LINE INTERFEROMETER PROGRAM

Mel Wright

and

Chuck Moore

Fourth Edition
September 1972

FOREWARD

Considerable effort has been made to make this document accurate. It is inevitable that mistakes will remain, therefore please advise the authors of any discrepancies.

The Program is very flexible and its design has followed the requirements of the line interferometer so far. There are many hidden features (e.g., tracking planets) and additions are readily made, so if you have a need not met by the program, again, consult the authors.

Efforts will be made to keep the documentation up-to-date.

15 January 1972

MCHW

CHM

PREFACE TO THIRD EDITION

A small number of additions and improvements have been made to the program and this edition updates and replaces the second edition. Three private disk packs are now in use and can be allocated to major users of the system. A small scratch data file "FREQS" is also available. Maps (HI or polarized S and X band) stored on disk through the program can be displayed on the printer, on the "ARDS" display or CalComp, or punched out on cards for input to other programs. Maps can also be "cleaned" and re-stored on disk.

Mel Wright

May 1972

PREFACE TO FOURTH EDITION

Several new subroutines have been added. The following are of general use:

STACK and STORE-FLUX (Section 6)

PLANE (Section 8).

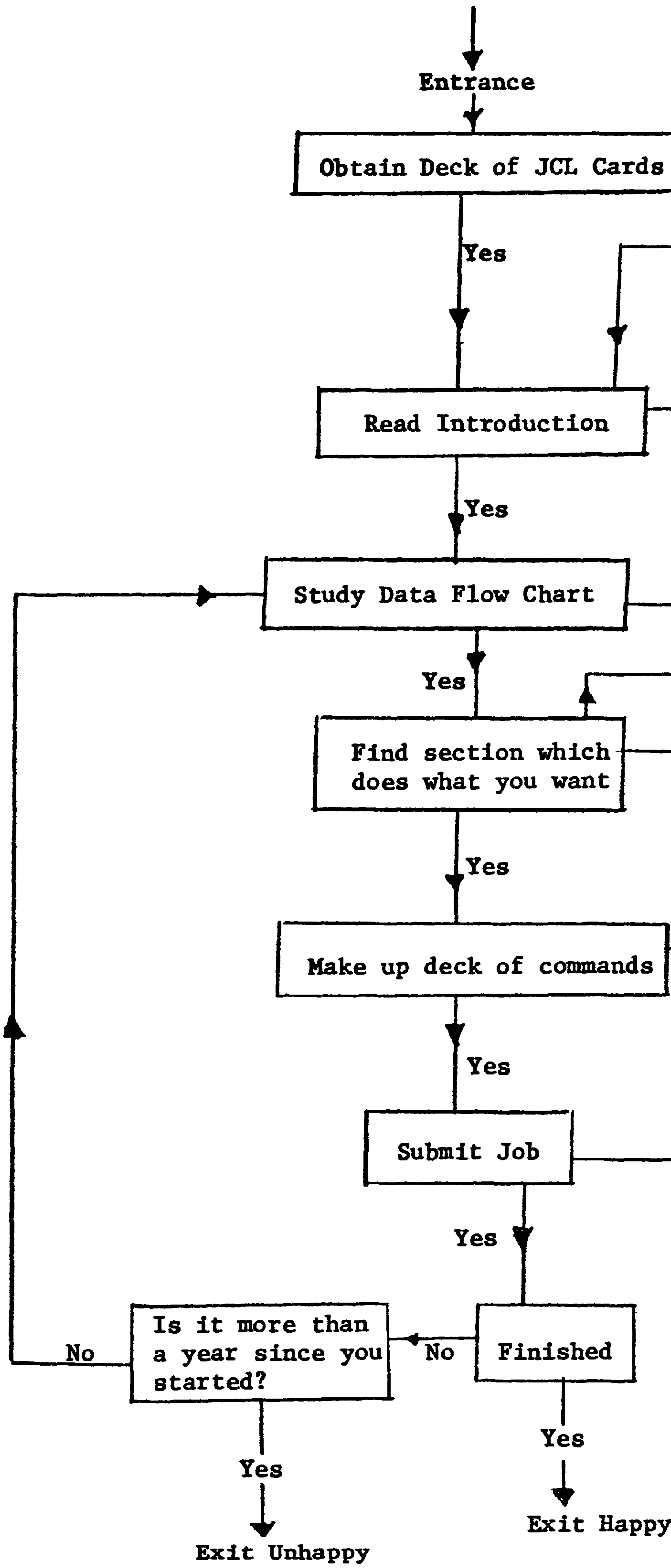
There are several new "Fixup" routines which have been moved into a module called FIXUP. (For example FLIP and DELAY-COR (Section 6)). The 180K inversion program can now apply a grading function to the u-v plane (Section 8). Six disk packs (GHC1 through GHC6) each 23179 blocks and FREQS 4096 blocks are now allocable to users. I have tried to keep this document concise and accurate. Please advise me of errors or incomprehendibilities!

Mel Wright

September 1972

CONTENTS

APPENDIX


Entrance
Obtain Deck of JCL Cards
Yes
Read Appendix 5
No
Read Introduction
No
Come here often?
Yes
Yes
Study Data Flow Chart
No
Ask Authors
Yes
NO
Yes
Find section which does what you want
No
Come here often?
Yes
Make up deck of commands
No
Consult Appendix 6
Yes
Submit Job
No
Yes
Is it more than a year since you started?
No
No
Finished
Yes
Yes
Exit Unhappy
Exit Happy

USERS GUIDE

1. INTRODUCTION

a) Purpose and Compatability

The Line interferometer program is designed to reduce and display data from the NRAO line interferometer. Many of the reduction programs have been re-coded from the Dual-Frequency system with particular attention to simplifying the input. The 360 program described here is separate from the Dual Frequency system, but users of the latter will be familiar with many of the job steps. Compatability with Dual Frequency data format has been maintained so that either Line or Dual Frequency data may be reduced using this program. In particular this program provides an automatic way of inverting, mapping and displaying polarized data.

b) Language

The program is a collection of Fortran and IBM - Assembler subroutines called by a Forth program. Essentially the Forth program provides a flexible language for defining operations and the program is directed by punching Verbs and Parameters in free format input. The program is compiled in an overlay structure so that one standard deck together with the Forth directives suffices for all use of the system.

c) The aim of this documentation is to provide a description of the program and a users' guide allowing him to use and extend the present program with his own subroutines.

The following sections provide a description of the Program and examples of its use to reduce the data.

d) Data Flow

1. The data from the telescope is recorded on 7-track tape data records.

2. The 7-track telescope tapes may be transferred to an "Observer" or "Back-up" 9-track tape in the original data format by a separate program, which rejects and gives advice of wrong length records, etc. No reformatting is done by this program which only serves to store the data on 9-track tape and release the 7-track tape.

3. Either 7-track telescope tape or 9-track observer tape may be loaded onto disk and converted into a catalogued disk data set. The data is reformatted at this time into 512 word records with the line data scaled into 16 bit words by the "GAIN" settings. Error messages are generated at the time the data is reformatted.

4. The data is catalogued, and can be accessed by Source and by Scan number.

5. The data can be inspected, edited and calibrated on the disk.

6. The data may be written out and reloaded from 9-track tape for storage at anytime.

7. Any continuum or averaged frequency channel(s) may be imaged into a gridded u-v plane which is also stored on disk.

8. The u-v planes may be Fourier inverted in-core in a separate 180K job.

9. The integer maps can be stored on disk, cleaned and the cleaned maps also stored for further processing and display.

10. A sequence of maps at an equal frequency interval may be transposed into space-frequency maps.

11. Either RA-DEC or RA-frequency maps may be displayed as printed maps or on the ARDS display as profiles or contours, or contoured through the "CalComp" routines.

12. Maps can be stored on 9-track tape.

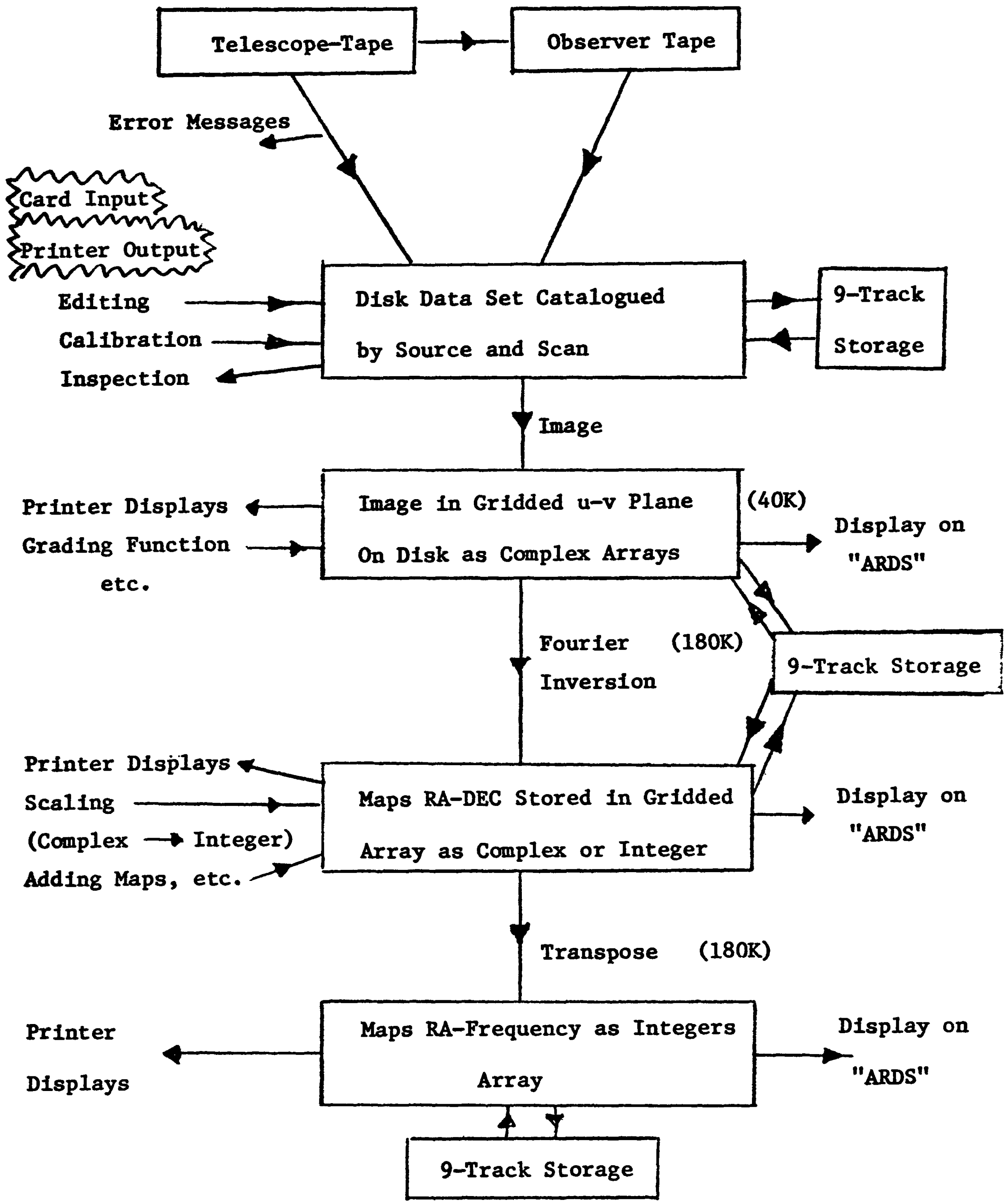


Figure 2. DATA FLOW

2. ORGANIZATION OF PROGRAM

a) Forth

This is the users interface with the program and provides a flexible free-format language for directing subroutine calls within the program.

Two types of words may be distinguised in the use of Forth.

1) Verbs cause some action - usually a subroutine call

2) Nouns set parameters to make choices and do not need repeating unless reset.

The user directs the program by typing a string of nouns to set parametric input followed by a verb to cause the action to take place. A fuller description of the Forth program is contained in Appendix 1 and a dictionary of Forth words is given in Appendix 2.

b) Interface

The Forth is interfaced to the subroutines in two ways.

i) through a Fortran main routine "INTER" which makes calls to the subroutines.

ii) through a common area "USER" which contains all the parameters which are set by Forth Nouns to be used by the subroutines.

c) Subroutines

The subroutines are written in Fortran or Assembler language and are compiled in a hierachical overlay structure. Additional subroutines may be easily added by the user and directions for doing this are contained in Appendix 3.

d) Data Flow

The data from the telescope consists of head records and data records which are recorded on 7 track tape, which is the starting point for this program. The 7 track tape is converted into a catalogued disk data set of constant length (512x16 bit integer) records and all disk access is handled by the routine "FREQS" through a common area "PRIME". FREQS is used to read and write records and index the data by scan # and by source name.

Display, calibration, and editing may all be performed from the disk data set without transfer to tape, but the data may be dumped or transferred to tape at anytime.

e) Mapping

A source may be arrayed into a gridded u-v plane inverted to obtain a map of the sky in any frequency channel (frequency channels can be combined). These maps are also stored on disk and a subroutine "ROWS" transfers the data through a common area "PLANE". In order to obtain a frequency spectra at a point in the sky, the required frequency channels must be separately inverted, stored and re-displayed as a spectra.

f) Disk Layout

The Forth Program, Index by Scan, Index by Source name, Data and Maps all reside on disk. The layout is shown in appendix 5. The user must himself allocate blocks between 110 and 159 to hold the indices for given sources, before a tape can be loaded.

3. LOADING DATA ONTO DISK

The line data is contained in a double precision word to avoid analogue gain setting for the correlator. This is converted to single precision (16 bits) at the time the telescope is loaded onto disk, the line data being divided by the analogue gain settings used for the broadband correlators and stored in the data records. The user may however provide his own scaling factors independent of the broadband gains used at the telescope by GAINS and the appropriate value of NU (see Section 7). For 10 NU = the broadband gains recorded on tape are used to scale the line data into single precision and the GAINS values are not used. The following steps are required to load data onto disk.

A. Establish an index of sources:

TAPES LOAD	load in program for tape I/O
ERASE-SOURCE	erases the previous source index
160 DEFINE AVAIL	sets the pointer to the next available block on disk back to 160 - the first available location for data (see Appendix 5).
DEFINE EVERYTHING	must be defined here

```
110 GAINS 0202   INCLUDE 3C48      now follows a list of sources to be indexed
111 GAINS 0202   INCLUDE 3C48-1    Disk blocks 110 through 159 are used to hold
112 GAINS 0000   INCLUDE P0237-23    source index.  These cards assign  blocks
          etc.                     to hold index for the sources to be included.
159 GAINS 0000   INCLUDE M33
```

FLUSH	terminate index: Flush the index onto disk.

Note 1

Additional sources may be added to an already established index by the appropriate INCLUDE cards and FLUSH

Note 2

You may add scans to an already established index of sources simply by reading the tape, i.e. omit all the above.

Note 3

More than one source (but not more than 256 scans) may be included in one block. In this case all the sources within the block are indexed by any of the names.

B. Read the tape and transfer the scans to disk

Only scans of INCLUDED sources are transferred. Data from either the telescope-tape or observer-tape may be read onto disk. For telescope tapes the whole scan range is read. For an observer-tape the given scan range is read. The appropriate JCL card must be included.

10 NU = TELESCOPE-TAPE	reads the sources above onto disk, reformats data and monitors ADC and discontinuities in data.
10 NU = 1544 1699 1544 1699 OBSERVER TAPE	transfers the given scan range to disk.

The next available block on disk is printed out by FIND AVAIL. This tells you how much data you have on disk. Maps must be stored after this location to avoid overwriting the data.

4. DATA DISPLAY

Data may be accessed by source and by scan number. Printed numbers or plots can be obtained for the different receivers or correlators.

Most of the displays are driven by SCANS. This is the verb which causes the selected action for each scan in the given range.

A. <u>Displays driven by SCANS</u>

HEADER
BASELINES
LOOK
DELAYS
SPECTRAL-DENSITY
PROFS
DISPLAY
DUMP
HEX-DUMP
MONITOR
DELAYS

Example 1: Displays of correlators may be printed as amplitude and phase (VECTOR) or real and imaginary (COMPLEX)

Choices are listed below:

FOR	EVERYTHING SOURCENAME	CONTINUUM RX-SUMS BROAD-BAND RX-A RX-AB RX-ABCD	VECTOR COMPLEX	AVERAGE n INT =	LOOK	m1 m2 SCANS m1 A-SCAN

The verb is SCANS which drives the display over the given range. One record (1 INT =) is the default value. Parameters remain constant unless reset.

Examples are:

1)	FOR EVERYTHING CONTINUUM VECTOR LOOK 3000 3024 SCANS	prints amplitude and phase of continuum channels (1 line per record)
2)	FOR NGC2403 RX-SUMS VECTOR LOOK 3000 3024 SCANS	prints added line channels for NGC2403 from scan 3000 to 3024 (1 line per record)
3)	FOR EVERYTHING AVERAGE BROAD-BAND COMPLEX LOOK 3124 3135 SCANS	prints the scan-average as real and imaginary for both continuum and rx-sums
4)	RX-CD VECTOR LOOK 1560 1780 SCANS	prints the line data 8 pairs to a line for the (second) 96 frequency channels
5)	DUMP 1010 1050 SCANS	the whole record is dumped in decimal
	HEX-DUMP 1030 A-SCAN	gives a hex instead of decimal dump.

Example 2: Plotted output. The following options are available:

1)	FOR	EVERYTHING SOURCENAME	CONTINUUM RX-SUMS BROAD-BAND	AMPL PHASE	AVERAGE n INT =	MONITOR	m1 m2 SCANS

Example 2: Plotted output

2) 0 30000 SCALE set scale

FOR	EVERYTHING SOURCENAME	AMPL PHASE	RX-AB RX-CD	SPECTRAL-DENSITY	m1 m2 SCANS

A matrix printout of 96 frequencies. The range given in "scale" is divided into 20 equal intervals which are printed as 0, blank, 1,blank,2,--------,9,blank.

3)	RX-AB RX-CD	PROFS	1030	1400	SCANS	Self-scaling amplitude and phase plot of scan averaged profile.

4) PLOT PRINT	REAL IMAG AMPL PHASE COMPLEX VECTOR	CONTINUUM LINE-SUM RX-AB FREQ 10,15,16,17,20,*		DISPLAY 1034 A-SCAN

Display prints up to 20 columns or plots up to 10 columns of numbers.

Example 3: Analog displays

HEADER	1010	1050	SCANS	a print out of header information
BASELINES	1010	1050	SCANS	prints out baselines
WATER	1010	1050	SCANS	water vapor receiver

DELAYS prints three chosen amplitudes and phases versus delays in binary. A correlator list is also required thus:

CONTINUUM

RX-SUMS DELAYS 1015 1050 SCANS

RX-CD FREQ 10, 11, 12,*

---

* N.B.: The final "," at the end of the list of scans or frequencies.

B) <u>Independent Displays not using SCANS</u>

1) LIST-SCANS lists scan numbers on disk

2) PROFILES

Profiles down the page may be obtained for selected scans by

SCAN-LIST 10, 100, 1000, 1001, 1002,

PRINT PLOT	REAL IMAG AMPL PHASE COMPLEX VECTOR	PROFILES

5. CALIBRATION

Five calibrations are envisaged:

i) interferometer baseline

ii) observing-frequency dependent instrumental phase

iii) instrumental phase

iv) shape of the IF passband

v) instrumental gain

i) Baseline Calibration

BASES LOAD — load program

5 VARIABLES

CONTINUUM — use continuum channels

CAL	3C48	100	200	SCANS	
CAL	3C309.1	150	300	SCANS	selects the sources to be used as baseline calibrations
CAL	3C418	170	270	SCANS	

SOLVE — performs a single least squares iteration to the phase of the continuum data to improve the 5 variables: phase center, Bx, By, Bz, K

ii) Combined Fit

A combined baseline and phase-frequency fit may be made by:

6 VARIABLES

RX-SUMS — use added line channels

CAL	3C48	500	1700	SCANS	
CAL	3C209.1	550	750	SCANS	sources to be used as calibrators
CAL	3C295	600	755	SCANS	

SOLVE performs a least squares iteration to the phase of the continuum or line-sum channels to improve ϕ_o, Bx, By, Bz, K, $d\phi/df$.

These calibrating programs print out the best fitting parameters, their rms deviation, and a correlation matrix of the variables. Inspection of the data as a phase-time plot can be obtained before and after the data has been corrected. The corrections are stored and may be applied directly:

BASE-COR	500	1000	SCANS	applies the stored baseline corrections to all bandwidths
"n" BANDWIDTH				
BASE-COR	600	700	SCANS	corrects the data for bandwidth "n" 1≡10 MHz thru 9≡39 KHz n=0 is the default value

An initial guess may, and should, be entered if there exists a lobe ambiguity in the baseline or phase-frequency parameters. Otherwise the least squares solution may find a false minimum. For this reason it is a good idea to observe a calibrator at a number of frequencies as a cross-check.

0	0	0	0	0	0	B12	
0	0	0	0	0	1413	B13	corrects all B/W's for initial values of all 6 variables, ϕo, ΔBx, ΔBy, ΔBz, ΔK, $d\phi/df$. units: millilobes, $\lambda/1000$ and millilobes per MHz respectively
0	0	0	0	0	-121	B23	
0	0	0	0	0	0	B14	

BASE-COR 100 1000 SCANS

iii) Instrumental Phase

A phase centre is calculated in the baseline fitting program, but the instrumental phase may be computed between jumps by:

1 VARIABLE

CAL	3C295	100	300	SCANS	
CAL	3C48	100	300	SCANS	determine instrumental phase from these calibrators
CAL	3C309.1	100	300	SCANS	

SOLVE
FOR EVERYTHING
BASE-COR 100 300 SCANS correct all B/W's by default

iv) IF Passband

PASS-BAND 144 A-SCAN	generates the shape of the scan averaged spectrum for scan 144 and stores its complex reciprocal on disk.
RX-AB EDIT-BAND 145 147 SCANS	multiplies the given range of scans by the stored reciprocals. Application of the calibrator to itself yields a corrected pass-band of constant amplitude and constant phase. A range of (-4, 4) is allowed in amplitude and the digital precision is limited to one part in 2^{12}. The average phase across the pass-band is preserved through this correction.

v) Gain Calibration

To inspect for system gain variations (actually system noise temperature variations if ALC is used):

1) Store the calibrator fluxes (in mfu) into the data records. e.g.:

FOR 3C48

15600 STORE-FLUX 1000 9000 SCANS

FOR 3C147

22500 STORE-FLUX 1000 9000 SCANS

2) The reciprocal gains i.e., 1000/(counts per mfu) can be printed out using:

FOR EVERYTHING AVERAGE BROADBAND RECIPROCAL-GAINS 1000 9000 SCANS
(Optional)

3) These are the numbers to be used in MULT-AMPL if you wish to correct for temporal gain variations (Section 6).

6. EDITING

Editing is provided by three subroutines "Edit", "Phase" and "Fixup" each with several entry points. Edit provides for Editing directly. Phase corrects for baseline, position, clock, atmospheric and observing frequency changes. Fixup is used for special corrections to the data.

A) Direct Editing

FETCH
STORE
DELETE
MULT-AMPL
ADD-PHASE
EDIT-BAND
STACK
STORE-FLUX

Examples

EDITS LOAD   load program

FOR 3C10
SCAN-LIST   120, 132, 115, 1040, into 1000 STACK

"STACK" vector averages given scan-list into a new scan stored on disk. This new scan can be displayed or used like any other scan.

FOR 3C48 15600 STORE-FLUX 1000 8500 SCANS. Store flux in mfu into the data records of 3C48.

4 FETCH 144 A-SCAN   print word 4 of every record in scan 144

50 INTO 4 STORE DATA 144 A-SCAN	store the value 50 into word 4 of data records
RX-A DELETE 140 150 SCANS	flags line receiver "bad"
COR-1 DELETE 140 A-SCANS	flags continuum "bad" This data is then ignored by subsequent phases of the program.
RX-AB 500 MULT-AMPL 144 170 SCANS	multiplies data by 500/1000
RX-AB 50 ADD-PHASE 144 170 SCANS	add 50 degrees of phase to RX-AB

B) Phase Corrections

```
CLOCK-COR

POSITION-COR

PHASE-COR
```

Examples

FOR EVERYTHING

PHASE-COR 140 5260 SCANS	corrects for atmospheric phase effects
10 CLOCK-COR 5260 5400 SCANS	adds the given correction (ms) to the clock

FOR 3C286

130 40 POSITION-COR

500 5000 SCANS

Adds corrections 130 x 1/100 secs of time in RA, 40 x 1/10 secs arc in DEC to positon of 3C286. Decimal points may be inserted wihout effect so the position changes may also be written 1.30 4.0. Trailing zeros must however be present.

C) Fixups

Please look at the subroutine listing.

-PHASE	changes the sense of the phase
FLIP	reverses the direction of the profiles

d1 d2 d3 DELAY-COR — corrects for delay errors. d_1 d_2 and d_3 are the octal delays actually set on the digi-switches.

All of the above corrections are driven by SCANS e.g.,

0 4463 4211 DELAY-COR 11130 11142 SCANS

Many more have been used and many more are possible. "Fixup" may be your answer!

7. TAPE TRANSFER

JCL cards must be included. The convention is to read tapes on //FT03F001 and write on //FT04F001.

TAPES LOAD			load program
10 NU = TELESCOPE-TAPE*			transfers the telescope-tape data to a catalogued disk data set
10 NU = 500	2700	OBSERVER-TAPE*	transfers given scan range on back-up tape to disk data set
500	2700	WRITE-NINE	writes a scan sequential 9 track tape from disk for scans 500 2700
500	2700	LOAD-NINE	loads the 9 track tape (written by above)
500	2700	1 BAND-WIDTH	as LOAD-NINE but loads only the 10 MHz Bandwidth data(2 ≡ 5 MHz, ..., 9 ≡ 39 kHz,). This allows sorting the data by bandwidth
500	2700	COPY-NINE	copies 9 track tapes
500	7000	TAPE-INDEX	list of scans on tape in given range

The disk-file may be dumped by block numbers onto tape and re-loaded through two separate programs. "DUMP" and "LOAD" (See Appendix 7).

The prefix "# NU =" must be used before reading in a telescope-tape or observer tape.

FOR "10 NU" = The broadband correlator gains, set at the telescope by the observer source card, are used to scale the line data into single precision. Also the interferometer baselines attached to the correlator are defined at the telescope by the digi-switches over the correlator. If however you wish to change this:

"17 NU =" applies the recorded gains and connections to the to the correlator but avoids correcting the baselines for frequency. This is appropriate to double sideband operation.

For NU less than 10 the user re-defines the gain settings and connections. The gain to be used for each baseline should now be included in the "GAINS" word. ($0 \equiv$ x1, $2 \equiv$ x.1, $3 \equiv$ x.01, $1 \equiv$ x.001) E.g.

GAINS 0032 scales BL12 and BL13 by x1
scales BL14 (autocorrelator) by x.01
and scales BL23 by x.1

The value of NU defines the connections.

Cases allowed for at present are:

a) 2 receiver mode

4 NU = channels 1-192 on BL12, channels 193-384 on BL23

5 NU = channels 1-192 on BL13, channels 193-384 on BL23

b) 3/4 receiver mode

6 NU = channels 1-96 on BL12, channels 97-192 on BL13 channels 193-288 on BL23, channels 289-384 are the autocorrelation which is best scaled by an extra factor of 10.

c) 1 receiver mode

7 NU = channels 1-384 on BL23

8.A. MAPPING

The mapping process has two stages:

1) Imaging the data for the desired source, correlators and baselines into a gridded u-v plane.

2) Fourier inverting the u-v plane to obtain a map of the brightness distribution in the x-y plane.

The Fourier inversion may be performed in the usual 90K job using disk storage but is better performed in a separate 180K job which also provides for storing the maps as integers and cleaning

A) 90K Mapping

DIMENSION
RESOLUTION
CONTINUUM
RX-SUMS
LINE-FREQ
ONE-FREQ
SHORT
IMAGE
MAP
PLANE
READMAP
TRANSFORM

Examples

MAPS LOAD	load program
127 DIMENSION	128 x 128 array (this is the default and the maximum value)
30 30 RESOLUTION	3" arc per cell in sky
3000 ARRAY =	starting address for map on disk (128 blocks per map)
CONTINUUM	use continuum correlators
RX-SUMS	use correlator sums
1 9 ONE-FREQ	u-v plane to be made from $\frac{1}{9}\sum_1^9 \text{frequ}_i$. Produces real map (imaginary part zero). Amplitude and phase in the u-v plane are meaningful.

1 9 LINE-FREQ	frequencies 1 thru 9 added together and end up in Real map, 10 thru 18 in Imaginary, "Amplitude" and "phase" in u-v plane are not valid.
SHORT	"SHORT" in all the above cases reduces the lists of correlators to the first member only, so that only the short baseline (BL23) is imaged into the u-v plane.
500 27000 IMAGE NGC2403	image the given scan range and source into a gridded u-v plane
0 100 SCALE	
AMPL | MAP	a printed plot of "1" thru "9" ranging over a previous set scale
PHASE |	
REAL | PLANE	a 2 digit display of left half plane
IMAG |	inverts the data and puts it back
TRANSFORM	into block 3000
0 10000 SCALE	re-set scale
44 76 45 77 OUTER-AREA	set area to be displayed
50 70 60 70 INNER-AREA	by read-map below
REAL READ-MAP	
IMAG READ-MAP	

Readmap is a 4 digit printout of the given outer-area (maximum 32 x 32) and also gives the sums and averages over the inner and outer areas specified.

B) Map Operations

BEAM
MOVE-MAP
MULT-MAP
ADD-MAP
DIVIDE-MAP

Examples

3000 INTO 3128 MOVE-MAP	enables one to preserve the u-v plane by moving it before executing one of the following.
3000 INTO 3128 ADD-MAP	adds 1st map into 2nd
3000 INTO 1328 MULT-MAP	multiplies 1st map into 2nd
3000 INTO 1328 DIVIDE-MAP	divides 1st map into 2nd
10 LOW = 3000 INTO 3128 BEAM	creates a u-v plane which when transformed gives the beam offset 10 cells in RA from the map center

8.B. 180K INVERSION PROGRAM

A separate 180K job is now used to perform an in-core Fourier transform. The previously stored u-v plane is first read into core as a floating-point array. The u-v plane can then be weighted by a Gaussian grading function and/or a beam can be added to real* data. The u-v plane is then Fourier transformed and scaled into two integer maps (real and imaginary parts) of transform) which can be displayed and stored onto disk taking 32 blocks each.

```
UV
SKY
GET
BEAM
GRADE
SQUEEZE
FLOAT
TRANSFORM
MAP
PLOT
SECOND MAP
PUT
TRANSPOSE
```

Example 1

UV 3000 GET	move into core the u-v plane stored at block 3000 on disk
TRANSFORM	Fourier transform
1000 100 SQUEEZE	condense to integer maps(divide real imaginary parts by 1000 and 100 respectively)
MAP	1 digit printout of real map as contours
4000 PUT	store integer map starting at block 4000
SECOND MAP	imaginary map printout
4032 PUT	store integer map (32 blocks per map)

* See overpage

Example 2: Grading

```
        30  30 RESOLUTION

UV      3000 GET
        BEAM
        50 GRADE
        TRANSFORM
        1000 100 SQUEEZE
        PLOT
        4064 PUT
        SECOND PLOT
        4096 PUT
```

BEAM, in the 180K mapping program adds 1000 into imaginary part before "TRANSFORM", and can be used for a real map to give the beam shape in the imaginary (or second) map. BEAM cannot be used with LINE-FREQ or polarized data (which already use the imaginary map.)

"GRADE" applies a Gaussian grading function to the u-v plane (and beam) such that the sky is effectively convolved by a 50 arc sec beam. It is necessary to first re-specify the x-y interval in 1/10 arc second units by "RESOLUTION".

"PLOT" is a 2 digit printout of the center half of the synthesized map.

Example 3: Re-inversion

SKY	"SKY" specifies an integer map.
4000 GET	Move integer map into core
FLOAT	"FLOAT" converts to floating point.
TRANSFORM	Re-transform sky map to u-v plane.
100 100 SQUEEZE	Scale back into integer maps.
PLOT	Plot real part in 2 digit numbers.
SECOND PLOT	Plot imaginary part.

The integer maps stored by the 180K program take 32 blocks each (for a 128 x 128 map). These maps can be used as input for an iterative source subtraction program (Section 10) which stores the cleaned maps back onto disk in a new location.

9. DUAL FREQUENCY DATA

The program has been designed to handle dual frequency data. Either 7 track telescope tapes or 9 track calibrated or uncalibrated tapes may be loaded onto disk and all the baseline fitting and display programs are available by specifying the appropriate correlators. In order to map a source from a calibrated dual-frequency tape the following steps are required.

A) Loading Tapes onto Disk

TAPES LOAD	
ERASE-SOURCE 160 DEFINE AVAIL	DEFINE EVERYTHING
110 GAINS 0000 INCLUDE 3C20	
FLUSH	
500 28000 DUAL-FREQ	loads 9 track tapes written by the dual frequency system

Display of Data Options

S-BAND	RR LL	AVERAGE	VECTOR	LOOK	500 28000 SCANS	
X-BAND	RL LR	n INT=	COMPLEX		1050 A-SCANS	Displays the scan-averaged correlators selected
			or			
			AMPL PHASE	MONITOR		Also DUMP, HEX-DUMP and other relevant displays work on dual-frequency data.

Editing and Calibrating Options

The relevant Editing and Calibrating described in Section 5 and 6 in principle also works on dual-frequency data.

B) <u>Making a U-V Plane on Disk</u>

A gridded u-v plane is made from the data stored on disk and the u-v plane is itself stored at a location specified by the user. The next available location after the data is printed out by FIND AVAIL. u-v planes are stored as a 128 x 128 complex array taking 128 blocks of storage. Each u-v plane should accordingly be allocated 128 blocks of disk storage, starting beyond the data.

```
MAPS LOAD
     10   10   RESOLUTION          1 x 1 arc cell size in sky
                                   appropriate for X-BAND
      0  100   SCALE               set scale for displays

        3000   ARRAY=              starting address of u-v plane.

     X-BAND      RR
                 LL  500 28000 IMAGE 3C20     Make gridded u-v plane for
     S-BAND      RL                           chosen band and correlator.

     500  28000 IMAGE 3C20
```

"RL" automatically gives RL and LR* in u-v plane for polarized data.

```
          AMPL   |
          PHASE  |  MAP       "MAP" gives a 1 digit display of the whole u-v plan
          REAL   |
                 |  PLANE     "PLANE" gives a 2 digit display of half the u-v
          IMAG   |              plane.
```

The polarized maps may be transformed and operated on as for line data. Note that the amplitude and phase of a polarized map are the intensity and twice the position angle of the polarization.

```
          RR  |
              |                     are defined to give maps from
          LL  |   SHORT
              |                     the 2-3 baseline only.
          RL  |
```

Dual-frequency maps may be inverted and cleaned through the 180K program exactly as for 21 cm maps.

10. CLEANING

(For philosophy see e.g. Interferometry and Aperture Synthesis, Fomalont and Wright in Galactic and Extragalactic Radio Astronomy)

CLEANF and CLEANH programs

Clean is the iterative source subtraction program which is interfaced with the H-Line system. It handles polarized or unpolarized data taking maps stored on disk, "cleaning" and placing cleaned maps in a new disk location. Cleaned maps may be displayed on the printer, storage tube display or CalComp and punched out on cards.

The program is overlayed, heavily optimized and runs in a 90K partition. The maximum of 250 iterations takes ≈ 6 mins. c.p.u. time per clean.

There are 5 control cards which are used by CLEAN, TAU (Hydrogen Line optical depth program) PPLOT (Polarized contouring program) HPLOT (H-Line contouring program and PMAP (polarization comparison program).

Card 1.

Format (1X, A8, R12.2, S12.1, 1X, L1, 1X, F7.1, 1X, 2E15.4)

SOURCE	source name
RA	ra
DEC	dec
POL N-FLAG	T for polarized otherwise F.
XYINT	Grid size in arcseconds used for map (as set in making u-v plane by RESOLUTION)
BEAMMAX MAPMAX	maxima printed out by INVERSION program.

Card 2

Format (4(1X, I2), 20X, 3(1X, F7.1))

XLO XHI YLO YHI	Cleaning area in units (1 to 64) printed on borders of maps printed maps.
BMAJ BMIN	Half-widths (in arcseconds) of Gaussian clean beam used to convolve clean map
PA	tangent of angle of parallelogram cleaning area. (See Figure 3.) CLEANF is suitable for a source elongated NORTH-SOUTH. CLEANH for a source elongated East-West. See Figure 3.

Card 3

Format (I4, 1X, F5.3, 1X, L1)

NIT	maximum number of iterations to be used
PERC	smallest source to be subtracted, as percentage of original maximum.
PUNCH	T for punched output otherwise F.

Card 4

Format (4(1X, I5))

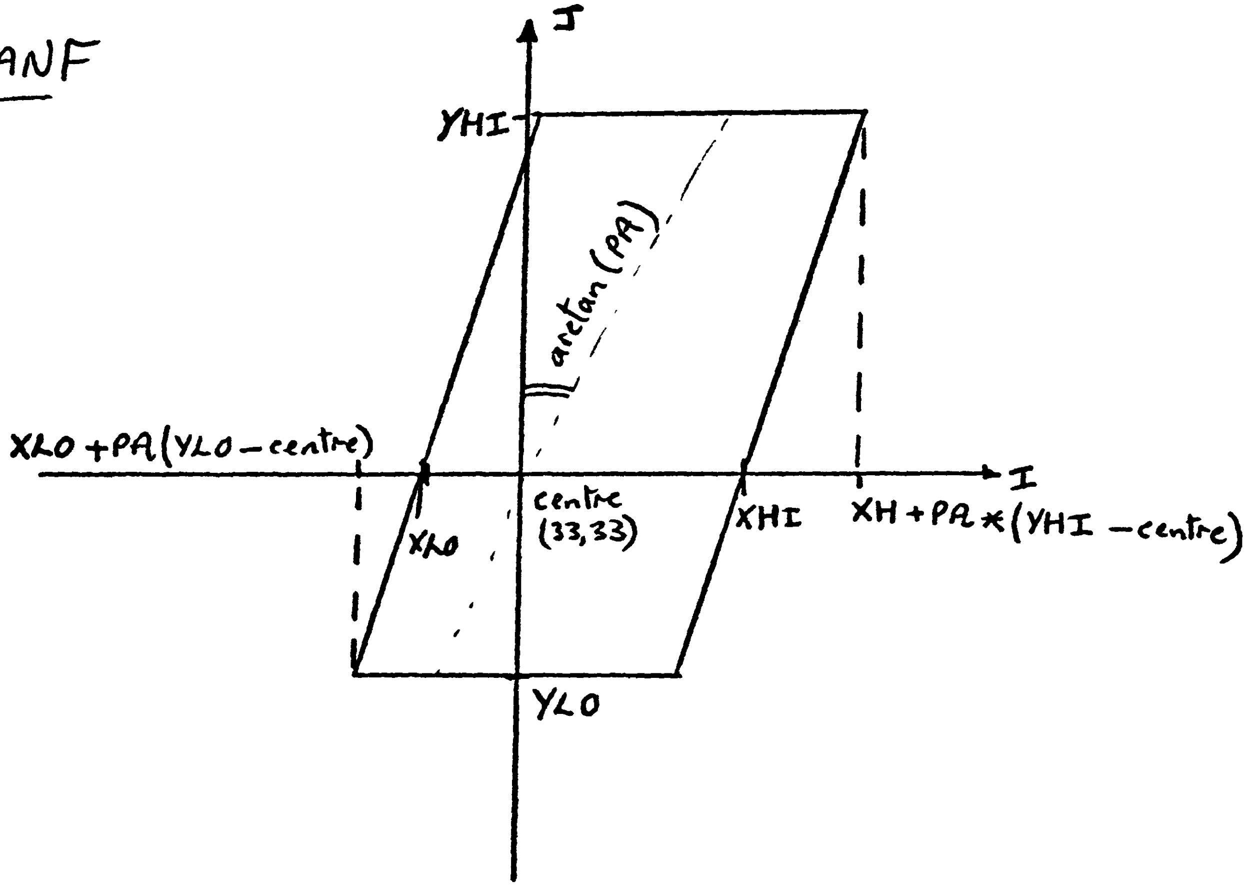
BEAM MAP	locations (starting block on disk) of dirty beam and map to be cleaned.
NUBEAM NUMAP	locations to be used for storing clean beam and map. (Each 32 blocks long) For a polarized map both real and imaginary maps are read and written on disk. The location of the real map should be given. The imaginary map is understood to be 32 blocks later.

Card 5

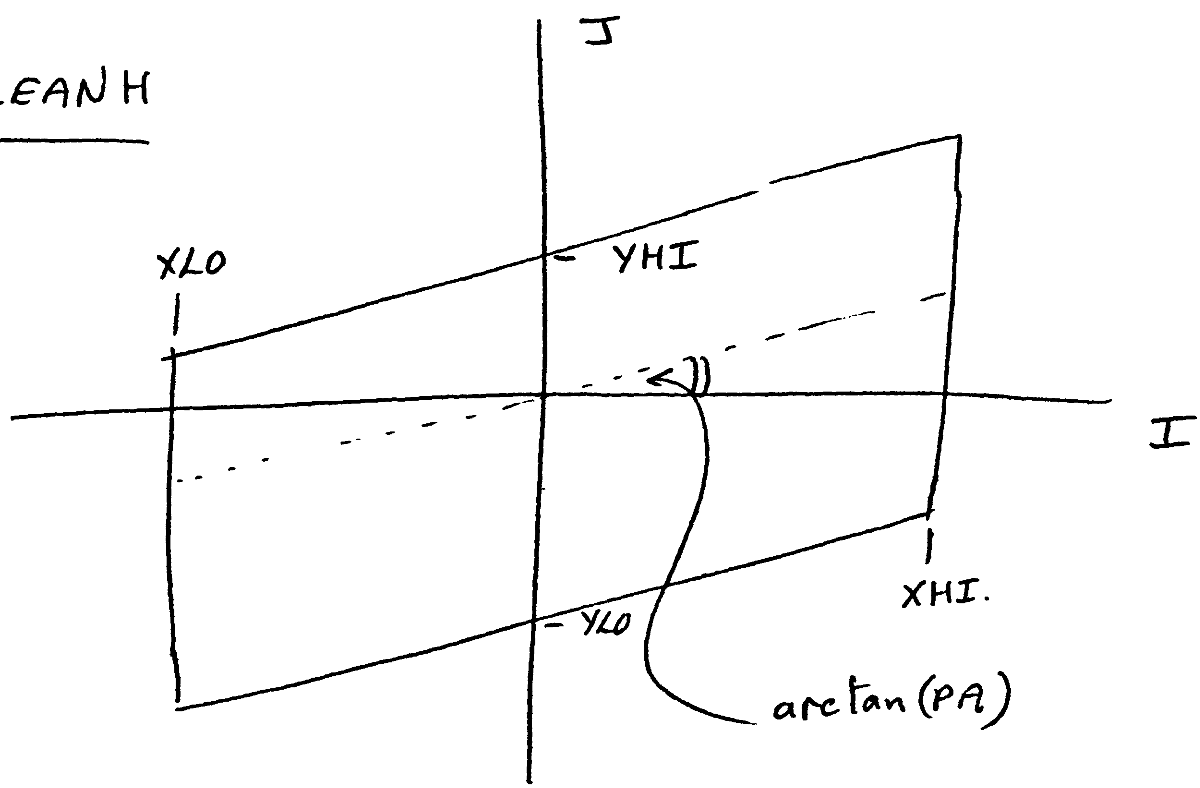
Blank. This card is used by TAU.

Figure 3. CLEANING AREA

a) CLEANF



b) CLEANH



APPENDIX 1

THE H-LINE FORTH LANGUAGE

The Forth language is a dictionary search for character sequences or "words" broken by spaces. Certain character sequences define basic machine operations and the Language is built up by defining other character sequences in terms of already defined words. The language is self-compiling and definitions may be mixed together with executable words.

The Forth used in this program is a particular application of the language to the IBM 360/50. In this application the Forth is interfaced with Fortran and is able to set parameters in the Fortran common area "/USER/" and to make subroutine calls from the Fortran main program. The parameters in common may all be directly set (for example: "45 THETA =") but are mostly combined into more mnemonic definitions. (For example, "RR" sets up the number and a list of the correlator positions for the RR data.) The subroutine calls are made by other words; the verbs of the language which cause actions to take place.

The program is run off-line by submitting a card deck of directives which starts:

```
      E LOAD
    H-LINE  LOAD
          |
          |-- Displays
          |-- MAPS LOAD ---------- mapping
          |-- TAPES LOAD --------- tape 1/0
          |
```

```
            ├── EDITS LOAD ────────── editing
            ├── BASES LOAD ────────── baseline fitting
            └── OPTIONAL LOAD ─────── extras
```

MAPS, TAPES, EDITS, BASES overlay each other. EDITS is the default.

The current definitions may be inspected at anytime by:

E LOAD

CARDS LOAD

20 40 SHOW

Definitions may be added or changed (caution) by the user

": WORD OTHER WORDS;"

defines "WORD" AS "OTHER WORDS."

Language Conventions used in H-LINE

Free-format on card-columns 1 through 64 with the following conventions:

a) Spaces must separate numbers

b) No spaces within a single word

c) All double words connected with a hyphen (-ve sign punch)

d) Statements may be separated onto several cards

Forth Diagnostics

OK	action completed
?	word not recognized skip to next command
EMPTY	parameter expected and not given

APPENDIX 2

H-LINE FORTH DICTIONARY

a) Common "/USER/"	USE
THETA	rotation of array
F1	starting frequency
NF	number of frequencies
DF	interval between frequencies
RES (4)	Δ_x, Δ_y, Δ_u, Δ_v
NS	count of items in List
LIST (38)	Lists of scans and frequencies
LOW	Lower Limit
HIGH	Upper Limit
INT	number of records integrated
C	correlator posiiton
SCAN	scan #
RECORD	record #
NU	frequency
P	polarization
SOURCE	used in disk index
ARRAY	statting block # for map
K	used for real, imag, etc.
ARG (3)	multiply used for parameter passing
Y-DIM	used in retangular FT
K1	entry point in "INTER"
NR	# of repitions of "SCANS" loop
ENTRY	entry point in subroutines

b) Words used to set parameters

INCLUDE	create a source index on disk
GAINS	set scaling factor for data transfer
DEFINE	a define file statement
AVAIL	a define file statement
INNER-AREA	inner area for readmap
OUTER-AREA	outer area for readmap
A-RECORD	sets record number
DATA	data records chosen by SCANS
HEAD	head record only chosen by SCANS
EVERY	every record chosen by SCANS
REAL	real part of data
IMAG	imaginary part of data
AMPL	amplitude of data
PHASE	phase of data
COMPLEX	real and imaginary of data
VECTOR	amplitude and phase of data
RX-A	frequency channels 1-48
RX-B	frequency channels 49-96
RX-C	frequency channels 97-144
RX-D	frequency channels 145-192
RX-AB	frequency channels 1-96
RX-CD	frequency channels 97-192
RX-ABCD	frequency channels 1-192
RESOLUTION	sets Δ_x, Δ_y, Δ_u, Δ_v for mapping
BASELINES	printout baselines

LOOK	printout given frequency list
HEADER	printout header
FREQ	list frequencies
SCAN-LIST	list scans
DUMP	prints whole record
PASS-BAND	store spectral pass-band or disk
WATER	printout water vapour rxs
DISPLAY	display frequency list
SPECTRA	averaged amplitude and phase
SPECTRAL-DENSITY	a matrix plot out
MULTIPLY	multiply and add phase to data
MULT-AMP	multiply data
ADD-PHASE	modify phase of data
PHASE-COR	apply atmospheric phase corrections
BASE-COR	apply baseline ocrrections
POSITION-COR	apply ΔRA, ΔDEC corrections
GRADING-FUNCTION	specify half widths of
PRIMARY BEAM	specify half widths of
INTO	dummie word
VARIABLES	no. of variables in baseline calibration
BANDWIDTH	bandwidth of data to be used in above

c) Words used to cause actions

FLUSH	moves included sources to disk
MAP	a printed contour map
READ-MAP	a numerical printout of map
ERASE-SOURCE	erase disk source indices
ERASE-ARRAY	zero array on disk
SCANS	a loop through scan #s
A-SCAN	a single scan
TELESCOPE-TAPE	load 7 track tape onto disk
OBSERVER-TAPE	load 9 track observer tape
DUAL-FREQ	load dual frequency data onto disk
LOAD-NINE	load 9 track tape onto disk
WRITE-NINE	write a 9 track tape
COPY-NINE	copy a 9 track tape
TAPE-INDEX	index a 9 track tape
ADD-MAP	add arrays together
IMAGE	make a gridded u-v plane
GENERATE	make a specified map
CAL	use source as calibrator
GET	move into core
PUT	store onto disk
BEAM	add 1000 j into u-v plane
GRADE	grade u-v plane
PLOT	2 digit plot of integer map
PLANE	" " " " floating-point u-v plane
SQUEEZE	scale floating-point to integers
FLOAT	FLOAT integer map
TRANSFORM	an FFT routine using disk access

APPENDIX 3

To Add A Fortran Subroutine To H-Line

1) Code the subroutine accepting all input through common /USER/. Compile it as a member of "MOORE·MODS".

2) Recompile subroutine "INTER" with a call to your subroutine.

3) Add your subroutine to an overlay statement with similar subroutines in the "LINK-EDIT" deck.

4) Add a Fortran definition of your subroutine to the "FORTH" deck, and/or a suitably memonic definition in the "FORTH" definitions.

APPENDIX 4

H-LINE RECORD FORMAT

Word	Content (units)	Unused fields
1	integer 1 (identifies header)	
2	scan no. (0 - 32767)	
3		360: no. of data records
4	Greenwich sidereal days since 2424832	
5-6	LST 116: BCD (.1^s) 360: integer (.1^s)	
7-10	source name (8 ANSI letters, even parity)	
11-12	RA (rev) 116: 30-place fraction 360: REAL*4	
13-14	Dec (rev) "	
15	return code: 0 1 2 3 4 5 6	
16		start LST (10^s)
17		stop LST (10^s)
18		mode
19		system
20		observer (2 ANSI letters)
21		operator "
22		
23	no. of phase corrections applied	
24		
25	" clock "	
26	" position "	
27		85-1 Dec (1/4 rev, 15-place fraction)
28		HA
29-30		85-2 "
31-32		85-3 "
33-34	line-of-sight standard-of-rest velocity (c) 116: 30-place fraction 360: REAL*4	
35-36	synthesizer frequency (Hz, integer)	
37-38	Z_{12} baseline (1/1024 cycle) 116: integer 360: REAL*4	
39-40	Z_{13} "	
41-42	Z_{23}	
43-44	Z_{14}	
45-46	X_{12}	
47-48	X_{13}	
49-50	X_{23}	
51-52	X_{14}	
53-54	Y_{12}	
55-56	Y_{13}	
57-58	Y_{23}	
59-60	Y_{14}	
61-62	k_{12}	
63-64	k_{13}	
65-66	k_{23}	
67-68	k_{14}	

69	source scaling divisor
70-100	
101-512	

Word	Content (units)	Unused fields
1	integer 2 identifies data	
2	scan no. (must match header)	
3		
4		sidereal day
5-6	LST 116: BCD (.1^s) 360: integer (.1^s)	
7	mode: 1 S-band 2 X-band 4 mixed	
8	receiver baselines (see word format)	
9	no. of distinct baselines	
10	correlator gains (see word formats)	
11	if selection "	
12	computer control "	
13	lock "	
14	range "	
15	faults "	
16	85-1 delay (us, 10-place fraction)	
17	85-2 "	
18	85-3 "	
19-20	local oscillator (Hz, integer)	
21-22	integration time (3.2 us, integer)	
23-24	signal power, channels 1-48	
25-26	" 49-92	
27-28	" 93-144	
29-30	" 145-192	
31-38	signal + noise power "	
	correlator words 1572-1587	
39-42	bandwidth code, channels 1-48, etc.	
43	mode	
44	noise tube state	
45	gain modulator code	
46	switches *	
47	switching rate code	
48	clipper test	
49	digital test	
50		
51-54	local oscillator, channels 1-48, BCD (10 Hz)	

55	1-2 continuum, real		
56		imaginary	
57		baseline #1	116: offset
58		RMS: 0 or negative - ignore data	
59-62	1-3 "	(#2)	
63-66	2-3 "	(#3)	
67-70	1-4 "	(#4)	

<u>360</u> <u>format</u>

71-74	first receiver, real
	imaginary
	baseline
	RMS
75-78	second "
79-82	third "
83-86	fourth
87	
88	
	A/D channels 24-63
89	85-1 R if monitor
90	" L "
91-92	85-2 "
93-94	85-3 "
95-96	42' "
97-104	" sync detector
105	microwave I phase
106	" Q "
107	
108	temperature (°C)
109	dew point (°C)
110	barometric pressure (mm)
111	cable pressure (?)
112	85-1 water vapor (?/K)
113	85-2 "
114	85-3 "
115	
116-118	" box temperature (?)
119-125	
126	42' water vapor (?/K)
127-128	
129-130	channel 1, real
	imaginary
131-132	channel 2 "
133-512	channels 3-192

<u>116</u> <u>format</u>

71-72	first, real
73-74	imaginary
75-76	$(2^{-7} * \text{RMS})^2$
77-82	second "
83-88	third "
89-94	fourth. "
95-134	A/D channels 24-63
135-150	correlator words 1588-1603
151-256	
257-258	channel 1, real
259-260	imaginary
261-264	channel 2 "
265-1024	channels 3-192

Word formats

Word	Use	Bit	Use	
8	receiver baselines	1-4	RX-A	
		5-8	B	
		9-12	C	
		13-16	D	
10	correlator gains	1-2	1-2 RR	00 multiplied by 1.
		3-4	LL	
		5-8	1-3	10 multiplied by .1
		9-12	1-4	11 " .01
		13-16	2-3	
11	if selected	1-3	85-1	001 XR-SL
		4-6	2	010 XR-XL
		7-9	3	100 SR-SL
		10	water vapor cal on	
12	computer control	1	85-1	
		2	2	
		3	3	
		4	85-1 receiver	
		5	2 "	
		6	3	
		7	85-1 delay	
		8	2 "	
		9	3	
		10	1-2 correlator gain	
		11	1-3 "	
		12	polarization and focus motors	
		13	1-4 correlator gain	
		14	2-3 "	
		15	2-4	
		16	3-4	
13	lock	10	42'	out of lock
		11	master lo	"
		12	85-1 lo	"
		13	2 "	
		14	3	
		15		
		16	42' lo	"

14	range	1	85-1 lo phase unlocked
		2	2 "
		3	3 "
		4	42' "
		5	
		6	master lo out of range
		7	if level "
		8	
		9	85-1 R if level "
		10	L "
		11-12	2 "
		13-14	3 "
		15-16	42' "
15	faults	1	85-1 box temperature out of range
		2	2 "
		3	3 "
		4	
		5	85-1 SR receiver fault
		6	2 SL "
		7	XR "
		8	XL "
		9-12	85-2 "
		13-16	3 "

APPENDIX 5

Disk Layout

Block	Content
0	Empty
1	Empty
2 – 99	Forth program
100	Name index
101	Calibrations
102 – 109	Scan index
110 – 159	Source index
160 –	Scans
⋮	Maps

4095 on FREQS

23379 on GHC disk packs

The next available location on disk for data is printed out by: " FIND AVAIL . " and must not exceed the length of the data set. Maps should begin after this location.

APPENDIX 6

Examples

```
//FRED   JOB (203,D,1,1,1),GHC4,MSGLEVEL=1,CLASS=D
// EXEC PGM=HLINE,COND=EVEN,ACCT=G
//STEPLIB DD DSNAME=MOORE.JOBS,DISP=SHR
//FT05F001  DD  DSN=BACK-UP,DISP=OLD,UNIT=TAPE,VOL=SER=2392
//FT04F001  DD  DSN=BACK-UP,DISP=NEW,UNIT=TAPE,VOL=SER=1675,LABEL=2,
//      DCB=(RECFM=VSB,LRECL=1028,BLKSIZE=2060)
//DISK DD DSNAME=MOORE.GHC4,DISP=OLD
//FT06F001 DD SYSOUT=A
//READER DD *
2 LOAD
H-LINE LOAD
TAPES LOAD
ERASE-SOURCE   160 DEFINE AVAIL   0 DEFINE EVERYTHING
110 GAINS 2222 INCLUDE 3C48
111 GAINS 2222 INCLUDE 3C273
112 GAINS 2222 INCLUDE 3C286
113 GAINS 2222 INCLUDE 3C279
114 GAINS 2222 INCLUDE 3C309REF
115 GAINS 2222 INCLUDE 3C309.1
116 GAINS 0000 INCLUDE FRED
117 GAINS 0000 INCLUDE 3C273.1
FLUSH
10 NO=
12143 12372 OBSERVER-TAPE
FIND AVAIL .
EDITS LOAD
: PB PASS-BAND ;  : EB RX-ABCD EDIT-BAND ;
: AS A-SCAN ;  : P PROFS ;
: UP RX-AB ADD-PHASE ;  : LP  RX-CD  ADD-PHASE ;
: SS SCAN-LIST ;
: S SCANS ;
95 UP 12143 AS
125 UP 12144 AS
-120 UP 12157 12160 S
-15 UP 12162 12165 S
58 UP 12166 12168 S
98 UP 12169 12171 S
-32 LP 12149 AS
120 LP 12150 12152 S
-65 LP 12153 12154 S
142 LP 12155 AS
PB 12328 AS EB 12326 12328 S
PB 12331 AS EB 12329 12331 S
PB 12334 AS EB 12332 12334 S
PB 12337 AS EB 12335 12337 S
PB 12340 AS EB 12338 12340 S
PB 12343 AS EB 12341 12343 S
PB 12346 AS EB 12344 12346 S
PB 12349 AS EB 12347 12349 S
SS 12149,12155,12152, INTO 1000 STACK
SS 12215,12218,12221,12224, INTO 1001 STACK
SS 12227,12230,12233,12236, INTO 1002 STACK
SS 12239,12242,12245,12248, INTO 1003 STACK
SS 12254,12257,12260,12263, INTO 1004 STACK
SS 12266,12269, INTO 1005 STACK
SS 12326,12329,12332,12335,12338, INTO 1006 STACK
SS  12341,12344,12347,12350, INTO 1007 STACK
SS 12353,12356,12359, INTO 1008 STACK
SS 12365,12368,12371, INTO 1009 STACK
: GO 1000 1010 S ;
FOR EVERYTHING
BROAD-BAND VECTOR LOCK GO
RX-AB PROFS GO
```

```
PAGE
RX-CD PROFS GO
TAPES LOAD
FOR EVERYTHING
700 12373 WRITE-NINE
DISCARD
GOODBY
```

```
//FRED  JOB (234,P,6,4,3),HLINE,MSGLEVEL=1,CLASS=C
// EXEC PGM=HLINE,COND=EVEN,ACCT=G
//STEPLIB DD DSNAME=MOORE.JOBS,DISP=SHR
//DISK DD DSNAME=MOORE.DISK,DISP=OLD,VOL=PRIVATE
//FTO6F001 DD SYSOUT=A
//READER DD *
2 LOAD
H-LINE LOAD
EDITS LOAD
FOR FRED
RX-AB -134 ADD-PHASE
7573 A-SCAN
RX-CD -23 ADD-PHASE
7573 A-SCAN
RX-AB -62     ADD-PHASE
8091 A-SCAN
RX-CD  65     ADD-PHASE
8091 A-SCAN
RX-AB -80     ADD-PHASE
8092 A-SCAN
RX-CD  43     ADD-PHASE
8092 A-SCAN
RX-AB -72     ADD-PHASE
8294 A-SCAN
RX-CD  55     ADD-PHASE
8294 A-SCAN
SCAN-LIST             3216,3367,4535,4666,5257,  5391,
INTO 1000 STACK
SCAN-LIST             6641,6644,7374,7573,8092,8294,
INTO 1001 STACK
SCAN-LIST 1000,1001, INTO 1003 STACK
: GO 1000 9000 SCANS ;
PAGE
RX-AB PROFS GO
PAGE
RX-CD PROFS GO
FOR 30454.3
: GO 1002 A-SCAN ;
SCAN-LIST 5215,5256,6640,7378, INTO 1002 STACK
PAGE
RX-AB PROFS GO
PAGE
RX-CD PROFS GO
DISCARD  GOODBY
```

```
//HLINE JOB (234,P,6,4,4),625KHZ,MSGLEVEL=1,CLASS=C
// EXEC PGM=HLINE,COND=EVEN,ACCT=G
//STEPLIB DD DSNAME=MCCRE.JOBS,DISP=SHR
//FT06F001 DD SYSOUT=A
//DISK DD DSN=MCCRE.GHC4,DISP=OLD
//READER DD *
E LOAD
H-LINE LOAD
EDITS LOAD
 :GO 1400 6511 SCANS ;
: RGAINS BROAD-BAND RECIPROCAL-GAINS ,
FOR RA0330 6000 STORE-FLUX GO
FOR 3C345 7000 STORE-FLUX GO
FOR 3C395 3400 STORE-FLUX GO
FOR 3C286 15200 STORE-FLUX GO
FOR 3C380 14400 STORE-FLUX GO
FOR EVERYTHING
AVERAGE
BROAD-BAND VECTOR LOCK GO
RGAINS GO
FIND AVAIL 160 -
DISCARD
GOODBY
```

```
//CYGAD JOB (234,P,6,13,4),DISK,MSGLEVEL=1,CLASS=F
// EXEC PGM=HLINE,COND=EVEN,ACCT=G
//STEPLIB DD DSNAME=MOORE.JOBS,DISP=SHR
//DISK DD DSNAME=MOORE.DISK,DISP=OLD
//FT06F001 DD SYSOUT=A
//READER DD *
E LOAD
H-LINE LOAD
MAPS  LOAD
30  50  RESOLUTION
RES 2( ?
C 10000 SCALE
:  L  1  LINE-FREC  1300  8500  IMAGE
 ARRAY @ .  AMPL  PLANE  ARRAY @ 128+  ARRAY=  ;
70 20 UNE-FREQ
10000 ARRAY= 1300 8500 IMAGE CYGA-D
AMPL PLANE
PHASE PLANE
10128 ARRAY=
 5 L CYGA-D
 7 L CYGA-D
 9 L CYGA-D
DISCARD
GOODBY
```

```
//CYGAD JOB (234,P,6,13,4),MAPS,MSGLEVEL=1,CLASS=L
//LINK EXEC PGM=IEWL,PARM='LET,MAP,LIST',ACCT=L
//SYSLIB DD DSNAME=SYS1.FORTLIB,DISP=SHR
// DD DSNAME=SYS1.GPSLMOD,DISP=SHR
//SYSUT1 DD UNIT=2314,SPACE=(CYL,(2,2))
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=3509)
//MODS DD DSNAME=MOORE.MODS,DISP=SHR
//JOBS DD DSNAME=MOORE.JOBS,DISP=SHR
//SYSLMOD DD DSNAME=&&TEMP(MAIN),DISP=(NEW,PASS),
// UNIT=DISK,SPACE=(CYL,(5,2,1))
//SYSLIN DD *
 INCLUDE JOBS(FORTHF)
 ENTRY BEGIN
 INCLUDE MODS(INTER,DAIO)
 INCLUDE MODS(EXPI,MOVE)
 INCLUDE MODS(PIGS)
// EXEC PGM=*.LINK.SYSLMOD,COND=EVEN,ACCT=G
//STEPLIB DD DSNAME=MOORE.JOBS,DISP=SHR
//DISK DD DSNAME=MOORE.DISK,DISP=OLD
//FT06F001 DD SYSOUT=A
//READER DD *
E LOAD H-LINE LOAD INVERT LOAD UV
10000 GET BEAM TRANSFORM 10000 100 SQUEEZE PLOT
16032 PUT
SECOND PLOT
16000 PUT
UV
10128 GET
TRANSFORM 10000 10000 SQUEEZE PLOT
16064 PUT SECOND PLOT
16096 PUT
UV
10256 GET TRANSFORM 10000 10000 SQUEEZE PLOT
16128 PUT SECOND PLOT
16160 PUT
DISCARD
GOODBY
```

```
//CLEAN JOB (234,P,6,13,4),CYGA-D,MSGLEVEL=1,CLASS=F
//  EXEC  PGM=CLEANH
//STEPLIB  DD  DSN=MOORE.JOBS,DISP=SHR .
//FT10F001  DD  DSNAME=MOORE.DISK,DISP=SHR,VOL=PRIVATE
//FT06F001  DD  SYSOUT=A
//FT07F001  DD  SYSOUT=B
//FT05F001  DD  *
 CYGADREF 19 57 44.50 +40 35 46.7 F 00005.0    580.0000E+01   579.9000E+01
 16 52 27 40                                15.0     15.0      0.4
 250 0.010 F
 16000 16032 19000 19032
     70    20  78.125    -78.7
 CYGAD05  19 57 44.50 +40 35 46.7 F 00005.0    580.0000E+01   473.5000E+01
 16 52 27 40                                15.0     15.0      0.4
 250 0.010 F
 16000 16064 19000 19064
      5     1  78.125    -78.7
 CYGAD06  19 57 44.50 +40 35 46.7 F 00005.0    580.0000E+01   495.2000E+01
 16 52 27 40                                15.0     15.0      0.4
 250 0.010 F
 16000 16096 19000 19096
      6     1  78.125    -78.7
```

```
//CYGAA      JOB (234,C,1,1,1),DISK,MSGLEVEL=1,CLASS=E
// EXEC PGM=HLINE,COND=EVEN,ACCT=G
//STEPLIB DD DSNAME=MOORE.JOBS,DISP=SHR
//FT06F001 DD SYSOUT=A
//DISK DD DSNAME=MOORE.DISK,DISP=OLD
//READER DD *
E LOAD
H-LINE LOAD
HERE .
FIND AVAIL .
MAPS LOAD
50  50  RESOLUTION
G 10000  SCALE
7000  ARRAY=
6  20  ONE-FREQ
1300  8500  IMAGE  CYGA-A
AMPL  PLANE
PHASE  PLANE
:  L  5  LINE-FREQ  1300  8500  IMAGE
 ARRAY @ .  AMPL  PLANE  ARRAY @ 128+  ARRAY=  ;
7128  ARRAY=
20  L  CYGA-A
30  L  CYGA-A
40  L  CYGA-A
50  L  CYGA-A
60  L  CYGA-A
70  L  CYGA-A
80  L  CYGA-A
```

```
//CYGAA        JOB (203,D,1,1,1),MAPS,MSGLEVEL=1,CLASS=L
//LINK EXEC PGM=IEWL,PARM='LET,MAP,LIST',ACCT=L
//SYSLIB DD DSNAME=SYS1.FORTLIB,DISP=SHR
// DD DSNAME=SYS1.GPSLMOD,DISP=SHR
//SYSUT1 DD UNIT=2314,SPACE=(CYL,(2,2))
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=3509)
//MODS DD DSNAME=MOORE.MODS,DISP=SHR
//JOBS DD DSNAME=MOORE.JOBS,DISP=SHR
//SYSLMOD DD DSNAME=&&TEMP(MAIN),DISP=(NEW,PASS),
// UNIT=DISK,SPACE=(CYL,(5,2,1))
//SYSLIN DD *
 INCLUDE JOBS(FORTHF)
 ENTRY BEGIN
 INCLUDE MODS(INTER,DAIC)
 INCLUDE MODS(EXPI,MOVE)
 INCLUDE MODS(PIGS)
// EXEC PGM=*.LINK.SYSLMOD,COND=EVEN,ACCT=G
//STEPLIB DD DSNAME=MOORE.JOBS,DISP=SHR
//DISK DD DSNAME=MOORE.GHC1,DISP=OLD
//FT06F001 DD SYSOUT=A
//READER DD *
E LOAD  H-LINE LOAD
INVERT LOAD
UV
7000  GET
BEAM
TRANSFORM  10000  100  SQUEEZE
PLOT
9032  PUT
SECOND  PLOT
9000  PUT
UV
7128  GET
TRANSFORM  10000  10000  SQUEEZE
PLOT
9064  PUT
SECOND  PLOT
9096  PUT
UV
7256  GET
TRANSFORM  10000  10000  SQUEEZE
PLOT
9128  PUT
SECOND  PLOT
9160  PUT
UV
7384  GET
TRANSFORM  10000  10000  SQUEEZE
PLOT
9192  PUT
SECOND  PLOT
9224  PUT
UV
7512  GET
TRANSFORM  10000  10000  SQUEEZE
PLOT
9256  PUT
SECOND  PLOT
9288  PUT
UV
```

```
7640  GET
TRANSFORM  10000  10000  SQUEEZE
PLOT
9320  PUT
SECOND  PLOT
9352  PUT
GV
7760  GET
TRANSFORM  10000  10000  SQUEEZE
PLOT
9384  PUT
SECOND  PLOT
9416  PUT
GV
7696  GET
TRANSFORM  10000  10000  SQUEEZE
PLOT
9448  PUT
SECOND  PLOT
9480  PUT
DISCARD GOODBY
```

```
//GRADED       JOB (203,D,1,1,1),MAPS,MSGLEVEL=1,CLASS=L
//LINK EXEC PGM=IEWL,PARM='LET,MAP,LIST',ACCT=L
//SYSLIB DD DSNAME=SYS1.FORTLIB,DISP=SHR
// DD DSNAME=SYS1.GPSLMOD,DISP=SHR
//SYSUT1 DD UNIT=2314,SPACE=(CYL,(2,2))
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=3509)
//MODS DD DSNAME=MOORE.MODS,DISP=SHR
//JOBS DD DSNAME=MOORE.JOBS,DISP=SHR
//SYSLMOD DD DSNAME=&&TEMP(MAIN),DISP=(NEW,PASS),
// UNIT=DISK,SPACE=(CYL,(3,2,1))
//SYSLIN DD *
 INCLUDE JOBS(FCRTHF)
 ENTRY BEGIN
 INCLUDE MODS(INTER,CAIO)
 INCLUDE MODS(EXPI,MOVE)
 INCLUDE MODS(PIGS)
// EXEC PGM=*.LINK.SYSLMOD,COND=EVEN,ACCT=G
//STEPLIB DD DSNAME=MOORE.JOBS,DISP=SHR
//DISK DD DSNAME=MOORE.DISK,DISP=OLD
//FT06F001 DD SYSOUT=A
//READER DD *
E LOAD  H-LINE LOAD
INVERT LOAD
150 150 RESOLUTION
UV 4000 GET
BEAM TRANSFORM 1000 100 SQUEEZE
MAP
7000 PUT
SECOND MAP
7032 PUT
UV 4000 GET
BEAM   60 GRADE
TRANSFORM  1000 100 SQUEEZE
MAP
7064 PUT
SECOND MAP
7096 PUT
UV 4000 GET
BEAM  90 GRADE
TRANSFORM  1000 100 SQUEEZE
MAP
7128 PUT
SECOND MAP
7160 PUT
DISCARD GOODBY
```

```
//POLN  JOB (203,D,1,1,1),3C244,MSGLEVEL=1,CLASS=F
// EXEC PGM=HLINE,COND=EVEN,ACCT=6
//STEPLIB DD DSNAME=MOORE.JOBS,DISP=SHR
//DISK DD DSNAME=MOORE.GHC2,DISP=SHR,VOL=PRIVATE
//FT06F001 DD SYSOUT=A
//FT03F001 DD UNIT=TAPE,VOL=SER=954,DSNAME=EDFM54,DISP=OLD
//READER DD *
E LOAD
H-LINE LOAD
TAPES LOAD
ERASE-SOURCE   160 DEFINE AVAIL   0 DEFINE EVERYTHING
110 GAINS 0000 INCLUDE 3C244.1
111 GAINS 0000 INCLUDE 3C27
FLUSH
550 30000 DUAL-FREQ
FIND AVAIL
MAPS LOAD
X-BAND
RL
VECTOR LOOK  550 600 SCANS
MAPS LOAD
30 30 RESOLUTION
S-BAND
RR
11000  ARRAY=
550 29103 IMAGE 3C244.1
0 100 SCALE
AMPL PLANE
PHASE PLANE
RL
11126  ARRAY=
550 29103 IMAGE 3C244.1
0 10 SCALE
AMPL PLANE
PHASE PLANE
LL
11700  ARRAY=
550 29103 IMAGE 3C244.1
0  100  SCALE
AMPL  PLANE
PHASE  PLANE
X-BAND
RR  SHORT
11256  ARRAY=
550 29103 IMAGE 3C244.1
0 100 SCALE
AMPL PLANE
PHASE PLANE
RL  SHORT
11384  ARRAY=
550 29103 IMAGE 3C244.1
0 10 SCALE
AMPL PLANE
PHASE PLANE
X-BAND
10  10  RESOLUTION
RR
11512  ARRAY=
550 29103 IMAGE 3C244.1
```

```
0  10  SCALE
AMPL  PLANE
PHASE  PLANE
RL
11640  ARRAY=
550 29163 IMAGE 3C244.1
0  10  SCALE
AMPL  PLANE
PHASE  PLANE
FIND AVAIL .
FLUSH GOODBY
```

```
//POLNMAPS    JOB (203,0,1,1,1),3C244,MSGLEVEL=1,CLASS=L
//LINK EXEC PGM=IEWL,PARM='LET,MAP,LIST',ACCT=L
//SYSLIB DD DSNAME=SYS1.FORTLIB,DISP=SHR
// DD DSNAME=SYS1.GPSLMOD,DISP=SHR
//SYSUT1 DD UNIT=2314,SPACE=(CYL,(2,2))
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=3509)
//MODS DD DSNAME=MOORE.MODS,DISP=SHR
//JOBS DD DSNAME=MOORE.JOBS,DISP=SHR
//SYSLMOD DD DSNAME=&&TEMP(MAIN),DISP=(NEW,PASS),
// UNIT=DISK,SPACE=(CYL,(5,2,1))
//SYSLIN DD *
 INCLUDE JOBS(FORTHF)
 ENTRY BEGIN
 INCLUDE MODS(INTER,DAIO)
 INCLUDE MODS(EXPI,MOVE)
 INCLUDE MODS(PIGS)
// EXEC PGM=*.LINK.SYSLMOD,COND=EVEN,ACCT=G
//FT06F001 DD SYSOUT=A
//DISK DD DSNAME=MOORE.GHC2,DISP=SHR
//READER DD *
E LOAD
H-LINE LOAD
INVERT LOAD
UV
11000  GET
BEAM
TRANSFORM  100  100  SQUEEZE
PLOT
12000  PUT
SECOND PLOT
12032  PUT
UV
11128  GET
TRANSFORM  10   10   SQUEEZE
PLOT
12064  PUT
SECOND PLOT
12096  PUT
UV
11256  GET
BEAM
TRANSFORM  100  100  SQUEEZE
PLOT
12128  PUT
SECOND PLOT
12160  PUT
UV
11384  GET
TRANSFORM  10   10   SQUEEZE
PLOT
12192  PUT
SECOND PLOT
12224  PUT
UV
11512  GET
BEAM
TRANSFORM  100  100  SQUEEZE
PLOT
12256  PUT
SECOND PLOT
```

```
12288  PUT
UV
11640  GET
TRANSFORM  1  1  SQUEEZE
PLOT
12320  PUT
SECOND PLOT
12352  PUT
UV
11706  GET
BEAM
TRANSFORM  100  100  SQUEEZE
PLOT
12384  PUT
SECOND PLOT
12416  PUT
DISCARD    GOODBY
```

APPENDIX 7

Utility and Other Programs

```
//GHC3 JOB (203,P,1,1,14),DUMP,MSGLEVEL=1,CLASS=D
// EXEC PGM=DUMP
//STEPLIB DD DISP=SHR,DSNAME=MOORE.JOBS
//FT06F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3458)
//FT01F001 DD DISP=SHR,DSNAME=MOORE.DISK
//FT02F001 DD UNIT=TAPE,DCB=(RECFM=FB,LRECL=1024,BLKSIZE=10240),
//    DISP=NEW,DSNAME=GHC1,VOLUME=SER=55[illegible],LABEL=3
//FT05F001 DD *
9-TRACK DUMP
 10000 23379
```

```
//HLINE6  JOB (173.C,6,7,14),WRITE,MSGLEVEL=1,CLASS=B
//STORE7  EXEC  PGM=STORE7
//STEPLIB  DD  DSN=MOORE.JOBS,DISP=SHR
//FT04F001 DD DSN=BACK-UP,DISP=MOD,UNIT=TAPE,
// VOL=(,RETAIN,SER=2491),
//     DCB=(RECFM=VSB,LRECL=2052,BLKSIZE=4108)
//FT06F001  DD  SYSOUT=A
//DDP116  DD  UNIT=TAPE7,DSN=INTERF.DDP116,DCB=(,DEN=1),DISP=OLD,
// LABEL=(2,BLP),VOL=SER=0500
//INDEX  EXEC  PGM=TAPEINDX
//SYSPRINT  DD  SYSOUT=A
//TAPE  DD  UNIT=TAPE,DISP=(OLD,KEEP),LABEL=(,BLP),
//     VOL=SER=2491
```

```
HASP-II JOB STATISTICS --     13 CARDS READ --       78 LINES PRINT
```

```
EOF
NO. HDRS=    134
NO. DATA=    897
NO. A/D=      0
NO. MSG=      0
NO. POINTING=      0

NO. 100 WORD DATA RECORDS=      0
NO. 1024 WORD DATA RECORDS=   897
```

```
VOL12491  0                                   NRAO

HDR1BACK-UP            2491   00010001        71270 000000000000

HDR2V04108020523OHLINE6  /STORE7      R

-----------------------------------------------------------------

            2875 BLOCKS

-----------------------------------------------------------------

EOF1BACK-UP            2491   00010001        72013 000000002875

EOF2V04108020523OHLINE6  /STORE7      R

-----------------------------------------------------------------

-----------------------------------------------------------------
```

```
//LIST76   JOB (173,0,6,7,14),WRITE,MSGLEVEL=1,CLASS=B
//   EXEC   PGM=LIST7
//STEPLIB   DD   DSN=MCCRF.JOBS,DISP=SHR
//FT06F001   DD   SYSOUT=A
//DDP116   DD   UNIT=TAPE7,DSN=INTERF.DDP116,DCB=(,DEN=1),DISP=OLD,
// LABEL=(2,BLP),VOL=SER=0500
```

```
//    EXEC  PGM=LIST9
//STEPLIB  DD  DSN=MOORE.JOBS,DISP=SHR
//FT03F001  DD  DSN=BACK-UP,DISP=OLD,UNIT=TAPE,
//     VOL=SER=2482
//FT06F001  DD  SYSOUT=A
//*  READS SCAN RANGE (I5,1X,I5)
//FT05F001  DD  *
 2735 2825
```

```
//PMAP  JOB  (203,P,6,8,4),MELVYN,MSGLEVEL=1,CLASS=C
// EXEC FORTGCLG
//FORT.SYSIN DD *
C    PROGRAM TO CALCULATE APPROPRIATE NUMBERS FROM THE PUNCHED OUTPUT OF THE
C    CLEAN PROGRAM., OR FROM DISK
C          THE PROGRAM EXPECTS ALL FOUR MAPS TO BE ENTERED:
C        1       S BAND INTENSITY
C        2       S BAND POLARIZATION
C        3       X BAND INTENSITY
C        4       X BAND POLARIZATION
C
C    POINTS WITH S & X BAND INTENSITIES > 5% ARE FLAGGED BY ** ON THE OUTPUT
C
       INTEGER*2 IPLOT(56)
        REAL*8 SNAME,RA,DEC
       COMPLEX SPCL,XPCL
       DIMENSION TNORM(4)
       INTEGER*2 SIXI(56,56,2)/6272*0/,SPXP(2,56,56,2)/12544*0/
       INTEGER*2 BLOCK(512),ROWS(128,4)
       LOGICAL STAR,PLOT,PFLAG,PUN,DISK,DIRTY,WRITE,PHASE
       DATA PI/3.141593/,CON1,CON2,CON3/.010310,1.098612,.0013690/
       EQUIVALENCE (BLOCK(1),ROWS(1,1))
CC
       DEFINE FILE 10 (23380,256,U,IO)
CC
C
        PRINT 1
1       FORMAT('      THE ORDER OF THE MAPS SHOULD BE:',
     *T40,'S BAND INTENSITY'//T40,'S BAND POLARIZATION'//
     *T40,'X BAND INTENSITY'//T40,'X BAND POLARIZATION'//
     *'0''**'' INDICATES BOTH S & X BAND INTENSITIES > IPCENT',//)
C
CCC    READ CONTROL CARD
C
    2 READ(5,1000,END=900) NMAPS,IPCENT,DISK,DIRTY,WRITE,PLOT
        PRINT 1100
        PRINT 1000,NMAPS,IPCENT,DISK,DIRTY,WRITE,PLOT
        IPCENT=IPCENT*100
       JPCENT=10*100
 1000   FORMAT(1X,I1,1X,I2,5(1X,L1))
 1100   FORMAT('1NMAPS,IPCENT,DISK,DIRTY,WRITE,PLOT',/)
C
CCCC       READ 'CLEAN' CARDS
        DO 20 I=1,NMAPS
       READ(5,2000) SNAME,RA,DEC,PFLAG,XYINT,BNORM,TMAX,MLO,MHI,NLO,NHI
      + ,BMAL,BMIN,PA,NIT,PERC,PUN,IBEAM,MAP,NUBEAM,NUMAP
       PRINT 1300
 1300   FORMAT(1H )
       WRITE(6,2000) SNAME,RA,DEC,PFLAG,XYINT,BNORM,TMAX,MLO,MHI,NLO,NHI
      + ,BMAL,BMIN,PA,NIT,PERC,PUN,IBEAM,MAP,NUBEAM,NUMAP
 2000 FORMAT(1X,A8,R12.2,S12.1,1X,L1,1X,F7.1,1X,2E15.4,/,
      + 4(1X,I2),20X,3(1X,F7.1),/,I4,1X,F5.3,1X,L1,/,4I6,/)
CC
        IF(NMAPS.EQ.4.AND.MOD(I,2) .EQ. 0 .AND. .NOT.PFLAG) GO TO 600
        TNORM(I)=TMAX/BNORM
CCC    READ DISK
C
       LOC=NUMAP+8
       IF(DIRTY) LOC=MAP+8
C
```

```
C
      L=1
      K=1
      IF(I .GT. 2) K=2
CCC     LOOP  ON  BLOCKS
    5 DO 9 J=1,14
      READ(10'LOC+J) BLOCK
      DO 9 J1=1,4
      N=4*(J-1)+J1
      IF(PFLAG) GO TO 7
      DO 6 M=1,56
      IPLOT(M)=ROWS(M+32,J1)/100
    6 SIXI(M,N,K)=ROWS(M+32,J1)
      GO TO 9
CCC       POLARIZED  MAP
    7 DO 8 M=1,60
    8 SPXP(L,M,N,K)=ROWS(M+32,J1)
    9 CONTINUE
      IF(.NOT. PFLAG) GO TO 10
      IF(L .EQ. 2) GO TO 10
      LOC=LOC+32
      L=2
      GO TO 5
   10 CONTINUE
CCC        PLOT  CUT  MAPS
      IF(.NOT. PLCT) GO TO 20
      PRINT 4000
 4000 FORMAT(40X, 'AMPLITUDE  NORMALIZED  TO  TMAX/100 ')
      PRINT 1400, I,(M,M=1,56)
      DO 12 NC=NLO,NHI
       N=NLO+NHI-NO
      DO 11 M=1,56
      IF(.NOT. PFLAG) IPLOT(M)=SIXI(M,N,K)/100
      IF(PFLAG)  IPLOT(M)=CABS(CMPLX((SPXP(1,M,N,
     +K)+0.),(SPXP(2,M,N,K)+0.)))/100.
   11 CONTINUE
   12 PRINT 1400, N,IPLOT
CCC     PLOT  POSITION  ANGLE  POLARIZED  MAP
      IF(.NOT. PFLAG) GO TO 20
      PRINT 5000
 5000 FORMAT(40X,' POSITION  ANGLE (0,180) IN TENS OF DEGREES')
      DO 13 NC=NLO,NHI
      N=NLO+NHI-NC
      DO 14 M=1,56
         TEMP=9./3.14159*ATAN2((SPXP(2,M,N,K)+0.),(SPXP(1,M,N,K)+0.01))
      IF(TEMP .LT. -0.5) TEMP=TEMP+18.
      IPLOT(M)=TEMP+0.5
   14 CONTINUE
   13 PRINT 1400, N,IPLOT
   20 CONTINUE
 1400  FORMAT(1X,I2,1X,56I2)
       IF (NMAPS .NE. 2) GO TO 140
C
COMPARISON OF TWO MAPS ONLY SET XBAND EQUAL TO SBAND
        TNORM(3)=TNORM(1)
        TNORM(4)=TNORM(2)
 140   CONTINUE
CC     NEED CHANGE  FOR  HORIZONTAL  CLEAN
CC   LOOP
     PRINT 3001
```

```
      DO 160 JJ=NLO,NHI
      DO 150 II=MLO,MHI
      J=JJ
      I=II+(J-33)*PA
       IF(I.LT.1) I=1
       IF(I.GT.64) I=64
       IF (NMAPS .NE. 2) GO TO 141
       SIXI(I,J,2)=SIXI(I,J,1)
       SPXP(1,I,J,2)=SPXP(1,I,J,1)
       SPXP(2,I,J,2)=SPXP(2,I,J,1)
 141   CONTINUE
       IF(SIXI(I,J,1).LE.JPCENT) GO TO 150
       IF(SIXI(I,J,2).LE.JPCENT) GO TO 150
       STAR=.TRUE.
       IF(SIXI(I,J,1).LT.IPCENT)STAR=.FALSE.
       IF(SIXI(I,J,2).LT.IPCENT)STAR=.FALSE.
       TEMP1=SPXP(1,I,J,1)
       TEMP2=SPXP(2,I,J,1)+0.01
       SPHS=ATAN2(TEMP2,TEMP1)
      IF(SPHS.LT.0.) SPHS=SPHS+2.*PI
       SPOL=CMPLX(TEMP1,TEMP2)*TNORM(2)
       TEMP1=SPXP(1,I,J,2)
       TEMP2=SPXP(2,I,J,2)+0.01
       XPHS=ATAN2(TEMP2,TEMP1)
      IF(XPHS.LT.0.)  XPHS=XPHS+2.*PI
       XPOL=CMPLX(TEMP1,TEMP2)*TNORM(4)
       SINT=SIXI(I,J,1)*TNORM(1)
       XINT=SIXI(I,J,2)*TNORM(3)
       ALPHA=XINT/SINT
       ALPHA=ALOG(ABS(ALPHA))/CON2
       SPERC=CABS(SPOL)/SINT
       XPERC=CABS(XPOL)/XINT
       RPOLN=CABS(XPOL)/CABS(SPOL)
       DPOL=SPERC/XPERC
       SPA=57.3*SPHS/2
       XPA=57.3*XPHS/2
      RPA=SPA-XPA
       RPA1=RPA+180.
       RPA2=RPA-180.
       RM=RPA/57.3/CON1
       RM1=RPA1/57.3/CON1
       RM2=RPA2/57.3/CON1
       PINT=XPA+57.3*RPA*CON3
       PINT1=XPA+57.3*RPA1*CON3
       PINT2=XPA+57.3*RPA2*CON3
      PRINT 3000,J,I,ALPHA,SPERC,XPERC,RPOLN,DPOL,SPA,XPA,RPA,RPA1,RPA2,
     *          RM,RM1,RM2,PINT,PINT1,PINT2
      IF(STAR) PRINT 3003
150    CONTINUE
      PRINT 3002
160    CONTINUE
3000   FORMAT(1X,2I4,F8.2,4F9.3,4F7.0,1X,7F7.0)
3001   FORMAT('-  J    I     ALPHA     SPERC     XPERC     RPOLN      DPOL
     *SPA     XPA     RPA  RPA+180 RPA-180    RM  RM+180 RM-180',
     *'  PINT   PINT+  PINT-'/)
3002   FORMAT(1X)
3003   FORMAT('+***')
      GO TO 2
C
 600   PRINT 1600
```

```
 1600    FORMAT(///,' MAPS OUT OF ORDER',///)
       RETURN
 900   STOP
      END
//GO.FT05F001  DD  *
 2 20 T F F T
 3C6.1     00 13 34.30 +79 00 11.0 F 00001.3    382.6000E+01   711.0000E+00
 27 39 20 50                               3.0      3.0     -0.5
 25C 0.010 F
 20288 20256 2C788 2C756

 3C6.1     00 13 34.30 +79 00 11.0 T 00001.3    382.6000E+02   366.0000E+00
 27 39 20 50                               3.0      3.0     -0.5
 25C 0.010 F
 20288 20320 2C788 2C820

 4 20 T F F T
 3C6.1     00 13 34.30 +79 00 11.0 F 00003.0    381.7000E+01   361.2000E+01
 29 37 26 39                              10.0     10.0     -0.5
 150 0.010 F
 20032 20000 20532 20500

 3C6.1     00 13 34.30 +79 00 11.0 T 00003.0    381.7000E+02   765.0000E+00
 29 37 26 39                              10.0     10.0     -0.5
 150 0.010 F
 20032 20C64 20532 20564

 3C6.1     00 13 34.30 +79 00 11.0 F 00003.0    148.4000E+01   392.0000E+00
 29 37 26 39                              10.0     10.0     -0.5
 150 0.010 F
 20160 20128 20660 20628

 3C6.1     00 13 34.30 +79 00 11.0 T 00003.0    148.4000E+02    18.3000E+01
 29 37 26 39                              10.0     10.0     -0.5
 25C 0.010 F
 20160 20192 2C660 20692

//FT10F001  DD  DSNAME=MOORE.GHC2,DISP=SHR,VOL=PRIVATE
```

```
//PPLOT    JOB  (205,F,6,8,4),WRIGHT,MSGLEVEL=1,CLASS=E
// EXEC FORTGCLG,ERRCR=E
//FORT.SYSIN DD *
        REAL*8 SNAME,RA,DEC
        COMPLEX PCL(64,64),TPOL(64,64)
       LOGICAL FLN,PFLAG
       INTEGER*2 ARRAY(512,32),IB(128,128)
        REAL*4 INT(64,64),PINT(64,64),CONT1(20),TMP(64,64)
        INTEGER*2  ZERC/0/
       LOGICAL DISK
       LOGICAL FUN,PFLAG
        LOGICAL CFLAG
        COMMON RA,DEC,XH,XL,YH,YL,XYINT,DRA,DDEC,ANGROT
       EQUIVALENCE (ARRAY(1,1),IB(1,1))
       EQUIVALENCE (TPCL(1,1),TMP(1,1),ARRAY(1,1))
       DEFINE FILE 10 (23360,256,U,IO)
        C1=180./3.14159
C       READ THE CCNTRCL CARD.  1=POLOT OF THE INTENSITY ONLY.  2=PLOT OF THE
C   POLARIZED INTENSITY.  3=POLOT OF TOTAL INTENSITY, POLARIZED INTENSITY AND
C   A POLARIZED LINE PLOT.  4=POLARIZED LINE PLOT ON TOTAL INTENSITY.
C   5=CALTEC LINE MAPS
C
   10 READ(5,1000,END=900) ITYPE,ANGROT,DISK
       PRINT 1001, ITYPE,ANGROT,DISK
 1000   FORMAT(I1,F10.1,1X,L1)
 1001   FORMAT(1X,I1,F10.1,1X,L1)
        GO TO (100,200,100,100,30),ITYPE
   30 READ(5,35) SNAME,NCH,OVMAX,XYINT,TAPER,SCALE,MLO,MHI,NLO,NHI
   35 FORMAT(2X,A8,1X,I2,1X,F6.1,3(1X,F4.1),4(1X,I3))
C   READ THE CATA
       TMP(1,1)=0.0
       CALL MOVE (TMP(1,1),TMP(2,1),16380)
       READ(5,31) ((TMP(J,I),I=MLO,MHI),J=NLO,NHI)
       DO 32 N=1,64
       I=N
       DO 32 K=1,64
       J=65-K
   32 INT(J,I)=TMP(N,K)
   31 FORMAT(26F3.0)
       WRITE(6,34) SNAME
   34 FORMAT (1H1,A4)
       MHI=64
       NHI=64
C   READ THE CCNTCURS
       READ(5,1030) NC1,(CONT1(I),I=1,NC1)
       WRITE (6,2010) (CONT1(I),I=1,NC1)
       READ (5,1080) DRA,DDEC
       WRITE (6,2080) DRA,DDEC
C   CALL THE PLOTTING ROUTINE
       CALL CNTPLT (.TRUE.,MHI,NHI,SNAME,NC1,CONT1,INT,PINT)
       WRITE (12,4011)
       WRITE (12,4012)
       GO TO 10
C   READ THE INTENSITY CATA
  100 IF(DISK) GO TO 101
C   READING CARCS
        READ(5,1010) SNAME,RA,DEC,PFLAG,XYINT,BNORM,XH,XL,YH,YL,MLO,MHI,
      1NLO,NHI,TMAX,XCHI,XCLO,YCHI,YCLO
 1010   FORMAT(A8,R12.2,S12.1,1X,L1,15X,F6.1,6X,E12.4,/,
     +  4F7.1,4I5,E12.4,/,4F7.1)
```

```
C    READ THE DATA
        TMAX=0.
        READ(5,1020) ((TMP(J,I),I=MLO,MHI),J=NLO,NHI)
      DO 400 N=NLO,NHI
      I=N
      DO 400 K=MLO,MHI
      J=K
  400 INT(J,I)=TMP(N,K)
 1020  FORMAT(26F3.0)
      GO TO 102
C       READ  DISK
  101 CONTINUE
      READ(5,1030) SNAME,RA,DEC,PFLAG,XYINT,BNORM,TMAX,MLO,MHI,NLO,NHI
     + ,BMAL,BMIN,PA,NIT,PERC,PUN,IBEAM,MAP,NOBEAM,NUMAP
      WRITE(6,432)
  432 FORMAT(1H1)
      WRITE(6,1030) SNAME,RA,DEC,PFLAG,XYINT,BNORM,TMAX,MLO,MHI,NLO,NHI
     + ,BMAL,BMIN,PA,NIT,PERC,PUN,IBEAM,MAP,NOBEAM,NUMAP
 1030 FORMAT(1X,A8,R12.2,S12.1,1X,L1,1X,F7.1,1X,2E15.4,/,
     + 4(1X,I2),20X,3(1X,F7.1),/,I4,1X,F5.3,1X,L1,/,4I6,/)
      MLO=1
      NLO=1
      MHI=64
      NHI=64
      XH=32*XYINT
      XL=-XH
      YH=XH
      YL=XL
      XCHI=(MHI-33)*XYINT
      XCLO=(MLO-33)*XYINT
      YCHI=(NHI-33)*XYINT
      YCLO=(NLO-33)*XYINT
      IMAP=NUMAP+1
      READ(10'IMAP) ARRAY
      DO 103 N=1,64
      DO 103 M=1,64
  103 INT(M,N)=IB(M+32,N+32)
  102 CONTINUE
C    RENORMALIZE DATA TO 100
        TN1=TMAX/BNORM
        DO 130 J=NLO,NHI
        DO 110 I=MLO,MHI
  110   IF(TMAX.LT.ABS(INT(J,I))) TMAX=INT(J,I)
  130   CONTINUE
        TMAX=TMAX/100.
        DO 140 J=NLO,NHI
        DO 140 I=MLO,MHI
  140   INT(J,I)=INT(J,I)/TMAX
        TN1=TN1*TMAX
      UVCELL=0.0
      TAPER=0.0
C    WRITE THE INPUT DATA
        WRITE(6,2000) SNAME,RA,DEC,ANGROT,XYINT,TN1,XH,XL,YH,YL
 2000        FORMAT(//,20X,'TOTAL INTENSITY DATA FOR ',A8,//,5X,
     +  'RA',R13.2,5X,'DEC',S13.1,10X,'MAP ROTATED',F7.1,//,
     +  10X,'INCREMENT',F7.1,'  NORMALIZATION',F6.4,
     +    '    BOUNDRIES',4F7.1)
        IF(ITYPE.EQ.4) GO TO 200
C    READ THE CONTOURS
        READ(5,1031) NC1,(CONT1(I),I=1,NC1)
```

```
 1031  FORMAT(I2,1X,20F3.0)
       WRITE(6,2010) (CCNT1(I),I=1,NC1)
 2010  FORMAT(1H-,10X,'CONTOUR LEVELS',/11X,20F5.0)
       READ(5,1080) DRA,DDEC
 1080  FORMAT(F3.0,1X,F3.0)
       WRITE(6,2080) DRA,DDEC
 2080  FORMAT(1H ,'GRID INCREMENT ',F5.0,1X,F5.0)
C   CALL THE PLOTTING ROUTINE
       CALL CNTPLT(.TRUE.,MHI,NHI,SNAME,NC1,CONT1,INT,PINT)
       WRITE(12,4011)
       WRITE(12,4012)
 4011  FORMAT('BRDR')
 4012  FORMAT('END ')
       IF(ITYPE.EQ.1) GO TO 10
       IF(ITYPE.EQ.3) GO TO 200
C    NOW WORK ON THE POLARIZED DATA
  200  IF(DISK) GO TO 201
CCC       READ  CARDS
       READ(5,1010) SNAME,RA,DEC,PFLAG,XYINT,BNORM,XH,XL,YH,YL,MLO,MHI,
     1NLO,NHI,TMAX,XCHI,XCLO,YCHI,YCLO
      TAPER=0.0
      UVCELL=0.0
       TNZ=TMAX/BNORM
C   READ THE DATA
       READ(5,1040) ((TPOL(J,I),I=MLO,MHI),J=NLO,NHI)
 1040  FORMAT(26F3.0)
      DO 500 N=NLO,NHI
      I=N
      DO 500 K=MLO,MHI
      J=K
  500 POL(J,I)=TPOL(N,K)
       GO TO 209
  201  CONTINUE
CCC    READ  DISK
      READ(5,3030) SNAME,RA,DEC,PFLAG,XYINT,BNORM,TMAX,MLO,MHI,NLO,NHI
     + ,BMAL,BMIN,PA,NIT,PERC,PUN,IBEAM,MAP,NOBEAM,NUMAP
      WRITE(6,3030) SNAME,RA,DEC,PFLAG,XYINT,BNORM,TMAX,MLO,MHI,NLO,NHI
     + ,BMAL,BMIN,PA,NIT,PERC,PUN,IBEAM,MAP,NOBEAM,NUMAP
 3030 FORMAT(1X,A8,R12.2,S12.1,1X,L1,1X,F7.1,1X,2E15.4,/,
     + 4(1X,I2),20X,3(1X,F7.1),/,I4,1X,F5.3,1X,L1,/,4I6,/)
      MLO=1
      NLO=1
      MHI=64
      NHI=64
CCC   READ  AND  STORE  REAL  MAP
       IMAP=NUMAP+1
       READ(10'IMAP) ARRAY
       DO 202 J=1,64
       DO 202 I=1,64
  202  POL(I,J)=CMPLX(0.,0.)
       DO 203 J=1,64
       DO 203 I=1,64
       RE=IB(I+32,J+32)
  203  POL(I,J)=POL(I,J)+CMPLX(RE,0.)
CCC    READ  IMAGINARY  MAP
       IMAP=IMAP+32
       READ(10'IMAP) ARRAY
       DO 204 J=1,64
       DO 204 I=1,64
       AIM=IB(I+32,J+32)
```

```
 204  POL(I,J)=PCL(I,J)+CMPLX(0.,AIM)
 209  CONTINUE
      TMAX=0.
CCC   RENORMALIZE  PCLN DATA  TO  100
      DO 230 J=NLO,NHI
      DO 210 I=MLO,MHI
      PINT(J,I)=CABS(PCL(J,I))
 210  IF(TMAX.LT.PINT(J,I)) TMAX=PINT(J,I)
 230  CONTINUE
      TMAX=TMAX/100.
      DO 240 J=NLC,NHI
      DO 240 I=MLO,MHI
      POL(J,I)=PCL(J,I)/TMAX
 240  PINT(J,I)=PINT(J,I)/TMAX
      TN2=TN2*TMAX
C   WRITE THE CATA
      WRITE(6,2C15) SNAME,RA,DEC,ANGROT,XYINT,TN2,XH,XL,YH,YL
 2015 FORMAT(//////,20X,'POL.  INTENSITY DATA FOR ',A8,//,5X,
     + 'RA',R13.2,5X,'DEC',S13.1,10X,'MAP ROTATED',F7.1,//,
     + 10X,'INCREMENT',F7.1,'  NORMALIZATION',F8.4,
     + '    BOUNDRIES',4F7.1)
      IF(ITYPE.EC.4) GO TO 300
C   READ THE CCNTCURS
      READ(5,1031) NC1,(CONT1(I),I=1,NC1)
      WRITE(6,2020) (CCNT1(I),I=1,NC1)
 2020 FORMAT(1F ,20F5.0)
      READ(5,1C3C) CRA,DDEC
      WRITE(6,2080) CRA,DDEC
C   PLOT THE POLARIZEC INTENSITY
      CALL CNTPLT(.TRUE.,MHI,NHI,SNAME,NC1,CONT1,PINT,INT)
      WRITE(12,4011)
      WRITE(12,4012)
C
      IF(ITYPE.EC.2) GO TO 10
C   PLOT THE LINE PCLARIZATION PLOT
 300  READ(5,1C31) NC1,(CONT1(I),I=1,NC1)
      WRITE(6,2010) (CCNT1(I),I=1,NC1)
      READ(5,1080) CRA,DDEC
      WRITE(6,2C80) CRA,DDEC
C   DO THE TOTAL INTENSITY PLOT
      CALL CNTPLT(.TRUE.,MHI,NHI,SNAME,NC1,CONT1,INT,PINT)
      WRITE(6,3000)
 3000 FORMAT(1F1,//)
      TMAX=TN2/TN1
C   SET UP ARRAYS FCR THE LINE PLOT
      DO 350 J=NLC,NHI
      DO 350 I=MLO,MHI
      IF(CABS(PCL(J,I)).NE.0.) INT(J,I)=ATAN2(AIMAG(POL(J,I)),
     + REAL(PCL(J,I)))*0.5
 350  CONTINUE
      CALL CNTPLT(.FALSE.,MHI,NHI,SNAME,NC1,CONT1,PINT,INT)
      WRITE(12,4011)
      WRITE(12,4012)
      GO TO 10
 900  WRITE(12,5000)
 5000 FORMAT('STOP')
      STOP
     END
C   SUBROUTINE CNTPLT
C
```

```
      SUBROUTINE CNTPLT(FLAG,IX,IY,SOURCE,NC,CONT,BITGRL,PANG)
      LOGICAL FLAG
      DIMENSICN BITGRL(64,64),PANG(64,64),CONT(20)
      REAL*8 SCURCE
      COMMON RA,DEC,XH,XL,YH,YL,XYINT,DRA,DDEC,ANGROT
      REAL*8 RA,CEC
C     INITIALIZE
      IF(.NOT.FLAG) GO TO 200
      WRITE(12,1000) SCURCE
 1000 FORMAT('JOB ',2X,A8)
C   ROTATE MAP
      WRITE(12,1005)
 1005 FORMAT('ANGL  180. 180.')
      XMIN=0.
      XMAX=IX-1
      XINC=1.
      YMIN=0.
      YMAX=IY-1
      YINC=1.
     XSCALE=4.C
     YSCALE=4.C
      X00=XMAX/XSCALE -5.0
      Y00=1.
      WRITE(12,1010) XSCALE,YSCALE,X00,Y00,XMIN,XINC,XMAX,
     + YMIN,YINC,YMAX
 1010 FORMAT('SIZX ',4F5.0,2(F10.0,F5.0,F10.0))
      WRITE(12,1020)
 1020 FORMAT('ARAY  C.C5 0.C7    1    2    1')
     WRITE(12,707)
  707 FORMAT('ARAY ','    1   64    1   64')
C    WRITE DATA
     DO 10 J=1,64
  10  WRITE(12,1030)((BITGRL(J,I),I=1,64))
 1030 FORMAT(('ARAY ',5E14.5,5X))
      WRITE(12,1C40)
 1040 FORMAT('BEND')
C   CONTOUR LEVELS
      DO 100 I=1,NC
  100 WRITE(12,1050) CCNT(I)
 1050 FORMAT('LEV  ',F5.1,39X,'1')
      WRITE(12,4317)
      CALL GRIC(RA,DEC,XH,XL,YH,YL,XYINT,DRA,DDEC,XMAX,YMAX)
      RETURN
C   DETERMINE THEPOLARZIATION LINE PLOT
  200 WRITE(12,4317)
 4317 FORMAT('PLCT')
      IYY=IY-2
     IXX=IX-2
      DO 275 J=3,IY
     Y=J-1
     DO 270 I=3,IX
     X=I-1
      SIZE=BITGRL(J,I)/100.
      PANGLE=PANG(J,I)-ANGROT*6.28315/360.
      DX=SIZE*COS(PANGLE)
     DX=-DX
      DY=SIZE*SIN(PANGLE)
     VLT =ABS(CX*DX+CY*DY)
     VL=SQRT(VLT)
CUT OFF IN POLARIZED LINE PLOT
```

```
      IF(VL.LT.C.125) GO TO 270
       X1=X+DX
       X2=X-DX
       Y1=Y+DY
       Y2=Y-DY
       WRITE(12,4318) X1,Y1,X2,Y2
 4318  FORMAT('LINE      1',4F5.2,19X,'1')
  270  CONTINUE
  275  CONTINUE
       CALL GRID(RA,DEC,XH,XL,YH,YL,XYINT,DRA,DDEC,XMAX,YMAX)
       RETURN
      END
       SUBROUTINE GRID(RA,DEC,XH,XL,YH,YL,XYINT,DRA,DDEC,XMAX,YMAX)
       REAL*8 RA,DEC,RAMAX,RAMIN,DECMAX,DECMIN,C1,ONE,FR
       DATA CNE/1.0D0/
       REAL*8 ZERC/0.0D0/
       IF(DRA*CDEC.EC.0.) RETURN
       C1=3.14159265/180./3600.
       DECMAX=CEC+YH*C1
       DECMIN=DEC+YL*C1
       FSH=DCOS(0.5*(DECMAX+DECMIN))
       RAMAX=RA+XH*C1/FSH
       RAMIN=RA+XL*C1/FSH
       DR=DRA*15.*C1
       DD=DDEC*C1
       RUNIT=XYINT*C1/FSH
       DUNIT=XYINT*C1
C    DO THE RA GRIC MARKS
      ITEMP=RAMAX/DR
       FR=ITEMP*DR
   10 CONTINUE
      X1=(RAMAX-FR)/RUNIT
       X2=X1
       Y1=0.
       Y2=0.5
       WRITE(12,4318) Y1,X1,Y2,X2
 4318  FORMAT('LINE      1',4F5.2,19X,'1')
       Y1=YMAX
       Y2=Y1-0.5
       WRITE(12,4318) Y1,X1,Y2,X2
      FR=FR-DR
      IF(FR .LT. RAMIN) GO TO 20
       GO TO 10
   20  CONTINUE
C    DO THE DECLINATICN GRID
       IF(DECMIN.LT.ZERO) ITEMP=ITEMP-1
       ITEMP=DECMIN/DC
       FR=ITEMP*DC
   30  FR=FR+DC
       IF(FR.GT.DECMAX) GO TO 40
       X1=0.
       X2=0.5
       Y1=(FR-DECMIN)/DUNIT
       Y2=Y1
       WRITE(12,4318) Y1,X1,Y2,X2
       X1=XMAX
       X2=X1-0.5
       WRITE(12,4318) Y1,X1,Y2,X2
       GO TO 30
   40  RETURN
```

```
      END
//FT06F001  DD  SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=133,BUFNO=1)
//GO.FT10F001  DD  DSNAME=MOORE.GHC2,DISP=SHR,VOL=PRIVATE
//GO.FT12F001  DD DSNAME=WRIGHT.POLAR1,UNIT=DISK,
//              DISP=(NEW,PASS),SPACE=(CYL,(2,2)),
//              DCB=(,RECFM=FB,LRECL=80,BLKSIZE=80)
//GO.SYSIN DD *
3             T
 3027      00 52 44.80 +68 06 02.0 F 00003.0    594.0000E+01   195.5000E+02
 28 37 31 34                             7.0      7.0      0.7
 150 0.010 F
 14032 14000 14532 14500

10 100+90+80+70+60+50+40+30+20+10
10. 30.
 3027      00 52 44.80 +68 06 02.0 T 00003.0    594.0000E+01   257.0000E+01
 28 37 31 34                             7.0      7.0      0.7
 150 0.010 F
 14032 14064 14532 14564

10 100+90+80+70+60+50+40+30+20+10
10. 30.
 5  90+70+50+30+10
10. 30.
3             T
 3027      00 52 44.80 +68 06 02.0 F 00003.0    294.0000E+01   261.0000E+01
 28 37 31 34                             7.0      7.0      0.7
 150 0.010 F
 14160 14128 14660 14628

10 100+90+80+70+60+50+40+30+20+10
10. 30.
 3027      00 52 44.80 +68 06 02.0 T 00003.0    294.0000E+01   265.6000E+01
 28 37 31 34                             7.0      7.0      0.7
 150 0.010 F
 14160 14192 14660 14692

10 100+90+80+70+60+50+40+30+20+10
10. 30.
 5  90+70+50+30+10
10. 30.
//KONTOUR  EXEC  PGM=KONTOUR
//FT04F001  DD  UNIT=DISK,SPACE=(CYL,(10,10))
//FT06F001  DD  SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=133,BUFNO=1)
//FT07F001  DD  SYSOUT=B
//PLOTTAPE DD SYSOUT=C,SPACE=(CYL,(5,5),RLSE)
//FT05F001   DD DSN=WRIGHT.POLAR1,UNIT=DISK,DISP=(OLD,DELETE)
```

```
//CASA JOB (234,P,6,13,4),'NEW TAU',MSGLEVEL=1,CLASS=C
//  EXEC   FORTCLG
//FORT.SYSIN  DD *
C                     PROGRAM TO FIND OPTICAL DEPTHS
C                     IN HYDROGEN LINE ABSORPTION
       REAL TNORM(2)
      INTEGER*2 IM(2,64,64),ARRAY(512,32),IB(128,128)
      EQUIVALENCE (ARRAY(1,1),IB(1,1))
      INTEGER*2 IMAP(2),DECK
       INTEGER*2 MAP(2),NUMAP(2)
      COMMON/PFLAG/DISK,DIRTY,LMAP
      LOGICAL PFLAG,PUN,DISK,LMAP,WRITE,PLOTFG
      LOGICAL DIRTY
       COMMON/HEADER/SNAME,RA,DEC,XYINT,BEAM,MLO,MHI,NLO,NHI
       REAL*8 SNAME,RA,DEC
       COMMON/INFC/ITYP,ICHAN,NCHANS,BW,VEL
C
 1000  FORMAT(1X,I1,5(1X,L1))
 1001  FORMAT(//,' ITYP,DISK,DIRTY,WRITE,PLOT',//)
 1010     FORMAT(A8,R12.2,S12.1,1X,L1,13X,F6.1,6X,E12.4,/,
     +  4F7.1,4I5,E12.4,/,4F7.1)
 1011 FORMAT(//,1X,A8,R12.2,S12.1,1X,L1,13X,F6.1,6X,E12.4,/,1X,
     + 4F7.1,4I5,E12.4,/,1X,4F7.1)
 1020 FORMAT(26I3)
C
  1   READ(5,1000,END=900) ITYP,DISK,DIRTY,WRITE,PLOTFG
       PRINT 1001
      PRINT 1000, ITYP,DISK,DIRTY,WRITE,PLOTFG
C
      DEFINE FILE 10 (23360,256,U,IO)
C
      L=1
    2 PRINT 1002
 1002 FORMAT(1H1)
      IF ( .NOT. DISK) GO TO 10
      LMAP=.TRUE.
      READ(5,1030,END=900) SNAME,RA,DEC,PFLAG,XYINT,BNORM,TMAX,
     +ML,MH,NL,NH
     + ,BEAM,BMIN,PA,NIT,PERC,PUN,IBEAM,IMAP(L),NUBEAM,NUMAP(L)
      WRITE(6,1030) SNAME,RA,DEC,PFLAG,XYINT,BNORM,TMAX,ML,MH,NL,NH
     + ,BEAM,BMIN,PA,NIT,PERC,PUN,IBEAM,IMAP(L),NUBEAM,NUMAP(L)
 1030 FORMAT(1X,A8,R12.2,S12.1,1X,L1,1X,F7.1,1X,2E15.4,/,
     + 4(1X,I2),20X,3(1X,F7.1),/,I4,1X,F5.3,1X,L1,/,4I6)
      MAP(L)=NUMAP(L)
      IF(DIRTY) MAP(L)=IMAP(L)
      TNORM(L)=TMAX
       READ(10'MAP(L)+1) ARRAY
      DO 5 J=33,96
      DO 5 I=33,96
    5 IM(L,I-32,J-32)=IB(I,J)
      IF(DIRTY) PRINT 1009
       IF(.NOT.DIRTY) PRINT 1019
 1009 FORMAT(1X,'COMPARISON OF DIRTY MAPS ')
 1019 FORMAT(1X,'COMPARISON OF CLEAN MAPS ')
      GO TO 21
   10 MAP(L)=L
  100  READ(5,1010) SNAME,RA,DEC,PFLAG,XYINT,BNORM,XH,XL,YH,YL,MLO,MHI,
     1NLO,NHI,TMAX,XCHI,XCLO,YCHI,YCLO
       PRINT 1011, SNAME,RA,DEC,PFLAG,XYINT,BNORM,XH,XL,YH,YL,MLO,MHI,
     XNLO,NHI,TMAX,XCHI,XCLO,YCHI,YCLO
```

```
C
      TNORM(L)=TMAX
      READ(5,1020,ERR=21) ((IM(L,I,J),I=1,64),J=1,64)
      MH=33+XCHI/XYINT
      ML=33+XCLO/XYINT
      NH=33+YCHI/XYINT
       NL=33+YCLO/XYINT
   21        IF(PLOTFG) CALL PLOT(0,L,MAP,IM,1,64,1,64,TNORM(L))
C
CALCULATE CENTRES OF CHANNELS
      READ(5,1040,END=900) ICHAN,NCHANS,BW,VEL
      DNUDV=-14.20405752/2.997930
      CENTRE=(2*ICHAN+NCHANS-1)/2.
      DNU=-BW/96.
      DV=DNU/DNUDV
      VELCHN=(CENTRE-49.)*DV+VEL
      WIDTH=NCHANS*DV
      FREQ=VEL*DNUDV-(CENTRE-49.)*DNU
      FWIDTH=-NCHANS*DNU
      PRINT 1050
      PRINT 1040    ,       ICHAN,NCHANS,BW,VEL
      PRINT 1060
      PRINT 1070,VELCHN,WIDTH,FREQ,FWIDTH
 1040 FORMAT(1X,2I6,2X,2F10.3)
 1050 FORMAT(//,' ICHAN,NCHANS,BW(KHZ),VELOCITY(KMS)',/)
 1060 FORMAT(/,' MAP VELOCITY,WIDTH(KMS),FREQUENCY,WIDTH(KHZ)',/)
 1070 FORMAT(1X,10F10.3)
C
   19 IF(L .EQ. 2) GO TO 20
CCC   DISPLAY  ALL MAPS  OVER  REFERENCE  AREA
      MLO=ML
      MHI=MH
      NLO=NL
      NHI=NH
      L=2
      GO TO 2
   20 CONTINUE
C
        TWO=   TNORM(2)/TNORM(1)
       THREE=     ALOG(TNORM(2)/TNORM(1))
      IF(DIRTY) TWO=1.
      IF(DIRTY) THREE=0.
      PRINT 1234, TWO,THREE
 1234 FORMAT(//,' SCALING FACTOR FOR MAP QUOTIENT',F10.4,//,' SCALING FA
     +CTOR FOR LOG QUOTIENT',F10.4)
C
      DO 500 N=NLO,NHI
      DO 500 M=MLO,MHI
      GO TO (71,72,73), ITYP
   71 IM(2,M,N)=IM(2,M,N)-IM(1,M,N)
      GO TO 500
   72 IM(2,M,N)=(0.+IM(2,M,N))*TWO/(IM(1,M,N)+0.01) *100
      GO TO 500
   73  IF (IM(1,M,N) .LE. 0 .OR. IM(2,M,N) .LE. 0) GO TO 173
      IM(2,M,N)=(ALOG(IM(2,M,N)+0.)-ALOG(IM(1,M,N)+0.)+THREE)*(-100.)
       GO TO 500
  173 IM(2,M,N)=999
  500 CONTINUE
      LMAP=.FALSE.
   23 CALL PLOT(ITYP,L,MAP,IM,MLO,MHI,NLO,NHI,1.)
```

```
      IF(WRITE) CALL CARDS(IM)
      GO TO 2
  900 STOP
      END
      SUBROUTINE PLOT(K,MAPNC,MAP,IM,MLO,MHI,NLO,NHI,TNORN)
      LOGICAL DISK,DIRTY,WRITE,PFLAG
      INTEGER*2 ICOMP(80),IM(2,64,64),MAP(2)
      LOGICAL LMAP
      COMMON/PFLAG/DISK,DIRTY,LMAP
      K1=K+1
      GO TO (21,22,23,24), K1
   21 PRINT 201, MAP(L)
      GO TO 15
   22 PRINT 202, MAP(2),MAP(1)
      GO TO 15
   23 PRINT 203, MAP(2),MAP(1)
      GO TO 15
   24 PRINT 204, MAP(2),MAP(1)
  201 FORMAT(///,'  THE MAP',I6,' RENORMALIZED TO MAX = 100',//)
  202 FORMAT(///,' THE DIFFERNCE OF MAP',I6,1X,'AND MAP',I6,//)
  203 FORMAT(///,' THE QUOTIENT OF MAP TIMES 10',I6,1X,'AND MAP',I6,//)
  204 FORMAT(///,' THE LOG OF QUOTIENT OF MAP',I6,1X,'AND MAP',I6,' ,= 1
     +0  TIMES OPTICAL DEPTH,TAU',//)
 2080 FORMAT(1H )
 2060 FORMAT(1H ,I2,1X,64I2)
   15 PRINT 2080
      PRINT 2060,I,(I,I=1,64)
      PRINT 2080
C
      LSCALE=10
      IF(DISK .AND. LMAP) LSCALE=100
      IF(DISK .AND. LMAP .AND. DIRTY) LSCALE=TNORN/100
C
      DO 61 N=1,64
   61 ICOMP(N)=0
      DO 62 N=1,64
      J=65-N
      IF(J .GT. NHI .OR. J .LT. NLO) GO TO 62
      DO 63 I=MLO,MHI
      ICOMP(I)=IM(MAPNC,I,J)/LSCALE
      IF( ICOMP(I) .LT. -9) ICOMP(I)= -ICOMP(I)
   63 CONTINUE
      PRINT 2060, J,(ICOMP(I),I=1,64)
   62 CONTINUE
      PRINT 2060,I,(I,I=1,64)
      RETURN
      END
      SUBROUTINE CARDS(IM)
      INTEGER*2 IM(2, 64, 64)
      COMMON/HEADER/SNAME,RA,DEC,XYINT,BEAM,MLO,MHI,NLO,NHI
      REAL*8 SNAME,RA,DEC
      COMMON/INFO/ITYP,ICHAN,NCHANS,BW,VEL
      COMMON/PFLAG/DISK,DIRTY,LMAP
      LOGICAL DISK,DIRTY,LMAP
      TNORM=100.
      PUNCH 4000,SNAME,RA,DEC,ITYP,DIRTY,XYINT,BEAM,TNORM,MLO,MHI,
     +NLO,NHI,ICHAN,NCHANS,BW,VEL
      PRINT 2000
      PRINT 4000,SNAME,RA,DEC,ITYP,DIRTY,XYINT,BEAM,TNORM,MLO,MHI,
     +NLO,NHI,ICHAN,NCHANS,BW,VEL
```

```
      DO 10 N=NLO,NHI
      DO 10 M=MLO,MHI
      IF(IM(2,M,N).GE.1000 ) IM(2,M,N)=999
      IF(IM(2,M,N).LT.1   ) IM(2,M,N)=0
 10   CONTINUE
      PUNCH 1000,((IM(2,M,N),M=MLO,MHI),N=NLO,NHI)
 1000 FORMAT(26I3)
 2000 FORMAT (//,' PUNCHED :- ',/)
 4000 FORMAT(1X,A8,R12.2,S12.1,I4,1X,L1,1X,2F4.1,F6.0,2X,4I6,/,
     +1X,2I6,2X,2F10.3)
      RETURN
     END
//GO.FT05F001  DD  *
 5 T F T F
 CASAREF 23 21 06.80 +58 32 46.9 F 00015.0    372.0000E+01   130.0600E+02
 24 40 23 42                              045.0 00045.0      0.0
 100 0.010 F
  3384  3416  3576  3608
     7     12  78.125    -48.8
 CASAD33  23 21 06.80 +58 32 46.9 F 00015.0    372.0000E+01   118.9400E+02
 24 40 23 42                              045.0 00045.0      0.0
 100 0.010 F
  3384  3448  3576  3640
    33      1  78.125    -48.8
 CASAD34  23 21 06.80 +58 32 46.9 F 00015.0    372.0000E+01   114.8400E+02
 24 40 23 42                              045.0 00045.0      0.0
 100 0.010 F
  3384  3480  3576  3672
    34      1  78.125    -48.8
//GO.FT10F001  DD  DSNAME=MCORE.GHC4,DISP=SHR,VOL=PRIVATE
```

THE END