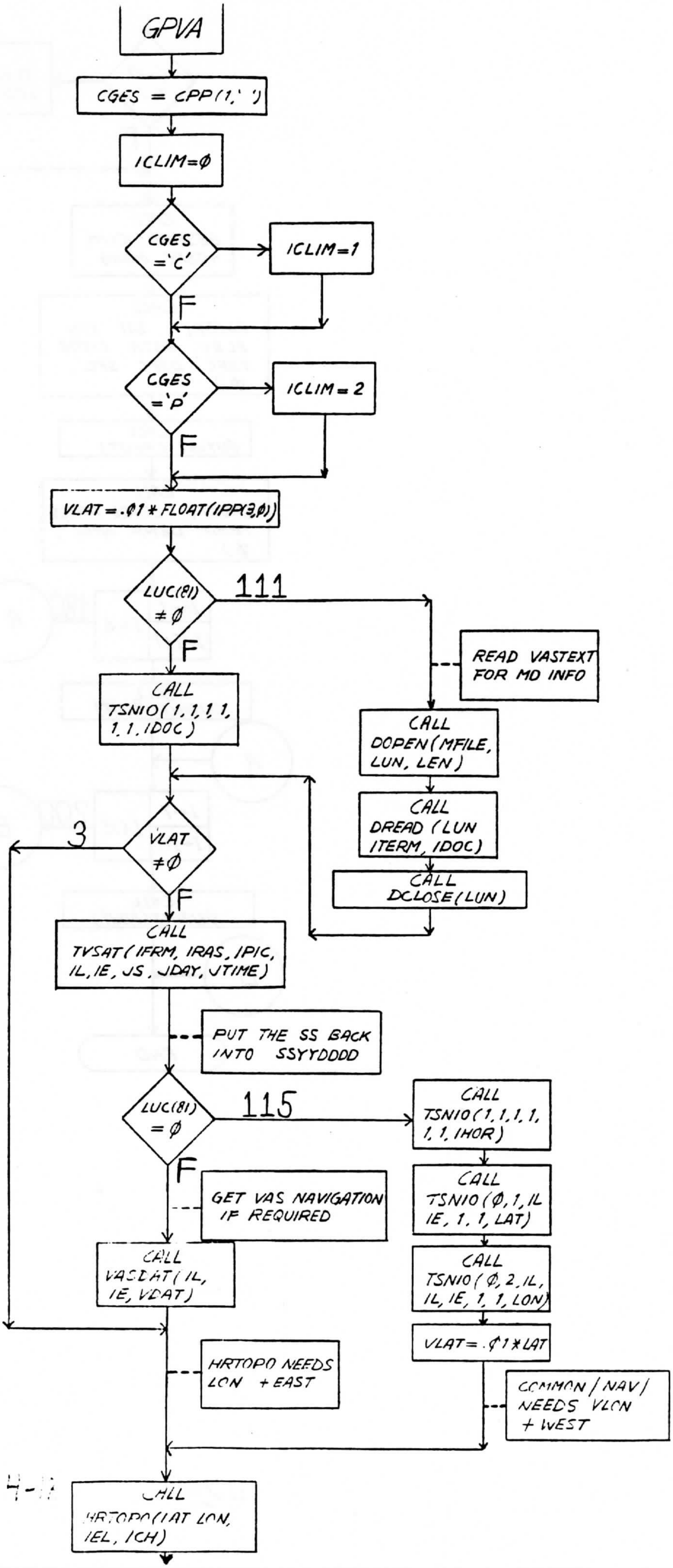
GPVA

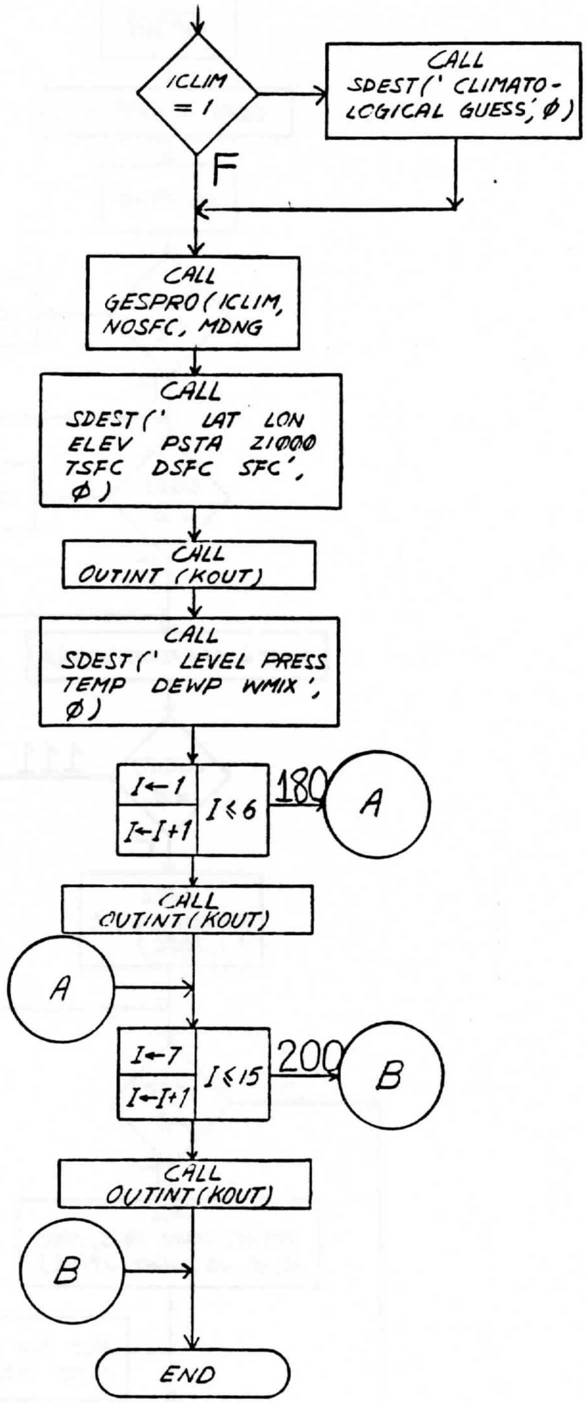
GPVA is used to display the VAS guess profiles of temperature, dewpoint and mixing ratio at either the cursor location (default) or a keyed-in latitude/longitude position. Three different types of guess profiles can be displayed: the guess profile in either a keyed in MD file or the MD file defined under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

First, the type of guess to be displayed (CGES), as well as the surface analysis and debug options, are set. In addition, variable ICLIM is set to 0,1 or 2, depending on the guess profile chosen. 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 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 more important items read from the VASTEXT file is the guess MD file number (MDNG). The rest of the discussion of GPVA, beyond code line 51, will be broken down into two sections, one section assuming the latitude and longitude for the guess have not been keyed in, the other assuming a keyin of both latitude and longitude has been made.

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 latitude and longitude information for the cursor. Next, subroutine HRTOPO returns the surface elevation and characteristic (land or sea), using a global 10 minute resolution topography. Finally, subroutine GESPRO gathers the rest of the guess data according to the guess option, including station 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 this is done, subroutines SDEST and OUTINT print the results on the CRT.

If the latitude and longitude values have been keyed in, there is no need to call VASDAT in code line 62, and therefore GPVA jumps immediately to HRTOPO. From this step to the end, this option proceeds identically to the previously-discussed no latitude/longitude keyin option.


GPVA
CGES = CPP(1,' ')
ICLIM=Ø
CGES ='C'
ICLIM=1
F
CGES ='P'
ICLIM=2
F
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)
CALL DREAD (LUN ITERM, IDOC)
CALL DCLOSE(LUN)
VLAT ≠Ø
3
F
CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, JS, JDAY, JTIME)
PUT THE SS BACK INTO SSYYDDDD
LUC(81) =Ø
115
F
CALL TSNIO(1,1,1,1, 1,1,IHOR)
CALL TSNIO(Ø,1,IL IE,1,1,LAT)
CALL TSNIO(Ø,2,IL, IL,IE,1,1,LON)
VLAT=.Ø1*LAT
GET VAS NAVIGATION IF REQUIRED
CALL VASDAT(IL, IE, VDAT)
HRTOPO NEEDS LON +EAST
COMMON/NAV/ NEEDS VLON +WEST
CALL HRTOPO(LAT LON, IEL, ICH)


ICLIM = 1
F
CALL SDEST(' CLIMATOLOGICAL GUESS', Ø)
CALL GESPRO(ICLIM, NOSFC, MDNG
CALL SDEST(' LAT LON ELEV PSTA Z1ØØØ TSFC DSFC SFC', Ø)
CALL OUTINT (KOUT)
CALL SDEST(' LEVEL PRESS TEMP DEWP WMIX', Ø)
I←1
I←I+1
I≤6
180
A
CALL OUTINT(KOUT)
A
I←7
I←I+1
I≤15
200
B
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) ENCCLR
18) LTQ
19) DECONV
20) FECONV
21) IECONV
22) ZECONV
23) MOVB
24) CLEANA
25) TQ
26) PRLINX
27) PRCLOS
28) PROPEN
29) PRPRPR
30) LOCK
31) PRWR
32) PRRD
33) UNLOCK
34) PRCL
35) POST
36) BLKA
37) LWCLOS
38) MOVWC
39) LWMOP
40) ITOC
41) GETFRM
42) TRMNL
43) TOMES
44) MOVCW
45) NVINIT
46) GETNAV
47) EPOCH
48) GETGAM
49) VASNAV
50) DDEST
51) CLEANW
52) EDEST
53) MOVW
54) READD
55) PLNKIV
56) OPNA
57) LWOPEN
58) LWGET
59) SATEAR
60) SATPOS
61) ANGGET
62) SOLARP
63) ANGLES
64) READOF
65) RBYTSX
66) ZEROS
67) DDEST
68) RDTRK
69) MOVC
70) STDATM
71) RAOBIN
72) CSRAOB
73) MNRAOB
74) SGRAOB
75) CSRAOS
76) CSRAOM
77) CSRAOZ
78) CSRAOP
79) CSRAOR
80) CSRAOI
81) WMIX
82) INTPTW
83) EXTEMP
84) CLMGES
85) CALDAY
86) VASGES
87) SURGES
88) GETSFV
89) IGNAME
90) PROFIX
91) LWPO
92) LWNEWF
93) LWSO
94) PRIOUT
95) TQSET
96) PUC
97) WD
98) JMBWTF

EXVA

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

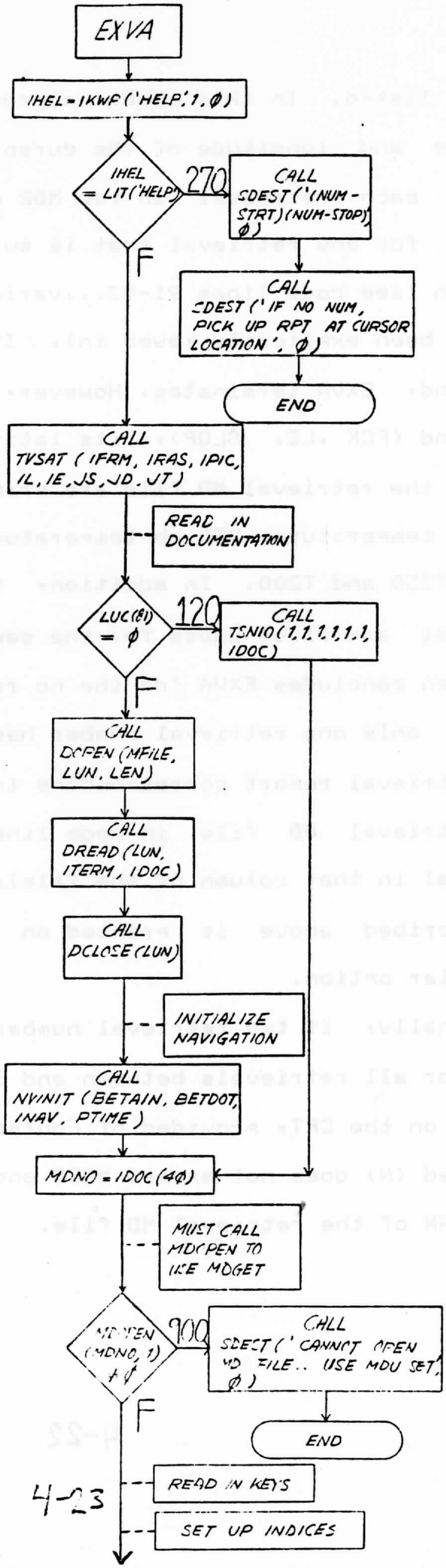
Initially, the cursor coordinates (in lines and elements) are determined, and the VASTEXT file is read. Then, after the navigation has been initialized, the retrieval MD file and row numbers are put, respectively, into the variables MDNO and MDR. These numbers are automatically taken from the VASTEXT 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 such things as the addresses (elements) within the retrieval MD file output array (IOUT) that contain latitude, longitude and surface temperature (NLAT, NLON and NDX, respectively). These addresses will be valid for any retrieval report in the retrieval MD file.

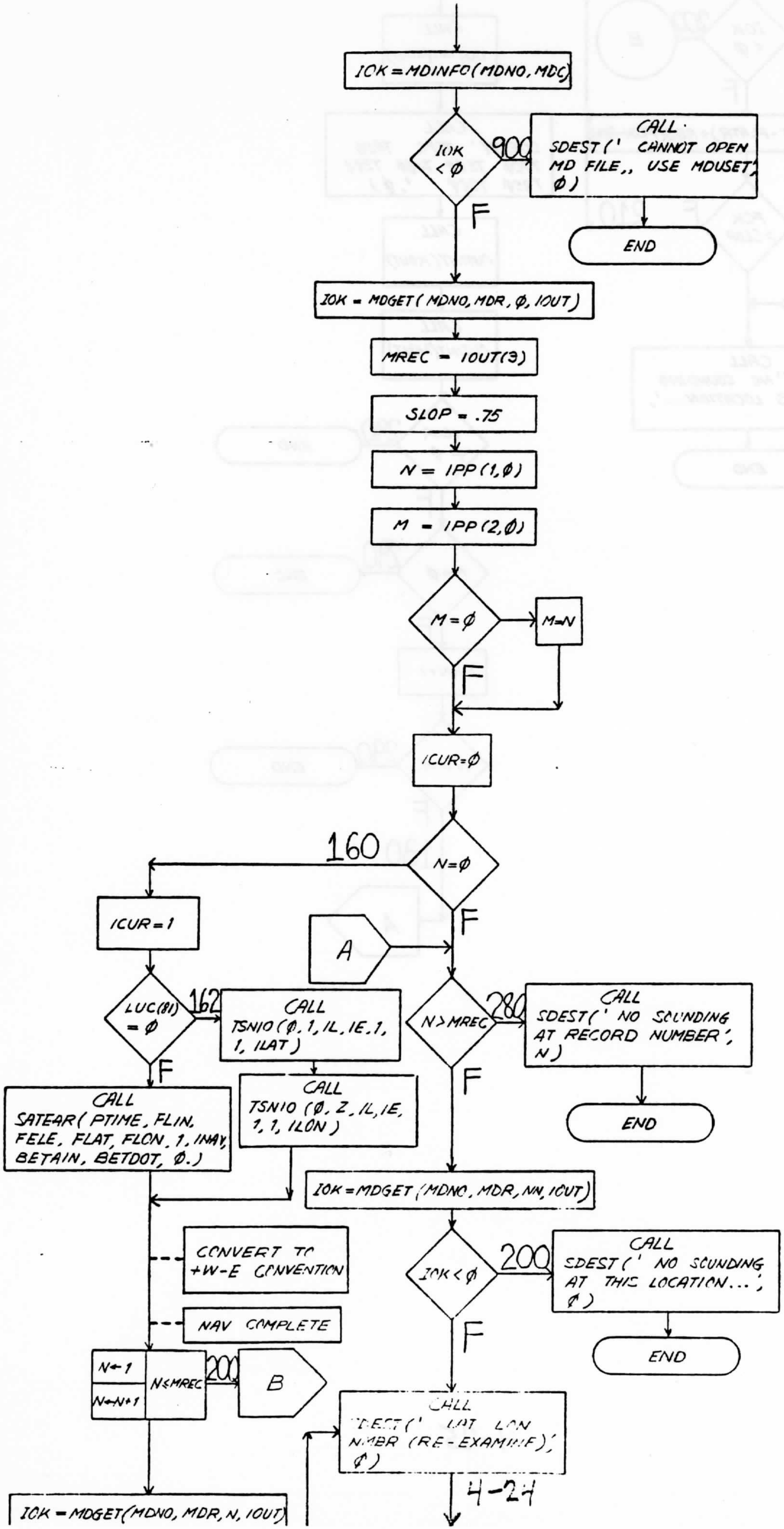
After the existence of the retrieval MD file has been checked with function MDINFO, the row header for row MDR of the file is read, which gives, among other things, the number of retrieval reports for that row of the file (stored in variable MREC). In addition, the numbers of the first and last retrievals whose data is to be displayed, if more than one retrieval report is desired, are read into variables N and M.

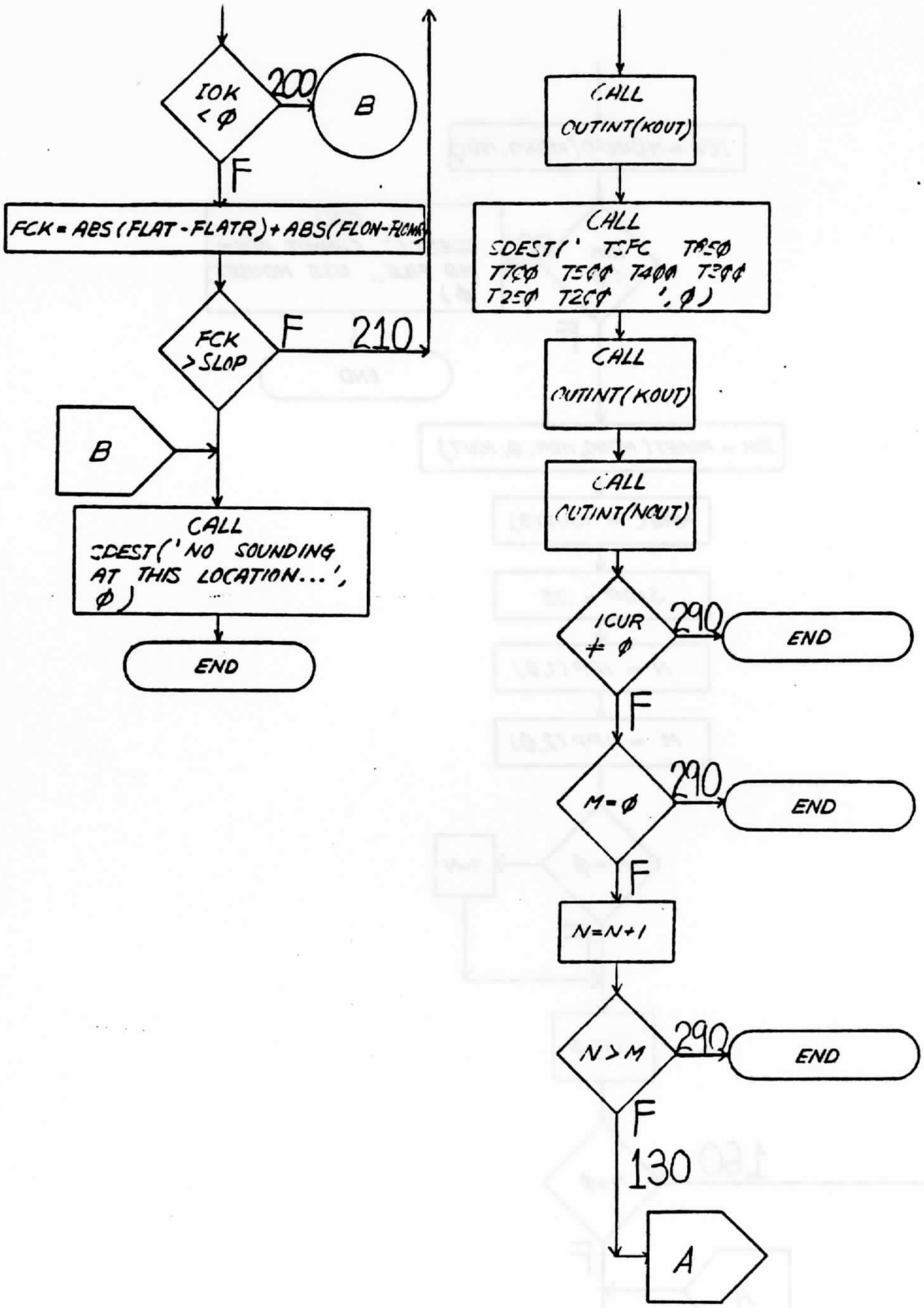
Assuming no retrieval number or numbers have been keyed in (N, M .EQ. 0), only the retrieval report at the cursor location will be listed. In this case, subroutine SATEAR calculates the latitude and longitude of the cursor. Then, DO LOOP 200 goes through each retrieval in row MDR of the retrieval MD file, looking for any retrieval that is suitably close to the cursor location (see code lines 91-92...variable SLOP=.75 degrees, if it has not been explicitly keyed in). 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 the retrieval MD file are printed on the CRT, as well as surface temperature, 850 mb temperature (T850), T700, T500, T400, T300, T250 and T200. In addition, the 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), the retrieval report corresponding to this number is read from the retrieval MD file in code line 66 (assuming there is a retrieval in that column of the file), and the same information as described above is printed on the CRT, concluding this particular option.

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 retrieval number to be displayed (N) does not exceed MREC and that there is a record in column NN of the retrieval MD file.


EXVA
IHEL=IKWF('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
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
CALL DOPEN (MFILE, LUN, LEN)
CALL DREAD (LUN, ITERM, IDOC)
CALL DCLOSE (LUN)
INITIALIZE NAVIGATION
CALL NVINIT (BETAIN, BETDOT, INAV, PTIME)
MDNO = IDOC(4Ø)
MUST CALL MDOPEN TO USE MDGET
MDOPEN (MDNO, 1) ≠ Ø
900
CALL SDEST(' CANNOT OPEN MD FILE.. USE MDU SET',Ø)
F
END
READ IN KEYS
SET UP INDICES


IOK = MDINFO(MDNO, MDC)
IOK < Ø
900
CALL SDEST(' CANNOT OPEN MD FILE,, USE MDUSET', Ø)
END
F
IOK = MDGET(MDNO, MDR, Ø, IOUT)
MREC = IOUT(3)
SLOP = .75
N = IPP(1,Ø)
M = IPP(2,Ø)
M = Ø
M=N
F
ICUR=Ø
N=Ø
160
F
ICUR=1
A
LUC(81) = Ø
162
CALL TSNIO(Ø, 1, IL, IE, 1, 1, ILAT)
CALL TSNIO (Ø, 2, IL, IE, 1, 1, ILON)
F
CALL SATEAR(PTIME, FLIN, FELE, FLAT, FLON, 1, INAV, BETAIN, BETDOT, Ø.)
N > MREC
280
CALL SDEST(' NO SOUNDING AT RECORD NUMBER', N)
END
F
IOK = MDGET (MDNO, MDR, NN, IOUT)
CONVERT TO +W-E CONVENTION
IOK < Ø
200
CALL SDEST(' NO SOUNDING AT THIS LOCATION...', Ø)
END
F
NAV COMPLETE
N←1
N←N+1
N≤MREC
200
B
CALL SDEST(' LAT LON NMBR (RE-EXAMINE)', Ø)
IOK = MDGET(MDNO, MDR, N, IOUT)


IOK
< Ø
200
B
F
FCK = ABS (FLAT - FLATR) + ABS (FLON-FLONR)
FCK
> SLOP
F
210
B
CALL
SDEST('NO SOUNDING
AT THIS LOCATION...',
Ø)
END
CALL
OUTINT(KOUT)
CALL
SDEST(' TSFC T85Ø
T7CØ T5CØ T4ØØ T3ØØ
T25Ø T2CØ ', Ø)
CALL
OUTINT(KOUT)
CALL
OUTINT(NOUT)
ICUR
≠ Ø
290
END
F
M = Ø
290
END
F
N=N+1
N > M
290
END
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) ENCCLR
35) DECONV
36) FECONV
37) IECONV
38) ZECONV
39) LWCLOS
40) MOVWC
41) LWMOP
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) LWPO
55) LWNEWF
56) LWSO
57) PRIOUT
58) TQSET
59) PUC
60) WD
61) JMBWTF

ESVA

This program edits surface reports at the cursor location, and can operate essentially in two modes. First, it can delete any combination of 1000 mb height, mean sea level temperature or station dewpoint depression 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. Furthermore, if there is no report at the cursor location, one is added. Moreover, there is also an option to convert surface temperature to mean sea level temperature. Finally, note that ESVA works only for retrieval surface MD files (schema RSVC).

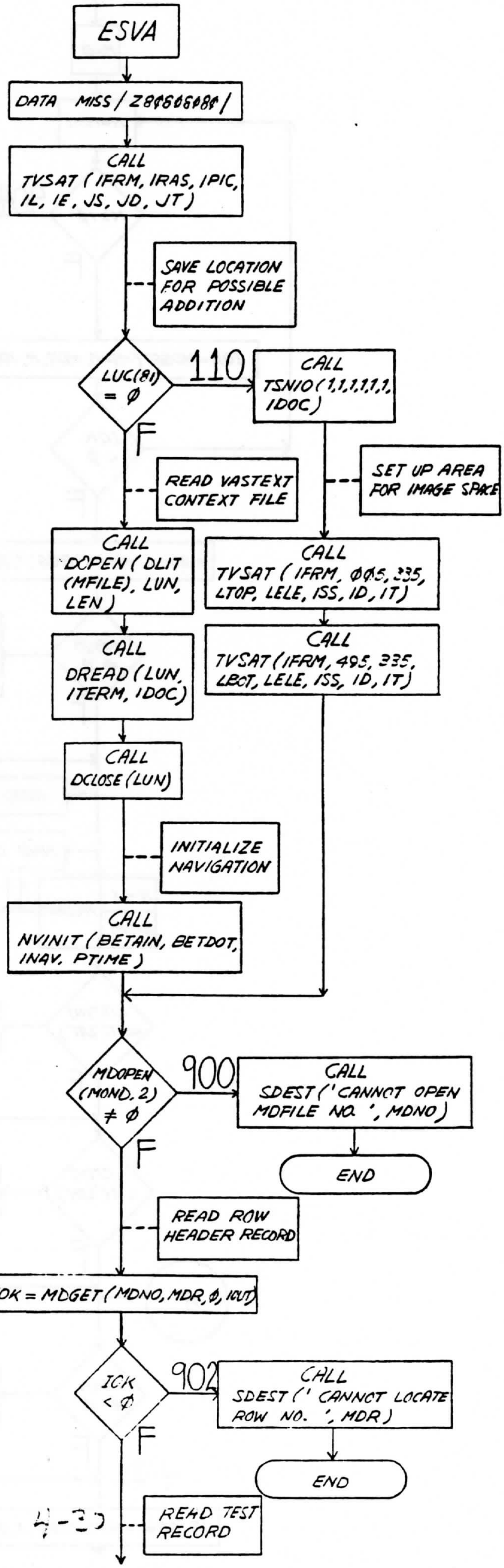
First, the code in lines 39-41 calculates the cursor line and element positions (IL,IE). Next, the VASTEXT file is read, the navigation is initialized, and MDNO is set to IDOC (36), which corresponds to the retrieval surface MD file in the VASTEXT file. This MD file is then opened for read/write, and the row header corresponding to variable MDR (and also to the valid time of the surface data) is read via function MDGET in code line 70. Then, after a test has been performed to assure that data exists in the retrieval surface MD file, the keys for the file are read in via function MDKEYS. This allows ESVA to locate the positions within the array IOUT (which will hold the surface data accessed by function MDGET) of the variables to be edited. Following DO-loop 11, the values keyed in for Z100, (1000 mb height), TSL (mean sea level temperature) and(or) DD (surface dewpoint depression) are read using function IKWP, and the value of variable ITEM is determined. Note that if "ESVA" alone is keyed in, the program will respond with "NO ACTION INDICATED AT THIS LOCATION" and terminate.

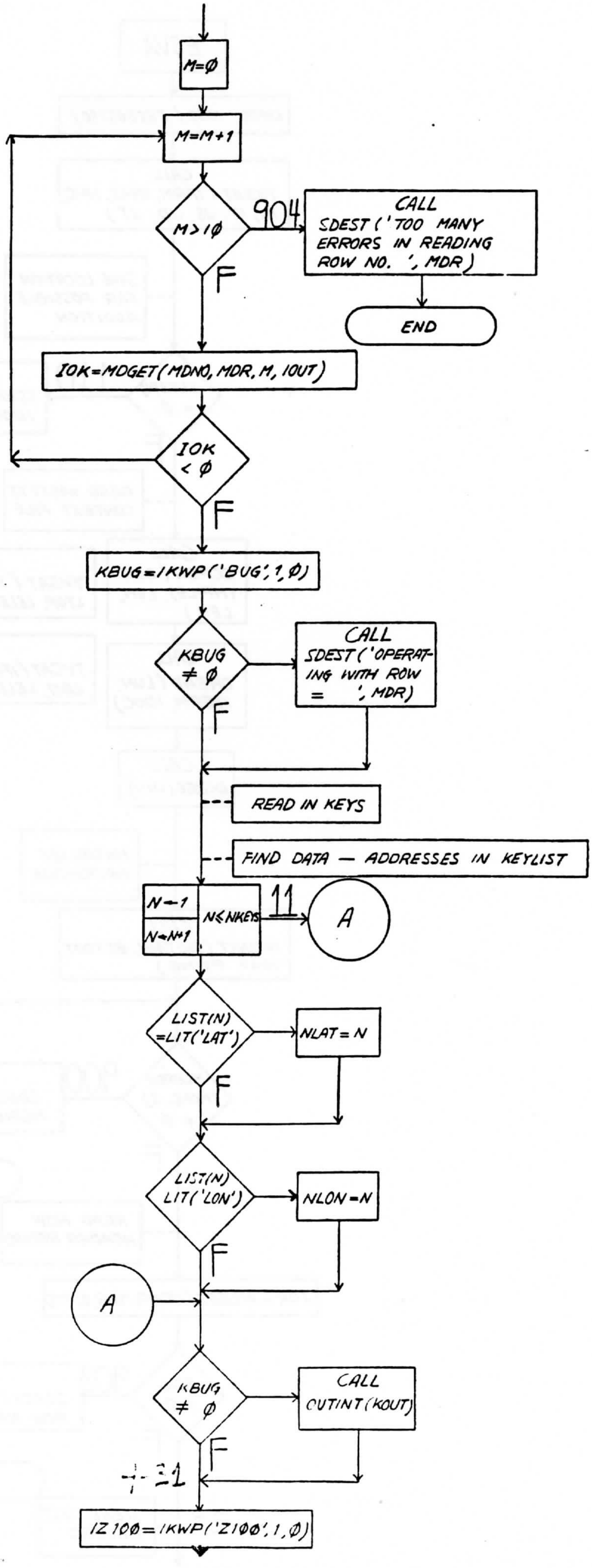
After frame information is accessed (GETFRM) and the plot package is started by subroutine INITPL, the cursor dimensions in latitude and longitude are determined, first by setting the NW corner of the cursor in satellite coordinates (line, element) and then earth coordinates (latitude, longitude), and then repeating the process for the SE corner of the cursor.

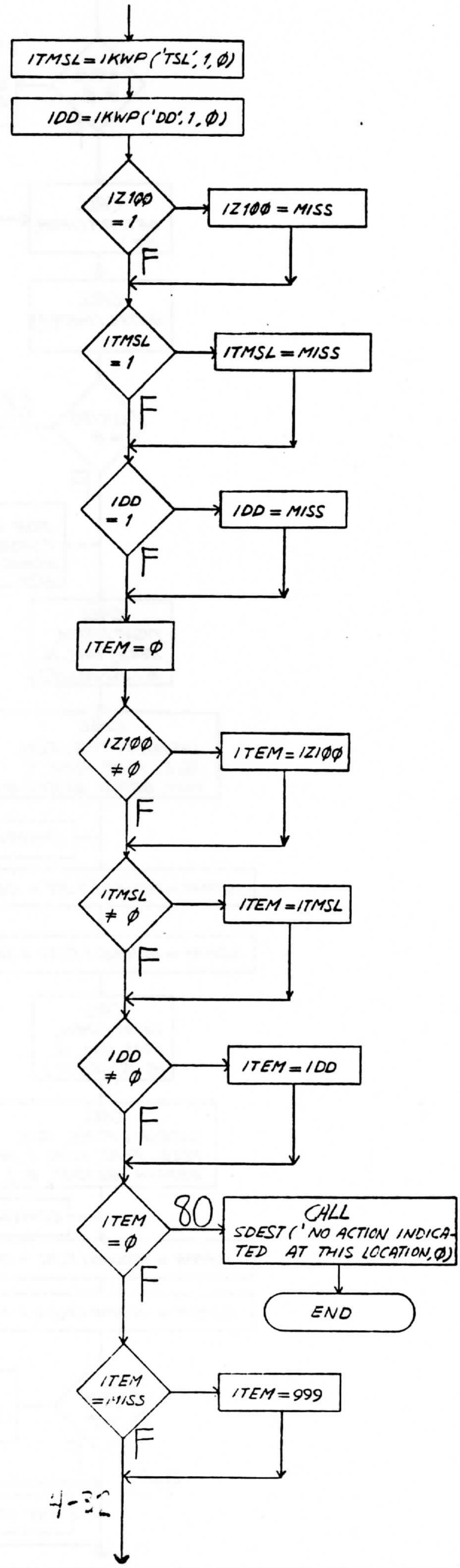
The next step involves the retrieval surface MD file itself. A large DO-loop (DO loop 70) reads MREC (= IOUT(3) in row header) MD file entries and checks to see if a given entry is within the cursor box (code lines 179-180). Note that the cursor dimensions, if large enough, may encompass more than one surface report. Following this, the TV coordinates of the report (raster, pixel) are obtained via subroutines SATEAR and SATTV. Now, the array IOUT is updated with the deletions or new values of Z100, TSL and(or) DD and is subsequently passed back into the retrieval surface MD file via function MDPUT in code line 209. In addition, variable ITEM is plotted on the video screen at the cursor location via subroutine PLTDIG, and control returns to code line 172, where the next surface report will be read.

After DO-loop 70 is exited, and assuming at least one report was located within the cursor box (NDAT .GT. 0), control transfers to statement 100, where an option to pack the retrieval surface MD file can 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 data within a cursor box. The following paragraph describes the option when data is to be edited into an area where no prior data existed.

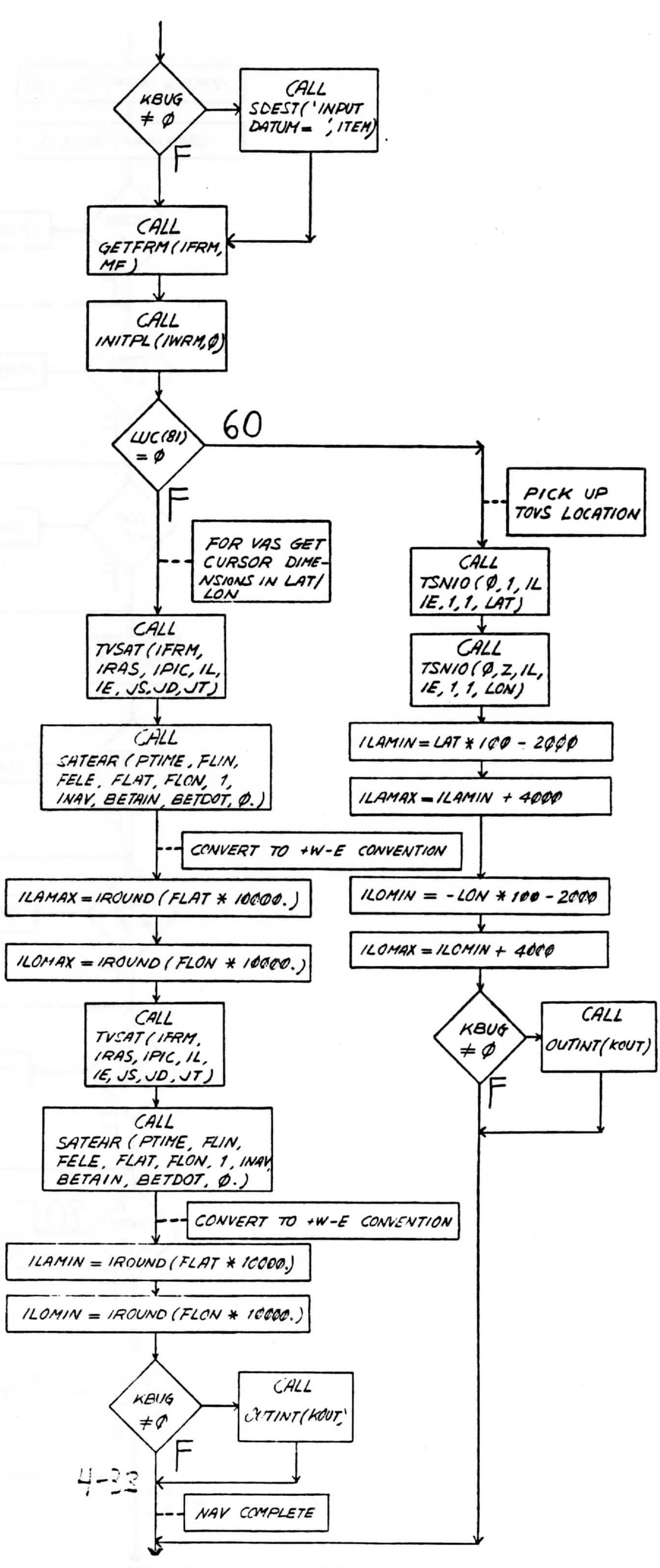
The discussion within this paragraph concerns the code beginning at line 223. ESVA functions up to DO-loop 70 in this mode in exactly the same way as it did when reports within the cursor were to be edited. However, DO-loop 70 will not do anything this time, since all the existing surface retrieval MD file reports will be outside the cursor boundaries. Hence, variable NDAT will equal 0 at code line 218, and control will transfer to code line 223. At this point, if a value or values of Z100, TSL, DD and(or) TSFC (surface temperature) 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 is now at code line 237, and the keyed in data is placed in the retrieval surface MD file one column beyond the last report (column IOUT(3) + 1). In addition, the row header is rewritten into the MD file, and variable ITEM is plotted on the video screen via subroutine PLTDIG. Finally, the PACK option is again encountered, after which ESVA terminates.


ESVA
DATA MISS / Z8Ø8Ø8Ø8Ø /
CALL TVSAT (IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)
SAVE LOCATION FOR POSSIBLE ADDITION
LUC(81) = Ø
110
F
CALL TSNIO (1,1,1,1,1,1, IDOC)
READ VASTEXT CONTEXT FILE
SET UP AREA FOR IMAGE SPACE
CALL DOPEN (DLIT (MFILE), LUN, LEN)
CALL TVSAT (IFRM, ØØ5, 335, LTOP, LELE, ISS, ID, IT)
CALL DREAD (LUN, ITERM, IDOC)
CALL TVSAT (IFRM, 495, 335, LBOT, LELE, ISS, ID, IT)
CALL DCLOSE (LUN)
INITIALIZE NAVIGATION
CALL NVINIT (BETAIN, BETDOT, INAV, PTIME)
MDOPEN (MOND, 2) ≠ Ø
900
F
CALL SDEST (' CANNOT OPEN MDFILE NO. ', MDNO)
END
READ ROW HEADER RECORD
IOK = MDGET (MDNO, MDR, Ø, IOUT)
IOK < Ø
902
F
CALL SDEST (' CANNOT LOCATE ROW NO. ', MDR)
END
READ TEST RECORD

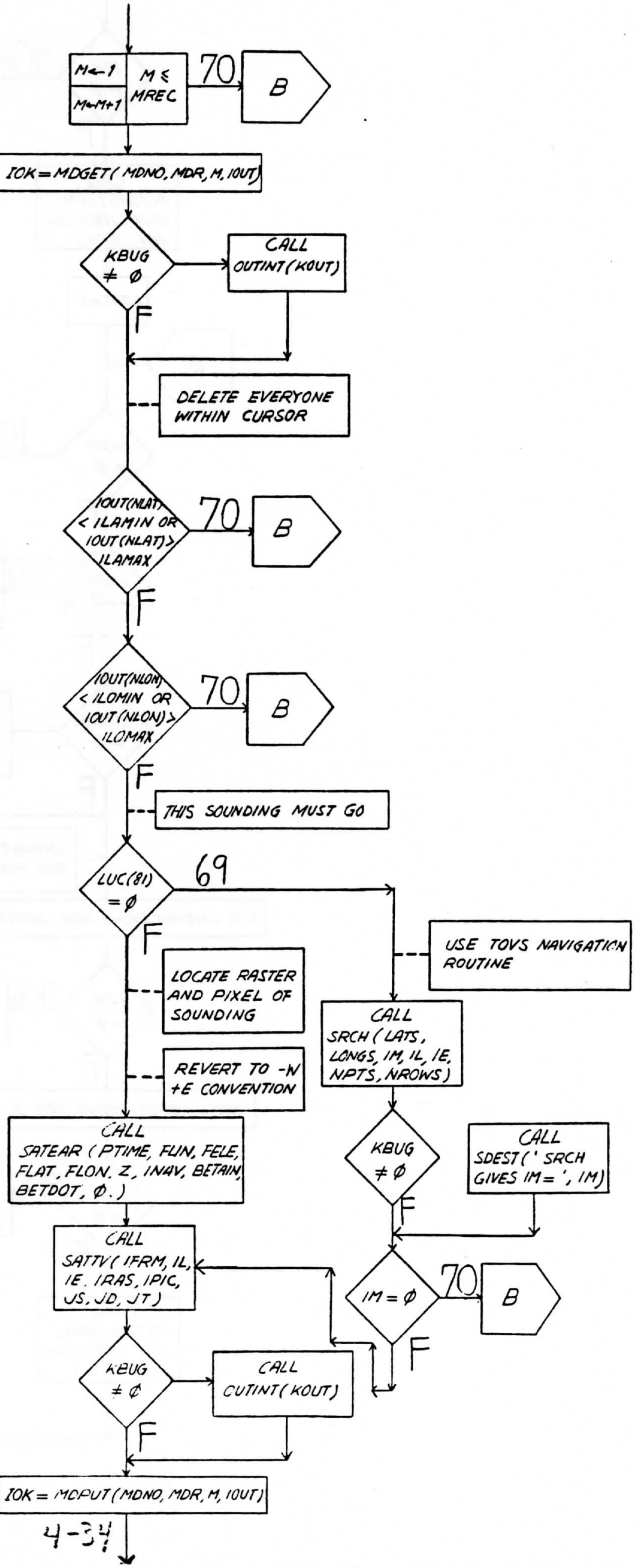

M=Ø
M=M+1
M>1Ø
904
CALL SDEST('TOO MANY ERRORS IN READING ROW NO. ', MDR)
END
F
IOK=MDGET(MDNO, MDR, M, IOUT)
IOK < Ø
F
KBUG=IKWP('BUG',1,Ø)
KBUG ≠ Ø
CALL SDEST('OPERATING WITH ROW = ', MDR)
F
READ IN KEYS
FIND DATA — ADDRESSES IN KEYLIST
N←1
N←N+1
N≤NKEYS
11
A
LIST(N) =LIT('LAT')
NLAT=N
F
LIST(N) LIT('LON')
NLON=N
F
A
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
21
IZ1ØØ=IKWP('Z1ØØ',1,Ø)



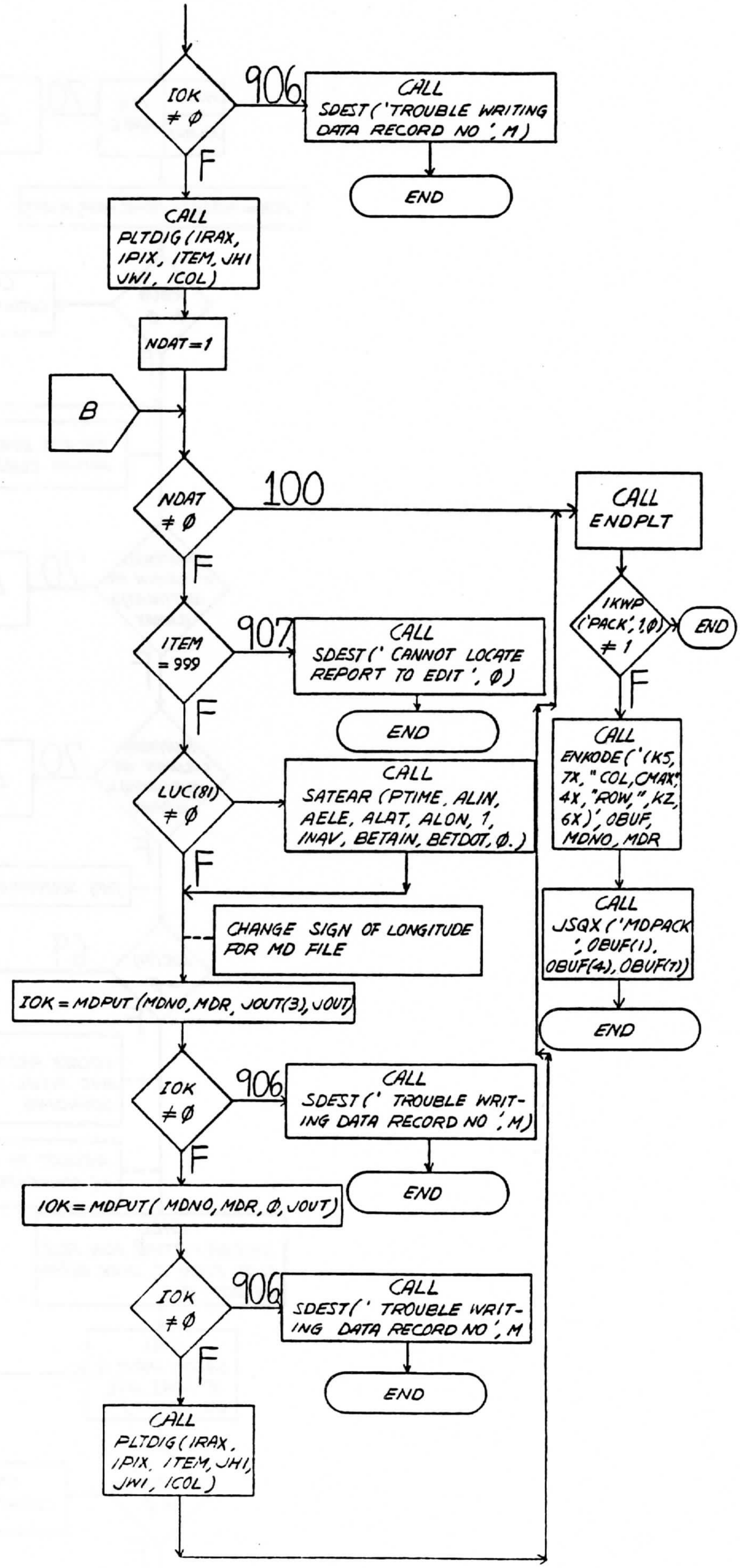
ITMSL=IKWP('TSL',1,Ø)
IDD=IKWP('DD',1,Ø)
IZ1ØØ =1
IZ1ØØ = MISS
F
ITMSL =1
ITMSL=MISS
F
IDD =1
IDD = MISS
F
ITEM=Ø
IZ1ØØ ≠Ø
ITEM=IZ1ØØ
F
ITMSL ≠ Ø
ITEM=ITMSL
F
IDD ≠ Ø
ITEM=IDD
F
ITEM =Ø
80
CALL SDEST('NO ACTION INDICATED AT THIS LOCATION,Ø)
END
F
ITEM =MISS
ITEM=999
F



KBUG ≠ Ø
CALL SDEST('INPUT DATUM= ', ITEM)
F
CALL GETFRM(IFRM, MF)
CALL INITPL(IWRM,Ø)
LUC(81) = Ø
60
F
PICK UP TOVS LOCATION
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 * 10ØØØ.)
ILOMIN = - LON * 1ØØ - 2ØØØ
ILOMAX = IROUND(FLON * 10ØØØ.)
ILOMAX = ILOMIN + 4ØØØ
CALL TVSAT(IFRM, IRAS, IPIC, IL, IE, JS, JD, JT)
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
CALL SATEHR(PTIME, FLIN, FELE, FLAT, FLON, 1, INAV, BETAIN, BETDOT, Ø.)
CONVERT TO +W-E CONVENTION
ILAMIN = IROUND(FLAT * 10ØØØ.)
ILOMIN = IROUND(FLON * 10ØØØ.)
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
NAV COMPLETE



M←1
M←M+1
M ≤ MREC
70
B
IOK = MDGET(MDNO, MDR, M, IOUT)
KBUG ≠ Ø
CALL OUTINT(KOUT)
F
DELETE EVERYONE WITHIN CURSOR
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
LOCATE RASTER AND PIXEL OF SOUNDING
REVERT TO -W +E CONVENTION
CALL SRCH (LATS, LONGS, IM, IL, IE, NPTS, NROWS)
CALL SATEAR (PTIME, FLIN, FELE, FLAT, FLON. Z, INAV, BETAIN, BETDOT, Ø.)
KBUG ≠ Ø
CALL SDEST(' SRCH GIVES IM= ', IM)
F
CALL SATTV(IFRM, IL, IE, IRAS, IPIC, JS, JD, JT)
IM = Ø
70
B
F
KBUG ≠ Ø
CALL CUTINT(KOUT)
F
IOK = MDPUT(MDNO, MDR, M, IOUT)


IOK ≠ Ø
906
CALL SDEST('TROUBLE WRITING DATA RECORD NO ', M)
END
F
CALL PLTDIG(IRAX, IPIX, ITEM, JHI JWI, ICOL)
NDAT=1
B
NDAT ≠ Ø
100
CALL ENDPLT
IKWP ('PACK', 1,Ø) ≠ 1
END
F
CALL ENKODE('(K5, 7X, "COL,CMAX" 4X, "ROW," KZ, 6X)', OBUF, MDNO, MDR
CALL JSQX('MDPACK', OBUF(1), OBUF(4), OBUF(7))
END
F
ITEM = 999
907
CALL SDEST(' CANNOT LOCATE REPORT TO EDIT', Ø)
END
F
LUC(81) ≠ Ø
CALL SATEAR(PTIME, ALIN, AELE, ALAT, ALON, 1, INAV, BETAIN, BETDOT, Ø.)
F
CHANGE SIGN OF LONGITUDE FOR MD FILE
IOK = MDPUT(MDNO, MDR, JOUT(3), JOUT)
IOK ≠ Ø
906
CALL SDEST(' TROUBLE WRITING DATA RECORD NO ', M)
END
F
IOK = MDPUT(MDNO, MDR, Ø, JOUT)
IOK ≠ Ø
906
CALL SDEST(' TROUBLE WRITING DATA RECORD NO ', M
END
F
CALL PLTDIG(IRAX, IPIX, ITEM, JHI, JWI, ICOL)

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) ENCCLR
43) DECONV
44) FECONV
45) IECONV
46) ZECONV
47) LWCLOS
48) MOVWC
49) LWMOP
50) GETNAV
51) EPOCH
52) GETGAM
53) VASNAV
54) DDEST
55) MOVCW
56) CLEANW
57) EDEST
58) MOVW
59) II
60) DWRITE
61) SATPOS
62) QGDASH
63) DSHOFF
64) PLOT
65) DSHON
66) PUC
67) WD
68) ENPT
69) PENADD
70) PACK
71) SENOUT
72) ATOE
73) TEKPUT
74) ISQX
75) TOKANL
76) SQSLED
77) LWPO
78) LWNEWF
79) LWSO
80) PAGE
81) PENMOV
82) BOX
83) PENBEG
84) WALK
85) PRIOUT
86) TQSET
87) JMBWTF

CHAPTER 5

Optional VAS Retrieval Software

The routines in this chapter are to be considered as being purely optional within the retrieval scheme. They are usually used only in certain specific situations, and do not have applications beyond this realm.

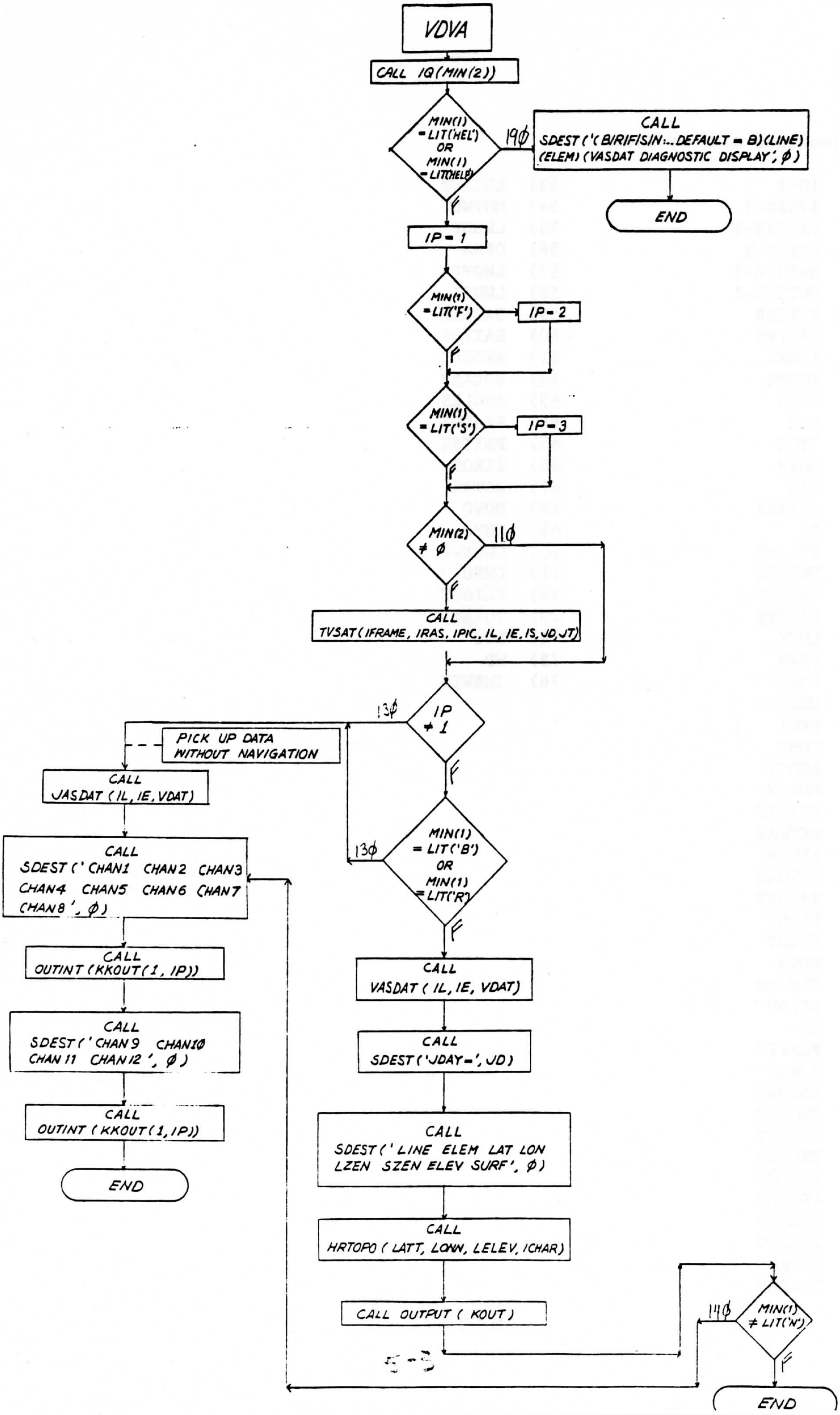
VDVA

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 necessary to the generation of a retrieval or area of retrievals; rather, VDVA can be used to check an individual sounding (field of view) to make sure no unreasonable brightness temperatures or other unreasonable quantities exist.

Five different quantities for each available VAS band can be printed, depending on the positional parameter denoted in the keyin: B (brightness temperature), R (radiance), F (filter), S (spin budget) and N (navigation). All five options (as well as the default option-- VDVA keyed in by itself) require the cursor position to be in satellite coordinates (line-IL, element-IE) in order to read the VAS data. If IL and IE have not been keyed in, subroutine TVSAT will calculate their values in code line 40.

For the default option (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 only 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 and satellite zenith angles, land/ocean 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/ocean elevation, and surface characteristic. Option R only returns radiance data for the twelve bands. Option F returns the filter (sensor) number being used for each band, while the last option, S, simply returns the spin budget for each band.

Finally, if one keys in VDVA followed by one of the five explicit positional parameters described above (B, N, R, F or S), two zeroes, and then a "1" (keyin of "VDVA B 0 0 1", for example), the information noted above for each parameter, as well as further diagnostic information about the sounder area directory (calibration, raw data counts, etc.) is printed on the CRT.


VDVA
CALL IQ(MIN(2))
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)
END
F
IP = 1
MIN(1) = LIT('F')
IP = 2
F
MIN(1) = LIT('S')
IP = 3
F
MIN(2) ≠ Ø
110
F
CALL
TVSAT(IFRAME, IRAS, IPIC, IL, IE, IS, JD, JT)
130
IP ≠ 1
F
PICK UP DATA WITHOUT NAVIGATION
CALL
VASDAT (IL, IE, VDAT)
CALL
SDEST (' CHAN1 CHAN2 CHAN3 CHAN4 CHAN5 CHAN6 CHAN7 CHAN8 ', Ø)
CALL
OUTINT (KKOUT(1, IP))
CALL
SDEST (' CHAN9 CHAN10 CHAN11 CHAN12 ', Ø)
CALL
OUTINT (KKOUT(1, IP))
END
130
MIN(1) = LIT('B') OR MIN(1) = LIT('R')
F
CALL
VASDAT (IL, IE, VDAT)
CALL
SDEST ('JDAY=', JD)
CALL
SDEST (' LINE ELEM LAT LON LZEN SZEN ELEV SURF', Ø)
CALL
HRTOPO (LATT, LONN, LELEV, ICHAR)
CALL OUTPUT (KOUT)
140
MIN(1) ≠ LIT('N')
F
END

Subroutines used by VDVA:

1) IO-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) ENCCLR
47) DECONV
48) FECONV
49) IECONV
50) ZECONV
51) DREAD
52) DCLOSE
53) LWCLOS
54) MOVWC
55) LWMOP
56) OPNA
57) LWOPEN
58) LWGET
59) SATEAR
60) SATPOS
61) ANGGET
62) SOLARP
63) ANGLES
64) READOF
65) RBYTSX
66) ZEROS
67) RDTRK
68) MOVC
69) LWPO
70) LWNEWF
71) LWSO
72) PRIOUT
73) TQSET
74) PUC
75) WD
76) JMBWTF

VTPX

VTPX is the program used for retrieving total precipitable water vapor from VAS radiance measurements from one individual field of view (FOV). Note that this is much different from the usual temperature retrievals discussed in SRAD and SRET, because those retrievals were made using brightness temperature measurements averaged over anywhere from 9 to 121 FOVs. Note also in code lines 23-27 which pointers must be set before the execution of the program. All of these pointers would be set by the time a typical sounder area was processed; however, if VTPX is used at some other time, any pointers not set must be set using program SPVA. In addition, the retrieval MD file used to store the results of VTPX must be created using schema "VTWV", as opposed to schema "VRET", which is the schema normally used for retrieval MD files.

Initially, the digital area number to be used for storing the image of the precipitable water vapor results is determined via keyword parameter "ARA" in code line 66. This should be included in the program keyin. Next, subroutine VRTIO reads both the VASTEXT file and the retrieval MD file (MDNR) row header, after which variables MDNG, MDNR and MDRR are set to VASTEXT values (through common block DOC and hence array IDOC). If the VASTEXT file has not been previously set, these three variables might not have the values the user desires. This is therefore a good example showing why the VASTEXT file must be set up with program SPVA before VTPX is executed, if the VASTEXT file has not been set up by prior processing. Then, after variable LRO is determined, the type of guess (climatology or grid (default)), as well as the surface analysis option, are stored in variables CGES and NOSFC. Next, the size in FOVs of one side of a given retrieval box is determined (11 by default) and stored in variable NBXS, and the line and element retrieval box spacing, last line and element defining a retrieval area, etc. are also set. Each retrieval box will consist of NBXS * NBXS individual retrievals (FOVs), meaning there may be as many as 121 retrievals in a given retrieval box.

After the row header for retrieval MD file MDNR has been written into the file via function MDPUT in code line 119, the initial line and element positions of the area to be processed are determined by calculating the line/element coordinates of the cursor using subroutine TVSAT. This assumes, of course, that the initial line and element positions have not been keyed in. Note also that this means the cursor should be placed near the NW corner of the sounder area listed in the VASTEXT file (IDOC(35)) before VTPX is run. However, there are provisions made in code lines 122-130 for either the case when the initial line and element have been keyed in, and the processing is being done on a video terminal, or the case when the initial line and element have been keyed in, but the processing is being done on a non-video terminal. Then, if the last line and element positions also have not been keyed in, only one retrieval at the cursor location will be performed. Normally, the last line and element positions (SE corner of sounder area image) will be keyed in, and the initial positions will be determined from the cursor location, similar to program SRAD. Note also that VTPX will normally be run on a video terminal (LUC(16)) .NE. 0), as opposed to a non-video terminal.

At this point, VTPX calls subroutine VASDAT, which returns navigation data only, because VDAT(1) is set to -1. before VASDAT is called. Important information returned 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 precipitable water vapor retrieval box (INCIL,INCIE).

Next, assuming the area number (NARA) .NE. 0, the exact line and element coordinates of the NW and SE corners of the image are calculated (coordinates of the center of a retrieval box returned by subroutine TVSAT) in code lines 162-172, and the number of lines and elements in the area to be processed are calculated and placed in variables MLIN and MELE, respectively. Then, a file and directory for the digital area that will be produced by VTPX (using the precipitable water vapor retrieval results) are generated via subroutines ARASIZ and ENAREA, respectively.

After variables KLINES and KELEMS (number of FOVs from top to bottom and side to side, respectively, in a retrieval box) are set in code lines 180-181, VTPX enters DO-loop 920, which comprises almost all of the remaining program. Nested within DO-loop 920 are DO-loops 880, 870 and 860. Together, these four loops access VAS data for the sounder area to be used in the precipitable water vapor retrievals. Data is accessed line by line within the sounder area, retrieval box by retrieval box within each line, and from the NW to the SE corner within each retrieval box, moving successively FOV by FOV to the right within each line of a given box. Note that a "sounder area" line is 11 FOVs wide from top to bottom, whereas a "retrieval box" line is only 1 FOV wide. The net result is a retrieval attempt for each FOV within the defined retrieval area (ILINE, LLINE, IELEM and LELEM).

The VAS data itself is accessed by a call to subroutine VASDAT in code line 215. Then, several tests are performed in code lines 217-222 to see if a retrieval should be attempted. Note especially that a retrieval will not be attempted if either the band 7 or 8 brightness temperatures are missing. DO-loop 340 then checks bands KTCH through 8 to determine if not totally unreasonable brightness temperature data exists in each of these bands. If there is reasonable data, variable MSAM is incremented once. In addition, the spin budget for each band not missing is saved in array NSPIN. If MSAM is never incremented (i.e., each band from KTCH through 8 either has bad data or is missing), VTPX will move to the next FOV to attempt a new retrieval. For the first FOV to be processed in a given retrieval box (the NW corner FOV), variable ITFLG=1, and subroutine GESPRO accesses the guess information, depending on the type of guess chosen. Within GESPRO, several surface parameters are determined, including surface pressure, temperature, dewpoint, and mixing ratio. It should be noted that only one guess profile is generated for each 11 X 11 retrieval box. When the first retrieval in a given box is completed successfully, variable ITFLG is set=0 by subroutine VASTPW, meaning the call to GESPRO (code line 240) will be skipped thereafter for the rest of the attempted retrievals within the box. After the surface parameters have been calculated, DO-loop 370 calculates the first pressure level below the surface for the retrieval box. Following this, surface information, in conjunction with the first pressure level below the surface, is used to adjust the lower levels of the guess temperature and moisture profiles in code lines 258-270.

After the guess profile (if LBUG .NE. 0) and other information concerning the FOV location (line, element, etc.) is (are) displayed (depending on the value(s) of variables LCRT and LBUG), subroutine VASTPW calculates the total precipitable water vapor for the FOV in question, returning the result via output parameter URET. The maximum possible value of URET allowed by VASTPW is 12 cm.

If VASTPW is successful, variable IFAIL=0, variable KFLAG=2, and variable NDONE is incremented by 1. Note that only every fourth retrieval is outputted to the retrieval MD file or, in other words, when variable MDOUT=1. The code from approximately lines 304 to 346 in part fills output buffer array IRET with retrieval data for insertion into column LRO of the retrieval MD file via function MDPUT in code line 341. In addition, the row header is updated to show the new number of retrievals through MDPUT in code line 344. Finally, if the plot option is in effect (IPLT .NE. 0), a call to subroutines SATTV and VASDIG in code lines 323 and 327 will output the value of precipitable water vapor to the video screen at the current FOV screen location.

Near the end of VTPX, after all the retrievals for a given sounder area line have been processed, DO-loops 900 and 890 prepare the precipitable water vapor results stored previously in array IARRAY for line by line input into subroutine PACK. (Note that the lines I speak of here are lines within the sounder area, as opposed to lines within retrieval boxes. Remember that these lines are 11 FOVs in width, while retrieval box lines are only 1 FOV in width). Subroutines PACK and WRITA write the precipitable water data into digital area NARA for each line of the sounder area. Then, a message saying "FINISHED LINE" is displayed on the CRT, and VTPX moves to the next line within the sounder area to attempt further retrievals (DO-loop 920 increments once). Therefore, after VTPX has finished, a call to system program DF, which will load digital area NARA into a given image frame, will display an image of precipitable water vapor on the video screen, as opposed to the original band 8 sounder area image.

After all the retrievals in the sounder area have been generated, the row header of the retrieval MD file is rewritten one last time, and the VASTEXT file is updated with a value indicating the total number of retrievals (stored in variable LRO). This is accomplished by the call to subroutine VRTIO in code line 373.

Finally, the output digital area (NARA) is closed, and a summary of the failures encountered during the processing of the sounder area (array NFAIL) is printed on the CRT.

Subroutines used by VTPX:

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) ENCCLR
25) LTQ
26) DECONV
27) FECONV
28) IECONV
29) ZECONV
30) MOVB
31) CLEANA
32) TQ
33) PRLINX
34) PRCLOS
35) LOCK
36) PRWR
37) PRRD
38) UNLOCK
39) PRCL
40) POST
41) PROPEN
42) BLKA
43) PRPRPR
44) DOPEN
45) DREAD
46) MOVCW
47) II
48) STC
49) DWRITE
50) ENCODE
51) DCLOSE
52) LWCLOS
53) MOVWC
54) LWMOP
55) LWPO
56) TRMNL
57) LWNEWF
58) DDEST
59) LWSO
60) CLEANW
61) MOVW
62) LWOPEN
63) LWGET
64) READDL
65) TQMES
66) ITOC
67) GETDAY
68) GETTIM
69) WRITDU
70) READD
71) RAOBIN
72) CSRAOB
73) CSRAOR
74) SGRAOB
75) CSRAOS
76) CSRAOM
77) CSRAOZ
78) CSRAOP
79) CSRAOI
80) MNRAOB
81) WMIX
82) INTPTW
83) EXTEMP
84) STDATM
85) CLMGES
86) VASGES
87) OUTINT
88) SURGES
89) GETSFV
90) IGNAME
91) PROFIX
92) SATEAR
93) SATPOS
94) ANGGET
95) SOLARP
96) ANGLES
97) HRTOPO
98) NVINIT
99) GETNAV
100) EPOCH
101) GETGAM
102) VASNAV
103) PLNKIV
104) OPNA
105) READOF
106) RBYTSX
107) ZEROS
108) RDTRK
109) MOVC
110) READDL
111) VASTAU
112) PRETAV
113) PREATV
114) ULMR
115) CO2TAV
116) H2OTAV
117) CONTAV
118) O3TAV
119) GAMTAV
120) PRECW
121) INITPL
122) PLTDIG
123) QGDASH
124) DSHOFF
125) PLOT
126) PAGE
127) WALK
128) PENMOV
129) PENBEG
130) PENADD
131) ENPT
132) SENOUT
133) ATOE
134) TEKPUT
135) BOX
136) DSHON
137) ENDPLT
138) WRTRK
139) CLOSA
140) EMES
141) PRIOUT
142) TQSET
143) PUC
144) WD
145) 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 Labroratory was indispensible in helping me understand the software itself, as were Dr. W. L. Smith, and Messrs. H. Woolf and F. Nagle, also of the NOAA/NESDIS Development Laboratory. Mr. G.S. Wade helped extensively with Chapter 1, in addition to answering questions from time to time. Mr. P. Rafael helped with the flowcharting, and Messrs. A. Schreiner and T. Achtor also provided timely assistance.

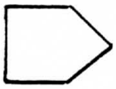
Many other people helped with various other aspects of the manual, including Dr. J.F. LeMarshall, Messrs. G. Callan, B. Howell, R. Dengel, M. Barrett, D. Santek and J. Benson, as well as Mrs. G. Dengel and Mses. D. Laitsch and A. Weickmann. Finally, I also thank Mrs. Gail Turluck , Mses. Jan Waite and A. North, Mr. Tony Wendricks, and Mr. Jeffer Lam for their help in preparing this manuscript.

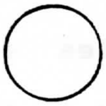
APPENDIX I

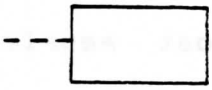
Level I Flowchart Constructs

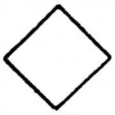
The following brief appendix contains examples and(or) descriptions of all the major FORTRAN constructs used in the VAS programs discussed in this manual. I think most people with FORTRAN experience will find this appendix self-explanatory.

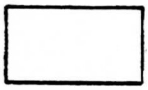
LEVEL 1 FLOWCHART CONSTRUCTS

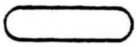
1.  : Offpage connector or beginning and end of DO-loops.

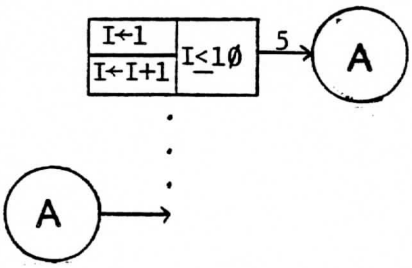
2.  : O-page connector or beginning and end of DO-loops.

3.  : Comment statement.

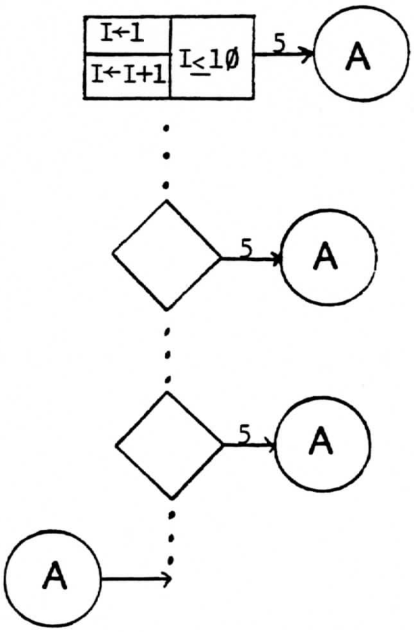
4.  : IF conditional.

5.  : CALL statement or arithmetic statement.

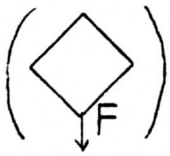
6.  : END statement.

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.

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 II

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 will be listed and described here.

Furthermore, the prefix or program/subroutine location within the McIDAS source library of each subroutine is listed. If the reader is referenced to a subroutine/program location, it means that subroutine is located in its entirety somewhere within that library member. For instance, 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 descripions of input and(or) output parameters and parameter lists whenever possible. Finally, the reader may occasionally be referred to the original source code 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	(MR)
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	(VR)
DDEST	(MR)
DECONV	(VR)
DIRADJ	(VR)
DOPEN	(VR)
DREAD	(VR)
DSHOFF	(in MRPLTPBK)
DSHON	(in MRPLTPBK)
DWRITE	(VR)
EDEST	(MR)
EMES	(MR)
ENAREA	(MR)
ENCCLR	(VR)
ENCODE	(VR)
ENCODX	(TR)
ENDPLT	(in MRPLTPBK)
ENKODE	(VR)
ENPT	(in MRPLTPBK)
EPOCH	(in MRGETNAV)
EXTEMP	(TR)
FBARN	(MR)
FECONV	(VR)
FILCLD	(in VLSRAD)
FILCLR	(in VLSRAD)
FILL	(TR)
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)
HTX	(in VLSRET)
IECONV	(VR)
IGNAME	(in MRIGMAKE)
INITPL	(in MRPLTPBK)
INTER	(in MRFBARN)
INTPTW	(TR)
IQ	(VR)
ISQX	(MR)
ITOC	(SR)
JMBWTF	(no source code found)
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)
PRIOUT	(VR)
PRLINX	(SR)
PROFIX	(TR)
PROPEN	(SR)
PRPRPR	(in SRPROPEN)
PRRD	(SR)
PRWR	(in SRPRRD)
PUC	(in MRLUC)
PUTCHR	(VR)
QGDASH	(in MRPLTPBK)
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	(no source code found)
SGRAOB	(MR)
SNDANL	(MR)
SOLARP	(MR)
SQSLED	(MR)
SRCH	(TR)
SRGSS	(in VLSRVA)
STC	(in SRIC)
STDATM	(TR)
SURGES	(TR)
TEKPUT	(in MRPLTPBK)
TOKANL	(MR)
TQ	(no source code found)
TQMES	(MR)
TQSET	(in MRLTQ)
TRMNL	(MR)
TSNIO	(TR)
TVSAT	(MR)
ULMR	(TR)
UNLOCK	(in SRLOCK)
VALUE	(in VLSRVA)
VASDAT	(in VLVDVA)
VASDIG	(VR)
VASGET	(VR)
VASNAV	(in MRGETGAM)
VASRTE	(VR)
VASTAU	(VR)
VASTPW	(VR)
VRTIO	(VR)
VTQ	(VR)
VTRET	(in VLSRET)
VWRET	(VR)
WALK	(in MRPLTPBK)
WD	(no source code found)
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 moved into it).

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 . . . 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 ≤ 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 Jan. and July data sets of T and w, 15°N is same data for both Jan. 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 . . . note ample description of input/output parameters in MRCSRAOB.

21) CSRAOI (NLEV, PRES, TEMP, TD, DIR, SPD, Z): routine to interpolate missing values of temperature, dewpoint, wind direction, wind speed . . . selected by option=4.

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', 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) DECONV (DGIVEN, NFRAC, JRMP, LBUF): used with subroutine ENCODE; this routine converts a real*4 or real*8 value to floating exponential decimal using the following algorithm:

Y*16**N=Y*10**(N LOG 16)=Y*10**X

=Y*10**(M+F)=(Y*10**F)*10**M=Z*10**M.

30) 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.

31) 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.

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

33) DSHOFF: dash off (--).

34) DSHON: dash on (--).

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

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

37) 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.

38) 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.

39) ENCCLR (JPUT, JWIDTH, BUFFER, SYMBOL): used with subroutine ENCODE . . . called by subroutine ENCODX, is part of encoding package.

40) 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.

41) 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.

42) 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.

43) 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 . . . 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.

44) 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 same as ENDPLT).

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

46) 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.

47) 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; final value is 0 (OK), or -1 (bad parameters) . . . for more information on parameters, see MRFBARN.

48) FECONV (GIVEN, NFRAC, JRMP, BUF): for use with ENCODE, called by ENCODX, is part of encoding package.

49) 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.

50) FILCLR (NSAM, NRAS, RADS, AVG, STD, BOAT, KM, NS): Also called by NSTAR, sets up file of clear radiances, is <u>not</u> registered; both FILCLD and FILCLR filter out noise as well.

51) FILL (IGRID, ITAG, IBUF, IFG, NROW, NCOL): fills missing elements of grid with appropriate values by linear extrapolation.

52) FLOC: gets latitude and longitude from line and element (for either TOVS or VAS).

53) 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, $\tau(I) = \tau(I)^G$).

54) 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).

55) GETDAY: get system date, in terms of integer values, from computer's own clock (format of YYDDD).

56) 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.

57) 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.

58) 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), is satellite dependent, not wavelength dependent.

59) GETSFV: gets surface data from gridfile for VAS or TOVS retrievals (1000 mb height, sea level temperature and dewpoint depression).

60) GETTIM: get system time, returns integer value in HHMMSS format (current time of day) from computers' clock.

61) GRADWI (JHITE, II, JJ, FCORIO, U, V, JHIT2, DELT): get gradient wind from analyzed field in height, wind direction returned is clockwise from grid north.

62) H2OTAV (PATH): calculates transmittances for H2O channels.

63) HEAPFY (XDA, XROW, XCOL, LL, K): sorting routine--sorts data according to latitude and longitude.

64) HRTOPO (MLAT, MLON, IELEV, ICHAR): takes input latitude and longitude and returns output as shown below . . . gives <u>H</u>igh <u>R</u>esolution <u>T</u>opography (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).

65) 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 surface).

66) IECONV (ITEM, JRMP, LBUF): for use with ENCODE, called by ENCODX, is part of encoding package.

67) IGNAME (GFNO, FILNAM): construct gridfile name from number; name is of form "GRIDNNNN".

68) 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.

69) INTER (XDA, XROW, XCOL, LL, K): is an entry point within subroutine HEAPFY.

70) INTPTW (PI, TI, WI, PF, TF, WF, NI, NF): interpolates temperature and mixing ratio in pressure.

71) 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).

72) 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).

73) 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.

74) JMBWTF: called within subroutine LWSO, does the actual writing of the block to disk.

75) 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; function value of 0 (JSQX was successful) or non-zero (JSQX was not successful); for further information, see VRJSQX.

76) LINFIL (LINE, N, IPASS): do linear interpolation on values along a single line.

77) 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.

78) 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.

79) 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.

80) 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). For more details on LW-routines, see DNLWDOC.

81) LWMOP (DFNAME): 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).

82) 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.

83) 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 is syscom so that the file may be closed automatically when the program terminates.

84) LWPO (ICB): write a file page out to disk; 'ICB' is the number of a buffer containing the file page.

85) LWSO (ISECTR, IBUF): writes block to disk (page), called by LWI and LWO, is part of McIDAS operating system.

86) MDCLOS (MDNO): close MD file(s); 'MDNO' is MD number of file to be closed, or 0 to close all.

87) MDNAME (MDNA, FILNAM): make name of MD file.

88) MNRAOB (MD, IDAY, ITIME, IDNO, ISTDAT, LEVDAT, ISTAT): gets mandatory level raob data; 'MD' is MD file#, '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). Further information on MNRAOB can be found in MRMNRAOB.

89) MOVB (NBYTES, FROM, TO, TOPOS): moves N bytes from word boundary to anywhere; 'TOPOS' is offset within 'TO' array (0-based), if 'NBYTES' is ≤ 0, nothing happens.

90) 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 ≤ 0, nothing happens. MOVC essentially moves N bytes from anywhere to anywhere.

91) 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.

92) MOVW (NWORDS, FROM, TO): moves N words from one array to another; if 'NWORDS' is ≤ 0, nothing happens.

93) MOVWC (IARR, CARR, N): move word data to character variable; 'IARR' is source array, 'CARR', 'N' represent character variable of length 'N'.

94) NSTAR (RADS, VCLR, IFAIL): infers clear column radiances from partially-clouded fields of view.

95) NVINIT (BETAIN, BETDOT, INAV, PTIME): refers to opening up of navigation (NAV) file.

96) O3TAV (TOTO, PATH): computes transmittances for channels affected by ozone absorption (especially channel 2 (≅690 cm^{-1})).

97) OPNA (IAREA): collect a list of the track groups assigned to the area and place them in the common block.

98) 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.

99) PACK (N, IS, ID): moves 'N' bytes from 'IS' at one per word to 'ID' at four per word.

100) 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.

101) PENADD (IVAL): interface subroutine to add 3-bit values.

102) PENBEG: new WRRRM interface subroutine to begin a new packet, deals with terminal protocol when new line is to be added.

103) PENMOV (ILINED, IPIXLD): interface subroutine to move the pen a "delta" (ERASEW, ENPT, PENMOV, PENPOS, PENBEG, PENADD subroutines are for new WRRRM interface).

104) PLNKIV (ISAT): brings in coefficients used for any Planck function calculations from disk.

105) PLOT (LIN, NEL, 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 'NEL' represent the logical end of line. See MRPLTPBK for more details.

106) PLTDIG (LINE, LEL, NVAL, LHGT, NFW, LEVEL): plots a single digit.

107) POST: part of PRINT package, "wakes up" printer handlers--is valid only in McIDAS.

108) PRCL: is an entry within PRRD on the line immediately below the PRWR entry; closes a transfer file.

109) PRCLOS (I): end a print spool string and connect it to que for printer 'I'.

110) PREATV (TEMP, WMIX): takes atmospheric information, gets quantities required for transmittance model (also computes derived functions of temperature and mixing ratio).

111) 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.

112) PRETAV (ISAT, KCHAN): more preparation for transmittance calculations, initializes pressure functions and reads in coefficients.

113) 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).

114) 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.

115) 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.).

116) PROPEN: start a print spool string.

117) PRPRPR (K): add a line of print to spool.

118) PRRD: printer spool read/write, reads record from transfer file, is not called by user.

119) PRWR: is also an entry within PRRD, writes record on transfer file, is not called by user.

120) PUC (FROM, INDEX): "put user common", pokes word into user common, subroutines LUC/PUC map user - common indices into indices in core-resident syscom; 'INDEX' is where (in user common) to do transfer, 'FROM' is fullword data value to transfer to user common, final value is word transferred.

121) PUTCHR (NBYTE, ARRAY, ITEM): stores simple byte in an array (the opposite of LCHAR).

122) QGDASH (JDASH): return current status of dash mode.

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 IO 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; <u>input parameters</u>: 'FRAM' = frame number, 'ILIN' - line satellite coordinate, 'IELE' = element satellite coordinate; <u>output parameters</u>: '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 -, 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) 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.

140) SRCH (LATS, LONGS, IM, ILIN, IELE, NPTS, NROWS): a subroutine which uses trigonometry to locate lat/long 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.

141) 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.

142) 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'.

143) 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.

144) 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).

145) TEKPUT (L,P): in graphics package, outputs to techtronics (4010) compatible displays.

146) TOKANL (CPARR, NTOK, NKW, NARR, IDEVAL): analyze token sequence produced by MCTOKN; 'CPARR' contains the token sequence, 'NTOK' is total number of tokens. <u>Outputs</u>: '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.)

147) TQ (BUFFER): involved with writing message to terminal, is part of operating system, sends a text message to CRT.

148) 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.

149) 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.

150) TRMNL (IS): returns the terminal number.

151) TSNIO (IOP, IFL, IR, IC, NR, NC, IA): TIROS-N data input/output (operates on orbit/image file).

152) 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.

153) 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).

154) UNLOCK: releases file for subsequent use by other programs.

155) VALUE (LAN, LAS, LOW, LOE, INC, IDL, FSCL, VAL): given latitude and longitude, returns a value interpolated from a grid.

156) 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 . . . NOTE: IL and IE are changed in this routine.

157) 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.

158) 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/.

159) VASNAV (FNUM, SYDREC, ISYD, IHMS): partially sets up navigation common block, is VAS navigation package (sounding mode).

160) 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:

$$W(p) = \frac{\partial \tau}{\partial \ln p}(\gamma,p,\theta) \cdot \frac{\partial B}{\partial T}(\gamma,T^n)\Big|_p \Big/ \frac{\partial B}{\partial T_B}(\gamma,T_B{}^n)\Big|_T$$

161) 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.

162) VASTPW (TBB, EBB, TOTO, TSFS, URET, ITMAX, ISAT, KCT, NP): retrieves total precipitable water from VAS radiances.

163) 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.

164) 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.

165) VTRET (KC1, KC2, DELB, KUSE, WG, IS, M, ER, IOK): VAS temperature retrieval enhancement routine.

166) VWRET (TS, TAUW, U, NL, NLS, TSP, USP): VAS water vapor retrieval enhancement routine.

167) WALK (DROW, DCOL): is internal subroutine related to wide line graphics, does actual plotting.

168) WD: retrieves words from user common (UC), is part of the scanner (operating system), alters syscom (systems communications region).

169) 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.

170) WRBOX (IR, IP, IZR, IZP, IC): writes (plots) a box.

171) WRITA (IAREA, NREC, IADDR): writes into an area.

172) WRITDU (ANUM, ENTRY): writes and unlocks area/sounding directory.
Input parameters: 'ANUM' = area number, 'ENTRY' = 64-word directory entry to be written (integer array).

173) WRMAR (IR, IP, IZ, IC, IN): program to operate "worm" (WRRRM-graphics package) and indicate 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.

174) WRTRK: writes onto a track, is an entry within SRRDTRK.

175) ZECONV (ITEM, NBYTES, JRMP, OUTBUF): used by subroutine ENCODE, is for conversion from integer to sexadecimal.

176) ZEROS: sends a bunch of zeroes to FORTRAN, makes all elements of an array equal to zero.

177) ZWIND (JHITE, I, J, JLAT, SPEED, DIREC, JHIT2, DELT): calculates a given type of wind (geostrophic, isallobaric, ageostrophic) according to the input arguments, also indirectly calculates gradient wind by calling subroutine GRADWI.

APPENDIX III

Function List and Descriptions

Appendix III 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 III cannot be considered as comprehensive as Appendix II.

As in the second appendix, the McIDAS source library location for each function is listed, as well as the parameter list and descriptions of the parameters for each function whenever possible. Finally, the purpose of each function is also detailed.

VAS Retrieval Software Functions

1)	AZMUTH	(VR)
2)	CKWP	(MR)
3)	CLIT	(SR)
4)	CPP	(MR)
5)	DEWPT	(TR)
6)	DLIT	(ML)
7)	DKWP	(MR)
8)	DPP	(MR)
9)	ICURG	(MR)
10)	IGGET	(in MRIGMAKE)
11)	IGPUT	(in MRIGMAKE)
12)	IGOPEN	(in MRIGMAKE)
13)	IKWP	(MR)
14)	ILALO	(MR)
15)	IPP	(MR)
16)	IROUND	(MR)
17)	ISATNV	(VR)
18)	IVASCL	(VR)
19)	LANSEA	(TR)
20)	LIT	(SR)
21)	LOGAND	(an ENTRY within VRLOGOR)
22)	LWI	(in MRLWSUBS)
23)	MDGET	(in MRMDMAKE)
24)	MDINFO	(MR)
25)	MDKEYS	(MR)
26)	MDOPEN	(in MRMDMAKE)
27)	MDPUT	(in MRMDMAKE)
28)	RADENC	(VR)
29)	VBRITE	(in subroutine PLNKIV)
30)	VBDTAU	(VR)
31)	VDBDTB	(in subroutine PLNKIV)
32)	VPLANC	(in subroutine PLNKIV)
33)	VSKINT	(VR)
34)	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) 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.

3) CLIT(I): type transformer: integer/real to character * 4.

4) CPP(I,CDEFLT): get positional parameter (character * 12); function value is 'I'th positional parameter, or 'CDEFLT' if parameter is missing.

5) DEWPT (P,T,W): calculates dewpoint using given P,T and mixing ratio.

6) DLIT(C): returns own argument (literal); result is real * 8 bitwise identical to 'C,' "c" is character * 8.

7) DKWP(CKW,I,DDEFLT): get keyword parameter (real*8), returns real*8 value.

8) DPP(I,DDEFLT): get a positional parameter (real * 8).

9) ICURG(IL,IE): 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.

10) 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).

11) 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 -, 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' ≤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).

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) 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.

14) ILALO(X): changes a floating point latitude-longitude to a packed integer (sign DDD‸MM‸SS); input parameter: 'X'=floating point latitude or longitude.

15) 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).

16) IROUND(X): rounds a floating point value; input parameters: 'X'-floating point value.

17) ISATNV(JSAT): obtain VAS satellite number from 'JSAT' (obtained via TVSAT).

18) 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.

19) 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 lat lon (positive E, negative W) in degrees * 1ØØ; final value=2 (ocean) or 1 (land).

20) 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.'

21) LOGAND (A,B): logical 'and' of two 4-byte arguments.

22) 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 filename (12 bytes), 'IBEGWD' is first virtual word in transfer (Ø based), 'NWDS' is number of words to transfer (if 'NWDS'≤0, nothing happens), 'IARR' is an array containing record to read/write; final value for LWI:Ø (last page 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)).

23) 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).

24) 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).

25) 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 ≤'NLIST;' if not =, it indicates that one or more keys were not found in schema. If 'NLIST'=-1, final value is total number of keys in schema (≤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.

26) 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).

27) 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).

28) RADENC(VALUE,CHANL,DELTAF,YSUBZ): converts radiance in given channel from counts to radiance in units of $mWm^{-2}\ st^{-1}\ cm^{-1}$.

29) VBRITE (R,K): calculates brightness temperature based on given radiance and wavenumber.

30) 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 $mWm^{-2}\ st^{-1}\ cm^{-1}$.

31) VDBDTB(TBB,K): computes derivative of Planck Function (radiance) with respect to brightness temperature.

32) VPLANC (T,K): calculates Planck radiance from given temperature and channel.

33) 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.

34) WSAT (P,T): calculates saturation mixing ratio for a given P and T.

APPENDIX IV

VAS Retrieval Software

The final appendix consists of copies of all the programs discussed in this manual. 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    SBG  12/23/83; ENTERED USER MANUAL CARD
//  EXEC  MCPRG,MOD=VPVA,LANGLVL=66
//FORT.SYSIN  DD *
@PROCESS SC(TQMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTOP,OPCOM,SQX,SQW)
@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
/*
```

```
//IDVA5910 JOB CLASS=B,MSGLEVEL=(0,0)
//* VLIDVA   CMH  01/24/84: MEMBER UPDATED
// EXEC MCPRG,MOD=IDVA
//FORT.SYSIN DD *
@PROCESS SC(TQ,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTOR,OPCOM,SQX,SQW,DOPEN,DREAD,DWRITE)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAINO
C ?   TO INITIALIZE VASTEXT DOCUMENTATION SECTOR (IF VAS PNTR SET)
C ?   OR TOVORB DOCUMENTATION
C ?     (MDNR) (MDRR) (LALONW) (LALOSE)
C ?     MDNR IS MD FILE FOR RETRIEVALS
C ?     MDRR IS ROW NO. (1-16)  MD ROW HEADER IS INITIALIZED
C ?        IF MDRR IS SPECIFIED,ROW AND POINTERS ARE 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 ?               INIT   NON ZERO INITIALIZES GUESS MD INFO
      DIMENSION VAS(13),MFR(64),KOUT(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/ Ø./,TOPE/360./,BOTW/Ø./
      COMMON /NAV/FLAT,FLON,ZENLOC,SZEN,IL,IE,IRAS,IPIC,IHMS,JT,JD
      COMMON /DOC/IDOC(112)
      COMMON /ARENT/IDIR(64)          DATA LETOP/Ø/,IEBOT/999999/
C     ADD COMMON BLOCKS FOR VASDAT    DATA VMISG/999999./
      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 TSNIO(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)
          IL=IL+ILR
          IF(LINNW.NE.0) IL=LINNW
          IDOC(3)=IL
          IE=IE+IER
          IF(LELENW.NE.0) IE=LELENW
          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
          LL=LL+ILR
          IF(LINSE.NE.0) LL=LINSE
          LE=IX+((LP-MFR(9))*IER)/MAG
          LE=LE+IER
          IF(LELESE.NE.0) LE=LELESE
      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 2 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 UPPER LEFT
          CALL VASDAT(IL,IEB,VAS)
          LAN=FLAT+0.5
          LOW=FLON+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 IF(ISEC.NE.0)GO TO 130
          ISEC=1
          GO TO 122
      130 VAS(1)=0.
          IL=IL-ILR
```

```
C       CHECK LAT/LON FORCE
        IF (LALONW.EQ.0)GO TO 138
        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        C   CHECK LAT/LON FORCE
                                             IF (LALONW.EQ.Ø) GO TO 138
        GO TO 133
    135 TOPW=FLON
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
        LEBOT=IEBOT
    156 LEBOT=LEBOT+IER
    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 159
  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.999999.)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    LAL
     *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 250
C     INITIALIZE RETRIEVAL MD FILE ROW
  210 CONTINUE
      IRET(1)=MOD(IDOC(1),100000)
      IHR=IDOC(2)/10000
      IRET(2)=IDOC(2)
  220 IF (IKWP('INIT',1,0).EQ.0)GO TO 250
C     INITIALIZE REMAINDER OF DOCUMENTATION
      DO 245 N=27,112
  245 IRET(N)=0
      IRET(3)=0
      IF (MDOPEN(MDNR,2).LT.0)GO TO 900
      IOK=MDPUT(MDNR,MDRR,0,IRET)
```

LEBOT = LEBOT - IER
IF (LEBOT .LT. IE) GO TO 400
GO TO 156

IF (MDOPEN(MDNR,2).LT.0) GO TO 900
IF (MDNR.EQ.0) GO TO 250

```
      IF (IOK.LT.0)GO TO 510
  250 CONTINUE
      IDOC(40)=MDNR
      IDOC(41)=MDRR          264 CONTINUE
C     SET STATUS FLAG TO READY
      IDOC(28)=1
C     PUT IN SNDR FILE NO
      NSND=LUC(81)
      IDOC(35)=NSND
C     SET LINE AND ELEMENTS FOR VRAD (actually SRAD)
      IDOC(50)=0
      IDOC(51)=0
C     SET SPACING DEFAULTS FOR VRAD (actually SRAD)
      IDOC(52)=10*16/ILR+0.5
C     FOLLOWING IS FOR BOX SIZE (PER RETRIEVAL..5*5 FOR RES 16)
      IDOC(53)=5*16/ILR+0.5
C     INDENT BY HALF THE BOX SIZE
      IOFF=IDOC(53)/2
      IF (MOD(IOFF,2).NE.1)IOFF=IOFF+1
      IOFF=IOFF*ILR
      IDOC(55)=LL-IOFF
      IDOC(56)=LF-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 TSNIO(0,1,K,1,1,NCOLS,LAT)
      IF (LAT(1).LE.9000.AND.LAT(NCOLS).LE.9000)GO TO 3
    2 CONTINUE
    3 CALL TSNIO(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 (MODE.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 VIA
     * KEYIN',0)
      CALL ABORT
  400 CALL SDEST('...PROBLEM LOCATING SE CORNER ... ENTER LINE + ELEM VIA
     * KEYIN',0)
      CALL ABORT
  900 CALL SDEST(' CANNOT OPEN RETRIEVAL MD FILE NO ',MDNR)
      ~~RETURN~~ GO TO 260
  910 CALL SDEST(' CANNOT WRITE MD ROW HEADER FOR ~~HOUR~~ ROW ',~~IHR~~ MDRR)
      RETURN
  920 CALL SDEST(' LALOSE HAS NOT BEEN SPECIFIED',0)
      RETURN
      END
```

```
//LOVA7000 JOB CLASS=A,MSGLEVEL=(0,0)
//* VLLOVA   SBG  12/23/83; ENTERED USER MANUAL CARD
// EXEC MCPRG,MOD=LOVA
//FORT.SYSIN DD *
@PROCESS SC(TQMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTOR,OPCOM,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 TOVORB 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 TQBUF(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  BEGE
     *  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/)',TQBUF,IYD,ITIM,
     1 IDOC(9),IDOC(10),IDOC(25),IDOC(26),IDOC(28),NSAT,NSND)
      CALL SDEST(TIT2,0)
      CALL ENKODE(' (6I8/)',TQBUF,IDOC(36))
      CALL SDEST(TIT3,0)
      CALL ENKODE(' (I8,-4I8/)',TQBUF,IDOC(42),IDOC(27))
```

```
      CALL SDEST(' NO. RETRIEVALS= ',NRET)
      IF (LUC(81).EQ.0)RETURN
      CALL SDEST(' CURRENT RETRIEVAL OPTIONS..',0)
      CALL SDEST(TIT4,0)
      CALL ENKODE(' ('11I6/)',TOBUF,IDOC(50))
      RETURN
      END
/*
```

```
//SPVA5680 JOB CLASS=A,MSGLEVEL=(0,0)
//* VLSPVA   AJS  01/06/84: MEMBER UPDATED
//* VLSPVA   SBG  12/23/83; ENTERED USER MANUAL CARD
// EXEC MCPRG,MOD=SPVA
//FORT.SYSIN DD *
@PROCESS SC(TQMES,EMES,DMES,MCOD,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/,LFN/100/
      DATA CPAR/'MDNS','MDRS','MDNG','MDRG','MDNR','MDRR',
     * 'NGFS','ZGRID','TGRID','DGRID'/
      KBUG=IKWP('BUG',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 TSNIO(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 TSNIO(3,1,1,1,1,1,IDOC)
C     CHECK ON RETRIEVAL NO. EDIT
   28 NRET=IKWP('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 VRTIO(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
```

```
//GSVA5910 JOB  CLASS=A,MSGLEVEL=(0,0)
//  EXEC  MCPRG,MOD=GSVA
//* VLGSVA    N    01/23/84: MEMBER UPDATED
//* VLGSVA   SBG  12/23/83; ENTERED USER MANUAL CARD
//FORT.SYSIN  DD  *
@PROCESS SC(TQMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTOR,OPCOM,SQX,SQW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW,DOPEN,DREAD,DWRITE,DCLOSE)
      SUBROUTINE MAINO
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 MDKEY 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 (IOK.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
      IOK=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 950
   12 CONTINUE
      CALL SDEST(' PREPARING GUESS FOR DAY ',IGDAY)
      CALL SDEST(' HOUR =  ',IGTIME)
      CALL SDEST(' VALID TIME = ',IGTINV)
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
      IOK=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
      IOK=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 20MB 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
      IOK=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)
      IOK=MDPUT(MDNG,NN,MM,IREC)
      IF (IOK.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 TSNIO(3,1,1,1,1,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',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
/*
//
```

```
//CSVA7000 JOB  CLASS=A,MSGLEVEL=(0,0)
//  EXEC  MCPRG,MOD=CSVA
//* VLCSVA   SBG  12/23/83; ENTERED USER MANUAL CARD
//FORT.SYSIN  DD  *
@PROCESS SC(TONES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,MDKEYS,MDGET,MDPUT,WTOR,OPCOM,SQX,SQW)
@PROCESS SC(MOVB,DOPEN,DREAD,DWRITE,MOVC,MOVW,CLEANW)
      SUBROUTINE MAIN0
C ?   LOAD SURFACE MD INTO SPECIAL EDIT MD FILE (MDOUT)
C ?   KEYIN: CSVA MDOUT
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/
C      CHECK ON AVAILABILITY OF OUTPUT FILE
      IPRNT=IKWP('BUG',1,0)
      KBUG=IPRNT
C     IF (IPRNT.EQ.0)IPRNT=1
C     CALL TOSET(IPRNT)
      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('SHIP',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('AUTO',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     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
      IOK= 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(MDO,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 MDNO  TO VASTEXT
      IDOC(27)=(IDAY+ITIME)*100
      IDOC(36)=MDO
```

```
      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,IDOC)
   60 CONTINUE
C     REWRITE ROW HEADER WITH NO OF REPORTS ADDED
      KBUF(3)=NOUT
      IOK=MDPUT(MDO,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
/
//
```

```
//SRVA5910 JOB CLASS=B,MSGLEVEL=(0,0)
//  EXEC  MCPRG
//* VLSRVA   CMH  01/23/84: MEMBER UPDATED
//* VLSRVA   SBG  12/23/83; ENTERED USER MANUAL CARD
//FORT.SYSIN  DD  *
@PROCESS SC(TQMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(LWOPEN,LWMAKF,LWQUIT,ISFILE,WTOR,GPCOM,SOX,SQU)
@PROCESS SC(ISCHAR,DOPEN,DREAD,DWRITE,ISAN,MCVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAINO
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 /DOC/IDOC(112)
      COMMON /DIMEN/NROWS,NCOLS
      COMMON/SRNAV/BETAIN,BETDOT,PTIME,INAV
      CHARACTER*8 MFILE
      CHARACTER*12 CGS,CKWP,CPP,CHAR
      DIMENSION KOUT(10),SDL(8),KBUF(200)
      DIMENSION LBUF(33)
      DIMENSION ITG(2400),IDL(2400)
      DIMENSION IUNI(4),DA(999),RW(999),CL(999)
      DIMENSION IREP(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/,NRMAX/999/
      DATA LBUF/33*Z40404040/,IBLNK/Z40404040/
      DATA SDL/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 TSNIO(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
      IOK=MDGET(MDNS,MDRS,0,KBUF)
      ISYD=KBUF(1)
      IHMS=KBUF(2)
      IF (IOK.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,MONG)
      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('NODAT',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.999999.)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
      IOK=MDGET(MDNS,MDRS,IREP(NN),KBUF)
```

```
      NOUT=NOUT+1
      KBUF(IP+6)=MISS
      IOK=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
      IOK=IGPUT(NGRFS,NGRNO,IDL,NROWS,NCOLS,IGHD,ISTAT)
      CALL TQMES(' 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 TSNIO(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                                       MDRS
  930 CALL SDEST(' NO DATA AVAILABLE FOR HOUR',NROW)
      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  12/23/83; 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 MDSFT 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 IARD(20),MDHD(64),FLA(4),FLO(4),KOUT(10)
      INTEGER*4 OBUF(20)
      REAL*8 DLIT
      CHARACTER*8 MFILE
      CHARACTER*12 CLIT,ICHR,CPP
      COMMON /DOC/IDOC(100)
      COMMON /NAV/FLAT,FLON,ZFNLOC,SZEN,IL,IE,IPAS,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 /FLOCNV/BETAIN,BETDOT,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(DLIT(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,BETDOT,INAV,PTIME)
      GO TO 130
  120 CALL TSNIO(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 GETFRM(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.IOUT(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=IPC+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 OUTINT(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)IOK=MDGET(MDNO,0,M,IOUT)
      IOK=MDGET(MDNO,NDR,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 40
      IF (IOUT(NLON).LT.ILOMIN(K).OR.IOUT(NLON).GT.ILOMAX(K))GO TO 40
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(PTIME,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 WPMAR(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,KTOT
      I=IADD(N)
      IOUT(I)=MISS
   58 CONTINUE
   60 CONTINUE
      IOK=MDPUT(MDNO,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
```

```
//SRAD5910 JOB CLASS=B,MSGLEVEL=(0,0)
//* VLSRAD   HMW  01/31/84: MEMBER UPDATED
// EXEC MCPRG
//FORT.SYSIN DD *
      SUBROUTINE MAIN0
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 (CURSOR))
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 ?            AUTO=1 (TAKE IL,IE,LL,LE FROM VASTEXT)
C ?            OLD=1  DATA IS BEFORE JAN 15 1982
      DIMENSION IFILNM(2),KOUT(10),ICU(3),ICT(6),IHUM(3)
     *,VDAT(13),BDAT(13)
     *,NSAM(13),AVG(13),RADS(121,13),NRAS(121)
     *,MF(64),LBUF(33)
      DIMENSION LC(13),NSFIN(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 CCES,CPP,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,IFLEV,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/IUCH(12,2)
      COMMON/DBUG/KBUG
      COMMON /JAN/IOLD
C     IF IOLD NE 0, WE USE THE FIXED DELTA F RATHER THAN LINE
```

```
C       DOCUMENTATION (DARA 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 ICU/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 ROG/29.2898/,GRAV/980.665/,TOTO/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.'O')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 VASDAT PROCESSING)
      IBOXS=1
C     SET UP PROCESSING DAY FOR DOCUMENTATION IN ROW 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('AUTO',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
      BDAT(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)CGOCTOC420CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCDOC340CI=1,3CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCICV=ICU(I)CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCIF(VDAT(ICV).NE.VMISG)CGOCTOC360CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CC340CCONTINUECCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCGOCTOC420CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC360CCONTINUECCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      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 CHNG
      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(NOSFC)
      TSTA=0.01*FLOAT(ITSFC)
      TMN=TSTA-10.
      IF (NOSFC.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.'M')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.RLO9) GO TO 670
```

```
      LC(9)=0
      LC(10)=0
  670 CONTINUE
      IF(RADS(I,8).NE.VMISG.AND.RADS(I,8).GE.RL08) GO TO 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.RL05) GO TO 680
      LC(4)=0
      LC(9)=0
      IF(RADS(I,4).NE.VMISG.AND.RADS(I,4).GE.RL04) GO TO 680
      LC(3)=0
      LC(10)=0
  680 DO 700 J=3,12
      IF(LC(J).EQ.0.OR.RADS(I,J).EQ.VMISG) GO TO 700
      NSAM(J)=NSAM(J)+1
      AVG(J)=AVG(J)+RADS(I,J)
  700 CONTINUE
  720 CONTINUE
  740 CONTINUE
      DO 760 I=1,12
      VDAT(I)=VMISG
      SRAD(I)=VMISG
      IF(NSAM(I).EQ.0.OR.IUSE(I).EQ.0) GO TO 760
      VDAT(I)=AVG(I)/NSAM(I)
  760 CONTINUE
      MSAM=NSAM(8)
      IF (MSAM.GE.5)GO TO 790
      IF(FORCE.NE.0..AND.CTYP.NE.'N')GO TO 790
      CTYP='N'
C * EXERCISE 'N*' OPTION
  770 DO 780 I=1,12
  780 IUCH(I,1)=IUSE(I)
      CALL NSTAR(RADS,NRAS,VDAT,IFAIL,NSMAX)
      IF(IFAIL.EQ.0) GO TO 790
      IF(KBUG.NE.0)CALL SDEST(' NSTAR FAILURE NO. ',IFAIL)
      GO TO 2190
  790 CONTINUE
      IF(KBUG.EQ.0) GO TO 800
      CALL ENKODE('(T2,"BRIGHT-TEMP",^12F8.2/)',LBUF,VDAT)
  800 CONTINUE
      DO 830 I=1,12
      EX(I)=100.
      IF(VDAT(I).EQ.VMISG) GO TO 830
      SRAD(I)=VPLANC(VDAT(I),I)
      DBDB=VDBDTB(VDAT(I),I)
      SPIN=NSPIN(I)-NSAM(I)
      IF(SPIN.EQ.0..OR.DBDB.EQ.0.) GO TO 830
      EX(I)=(EV(I)/DBDB)/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(IDOC(33),1000)
      IRET(8)=IDOC(34)+IGD*10000
      IRET(9)=ISDOC(6)
      DO 2202 K=1,5
 2202 IRET(9+K)=ISDOC(K)
      IRET(15)=0
      IRET(16)=IOPROS
C     FILL IN USER MOD FLAG
      IRETD(3)=-999
      IRETD(4)=LASRET+1
      IRETD(5)=LAT*100
      IRETD(6)=LON*100
      IRETD(7)=IDOC(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
      IDOC(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)=IDOC(2)
      IRET(3)=LASRET
      CALL VRTIO(IRET,0,1)
      IF(LASRET.EQ.MAXRET) GO TO 2500
 2400 CONTINUE
 2500 CALL SDEST(' NO. OF SNDCS. IN RTVL. FILE = ',LASRET)
      RETURN
      END
```

```
//SRET6910 JOB CLASS=B,MSGLEVEL=(0,0)
//* VLSRET   CMH  02/06/84: MEMBER UPDATED
// EXEC MCPRG
//FORT.SYSIN DD *
@PROCESS SC(TOMES,EMES,DMES,NCCD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTOR,OPCOM,SQX,SQW)
@PROCESS SC(DOPEN,DREAD,DWRITE,MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAIN0
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),IHUM(3)
     *,B(40),S(40),U(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),RPHO(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 /GESTBB/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),DGES(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/IUCH(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 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 ICU/7,9,10/
      DATA ITCH/6/
      DATA ROG/29.2898/,GPAV/980.665/,TOTO/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/)',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
      ISATEL=IDOC(1)/100000
      ISAT=ISATNV(ISATEL)
      CALL PLNKIV(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 PLNKIV 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.'O')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 GUESS PROFILE WITH RETRIEVAL AT
C     EACH ITERATION..BLENDING IS A LEVELS BELOW PREF
      LCRT=IKWP('BUG',1,0)
      IF (LCRT.NE.0)CALL TQSET(LCRT)
C GET TERMINAL NO..THIS HAS ALREADY BEEN LOADED IN VRTIO,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,BETDOT,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 VRTIO(IRET,NN,0)
C     LOOK AT USER MOD FLAG; LT 0 MEANS IT HASN'T BEEN RETRIEVED
C     GT 0 MEANS ITS EITHER FAILED OR BEEN EDITED, =Ø MEANS RETRIEVED
      IF (IRETD(3).GE.0)GO TO 2400                 ALREADY BUT OK
      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)
      NOGO=1
      CALL GESPRO(IGES,NOSFC,MDNG)
      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)
      DTDF=DTS/(P(IS)-P(ILO))
      DO 440 I=ILO,IS
  440 T(I)=T(I)+DTDF*(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.'O')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
        IF(RWV.LT.RRHO(2))GO TO 885
        TGR=(RWV-RRHO(1))/(RRHO(2)-RRHO(1))
        TGR=AMAX1(TGR,-10.)
        RH=RHO(1)*((RHO(2)/RHO(1))**TGR)
        GO TO 1090
  885   CONTINUE
        DO 890 I=26,NL
  890   S(I)=RHO(ISTEP)*WSAT(P(I),T(I))
        CALL PRECW(P,S,U,NL)
        CALL VASTAU(T,S,TOTO,VZEN,TAU,ISAT,ICV)
 1080   RRHO(ISTEP)=VBDTAU(TAU,B,BS,ICV,NLS)
        TGR=(RWV-RRHO(2))/(RRHO(3)-RRHO(2))
        TGR=AMAX1(TGR,-10.)
        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 PWV*100 R50*100 TGR*100  RH 100")',
     * LBUF)
      CALL VTQ(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 VTQ(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 VASTAU(T,W,TOTO,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=IKWP('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 H20..CONVERGENCE
C          TEST IS @ 1920
C           =1   ENHANCE TEMP, LAYER 1
C           =2                 LAYER 2
C           =3                 LAYER 3
C           =4   REPEAT T ITERATION IF ENHANCE FAILED,REPEAT H20,EXIT
C     HEAVE ANY CLOUDY/CLIMATOLOGY SOUNDING
       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,TOTO,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.NOSFC.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 (IOK.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
      IOK=-1
      IF (IENH.NE.1)
     *CALL VTRET(KC1,KC2,DELT,KUSE,WG,IS,IPASS,EX,IOK)
      IF (IOK.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
      DTDF=DTS/(P(IS)-P(37))
      DO 1780 I=37,IS
 1780 T(I)=T(I)+DTDF*(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
      IOK=0
      ITLIM=ITLIM+3
C     RAISE ITERATION LIMIT WHEN WE GO TO SECOND PASS OF ITERATIVE
      GO TO 1670
 1885 CONTINUE
C **  CHECK CONVERGENCE
      IOK=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)
      IF(SUM.LT.0.025) GO TO 1940
      ITER=ITER+1
      IF (KBUG.NE.0)CALL SDEST(' *** 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=MIN0(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 SNDANL(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,"TOT =",F7.2/)',LBUF,TOT)
 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 850-500
     * 850-200")',LBUF)
      CALL VTQ(LBUF)
      CALL ENKODE('(6I8)',LBUF,IT850,ID850,IT500,ITOTLS,KDZ1,
     * IDZ2)
      CALL VTQ(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(PTIME,FLIN,FELE,VLAT,SLON,2,INAV,BETAIN,BETDOT,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 VRTIO(IRET,NN,1)
 2380 CONTINUE
 2400 CONTINUE
 2500 CALL SDEST(' LAST RETRIEVAL NO. PROCESSED WAS ',NN)
      RETURN
      END
```

↑ Same as 2/8/84

```
//PLVA5910 JOB CLASS=B,MSGLEVEL=(0,0)
//* VLPLVA   AJ   01/23/84: MEMBER UPDATED
// EXEC MCPRG,MOD=PLVA
//FORT.SYSIN DD *
@PROCESS SC(TQMES,LIT,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,DOPEN,DCLOSE,DREAD,DWRITE,WTOR,OPCOM,SQX,SQW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW)
      SUBROUTINE MAINO
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
      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 KOUT(10),MDHD(64)
      CHARACTER*12 CCHR,LCHR,CPP,CLEV
      CHARACTER*8 MFILE
      CHARACTER*4 CLIT
      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/NODE
      COMMON /TIGHT/ITOL
      COMMON /ENTRY/INIT
      COMMON/ORIENT/YCOORD,XCOORD
      DATA SATPOS/135.,75./
      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)
      IWRM=LUC(-5)
      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 TSNIO(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 (MCHD(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
        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 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.IOUT(I))GO TO 31
        IF (CLEV.EQ.CLIT(IOUT(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.IOUT(I))GO TO 51
      IF (CLEV.EQ.CLIT(IOUT(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 (IOUT(IADD).EQ.MISS)GO TO 70
C     LOCATE RASTER AND PIXEL OF SOUNDING
      FLAT=IOUT(NLAT)*.0001
      IF (FLAT.GT.LAN.OR.FLAT.LT.LAS)GO TO 70
      FLON=IOUT(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(PTIME,FLIN,FELE,FLAT,SLON.2,INAV,BETAIN.BETDOT,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+LTOP-1
      IF (IL.LT.LTOP.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 58
      FLON=-FLON
      ADJ=DIR
      CALL DIRADJ(FLAT,FLON,SATLON,DIR,ADJ)
   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
      CALL VASDIG(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
```

```
//BNVA5910 JOB  CLASS=E,MSGLEVEL=(0,0)
//  EXEC  MCPRG
//> VLENVA   GMC  01/30/84: 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
C ?           <NGG>   GRID NO OF GUESS GRID
C ?           LEV MAY BE 'SFC', MB LEVEL, OR 'DIFF' FOR TCP DIFFERENCES
C ?     KEYWORDS
C ?           LAT   MIN AND MAX LATS
C ?           LON   MIN AND MAX LONGITUDES
C ?           INC   INCREMENT 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 ?                 PASS 1 IS 1ST ARGUMENT, 2 IS SECOND
C ?           BIAS   VALUE TO BE SUBTRACTED FROM OBS
C ?           EDIT  POSITIVE EDITS MDNR BY DATA/ANAL COMPARISON
C ?                 NEGATIVE 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,ICHP,LCHP,CGS
      REAL*8 DLIT
      REAL*8 FN,SUM,SUMS
      DIMENSION KOUT(8),MDHD(64),KBUF(400)
      DIMENSION LBUF(33)
      DIMENSION ITG(2400),IDL(2400)
      DIMENSION IUNI(4),IREP(3000),DA(3000),PW(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)
      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 TSNIO(1,1,1,1,1,1,IDOC)
      GO TO 112
```

```
105      111 CALL DOPEN(DLIT(MFILE),LUN,LEN)
106          CALL DREAD(LUN,ITERM,IDOC)
107          CALL DCLOSE(LUN)
108      112 MDNO=IKWP('MDNR',1,0)
109          IF(MDNO.EQ.0)MDNO=IDOC(40)
110          NGFG=IKWP('NGFG',1,0)
111          IF (NGFG.EQ.0)NGFG=IDOC(42)
112          MDR=IKWP('MDRR',1, -1)
113          IF(MDR.LT.0)MDR=IDOC(41)
114          MDNG=IDOC(38)
115          MDNG=IKWP('MDNG',1,MDNG)
116          ICHR=CPP(1,'Z')
117          LCHR=ICHR
118          ILEV=IPP(2,0)*10
119          LPR=ILEV/10
120          IF (MDINFO(MDNO,MDHD).NE.0)GO TO 900
121          IF (MDOPEN(MDNO,2).NE.0)GO TO 900
122    C     READ ROW HEADER RECORD
123          IOK=MDGET(MDNO,MDR,0,KBUF)
124          IF (IOK.LT.0)GO TO 902
125          ISYD=KBUF(1)
126          IHMS=KBUF(2)
127    C     READ TEST RECORD
128          M=0
129        1 M=M+1
130          IF (M.GT.100)GO TO 904
131    C     READ COLUMN HEADER FOR RAOB
132          IF (MDHD(8).NE.0)IOK=MDGET(MDNO,0,M,KBUF)
133          IOK=MDGET(MDNO,MDR,M,KBUF)
134          IF (IOK.LT.0)GO TO 1
135          MMAX=MDHD(5)
136          IF (KBUG.GT.0)CALL SDEST(' MMAX IS ',MMAX)
137    C     MMAX IS MAX POSSIBLE DATA ENTRIES.USED TO AVOID
138    C     INFINITE LOOPING AT SN 148
139    C     FIND ADRESS FOR LAT AND LON
140          NKEYS=MDKEYS(MDNO,-1,LIST,ISCL,IUN,LOCS)
141          NCA=0
142          DO 2  N=1,NKEYS
143          IF (LIST(N).EQ.LIT('MOD '))NMOD=N
144          IF (LIST(N).EQ.LIT('CMAX'))NCA=N
145          IF (LIST(N).EQ.LIT('NREC'))NCA=N
146          IF (LIST(N).EQ.LIT('LAT '))NLAT=N
147          IF (LIST(N).EQ.LIT('LON '))NLON=N
148    C     SAVE REPRESENTATIVE ENTRIES FOR NODATA CASE
149          IF (LIST(N).EQ.LIT(ICHR))NADD=N
150        2 CONTINUE
151          NRPT=500
152          IF (NCA.NE.0)NRPT=KBUF(NCA)
153          NRPT=IKWP('LAST',1,NRPT)
154          IF (LCHR.EQ.'U'.OR.LCHR.EQ.'V')ICHR='SPD'
155          IBIAS=0
156          IF (ICHR.EQ.('T   ').OR.ICHR.EQ.('TD  '))
```

```
      * IBIAS=27312
C       CHECK ON KEYIN OVERRIDE
        IB=IPP(4,-1)
        IF (IB.GE.0)IBIAS=IB
        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
      ICK=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,LPR,
     *ICHR,MDNG,NOSFC)
  110 NB=0
      IF (IKWP('NODAT',1,0).NE.0)GO TO 150
      IF (KBUG.GT.0)CALL SDEST(' NO. OF REPORTS IS ',NRPT)
      SUM=0.
      SUMS=0.
C     BEGIN IMPLICIT DO LOOP ON REPORTS
      NN=1
      LL=0
C     NN IS USED TO COUNT ALL REPORTS
C     NB IS USED TO COUNT REPORTS FOR BARNES ANALYSIS
C     LL IS USED TO COUNT ALL VALID REPORTS (INCLUDING GROSS ERROR EDIT)
C     FOR THE FINAL EDIT STEP,IF REQUESTED
      IADD=0
      ERR=999999.
      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
      IADD=LOCS(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
      IADD=LOCS(N)
      IS=ISCL(N)
      IF (LIT(ICHR).EQ.LIST(N))GO TO 48
   47 CONTINUE
      GO TO 148
   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.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 IF (KBUF(IADD).EQ.MISS)GO TO 148
      FSCL=10.**IS
      IDAT=KBUF(IADD)
      IF (IAD2.NE.0)IDAT=IDAT-KBUF(IAD2)
      IF (IDAT.EQ.MISS)GO TO 148
      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
      IREP(LL)=NN
C     ABOVE SAVES ORIGINAL SUBSCRIPT FOR EDITING
C     ABOVE ARE SAVED FOR EDITING
      IF (CGS.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(N-1)
      SSCL=10.**LS
      DIR=FLOAT(IDIR)/SSCL*.0174563
      U=-SPD*SIN(DIR)
      V=-SPD*COS(DIR)
      DAT=U
      IF (LCHR.EQ.'V')DAT=V
   70 CONTINUE
C     ADD ARRAY FOR BARNE3
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 (NN.GT.NRPT)GO TO 140
      IF (NN.LE.MMAX)GO TO 15
  140 IF (IDATA.EQ.77777.AND.IGB.NE.1)GO TO 400
      IF (IDATA.EQ.77777)GO TO 150
      FN=NB
      SUM=SUM/FN
      SUMS=SUMS/FN
      SD=DSQRT(SUMS-SUM*SUM)
      SD=AMIN1(ERR,SD)
      SD=AMAX1(1.,SD)
      ISD=SD*100.
      IF (ERR.EQ.999999.)ERR=SD
      IF (KBUG.GT.0)CALL SDEST('DARA TOSSOUT 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
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 GPID
      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
      IOK=MDGET(MDNO,MDR,IREP(NN),KBUF)
      KBUF(NMOD)=MMOD
      IOK=MDPUT(MDNO,MDR,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=LUC(6)
C     FOLLOWING IS TO STORE RESULT IN SAME GRID FILE AS GUESS GRID
C     AS EXPECTED BY "UGVA"
      IF (NGG.NE.0)NGRF=NGFG
      NGRNO=0
      IOK=IGPUT(NGRF,NGRNO,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
      IOK=IGPUT(NGRF,NGRNO,IDL,NROWS,NCOLS,IGHD,ISTAT)
      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
      IOK=IGPUT(NGRF,NGRNO,IDL,NROWS,NCOLS,IGHD,ISTAT)
      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 ',M)
```

```
521         RETURN
522         END
```

```
//GWVA7000 JOB CLASS=B,MSGLEVEL=(0,0)
//* VLGWVA    SBG  12/23/83; ENTERED USER MANUAL CARD
// EXEC MCPRG,MOD=GWVA
//FORT.SYSIN DD *
@PROCESS SC(TQMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,WTOR,OPCOM,SQX,SQW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW)
@PROCESS SC(DOPEN,DREAD,DWRITE,ISFILE,WTOR,OPCOM,SQX,SQW)
@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 MDSET AND 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 MULTIPY 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,IPIC,IHMS,JT,JD
      COMMON /WTYPE/KGEO
      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 /ORBIT/NODE
      COMMON /TIGHT/ITOL
      COMMON /ENTRY/INIT
      EQUIVALENCE (IDOC(40),MDMR),(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)),(IGTINV,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)
      MDNO=LUC(5)
      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
      DO 1 K=36,41,2                 MDNO = IDOC(40)
      MDR=IDOC(K+1)                  MDR= IDOC(41)
      IF (MDNO.EQ.IDOC(K))GO TO 2
    1 CONTINUE
```

```
      GO TO 900
    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.'MC')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
      MDNO=IDOC(40)
      IF (MDINFO(MDNO,MDHD).NE.0)GO TO 900
      NKEYS=MDKEYS(MDNO,-1,LIST,ISCL,IUN,LOCS)
C     READ TEST RECORD
      M=0
  101 M=M+1
      IF (M.GT.100)GO TO 904
C     READ ROW HEADER
      IOK=MDGET(MDNO,MDR,0,IBUF)
C     READ COLUMN HEADER IF IT EXISTS
      IF (MDHD(8).NE.0)IOK=MDGET(MDNO,0,M,IBUF)
      IOK=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.)IOK=IGGET(NGRF,NGR,2400,JHIT2,NR,NC,IGHD)
      IF (IOK.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 (MDHD(8).NE.0)IOK=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,JP,IP,JS,JD,JT)
  168 I=(LOW-LON/10-INC2)/INC+1
      J=(LAM-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,KCL,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,NDR,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  IOK= ',IOK)
      RETURN
C
      END
```

```
//UGVA5910 JOB CLASS=A,MSGLEVEL=(0,0)
//* VLUGVA    JPN  01/23/84: MEMBER UPDATED
//* VLUGVA    SBG  12/23/83; ENTERED USER MANUAL CARD
// EXEC MCPRG,MOD=UGVA
//FORT.SYSIN DD *
@PROCESS SC(TQMES,LIT,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,TQ,DOPEN,DCLOSE,DREAD,DWRITE,WTOP,OPCOM,SQX,SQV)
@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> <NGFG>
C ?          POSITIONAL PARAMETERS:
C ?          <NGRID> BEGINNING NO. OF GRIDS TO BE UPDATED (NO DEFAULT)
C ?          <NGFG> GRID FILE CONTAINING GUESS GRIDS (DEFAULT VASTEXT)
C * SSEC/MCIDAS USER MANUAL - CHAP12
      COMMON /DOC/IDOC(100)
      DIMENSION LIST(26),KOUT(10),ISCALE(26),IUNIT(26),LOCS(26)
C     ABOVE TO RECEIVE MDKEY INFO
      INTEGER*4 OBUF(20)
      DIMENSION IGRID(3200),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)),(IGTINV,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 OBUF/20*'    '/
            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/
            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)
            NGFG=IPP(2,NGFG)
        100 NGE=NGB+70
            L=0
            DO 10 N=NGB,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
            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
    5 CONTINUE
      GO TO 10
    6 CONTINUE
      DO 80 I=1,64
   80 IGHD(I)=IGHDT(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 TQ FOR GUESS UPDATE
      CALL ENKODE(' (A4,8X,K5,7X,K5,7X,K5,7X)',OBUF,JCHAR(JB),ITLEVL,N,
     +IBIAS)
      CALL TQ(OBUF)
      CALL JSQX('BNVA    ',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
/*
//
```

```
//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(TQMES,EMES,DMES,MCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(ISFILE,TSWIO,WTOR,OPCOM,SGX,SOW)
@PROCESS SC(MOVB,MOVC,MOVW,CLEANW,DOPEN,DREAD,DWRITE,DCLOSE)
      SUBROUTINE MAINO
C *** CRT DISPLAY OF VAS-RETRIEVAL GUESS AT CURSOR POSITION
C ? DISPLAY VAS GUESS PROFILE AT CURSOR LOC OR AT SPECIFIED LAT/LON (CMH)
C ? KEYIN: GPVA <GSS> <SFC> <LAT> <LON> <MDNG>
C ?    POSITIONAL PARAMETERS
C ?         GSS - 'C'  'P' FOR CLIM OR PROF..DEFAULT TO MOIS
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 TSWIO(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     DSFC
     *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
/*
```

```
//EXVA7000 JOB  CLASS=B,MSGLEVEL=(0,0)
//  EXEC  MCPRG
//* VLEXVA   SBG  12/23/83; ENTERED USER MANUAL CARD
//FORT.SYSIN  DD *
@PROCESS SC(TQMES,EMES,DMES,NCOD,ENCODE,DECODE,LWGET,LWPUT,LWCLOS)
@PROCESS SC(DOPEN,DCLOSE,DREAD,DWRITE,ISFILE,VTOR,OPCOM,SQX,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 MDC(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(RETAIN,BETDOT,INAV,PTIME)
      GO TO 122
  120 CALL TSNIO(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
      IOK=MDINFO(MDNO,MDC)
      IF (IOK.LT.0)GO TO 900
      IOK=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
      IOK=MDGET(MDNO,MDR,NN,IOUT)
      FLATR=.0001*IOUT(NLAT)
      FLONR=.0001*IOUT(NLON)
      IF (IOK.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(PTIME,FLIN,FELE,FLAT,FLON,1,INAV,RETAIN,RETDOT,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
      IOK=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      T20
```

```
         *250    T200            ',0)
          KOUT(1)=8
          NOUT(1)=8
          L=1
          DO 220 K=1,8
          L=L+1
          MM=(ILV(K)-1)*9+NDX
          KOUT(L)=999
          NOUT(L)=999
          IF (IOUT(MM).EQ.MISS)GO TO 220
          KOUT(L)=IOUT(MM)
          NOUT(L)=IOUT(MM)-IOUT(MM+5)
      220 CONTINUE
          CALL OUTINT(KOUT)
          CALL OUTINT(NOUT)
          IF(ICUR.NE.0) GO TO 290
          IF(M.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
/*
//LKED.SYSIN DD *
 INCLUDE SYSLIB(NVLOPE,SCPACK,UTPACK,LWPACK)
 ENTRY NVLOPE
 NAME EXVA(R)
/*
```

```
//ESVA5910 JOB CLASS=A,MSGLEVEL=(0,0)
//* VLESVA   HBH  02/02/84: 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 OBUF(20)
         CHARACTER*8 MFILE
         REAL*8 DLIT
         COMMON /DOC/IDOC(112)
         COMMON /NAV/FLAT,FLON,ZENLOC,SZFN,IL,IE,IPAS,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/,NCCLS/56/,ICOL/1/,NDAT/0/
         MONO=LUC(5)
         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(BETAIN,BETDOT,INAV,PTIME)
      GO TO 120
  110 CALL TSNIO(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,LFLE,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,LOCS)
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(PTIME,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(PTIME,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 TSNIO(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,MDP,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(PTIME,FLIN,FELE,FLAT,FLON,2,INAV,BETAIN,BETDOT,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,BETDCT,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 JSQX('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 ',0)
      RETURN
      END
```

```
//VDVA7000 JOB CLASS=A,MSGLEVEL=(0,0)
//* VLVDVA   SBG  12/23/83: ENTERED USER MANUAL CARD
// EXEC MCPRC,MOD=VDVA
//FORT.SYSIN DD *
@PROCESS SC(TQMES,EMES,DMES,MCCD,ENCODE.DECODE.LWGET.LWPUT.LWCLOS)
@PROCESS SC(ISFILE,WTOR,OPCOM.SQX,SQW)
@PROCESS SC(DOPEN.DREAD.DWRITE.MOVB.MOVC.MOVW.CLEANW)
      SUBROUTINE MAINO
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 /SFCGNO/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     E
     *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   CHA
     *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/F/F/S/N...DEFAULT=B) (LINE) (ELEM) (VASDAT DIAGNOST
     *IC DISPLAY',0)
C     CALL SDEST('... N (NAV) GIVES SFC DATA IF AVAIL ... RTVL-FILE PTR
C    *MUST BE SET',0)
      RETURN
      END
```

```
//VTPX5910 JOB CLASS=B,MSGLEVEL=(0,0)
//* VLVTPX   HMW  02/04/84: MEMBER UPDATED
// EXEC MCPRG,MOD=VTPX
//FORT.SYSIN DD *
      SUBROUTINE MAIN0
C ?   VAS TOTAL PRECIPITABLE WATER VAPOR RETRIEVAL (H.M.WOOLF)
C ?      KEYIN: VTPX <KEYWORDS>
C ?        KEYWORDS (DEFAULTS IN PARENTHESES):
C ?          ARA=NUMBER OF DIGITAL AREA FOR IMAGING (0)
C ?          BOX=N (SIZE IN FOV'S OF SQUARE BOX (11))
C ?          SPC=IL IE (LINE AND ELEMENT SPACING OF RTVLS (BOX BOX))
C ?          BEG=IL IE(FIRST LINE AND ELEMENT TO DEFINE AREA (0 0))
C ?          END=LL LE (LAST LINE AND ELEMENT TO DEFINE AREA (0 0))
C ?          GSS=C (CLIM) G (GRID,DEFAULT)
C ?          SFC=1 (NO SURFACE ANALYSIS (0))SFC=3 SKIPS TRANSFER CALC.
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 ?          SSD=SS YYDDD - FOR NON-VIDEO TERMINAL (0 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)
      DIMENSION IRET(250),IAMES(8),IARET(220),IABUF(55),NFAIL(7)
C
      INTEGER 2 IARRAY(11,11,20)
C
      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/DBUG/LBUG
      COMMON/DOC/IDOC(100)
      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,IPAS,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,IRTE
      DATA CBLNK/' '/,CLETC/'C'/,CLETG/'G'/
      DATA ICV/7/,ICW/8/
      DATA TOTO/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=84035
      MCINIT=0
      IF(LUC(-25).EQ.1) MCINIT=LUC(-23)
      CALL CALDAY(IVER,IVY,IVM,IVD,IVMO)
      CALL ENKODE('(132X,T1,"BEGIN - VTPX - 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 1000
C * CHECK FOR OVERRIDE OF TERMINAL NUMBER
      IDEF=LUC(-20)
      ITERM=IKWP('TER',1,IDEF)
C * READ 'VASTEXT' DOCUMENTATION RECORD & EXTRACT NEEDED INFO
      CALL VRTIO(IRET,0,0)
      MDNG=IDOC(38)
      MDNR=IDOC(40)
      MDRR=IDOC(41)
C * CHECK FOR 'NO-OUTPUT-MEDIUM'
      IF(NARA.EQ.0.AND.MDRR.LT.0) GO TO 1020
      LRO=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
      IDEF=0
      NOSFC=IKWP('SFC',1,IDEF)
      IRTE=10
      IF(NOSFC.NE.3)GO TO 110
      NOSFC=0
      IRTE=0
  110 CONTINUE
      IDEF=11
      NBXS=IKWP('BOX',1,IDEF)
      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)
      LLINE=IKWP('END',1,0)
      LELEM=IKWP('END',2,0)
      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)
      ILINE=IKWP('BEG',1,0)
      IELEM=IKWP('BEG',2,0)
      IF(MDRR.LT.0) GO TO 120
      JDAY=IDOC(1)
      JTIME=IDOC(2)
      IRET(1)=JDAY
      IRET(2)=JTIME
      IRET(3)=LRO
      IOK=MDPUT(MDNR,MDRR,0,IRET)
      IF(IOK.NE.0) GO TO 980
  120 IF(ILINE*IELEM.EQ.0) GO TO 160
      IF(LUC(16).NE.0) GO TO 140
C * NON-VIDEO TERMINAL OR BACKGROUND - OBTAIN PARAMETERS FROM KEYIN
      JSAT=IKWP('SSD',1,0)
      JDATE=IKWP('SSD',2,0)
      GO TO 180
  140 CALL GETFRM(IFRM,IDIR)
      JSAT=IDIR(1)
      JDATE=IDIR(2)
      GO TO 180
  160 CALL TVSAT(IFRM,INRAS,INPIC,ILINE,IELEM,JSAT,JDATE,JTIME)
C * NEED 'OLDSTYLE' JDAY FOR NAVIGATION WITHIN VASDAT
  180 JDAY=JSAT*100000+JDATE
      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*INCRE*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.220) GO TO 1040
      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 VTQ(LBUF)
  220 CONTINUE
C
      NCONE=0
      LINET=-1
      DO 920 IL=ILINE,LLINE,INCIL
      LINET=LINET+1
      IF(LCRT.EQ.0) GO TO 240
      CALL ENKODE('("BEGIN LINE ",I4)',LBUF,IL)
      CALL VTQ(LBUF)
  240 LASLIN=-1
      KBOX=0
C
      DO 880 IE=IELEM,LELEM,INCIE
      KBOX=KBOX+1
      LASELE=-1
      ITFLG=1
      DO 870 KLIN=1,KLINES
      DO 860 KELE=1,KELEMS
      IURET=MAXWV
      DO 260 I=4,LENV
```

```
  260 IRET(I)=MISG
      DO 300 I=1,13
      VDAT(I)=VMISG
  300 NSPIN(I)=0
      ILINES=IL
      IELES=IE
      CALL VASDAT(ILINES,IELES,VDAT)
      IFAIL=1
      IF(ABS(VLATS).GE.90.) GO TO 850
      IF(ABS(VZEN).GT.60.) GO TO 850
      IF(LELEV.EQ.999999) GO TO 850
      IF(VDAT(1).LT.0.) GO TO 850
      IF(VDAT(ICV).EQ.VMISG) GO TO 850
      IF(VDAT(ICW).EQ.VMISG) GO TO 850
      IMF=1
      MSAM=0
      DO 340 I=KTCH,ICW
      IF(VDAT(I).LT.180.OR.VDAT(I).GT.330.) VDAT(I)=VMISG
      IF(VDAT(I).EQ.VMISG) GO TO 340
      IF(IMF.EQ.0) GO TO 320
      MSAM=MSAM+1
      IMF=0
  320 CONTINUE
C     SAVE REPRESENTATIVE SPIN CLOCK (ASSUME SPIN BUDGET INVARIANT)
      NSPIN(I)=ISPIN(I)
  340 CONTINUE
      IELEV=LELEV
  360 CONTINUE
      IFAIL=2
      IF(MSAM.EQ.0) GO TO 850
      IF(ITFLG.EQ.0) GO TO 420
      CALL GESPRO(IGES,NOSFC,MDNG)
      IFAIL=3
      IF(TGES(1).LE.0.) GO TO 850
      PSTA=IPSTA
      TSTA=.01*FLOAT(ITSFC)
C
      TDSTA=0.01*FLOAT(IDSFC)
      E=SATVAP(TDSTA)
      WSTA=(622.*E)/(PSTA-E)
      LS=NL
      DO 370 J=1,20
      I=NLP1-J
      DP=P(I)-PSTA
      IF(DP.GT.0.)GO TO 370
      IS=I+1
      IS=MIN0(IS,NL)
      GO TO 375
  370 CONTINUE
  375 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 380 I=ILO,40
      W(I)=W(I)*WSTA/W(NLS)
      T(I)=T(I)+DTDF*(P(I)-P(ILO))
      IF(I.LE.NLS)GO TO 380
      T(I)=TSTA
      W(I)=WSTA
  380 CONTINUE
  400 CONTINUE
      IF(LBUG.EQ.0) GO TO 420
      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,IZ10,ITSFC,IDSFC)
  420 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 720
      CALL ENKODE('(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 ENKODE('(132X,T1,6I8)',LBUF,IL,IE,LAT,LON,IVZ,ISZ)
      IF(LBUG.NE.0) CALL LTQ(LBUF)
      IF(LCRT.EQ.0) GO TO 720
      CALL VTQ(LBUF)
      CALL SDEST('  SAMPLE = ',NSAM)
  720 CONTINUE
C * DATA ACQUISITION COMPLETE
      CALL VASTPW(VDAT,TOTO,TSFS,URET,ITFLG,ISAT,KTCH,NLS)
      IFAIL=7
      IF(URET.EQ.VMISG) GO TO 850
      IFAIL=0
      KFLAG=2
C * OUTPUT ONLY EVERY FOURTH RETRIEVAL TO MD-FILE
      NDONE=NDONE+1
      MDOUT=0
      IF(MOD(NDONE,4).EQ.1) MDOUT=1
C * MAXIMUM PERMISSIBLE VALUE OF 'URET' IS 12 CM
      IURET=IROUND(URET*20.)
C
      IF(MDRR.LT.0) GO TO 800
      IF(MDOUT.EQ.0) GO TO 800
      K=10
      DO 780 I=KTCH,8
      IF(KTCH.EQ.5.AND.I.EQ.6) GO TO 780
      K=K+1
```

```
      IF(VDAT(I).EQ.VMISG) GO TO 780
      IRET(K)=IROUND(VDAT(I)*100.)
  780 CONTINUE
      IRET(14)=IROUND(TSFS*100.)
  800 IF(LBUG.EQ.0.AND.LCRT.EQ.0) GO TO 820
      CALL ENKODE('(132X,T1," *** T-P-W = ",F6.3)',LBUF,URET)
      IF(LCRT.NE.0) CALL VTG(LBUF)
      IF(LBUG.NE.0) CALL LTQ(LBUF)
  820 CONTINUE
      IF(IPLT.EQ.0) GO TO 840
      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)
  840 CONTINUE
      IF(MDRR.LT.0) GO TO 850
      IF(MDOUT.EQ.0) GO TO 850
      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(WSTA*1000.)
      LRO=LRO+1
      IOK=MDPUT(MDNR,MDRR,LRO,IRET)
      IF(IOK.NE.0) GO TO 960
      IRET(3)=LRO
      IOK=MDPUT(MDNR,MDRR,0,IRET)
      IF(IOK.NE.0) GO TO 980
  850 CONTINUE
      IF(IFAIL.NE.0) NFAIL(IFAIL)=NFAIL(IFAIL)+1
      IF(NARA.EQ.0) GO TO 860
C
      IARRAY(KELE,KLIN,KBOX)=IURET
C
  860 CONTINUE
  870 CONTINUE
  880 CONTINUE
      IF(NARA.EQ.0) GO TO 910
C
      DO 900 J=1,KLINES
      IA=0
      DO 890 K=1,KBOXES
      DO 890 I=1,KELEMS
      IA=IA+1
      IARET(IA)=IARRAY(I,J,K)
  890 CONTINUE
      CALL PACK(MELE,IARET,IABUF)
```

```
      CALL WRITA(NARA,IAREC,IABUF)
      IAREC=IAREC+1
  900 CONTINUE
C
  910 CALL EDEST('FINISHED LINE ',IL)
  920 CONTINUE
      IF(MDRR.LT.0) GO TO 940
      IDOC(100)=LRO
      CALL VRTIO(IRET,0,1)
  940 IF(NARA.NE.0) CALL CLOSAO(NARA)
      CALL EDEST('  *** ALL DONE *** ',0)
      CALL ENKODE('("FAILURE SUMMARY ",7I5)',LBUF,NFAIL)
      CALL VTQ(LBUF)
      RETURN
  960 CALL EDEST('TROUBLE WRITING COLUMN(RECORD) ',LRO)
      GO TO 1060
  980 CALL EDEST('TROUBLE WRITING HEADER FOR ROW ',MDRR)
      GO TO 1060
 1000 CALL EDEST('INVALID AREA NUMBER ',NARA)
      GO TO 1060
 1020 CALL EDEST('NO OUTPUT MEDIUM (AREA OR MD) SPECIFIED ***',0)
      GO TO 1060
 1040 CALL EDEST('IMAGE TOO WIDE - MAXELE IS 220, ESIZ = ',NELE)
 1060 CALL ABORT(0)
      RETURN
      END
```

References

MacMillin, L.M., 1978: An improved technique for obtaining clear radiances from cloud contaminated rqdiances. 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.