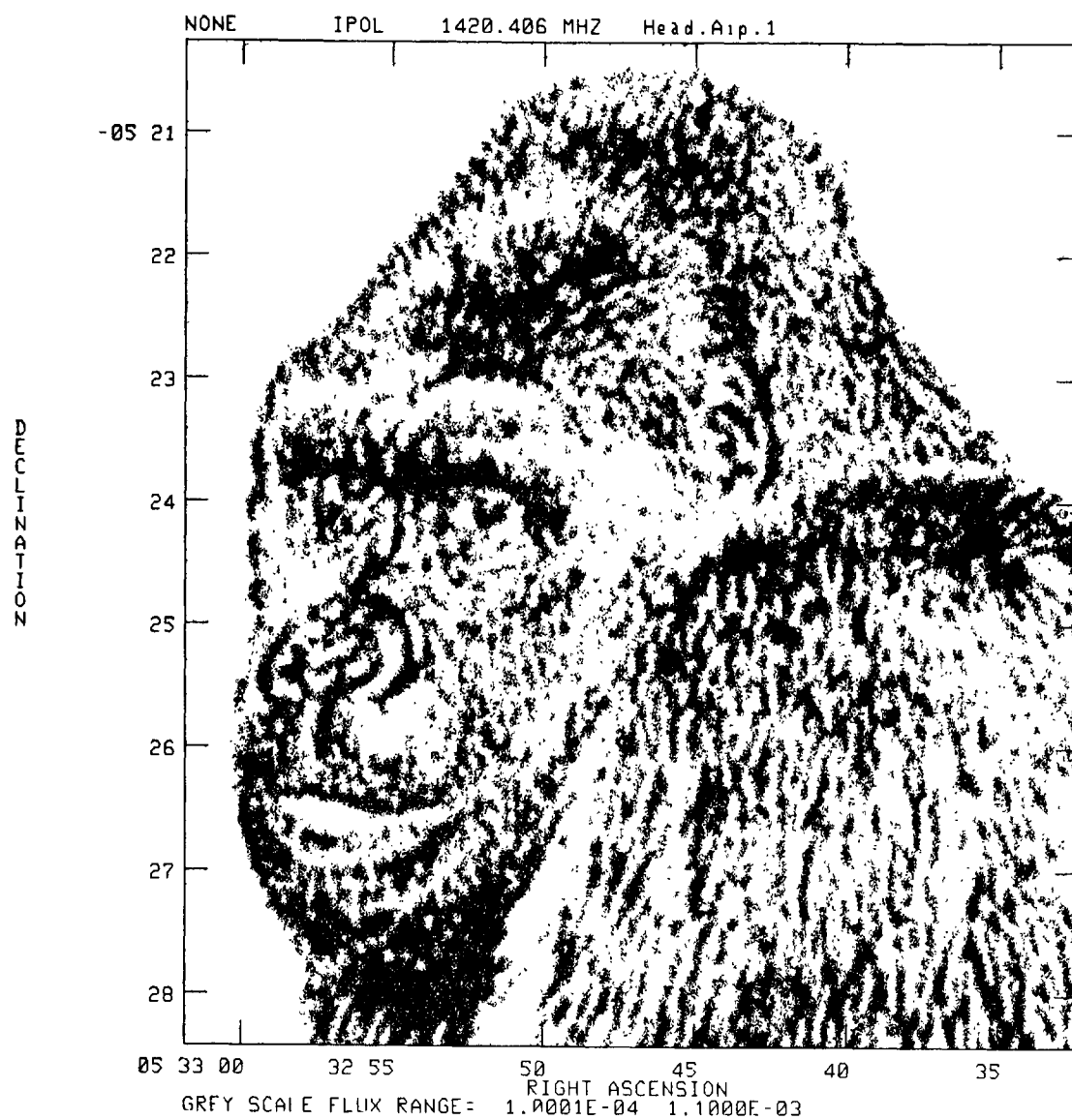


# AIPS COOKBOOK

15-Jan-1983



The National Radio Astronomy Observatory

Edgemont Road

Charlottesville, VA 22901

Operated by Associated Universities, Inc.

under contract with the National Science Foundation

## ACKNOWLEDGEMENTS

The *COOKBOOK* cover design is by Pat Smiley of the NRAO Graphic Arts Department. It is based on a design suggested by John Bally of Bell Labs.

The digital image on the title page was plotted on a Versatec by the *AIPS* tasks **GREYS** and **PRTPL**. It is a processed subimage of a digital image which was sent to us by David Garrett of the University of Texas.

The Editors are grateful.

# TABLE OF CONTENTS

1.	INTRODUCTION	1
1.1.	What <i>AIPS</i> can do	2
1.2.	Organization of the <i>COOKBOOK</i>	2
2.	STARTING UP <i>AIPS</i>	3
2.1.	Signing up for <i>AIPS</i> time	3
2.2.	Using the terminals	3
2.3.	Logging in	4
2.4.	Hardware tape mount	5
2.5.	Software tape mount	5
3.	MAKING YOUR FIRST MAPS	6
3.1.	Making maps with data storage on disk	6
3.1.1.	EXIND and PRTP	6
3.1.2.	UVLOD	7
3.1.3.	UVSRT	8
3.1.4.	UVMAP	9
3.2.	Making maps directly from tape— <i>APMAP</i>	10
4.	SOME IMPORTANT UTILITIES	11
4.1.	The Message file	11
4.2.	The Catalog file	11
4.2.1.	Renaming data files	12
4.2.2.	Speedy data file selection	12
4.2.3.	Catalog entry status	13
4.2.4.	Other catalog listings	13
4.3.	SAVE, GET, TPUT, and TGET	13
4.4.	Monitoring the disk space	14
4.5.	Finding information in <i>AIPS</i>	14
5.	IMPROVING MAPS - CLEAN, SELF CAL, EDITING	15
5.1.	CLEANing maps	15
5.1.1.	Restarting CLEANs	17
5.1.2.	Listing CLEAN components	17
5.2.	Self-calibration	17
5.3.	Editing UV data	19
6.	READING AND DISPLAYING IMAGES	20
6.1.	Loading an image from tape to disk	20
6.2.	Loading an image from disk to the TV display	21
6.2.1.	Alternative color coding	22
6.2.2.	Transfer functions	22
6.3.	Reading out image data using the trackball/cursor	23
7.	TAKING HARD COPY OF IMAGES	24
7.1.	Ordinary contour plots (CNTR)	24
7.2.	Contour plots with polarization vectors (PCNTR)	25
7.3.	Deleting unwanted plot files (EXTDEST)	25
7.4.	SLICE files (profile plots)	26
7.5.	Dicomed copies of images	27
8.	ANALYZING IMAGES	28
8.1.	Combining images (COMB)	28
8.1.1.	Polarized intensity and p.a. maps	28
8.1.2.	Other map combination options	29

## TABLE OF CONTENTS

---

8.2.	Map statistics and flux integration ( <b>IMEAN</b> )	29
8.3.	Fitting of images	30
8.3.1.	Parabolic fit to maximum ( <b>MAXFIT</b> )	30
8.3.2.	Two-dimensional Gaussian fitting ( <b>IMFIT</b> )	30
8.3.3.	Gaussian fits to slices ( <b>SLFIT</b> )	31
9.	<b>SPECTRAL LINE SOFTWARE</b>	32
9.1.	Reading in multichannel UV data	32
9.2.	Making spectral line maps	32
9.3.	Building the cube	33
9.4.	Transposing the cube	34
9.5.	Further display and processing	34
10.	<b>TIDYING UP AND EXITING AIPS</b>	35
10.1.	Backups	35
10.2.	Deleting your data	35
10.3.	Sending comments to the <i>AIPS</i> programmers	36
10.4.	Exiting	36
11.	<b>PANIC SECTION</b>	37
11.1.	My printout is semi-infinite!!	37
11.2.	The TEK screen will not print hard copy!!	37
11.3.	My tape refuses to be read!!	37
11.4.	My data catalog has vanished!!	38
11.5.	My program crashed for lack of disk!!	38
11.6.	My terminal has hung itself up!!	38
12.	<b>AIPS FOR THE MORE SOPHISTICATED USER</b>	39
12.1.	<i>AIPS</i> syntax	39
12.2.	Datafile formats	40
12.3.	<i>AIPS</i> language	41
12.4.	Processing loops	41
12.5.	Procedures	41
12.6.	<b>RUN</b> files	44
12.7.	Batch jobs	45
12.8.	Writing your own programs with <b>POPS</b>	46
13.	<b>Current AIPS software</b>	48
	HELP TAPU	48
	HELP UVPR	49
	HELP MAPETC	50
	HELP APTASKS	50
	HELP GENERAL	51
	HELP CATINFO	52
	HELP TVGEN	52
	HELP TVINTER	53
	HELP CURSOR	54
	HELP TVCOLOR	54
	HELP PL2D	55
	HELP SL1D	55
	HELP ANALYSIS	56
	HELP DELETE	56
	HELP CUBE	57
	HELP VLBI	57
	HELP POPSYM	58

---

HELP INDEX . . . . .	80
G. GLOSSARY . . . . .	87
Z. SYSTEM-DEPENDENT AIPS TIPS . . . . .	78
Z.1. VLA Site VAXes . . . . .	78
Z.1.1. Signing up for AIPS time at the VLA . . . . .	78
Z.1.2. Using the Tektronix 4025 terminals at the VLA . . . . .	78
Z.1.3. Logging in at the VLA . . . . .	79
Z.1.4. Hardware tape mount at the VLA . . . . .	80
Z.1.4.1. Mounting tapes on TU77s and TU78s . . . . .	80
Z.1.4.2. Mounting tapes on Telex 6250s . . . . .	80
Z.1.5. Software tape mount at the VLA . . . . .	81
Z.1.6. Monitoring disk space at the VLA . . . . .	81
Z.1.7. Solving problems at the VLA . . . . .	82
Z.1.7.1. Stopping excess printout at the VLA . . . . .	82
Z.1.7.2. Hard copy device . . . . .	82
Z.1.7.3. Booting the VLA VAX systems . . . . .	83
Z.1.7.4. Other problems at the VLA . . . . .	83
Z.1.8. Editing RUN files at the VLA . . . . .	83
Z.2. Charlottesville VAX . . . . .	84
Z.2.1. Signing up for AIPS time on the CV VAX . . . . .	84
Z.2.2. Using the terminals on the CV VAX . . . . .	84
Z.2.3. Logging in to AIPS on the CV VAX . . . . .	85
Z.2.4. Hardware tape mount on the CV VAX . . . . .	86
Z.2.4.1. Mounting tapes on TU77 . . . . .	86
Z.2.4.2. Mounting tapes on the System Industries drive . . . . .	86
Z.2.5. Software tape mount on the CV VAX . . . . .	86
Z.2.6. Monitoring disk space on the CV VAX . . . . .	87
Z.2.7. Solving problems on the CV VAX . . . . .	88
Z.2.7.1. Stopping excess printout on the CV VAX . . . . .	88
Z.2.7.2. Hard copy device . . . . .	88
Z.2.7.3. Booting the CV VAX system . . . . .	88
Z.2.7.4. Other problems on the CV VAX . . . . .	89
Z.2.8. Editing RUN files on the CV VAX . . . . .	89
Z.3. Charlottesville MODCOMP . . . . .	90
Z.3.1. Signing up for AIPS time on the MODCOMP . . . . .	90
Z.3.2. Using the Tektronix terminals on the MODCOMP . . . . .	90
Z.3.3. Logging in to AIPS on the MODCOMP . . . . .	91
Z.3.4. Hardware tape mount on the MODCOMP . . . . .	92
Z.3.5. Software tape mount on the MODCOMP . . . . .	92
Z.3.6. Monitoring disk space on the MODCOMP . . . . .	92
Z.3.7. Solving problems on the MODCOMP . . . . .	93
Z.3.7.1. Stopping excess printout on the MODCOMP . . . . .	93
Z.3.7.2. Hard copy device . . . . .	93
Z.3.7.3. Booting the MODCOMP system . . . . .	94
Z.3.7.4. Other problems on the MODCOMP . . . . .	94
Z.3.8. Editing RUN files on the MODCOMP . . . . .	95

THIS PAGE DELIBERATELY LEFT BLANK.

## 1. INTRODUCTION

This *COOKBOOK* is meant to help beginning users of the *Astronomical Image Processing System (AIPS)* by providing a recipe approach to the most basic *AIPS* operations. While it illustrates some aspects of *AIPS*, it does not pretend to be complete. It does however include detailed instructions for running the most important items of *AIPS* software. With these as a model, the user should be able to run other *AIPS* software aided by the *HELP* and *INPUTS* files and the complete index of software given in § 13 of the *COOKBOOK*.

*AIPS* software is continually changing and growing. (This edition describes the 15 January 1983 release of *AIPS*, frequently referred to as 15JAN83.) Users are encouraged to recommend new and better analysis and display tools, and to help debug the existing software by filling out "Gripes" (see § 10.4). Please note that examples of bugs that are documented by printouts of inputs, message logs, etc. are most useful to the programmers. Also note that written bug reports are much more effective than verbal reports.

The original *COOKBOOK* was written by Alan Bridle as he first encountered *AIPS*. It was then substantially rewritten by Alan Bridle and Ed Fomalont, with assistance from Tim Cornwell and Jacqueline van Gorkom. The present edition was *T<sub>E</sub>X*set and revised by Eric Greisen with the assistance of Don Wells, Fred Schwab, and Alan Bridle. It has been updated (and corrected) to reflect the state of *AIPS* at 15 January 1983 and an attempt has been made to separate into Appendix Z instructions and advice peculiar to the individual *AIPS* sites of the NRAO. It is our intention to have future editions edited by Eric Greisen and Don Wells, with scientific guidance from Alan Bridle (Charlottesville) and Ed Fomalont (VLA). Please report any inadequacies or errors in the *COOKBOOK* to Alan Bridle or Ed Fomalont, preferably in writing.

This edition contains a new section which we recommend highly — a Glossary of astronomical and computing terms written by Fred Schwab. In this edition of the *AIPS COOKBOOK*, the Glossary is in a preliminary and incomplete version. We solicit comments on the contents of this document and the suitability of placing it in the *COOKBOOK*.

As a result of typesetting, the *COOKBOOK* no longer exists in a form which can be listed intelligibly on a line printer. Copies may be obtained by writing to the NRAO Computer Division at Charlottesville or at the VLA. A faster method is to use your own copying machine.

Additional written documentation on *AIPS* is available in the form of a three-volume manual called *AIPS: A Manual, NRAO Computer Division Users' Manual Series*, Number 31. The original text files and a printed copy should be available at your installation. *AIPS* also provides run-time documentation in the form of *HELP* and *INPUTS* files which may be viewed at the terminal. Should these not suffice, consult your local *AIPS* Manager and then, if needed, call the *AIPS* programmers in Charlottesville. Although individual *AIPS* programs have often been written and are best understood by a single programmer, the *AIPS* group as a whole assumes responsibility for all released software. Anyone in the group will attempt to help you or, at least, to put you in touch with some other member of the group better able to help you.

### 1.1. What AIPS can do

AIPS is a system of hardware and software for:

1. Processing UV data to make maps (images) and to improve their quality.
2. Displaying images in a variety of ways.
3. Analyzing and combining images to obtain useful parameters.

Input to and output from AIPS is by magnetic tape. The UV data can be read or written in the so-called EXPORT format which can be generated by the DEC-10 at the VLA or the IBM in Charlottesville. UV data can also be read and written in the "UV" extension of the FITS format. Maps can be read or written in the FITS format. A special Charlottesville IBM format for maps is also supported.

### 1.2. Organization of the COOKBOOK

Section 2 of the COOKBOOK describes in general terms how to get started in AIPS — signing up, logging in, mounting tapes, etc. Appendix Z gives the details of these operations specific to NRAO's various AIPS sites. Your local AIPS Manager may be able to provide you a version of this Appendix appropriate to your system. Section 3 will lead you through the basics of reading in your UV data and making maps — the operations which beginning users are most likely to encounter first. Section 4 describes a few useful features of AIPS which you will need to use it efficiently. This Section is best read after you have acquired some experience with AIPS either by following through § 3 or by some other means. Sections 5 through 8 then introduce the basic AIPS tools for improving images, for making interactive and hard-copy displays of them, and for analyzing them. Section 9 contains hints and further AIPS tools of particular interest to spectral-line users and other observers who have data of more than 2 dimensions. Section 10 deals with exiting from AIPS and tidying up your disk areas to be polite to the other users. Section 11 suggests some cures for common hang-ups and miscellaneous "disasters" which seem to afflict AIPS users. No such list can be made comprehensive or sufficiently general to cover all the computer systems now running AIPS. You will need to consult with your local AIPS Manager or other users if you encounter an unlisted problem.

Section 12 is intended for the "mature" AIPS user who wishes to learn about data formats, procedures, RUN files, and various subtleties of AIPS syntax. We recommend that you read this after becoming familiar with the operations described in §§ 3 through 8. Section 13 contains lists of all available routines broken down by categories and in a complete, alphabetic list. Appendix G presents Fred Schwab's Glossary of radio astronomy data processing terminology.

Throughout this COOKBOOK, RESPONSES TO BE TYPED BY THE USER APPEAR IN THE PRESENT FONT. Prompts provided by the operating or AIPS systems are left-justified on the same line, e.g. system prompt \$, AIPS prompt >. THIS IS THE FONT USED FOR SAMPLE OUTPUTS FROM THE COMPUTER. A lower-case italic font, *such as this*, is used for numeric and character parameter values which must be supplied by the user. The symbol AIPS refers to the program called AIPS, while the symbol *AIPS* refers to the full system (the AIPS program, numerous other programs, and, the hardware configuration). The symbol `CR` means "hit the RETURN key on the terminal".

In this COOKBOOK, we also use the minimum-matching capability of AIPS to abbreviate the instructions needed to run the programs. This speeds up your activity at the terminal while working in AIPS. The full names of some of the AIPS instructions may, however, be easier to learn and remember. They are given in § 13.



## 2. STARTING UP AIPS

This Section contains general information concerning the steps needed to obtain access to and use an *AIPS* system. It attempts (as does the design and coding of *AIPS* itself) to avoid specific references to particular computer devices and to the peculiarities of any one *AIPS* installation. However, some installation-specific practicalities must always remain. For the NRAO installations, these are described in Appendix Z.

### 2.1. Signing up for AIPS time

Most computer systems cannot support more than a few simultaneous users of *AIPS*. Thus, most locations are obliged to institute a mechanism for distributing the available *AIPS* time to the people desiring it. At NRAO, sign up sheets and rules for their use are normally posted in or near the principal "*AIPS* Caige" (user-terminal room). To promote fair and efficient use of the system, there are often some restrictions on the amounts of time that any one user or user group may reserve.

*AIPS* can support several users, whom it designates *AIPS1*, *AIPS2*, etc. The user designated *AIPS1* has priority for the use of the displays, array processor, and tape drives. In many cases however, the user designated *AIPS2* may carry out similar processing tasks, by verbal negotiation with the *AIPS1* user over the use of these resources. The facilities available through *AIPS3* and higher are more limited and most users will require either *AIPS1* or *AIPS2*.

Your installation may have separate sign-up sheets and terminals for *AIPS1*, *AIPS2*, etc. with rules reflecting these priorities. Consult your local *AIPS* Manager for details.

### 2.2. Using the terminals

The way that a terminal behaves is a function both of the type of terminal and of the computer system to which it is attached. It is important for beginning users to master the foibles of the terminal(s) they will be using. For example, on some terminals/systems CTRL H is used to back over and delete characters which have been typed. Other systems require instead the use of the RUB OUT key or the BACK SPACE key for this purpose. Some systems erase deleted characters while others merely reposition the cursor so that deleted characters may be overwritten. Appendix Z describes these characteristics for the NRAO hardware.

Some computer systems are very friendly to the users. They allow, for example, the XON / XOFF protocol by which the user may temporarily suspend the execution of his programs while he examines the current contents of the terminal display and then resume the execution. They also recognize interrupt commands which, usually with little ceremony, will abort the current program. Such friendly systems also normally allow "type-ahead" by which the user may type the next command line(s) while the current line is being executed.

More expensive terminals are now available with more than one screen's worth of memory. Such terminals allow the user to scroll and/or page back to look at earlier output. We recommend such terminals for *AIPS* since its *HELP* and *INPUTS* files are sometimes longer than the usual number of lines (24) on a terminal screen.

### 2.3. Logging in

Logging in may be the most machine-dependent step in using *AIPS*. Most systems require you to signal your intention to use the computer from a particular terminal in order to assign that terminal to you and to begin any accounting processes. Then, normally, you must ask some batch or job control process in the system to run the *AIPS* program for you. Finally, you must tell the *AIPS* program who you are. Details for the VLA VAXes, the Charlottesville VAX, and the Charlottesville MODCOMP are given in Appendix Z. On other installations, consult your local *AIPS* Manager before logging in.

As you enter the commands needed to log in to your system, please read all messages which appear. They are often important and relate to current system, disk, and *AIPS* problems which may affect your reductions.

After the system messages the question **Enter AIPS version you want to run. NEW, OLD, or NONE:** will appear on those systems which offer a choice. Type in your selection. The OLD version is likely to be relatively free of bugs (provided the *AIPS* version in NEW does not prescribe format changes which prevent OLD from working), but the NEW version will contain the latest improvements. (Note that this choice affects only the version of the *AIPS* program itself. You may choose NEW or OLD versions of the *AIPS* reduction programs at a later time.)

Following this (on systems requiring software tape mounts at job control level) is a message about tape mounting. This is the most convenient time to mount a tape if you will be reading in images from a FITS- or IBM-format tape or UV data from an EXPORT- or FITS-format tape. The tape drive(s) may already be assigned to another terminal, however. Check with other active users if in doubt at this point, then proceed with steps 1 and 2 below if you indeed want to mount a tape. If not, skip to step 2.

1. Physically mount the tape as described in § 2.4 below. You must mount the tape before typing in the tape drive number (the next step).
2. Type `CR` if you do not want to mount a tape.  
Type `n CR` to software-mount the tape drive labeled `n`.

You should then see the message:

**BEGIN THE ONE TRUE AIPS NUMBER `n`**

where `n` is a number between 1 and 6. Check that this "*AIPS* number" is indeed the one for which you signed up. If not, you are probably on the wrong terminal: on many systems *AIPS1* and *AIPS2* are assigned to specific terminals, labeled as such, while other *AIPS* numbers are available on a first-come-first-served basis. *AIPS* will now ask you for your user number and provide a ? prompt:

**AIPS: ENTER USER ID NUMBER**

**? `uuu` C<sub>R</sub>**

where `uuu` is the number assigned to you for the local *AIPS* system. The *AIPS* prompt `>` should now appear.

## 2.4. Hardware tape mount

On some *AIPS* systems, tapes are handled by designated operators. Before mounting tapes read Appendix Z (for NRAO sites) or obtain directions from your local *AIPS* Manager or operators for methods by which tapes are to be handled. Most *AIPS* systems, however, are on the self-service plan. In that case, go to the room in which the tape drives are located and find one which is not in use. Mount the tape physically on the drive following the mounting instructions in Appendix Z or those posted at your installation for the particular kind of tape drive. Don't forget to insert a write ring if you intend to write on the tape or to remove any write ring if you intend only to read the tape. Note the number *n* marked on the drive you are using, as you will need to provide that number to the software for mounts and for executing *AIPS*.

## 2.5. Software tape mount

When you have the tape physically mounted on the tape drive, most computer systems must also be told of your decision. This step is called a "software tape mount". If you are logging in to *AIPS* on such a system, go to step 2 of the login tape mount procedure described in § 2.3 above. If you are logged in to *AIPS* without having mounting a tape, type:

> STORE 1 C <sub>R</sub>	to save your <i>AIPS</i> inputs.
> EXIT C <sub>R</sub>	to exit to monitor level.
\$ @MOUNT <i>n</i> C <sub>R</sub>	to mount the tape in software on the drive labeled <i>n</i> .
\$ @AIPS C <sub>R</sub>	to reactivate <i>AIPS</i> .

The message **Enter the AIPS version you want, NEW or OLD** will appear. Give the same answer you gave when first starting *AIPS*. Then the message **AIPS: ENTER USER ID NUMBER** will appear, followed by a ? prompt. Type your user number, the same as before, followed by C<sub>R</sub>. Then:

> RESTORE 1 C <sub>R</sub>	to recover your <i>AIPS</i> inputs.
----------------------------	-------------------------------------

Note that the two commands above prompted by a \$ are job control commands appropriate to some, but not all, *AIPS* systems. Consult your local *AIPS* Manager if they do not appear to be recognized by your system.

### 3. MAKING YOUR FIRST MAPS

A complete list of the software dealing with UV data can be obtained by typing `HELP UVPR CR` or `HELP MAPETC CR`. This list is also given in § 13 of the *COOKBOOK*.

Maps are made by *AIPS* from UV data on EXPORT tapes prepared by running `EXPREP` and `EXPVIS` on the DEC-10, or from the Charlottesville IBM, or from previous *AIPS* sessions whose UV data sets were saved on tape. UV FITS format tapes written by the IBM, by *AIPS*, or by other systems may also be used. Bring your data tape to the *AIPS* processor and follow the tape mounting instructions in §§ 2.4 and 2.5 above. Once your tape is hardware and software mounted, you have two options:

1. You can write the UV data to disk with `UVL0D`, sort it with `UVSRT` and keep the sorted data set on disk while you make a number of different maps with `UVMAP`. As the sort step is usually the major fraction of the computing time when making maps from large data bases, this option is recommended. Furthermore, self-calibration requires that the data sets reside on disk.
2. You can use `APMAP` to make maps directly from UV data on tape without storing the data on disk. This option is suitable only for small data sets, or when you know you will make only *one* set of maps from the data and thus sort it but once. `APMAP` works only on data tapes in the EXPORT format and does not do shifts and rotations correctly. (It is not being fully maintained by the *AIPS* group and its future is problematical.)

#### 3.1. Making maps with data storage on disk

To make maps and store the UV data on disk, the following sequence of *AIPS* programs is needed:

1. `EXIND` or `PRTTP` to check data on tape and find the number of visibility data points to be processed.
2. `UVL0D` to copy data from tape to disk.
3. `UVSRT` to sort data into proper order for the FFT.
4. `UVMAP` to make the map(s) and beam.

##### 3.1.1. EXIND and PRTTP

The following might be used as a set of inputs to `EXIND`:

- |                         |   |
|-------------------------|---|
| > TASK 'EXIND' ; INP CR | to list the required inputs on the terminal.  |
| > INTAPE <i>n</i> CR    | to specify the use of drive number <i>n</i> —make this specification compatible with your tape mount! |
| > NFILE 3 CR            | to skip 3 files before getting data (default: 0).   |
| > GO CR                 | to run <code>EXIND</code> with these inputs.  |

The tape should begin to move. On the *AIPS* "monitor" (a terminal screen near your primary terminal or, in its absence, your own terminal) a listing of the source name, RA, Dec, frequency and number of visibility points should appear for each source in the file. The *AIPS* prompt > should have already returned

on your terminal, however, since **EXIND** is running as a detached *task*. Tasks are the more complicated *AIPS* programs, run by the **GO** command after setting the task name with **TASK 'taskname'**. They are shed from the terminal, as **EXIND** has been here, allowing you to use *AIPS* for any other processing except running the *same* task at the same time.

To run **EXIND** again, this time on the next file, enter:

> NFILE 0 ; GO C<sub>R</sub>

once **EXIND** has finished on the first file. If you have reset the taskname in the meanwhile, enter:

> NFILE 0 ; GO EXIND C<sub>R</sub>

instead. To get hard copy of the **EXIND** output for later use, follow the instructions in § 4 dealing with the message file.

The program **EXIND** works on only one file at a time and works only on EXPORT format UV data tapes. The program **PRTTP**, however, reads a full tape and prints out a summary of all the UV and map images in any of the supported formats. To use this program, type:

> INTAPE *n* C<sub>R</sub> to specify the tape drive number.

> NFILES 0 C<sub>R</sub> to do all files.

> GO PRTTP C<sub>R</sub> to run the program.

The tape will be rewound and then all files will be read. The print out provides a valuable summary of your data tape.

### 3.1.2. UVLOD

Copying the UV data from tape to disk is accomplished by **UVLOD**. The following shows inputs to **UVLOD** for reading data on 3C138 at C band from the third file on a tape mounted on tape unit number 2:

> TASK 'UVLOD' ; INP C<sub>R</sub> to review the required inputs and set taskname.

> SOURCE '3C138' C<sub>R</sub> to load only 3C138 data.

> BAND 'C' C<sub>R</sub> to load only 6-cm data.

> INTAPE 2 C<sub>R</sub> must be same as tape mount number.

> NFILES 2 C<sub>R</sub> to skip to the third file on the tape.

> OUTN '3C138 A' C<sub>R</sub> to bookkeep that this is A array data (for example).

> OUTC 'UVDATA' C<sub>R</sub>

> OUTDISK 3 C<sub>R</sub> to specify writing to disk 3, e.g. to select a disk with sufficient free space. (See § 4.4 for help in monitoring free disk space).

> NPOINTS 17 C<sub>R</sub> to specify the number of visibility points in thousands (a 20% margin is allowed). Get this number from the **EXIND** output.

> REWIND C<sub>R</sub> to rewind tape before skipping files.

> GO C<sub>R</sub> to run **UVLOD**.

The tape will begin to move and appropriate messages should appear on the *AIPS* monitor. When the prompt > appears on your terminal, you are free to use *AIPS* for other purposes. Only one source and band will be loaded with each run of **UVLOD** as set up here. However, more than one source or band may be loaded under control of additional parameters: type **HELP UVLOD C<sub>R</sub>** for details. **UVLOD** will also read FITS format UV data tapes.

Once UVLOD has finished, check that your disk catalog now contains the UV data you have just tried to load, by:

> INDI 0 ; UCAT CR

which will list all UV data sets in your disk catalog. This list should look something like:

CATALOG ON DISK 3

CAT	USID	MAPNAME	CLASS	SEQ	PT	LAST ACCESS	STAT
32	76	3C138 A	.UVDATA	.	1 UV	22-OCT-1982 12:33:34	

Alternatively, get terminal *and* hard-copy listing of your catalog by:

> INDI 0 ; INTY 'UV' CR                      to list all disks, UV files only.  
 > CATALOG CR                                  to put the catalog listing in message file.  
 > PRTMSG CR                                   to print the message file.

This sequence takes a little longer to execute, but the hard-copy list (sent to the an appropriate printer) may be useful if your catalog is a long one.

Note that the catalog has assigned an ordinal number to the data set in the first (CAT) column of the listing. This number and the disk number (3) should be noted for future reference as they are useful when selecting this data set for further processing.

On systems with a job control level software tape mount, you should deassign the tape drive once you are completely done with tape data. Use:

> STORE 1 CR                                  to save your AIPS inputs.  
 > EXIT CR                                    to exit temporarily.  
 \$ @DMOUNT *n* CR                            to dismount the tape drive labeled *n*.  
 \$ @AIPS CR                                  supply your user number as asked, after this.  
 > RESTORE 1 CR                              to recover your AIPS inputs.

where the two commands above prompted by a \$ are job control commands appropriate to some, but not all, AIPS systems. Consult your local AIPS Manager if they do not appear to be recognized by your system.

### 3.1.3. UVSRT

The next step in map-making is to sort the data for the FFT. This is done with UVSRT:

> TASK 'UVSRT' ; INP CR                    to set the taskname and list the input parameters.  
 > INDI *n* ; GETN *ctn* CR                   where *n* is the disk number with the UV data and *ctn* is its catalog number on that disk. (*n* = 3 and *ctn* = 32 from our UCAT example).  
 > OUTN INNA ; OUTCL 'UVSRT' CR           to set the output file name and class.  
 > SORT 'XY' CR                            to select the sort type required for map making.  
 > INP CR                                   to review the inputs you have selected. *N.B.* check them carefully since the sort can be a very time-consuming step for large data sets.  
 > GO CR                                    to run the task UVSRT.

On a Vax 780 under VMS, UVSRT will take about 15 minutes to sort a data set with 100,000 visibility points (longer if the machine is being heavily used) and will need two scratch files of about 15,000 blocks each. If the disks are filled, the job may get terminated. Check in § 4.4 for things to do if this happens.

Once **UVSRT** has finished, check that a UV data base with the "class" **UVSRT** has appeared in your disk catalog by:

```
> INDI 0 ; UCAT CR
```

The catalog listing might now look like:

```
CATALOG ON DISK 3
CAT USID MAPNAME      CLASS  SEQ  PT    LAST ACCESS      STAT
 32   76 3C138 A      .UVDATA .   1 UV 22-OCT-1982 12:33:34
 35   76 3C138 A      .UVSRT  .   1 UV 22-OCT-1982 12:56:50
```

Note that the catalog number of the sorted file is not contiguous with that of the unsorted—slots 33 and 34 are already occupied. This situation may arise in *AIPS* systems which are configured to have "public" catalog files. A number of *AIPS* installations, including the VAX systems at the VLA and Charlottesville, now have "private" catalog files. On those systems, your UV files will have contiguous catalog numbers starting from 1 when you first write UV data to disk.

Once the sorted data base is made, please delete the unsorted data unless you plan to self-calibrate your data. Data bases can be very large and there is no need to keep the unsorted data on disk since it can be reloaded quickly with **UVLOAD**. To do the deletion:

```
> INDI n ; GETN ctn CR
```

where *n* and *ctn* select the data set to be deleted. (*n* = 3, *ctn* = 32 here.)

**AIPS** will return the name of the selected data set. Check this, then:

```
> ZAP CR
```

will do the deletion.

### 3.1.4. UVMAP

The task which does the gridding and Fourier transform of the sorted data is **UVMAP**. Examples of the inputs follow:

```
> TASK 'UVMAP' ; INP CR
```

tells you what you have to specify.

```
> INDI n ; GETN ctn
```

to select the *sorted* data set (*n* = 3, *ctn* = 35 here).

```
> STOKES 'IQU'
```

to make a Beam, I, Q, and U maps of source (the default 'I' makes only an I map and beam).

*N.B.* The default **STOKES 'I'** will select UV points where LL and RR data are present *and* points where only LL or only RR are present. **STOKES 'IQU'** requires both LR and RL to be present and either LL or RR or both. As these criteria are not identical, beams output with class **IBEM** from **STOKES 'I'** may not be identical to those output as **IBEM** from **STOKES 'IQU'**, and you may wish to distinguish them later for **CLEAN**ing purposes, using the **RENAME** command (see § 4.2 on the Catalog File).

```
> IMSIZE 512, 256 CR
```

to make a 512 by 256 map—**IMSIZE** parameters must be powers of 2 from 32 to 2048.

```
> CELL 0.5 CR
```

to select 0.5 arc sec cell size (i.e. 256 by 128 arcsec field of view with the 512 by 256 **IMSIZE** specified above).

```
> UVTAPER 50, 40 CR
```

to specify the widths to 30% of the Gaussian taper in *x* and *y*, in units of kilowavelengths (i.e. the **UVTAPER** specified here is 30% at 50000 and 40000 wavelengths). Default: no **UVTAPER**ing.

```
> UVRANGE 10, 100 CR
```

to restrict the projected baselines used to the range 10000 to 100000 wavelengths. Default: no baseline restriction.

```
> UVWTFN 'NA' CR
```

to specify "natural" weighting. The default, ' ', specifies "uniform" weighting.

Other inputs, as well as the last 3, are defaulted sensibly. Use:

> HELP *xxx* *CR*

where *xxx* is a parameter name to get useful information on what they specify. The default UV convolution function is  $\text{sinc} \times \exp$  (XTYPE, YTYPE = 4) and has good characteristics for suppressing aliasing. Check that you are satisfied with the inputs by:

> INP *CR*

then:

> GO *CR* will run **UVMAP**.

The *AIPS* monitor provides some important messages while **UVMAP** is running. When you see **UVMAP: SUCCESSFUL COMPLETION** on this monitor, you should find the requested maps in your catalog using:

> INDI 0 ; MCAT *CR*

This would produce a listing such as:

CATALOG ON DISK 2									
CAT	USID	MAPNAME	CLASS	SEQ	PT	LAST ACCESS		STAT	
42	76	3C138 A	.IMAP	.	1	MA	22-OCT-1981	13:50:10	
43	76	3C138 A	.IBEM	.	1	MA	22-OCT-1981	13:59:58	
44	76	3C138 A	.QMAP	.	1	MA	22-OCT-1981	14:10:10	
45	76	3C138 A	.UMAP	.	1	MA	22-OCT-1981	14:19:19	

To get a hard copy listing of the catalog type:

> INDI 0 ; CLRN *CR* to clear file naming adverbs.

> INTY 'MA' ; CAT ; PRTM *CR* to catalog maps and print.

The **UVMAP** messages which contain vital parameters of your mapping task are logged in a "history" file that is cataloged on disk as an extension of your output map file. This history file can be printed out by:

> INDI *n* ; GETN *ctn* ; PRTHI *CR* where *n* and *ctn* are the disk and catalog number of the desired file (*n* = 3, *ctn* = 42 here would select the I map).

### 3.2. Making maps directly from tape—APMAP

Maps can be made directly from the EXPORT tape using the task **APMAP**. The inputs for **APMAP** are very similar to those above for **UVMAP** with the addition of a couple from **UVL0D**. The use of **APMAP** is not advised unless the data base is small (*e.g.* VLA "snapshots"). This program has not been maintained as well as **UVMAP**. Do not shift or rotate the maps. Check your output carefully:

> TASK 'APMAP'; INP *CR* to review the required inputs.

Sample inputs for **APMAP**:

> SOURCE = '1200+519' to select source 1200+519 from the data on tape.

> INTAPE = *n* *CR* to specify the tape unit to be used as that labeled *n*.

> NFILES 1 *CR* to skip one file before reading the tape data.

> BAND 'C' to select 6cm data; default is 'L'.

The other inputs are like those of **UVMAP** above. Use

> INP *CR* to review them.

> GO *CR* to run **APMAP** when you are satisfied with the inputs.

> REWIND *CR* to rewind your tape *when APMAP and you are done*.



## 4. SOME IMPORTANT UTILITIES

Before going on to further map processing, you should become more familiar with a few *AIPS* and system utilities, some of which have already been encountered in the map-making steps of §3 above. Lists of available utilities can be obtained by typing `HELP CATINFO CR` and `HELP GENERAL CR`. See also §13.

### 4.1. The Message file

Many *AIPS* programs will produce messages on the *AIPS* monitor—the black-and-white alphanumeric screen near your terminal or, in its absence, your terminal itself. The messages that fly by on this monitor are also written to disk in a file called the message file. The disk message file can be printed out by typing `PRTMSG CR` following a `>` prompt at your terminal. `PRTMSG` is an example of an *AIPS* “verb”. Unlike an *AIPS* “task”, it does not need a `GO` in order to execute and it is *not* shed from your terminal. Because they tie up the terminal while executing, “verbs” are generally simple programs which do not take long to execute.

Each entry in the message file has an associated priority: 0 for user input, 2 for “unimportant” messages, 5 for general output messages, and 8 for serious error messages. To set the priority level for hard copy output of messages type:

```
> PRIORITY np CR
```

where *np* is the desired minimum level.

before running `PRTMSG`; then only messages at this level or above will be listed on the printer. All messages are deleted from the file after they are printed. The file is also automatically printed (with user-specified `PRIORITY`) and cleared when you `EXIT` from *AIPS*. To clear your message file with no printing, set:

```
> PRIORITY 11 CR
```

and then type:

```
> PRTMSG CR
```

### 4.2. The Catalog file

A summary record of your *AIPS* data sets (UV data, maps, beams) is kept in a disk file called the catalog file. To interrogate your disk catalog, use:

```
> INDI 0 ; MCAT CR
```

to list all maps.

```
> INDI 0 ; UCAT CR
```

to list all UV data sets.

A complete listing of the catalog file, which may be printed with `PRTMSG`, can be generated by:

```
> CLRNAME CR
```

to reset adverbs `INNAME`, `INCLASS`, and `INSEQ`.

```
> INDI 0 ; INTY ' ' CR
```

to get all disks and types.

```
> CAT CR
```

to generate the listing.

which will list all of your disk data sets. To limit the listing to a particular name, class, sequence number, type, and/or disk, use an appropriate combination of the parameters `INNAME`, `INCLASS`, `INSEQ`, `INTYPE`, and `INDISK`, respectively. Unless a hard copy or a limited part of the catalog is desired, it is faster to use `MCAT` and `UCAT`. A typical listing looks like:

```

CATALOG ON DISK 1
CAT USID MAPNAME      CLASS  SEQ PT    LAST ACCESS    STAT
 18   76 1665KM.MIL    .IMAP  .   1 MA 27-FEB-1982 22:30:18
 19   76 1665KM.BIL    .IBEM  .   1 MA 27-FEB-1982 23:02:14
 22   76 1665KM.MIL    .IMAP  .   2 MA 28-FEB-1982 15:30:45
CATALOG ON DISK 2
CAT USID MAPNAME      CLASS  SEQ PT    LAST ACCESS    STAT
 22   76 1200+519      .IMAP  .   1 MA 01-MAR-1982 23:50:10 READ
 23   76 1200+519      .IBEM  .   1 MA 01-MAR-1982 23:59:58 READ
 24   76 1200+519      .QMAP  .   1 MA 28-FEB-1982 00:10:10
 25   76 1200+519      .UMAP  .   1 MA 28-FEB-1982 00:19:19
 28   76 1200+519      .ICLN  .   1 MA 02-MAR-1982 00:35:20 WRIT
CATALOG ON DISK 3
CAT USID MAPNAME      CLASS  SEQ PT    LAST ACCESS    STAT
 35   76 3C138 A       .UVSRT .   1 UV 22-FEB-1982 12:56:50

```

This user (number 76) has eight image files, three on disk 1 and six on disk 2. He also has a sorted UV data set on disk 3. Image data files (maps and beams) are distinguished by the type code **MA** in this list. The UV data file is distinguished by the type code **UV**.

Note that this user has given the image files on disk 1 VLA 11/70 style file names embedded in their *AIPS* mapnames. Note how **SEQ** numbers are used to distinguish different versions of a file with the same name. Careful choice of filenames for output from **UVMAP** or **APMAP** can save you a lot of other bookkeeping. The mapname can be any valid string up to 12 characters in length.

#### 4.2.1. Renaming data files

Files may also be renamed, after they have been cataloged, using the *AIPS* verb **RENAME**. Typical inputs might be:

```

> INDI 2 ; INNA '1200+519' CR           to set input (old) name.
> INCL 'IMAP' ; INSEQ 1 CR
> OUTN '1200+519LOW' ; OUTSEQ 2 CR      to set desired name.
> INP RENAME CR                       to review the inputs to the verb.
> RENAME CR                           to rename the I map to 1200+519LOW and reset its sequence
                                         number to 2.

```

#### 4.2.2. Speedy data file selection

Associated with each catalog entry is an identification number called the "catalog slot number". The **CAT** listing in the column at the left shows these catalog numbers. They can be used conveniently to initialize inputs for *AIPS* programs which read data sets from disk. Use:

```

> INDI n1 ; GETN ctn1 CR             where n1 selects the disk and ctn1 is the catalog slot number.
The verb GETNAME (abbreviated through minimum match as GETN above) sets the parameters INNAME,
INCLASS, INSEQ, and INTYPE used by many tasks and verbs. Some tasks require a second and even a third
set of input image name parameters. For these, use:
> IN2D n2 ; GET2N ctn2 CR           to set the second set.
> IN3D n3 ; GET3N ctn3 CR           to set the third set.

```

### 4.2.3. Catalog entry status

Note that catalog slots 22, 23 and 28 on disk 2 in our example above do not have blank entries in the **STAT** column. This listing could have been taken while the user was running a **CLEAN** on the **IMAP** in slot 22 using the **IBEM** in slot 23, as these files are opened for **READING**. The **CLEAN** map file 28 **ICLN** is opened for **WRITing**. Procedures which attempt to read files which are opened for writing, or vice versa, will be rejected with appropriate error messages, so you should note any non-blank entries in the **STAT** column carefully. In some situations, involving abortion of tasks, files may be left in **READ** or **WRIT** status indefinitely. The file status may be reset in such cases by use of **CLRSTAT C<sub>R</sub>** following the appropriate **INDISK** and **GETNAME**.

Check that a data set you are about to use has a clear status before running any **AIPS** programs that will use it. (It is often possible to let two tasks read the same file at the same time, but this is not recommended.) Also note its ordinal number in the catalog, as this will be useful.

### 4.2.4. Other catalog listings

You can get a terminal listing of the image header file by following the **GETNAME** step above with

> **IMHEAD C<sub>R</sub>**

The output of this verb may also be printed via **PRTMSG**.

### 4.3. SAVE, GET, TPUT, and TGET

All input parameters ("adverbs") are global throughout **AIPS**. When a parameter value is specified for any program, it will remain at that value in any other program which has an input of the same name until you change it. This global nature of the input parameters is useful in some cases, but can be inconvenient in others. Before running any task or verb check the input parameters carefully with:

> **INP *xxxx* C<sub>R</sub>**                                where *xxxx* is the program name.

The inputs you have specified for **AIPS** at any given time may be saved on disk by typing:

> **SAVE *aaaa* C<sub>R</sub>**                                where *aaaa* is any string of up to 12 characters.

> **GET *aaaa* C<sub>R</sub>**                                will restore these inputs at a later time.

These commands save or restore all the inputs and the rest of your **AIPS** "environment". For this reason, they must be the only commands on the input line.

The input parameter values associated with a task or a verb can be stored by the command:

> **TPUT *taskname* C<sub>R</sub>**                                where *taskname* is the verb or task name.

and retrieved by the command:

> **TGET *taskname* C<sub>R</sub>**

This avoids to some extent the global nature of the adverb values in **AIPS**. Whenever a task is executed (by the verb **GO**), **TPUT** is automatically run. **TGET** will therefore produce the set of input parameters used for the last execution of the task, unless you deliberately overwrite them with a **TPUT** of your own.

#### 4.4. Monitoring the disk space

Disk space is generally at a premium and there are several ways to monitor the available amount of space. Within AIPS there is:

> USER 32000 ; DISKUSE C<sub>R</sub>

This will (eventually) list on your terminal the rough amount of data space in use by each user for all AIPS disks. Identify the worst disk hogs and apply appropriate peer pressure.

Sometimes the available disk space has been eaten up by AIPS scratch files which are no longer being used. Tasks which abort and other mysterious events can cause this situation to arise. To delete the scratch files from a specific task, type:

> SPY C <sub>R</sub>	to see if the task is running.
> TASK 'taskname' C <sub>R</sub>	to set the required adverb.
> SCRD C <sub>R</sub>	to delete the files.

This verb will only work if no one is running the specified task at the moment. Useful *taskname* values seem to be APCLN, UVMAP, and ASCAL.

The verb TIMDEST will destroy all data sets which have not been used in some minimum time interval. In standard versions of AIPS, this time interval is 14 days. Your local AIPS Manager may select some other limit, however.

At present, all other methods for monitoring disk space involve system-dependent commands of one sort or another. § 10 of this COOKBOOK tells you how to backup or delete your own data to relieve disk crowding. To use these methods:

> STORE 1 C <sub>R</sub>	to save your AIPS inputs.
> EXIT C <sub>R</sub>	to exit from AIPS.

Then consult Appendix Z for instructions appropriate to your machine.

#### 4.5. Finding information in AIPS

Much of the documentation of AIPS can be displayed on the terminals by typing `HELP word`, where *word* can be a verb, task or adverb. The information given will supplement that given in the INPUTS for a verb or task. It is the only source of information on the adverbs.

There are more general help files which list a variety of information. Some illustrate the use of features in AIPS (try `HELP BATCHJOB CR` for a description of the batch processor). Others list the currently working software (e.g. `HELP ANALYSIS CR` gives a listing of software which analyzes images). These files should be consulted often. All of these general `HELP` files are listed in § 13 of this COOKBOOK and give a summary of the software which is currently available in AIPS. They are updated as new software is developed. `HELP` has other capabilities as well. Type:

> HELP C <sub>R</sub>	or
> HELP HELP C <sub>R</sub>	to get a complete summary.

## 5. IMPROVING MAPS - CLEAN, SELFAL, EDITING

This Section deals with the major programs which can be used to improve the quality of dirty map images which are not noise-limited but which suffer from dynamic range problems. See Lectures 9-13 in the *NRAO-VLA Workshop on Synthesis Mapping* for general discussion and guidance about image defects and the strategies for reducing them.

### 5.1. CLEANing maps

The CLEAN implemented in AIPS is an array-processor CLEAN of the type devised for the PDP11/70 by Barry Clark (*Astron. Astrophys.*, **89**, 355 (1980)). CLEAN components are found during "minor" iteration cycles by CLEANing the brightest parts of the residual map in the array processor with a "beam patch" of limited size. More precise CLEANing is achieved at the ends of "major" iteration cycles when the Fourier transform of the CLEAN components is computed, multiplied by the transform of the beam, transformed back to the map plane, and then subtracted from the dirty map. The rule for deciding when a major iteration should end in order to achieve a desired accuracy is not well defined (see the Clark paper).

The APCLN task allows the user to vary the rule somewhat to suit the requirements of his map and to control all the usual options associated with CLEANing. Enter:

- > TASK 'APCLN' ; INP C<sub>R</sub>                      to tell you what you have to specify.
  - > INDI *n1* ; GETN *ctn1* C<sub>R</sub>                      where *n1* is the disk on which the dirty map is stored and *ctn1* is its catalog number there.
  - > IN2D *n2* ; GET2N *ctn2* C<sub>R</sub>                      where *n2* and *ctn2* here select the dirty beam.
- Examples of other inputs to APCLN are:
- > OUTN 'M84LOW' C<sub>R</sub>                      CLEAN map name - default: same as INNAME.
  - > OUTS 0 C<sub>R</sub>                                      to create a new output file. If OUTSEQ  $\neq$  0, then the specified value is used. This parameter must be set to restart a CLEAN.
  - > OUTC 'ICLN1' C<sub>R</sub>                              defaults to ICLN for INCLASS IMAP, QCLN for QMAP, UCLN for UMAP, and VCLN for VMAP.
  - > GAIN 0.2 C<sub>R</sub>                                      loop gain parameter, defaults to 0.1. Values in the range 0.15 to 0.25 may be suitable for complex sources, higher values are ok for simple sources.
  - > FLUX 0.003 C<sub>R</sub>                                      to stop CLEANing when peak of residual map falls to 3 mJy.
  - > NITER 500 C<sub>R</sub>                                      to stop CLEANing when 500 components have been subtracted (default 100).
  - > BMAJ 1.25 C<sub>R</sub>                                      to set the major-axis FWHM of restoring beam to 1.25 arcsec. 0 here specifies that beam is to be fitted. A negative number means that the *residual* map is to be stored in the output file.
  - > BMIN 1.10 C<sub>R</sub>                                      to set the minor-axis half-width of restoring beam to 1.10 arcsec.
  - > BPA 52.8 C<sub>R</sub>                                      to set the position angle of beam axis to 52°48' measured CCW from North.
  - > INVERS 1 C<sub>R</sub>                                      to specify which version of the CLEAN components file(s) is desired. Default is 1.

- > NBOXES 1  $\mathcal{C}_R$  to set the number of boxes in which to search for CLEAN components. Must be  $\leq 10$ , default is 1.
- > BOX 156, 156, 356, 356  $\mathcal{C}_R$  CLEANing box is 200 x 200 centered on pixel (256, 256). Default 0, 0, 0, 0 specifies inner quarter of map. Fill the array with more windows if NBOXES > 1.

The TV cursor can be used to set the CLEANing boxes if the TV display shows either the dirty map or a previous version of the CLEAN map. Type:

- > NBOXES ( $n$ )  $\mathcal{C}_R$  to set the number of boxes.
- > TVBOX  $\mathcal{C}_R$  to begin an interactive, graphical setting of the  $n$  boxes.

Position the TV cursor at the bottom left corner of the first CLEANing box and press a trackball button. Then position the cursor at the top right corner of the box and press Button B. Repeat for all desired boxes. This will fill the box array for APCLN. Note that the terminal will display some additional instructions. These may be followed to reset any of the previously set corners should you need to do so.

To continue the example parameters:

- > MINPAT 51  $\mathcal{C}_R$  to set the minimum half-width of the beam patch (in pixels) allowed in a minor cycle. Default = 21, which may be too small for accurate CLEANing of complex maps.
- > DOTV TRUE  $\mathcal{C}_R$  to show residual maps on TV in interactive AIPS. The residual map is displayed on the TV after each major cycle and the CLEAN map is displayed at end, if DOTV is set to TRUE (+1.0). Default = FALSE (-1.0) for no display. (See also below.)
- > INP  $\mathcal{C}_R$  to review what you have specified.

Note that specifying INNAME and (usually) NITER may be enough to run the program if your dirty map and beam have the same image mapname (differing only by their classnames, e.g. IMAF and IBEM) and if you are content to use the system defaults for the CLEANing parameters. These defaults are reasonably chosen for CLEANing a complicated image.

The FACTOR parameter in APCLN can be used to speed up or to slow down the CLEANing process by increasing or decreasing the number of minor cycles in the major cycles. The default FACTOR 0 causes major cycles to be ended using Barry Clark's original criterion (as in the 11/70 CLEAN). Setting FACTOR in the range 0 to +1.0 will speed up the CLEAN, by up to 20% for FACTOR 1.0, at the risk of poorer representation of extended structure. Setting FACTOR in the range 0 to -1.0 will slow it down, but gives better representation of extended structure. For further details on the use of non-standard FACTORS, consult with Bill Cotton in Charlottesville.

When you're satisfied with the inputs listed by INP, then:

- > GO  $\mathcal{C}_R$  starts the CLEAN.

The AIPS monitor displays useful information about the progress of the CLEANing, and the TV itself displays the residual maps after each major cycle (if DOTV was set to TRUE. You may change the image display parameters (e.g. coloring, transfer function, zoom, etc.) using the display modification commands while the CLEAN is in progress to keep the residual map looking intelligible. When using the TV, APCLN schedules 15-second pauses after each major cycle. The new residual map will be displayed on the TV monitor at these times. Pressing trackball button D during the 15 seconds will end APCLN in an orderly fashion at that point. Pressing any of buttons A, B, or C causes the execution to resume (as does waiting 15 seconds). The AIPS monitor prompts you for this action, which allows quasi-interactive CLEANing. Its use is encouraged when maps are being CLEANed for the first time.

By default, **APCLN** will fit a Gaussian **CLEAN** beam to the central portion of the dirty beam. The results may not be desirable since the central portions of many dirty beams are not well represented by a single Gaussian and since the present fitting algorithm is not very elaborate. If you use the default, check that the **CLEAN** beam is an accurate representation of the dirty beam. Use task **PRTIM** on the central part of the dirty beam for this purpose.

When **APCLN** terminates, a record of the **CLEANed** map is entered into your disk catalog and the map can then be displayed, contoured, etc. as described in §§ 6 and 7 below. The most important parameters of the **CLEAN** are logged in a "history" file that is cataloged as an extension of the **CLEANed** map file. This history file can be printed by:

```
> INDI n ; GETN ctn CR           where n and ctn select the disk and catalog numbers of the
                                     CLEAN map.
> PRTHI CR
```

### 5.1.1. Restarting **CLEANs**

**CLEANing** can be restarted with the parameter **BITER** in the **APCLN** inputs. Set **BITER** equal to the number of components to be used from the previous **CLEAN** when **CLEANing** is restarted. When you are restarting a **CLEAN**, the **OUTNAME** and **OUTSEQ** parameters *must* be set explicitly to those of the previous **CLEAN** map since the **CLEAN** component list is associated with this map file (as its **CC** extension). A map can be convolved by setting **NITER** = **BITER** and specifying the desired (new) **CLEAN** beam. The previous **CLEAN** map will be overwritten by the new one.

### 5.1.2. Listing **CLEAN** components

The list of **CLEAN** components associated with a **CLEAN** map can be printed by:

```
> INDI n ; GETN ctn CR           where n and ctn select the disk and catalog numbers of the
                                     CLEAN map.
> BITER n1 ; NITER n2 ; XINC n3 CR   to list CLEAN components from n1 to (n1 + n2) with incre-
                                     ment n3.
> GO PRTCC CR                       to execute the task.
```

## 5.2. Self-calibration

The task **ASCAL** is a tool for obtaining maps with high dynamic range when there is sufficient signal to noise. **ASCAL** takes two input files, the UV data in time-baseline order (i.e. **SORT** 'TB', not the file sorted for mapping) and a **CLEANed** map. The **CLEAN** components of this map are used to generate the input source model needed for the self-calibration process. The output file is a recalibrated UV data base and will need to be sorted (**SORT** 'XY') before running **UVMAP**.

If a single point source can be used as the input source model, then **ASCAL** need not be given a **CLEAN** map and the parameters for the model point source can be supplied directly. Do not use **ASCAL** unless there is sufficient signal to noise in your data to warrant improvement. If your maps are limited by receiver noise, then self-calibration may produce erroneous results. See Lecture 13 by Tim Cornwell in the *NRAO-VLA Workshop on Synthesis Mapping* for guidance on whether to use self-calibration.

The following input parameters are used by **ASCAL**:

> INDI <i>n1</i> ; GETN <i>ctn1</i> C <sub>R</sub>	to select the 'TB' sorted UV data base.
> IN2D <i>n2</i> ; GET2N <i>ctn2</i> C <sub>R</sub>	to select the CLEAN map to use.
> NITER = <i>n</i> C <sub>R</sub>	to cut off the model at the <i>n</i> th CLEAN component associated with this map. If <i>n</i> < 0, then use up to ABS( <i>n</i> ) components, but stop at the first negative component.
> DOCAT 1 C <sub>R</sub>	to catalog the gain file as an extension to the input UV data base. This is necessary to self-cal a spectral line data base and to display the gain solutions.
> APARM(1) = <i>x1</i> C <sub>R</sub>	minimum baseline (in kilowavelengths) to be given full weight.
> APARM(2) = <i>x2</i> C <sub>R</sub>	maximum baseline (in kilowavelengths) to be given full weight.
> APARM(3) = <i>x3</i> C <sub>R</sub>	weight for baselines outside the APARM(1) ⇒ APARM(2) range.
> APARM(4) = <i>n4</i> C <sub>R</sub>	reference antenna (choose a known good one for best results).
> APARM(5) = <i>n5</i> C <sub>R</sub>	minimum number of antennas to be present for a solution.
> APARM(6) = 1 C <sub>R</sub>	print out the details of the solution: <i>a lot!!</i>
> APARM(6) = 0 C <sub>R</sub>	print out only a summary of the solution: <i>usually enough</i> .
> APARM(7) = 1 C <sub>R</sub>	means solve for amplitude and phase corrections.
> APARM(7) = 2 C <sub>R</sub>	means solve for phase weighted by amplitude.
> APARM(7) = 3 C <sub>R</sub>	means solve for phase ignoring amplitude.
> APARM(8) = 0 C <sub>R</sub>	means solve for RR and LL separately.
> APARM(8) = 1 C <sub>R</sub>	means average RR and LL correlators before solving.
> APARM(9) = <i>x9</i> C <sub>R</sub>	length of the solution interval (in minutes).
> APARM(10) = 0 C <sub>R</sub>	when APARM(7) = 1, scale the gain corrections by the mean modulus of all gains to keep the flux density scale from drifting.
> APARM(10) = 1 C <sub>R</sub>	do not apply any gain normalization.
> BPARAM(1) = 0 C <sub>R</sub>	use CLEAN components as a model.
> BPARAM(1) = 1 C <sub>R</sub>	use a single point source model instead.
> BPARAM(2) = <i>y2</i> C <sub>R</sub>	flux density of point source model (if BPARAM(1) = 1).
> BPARAM(3) = <i>y3</i> C <sub>R</sub>	east offset of point source model in arcsec (if BPARAM(1) = 1).
> BPARAM(4) = <i>y4</i> C <sub>R</sub>	north offset of point source model in arcsec (if BPARAM(1) = 1).
> BPARAM(6) = 0 C <sub>R</sub>	use a least squares gain solution routine.
> BPARAM(6) = 1 C <sub>R</sub>	use an experimental L1 minimization routine which is rather slower, but much less sensitive to wild data values. (Consult Fred Schwab in Charlottesville for details.)
> BPARAM(7) = 0 C <sub>R</sub>	do no editing of data.
> BPARAM(7) = 1 C <sub>R</sub>	to flag "bad" data under control of BPARAM(8) and BPARAM(9).
> BPARAM(8) = <i>y8</i> C <sub>R</sub>	if BPARAM(7) = 1, flag full weight points that differ from the model by BPARAM(8) times the rms residual.
> BPARAM(9) = <i>y9</i> C <sub>R</sub>	if BPARAM(7) = 1, calculate the antenna-IF running mean amplitude correction and flag an antenna-IF if that correction is more than BPARAM(9) × σ away from the mean.

Note that the parameter **BPARAM(5)** is not used.



### 5.3. Editing UV data

There are many programs which aid in the processing, display and editing of UV data. This software is listed by:

> HELP UVPR  $C_R$

and in § 13 of the *COOKBOOK*. In particular, there are facilities in **ASCAL** and **CLIP** to flag UV data in *AIPS* based on deviations from specified norms. There is also a task, **UVFLG**, which allows flagging and unflagging by antenna-IF or by correlator. Type **HELP ASCAL**, **HELP CLIP**, or **HELP UVFLG** for details. The task **UVPLT** plots various combinations of UV data—type **HELP UVPLT  $C_R$**  for details. The task **UVFND** is also recommended for printing out suspicious portions of the data base.

**CLIP** is a very useful task for flagging data on the basis of its discrepancies from the visibility predicted by a set of **CLEAN** components. The task **UVSUB** can be used to subtract the Fourier transform of the **CLEAN** components from the visibility data. **UVPLT** enables the residuals to be displayed and **CLIP** can be used to flag abnormally high points. Cautious users may prefer to run **UVFND** to display such points before running the automatic **CLIP** task. Of course, after flagging, **UVSUB** should be run again to add the transform of the **CLEAN** components back into the data (using the input **FACTOR** =  $-1.0$ ).

## 6. READING AND DISPLAYING IMAGES

In many cases, images will be made outside of *AIPS* (e.g. on the VLA 11/70 mapping system or the Charlottesville IBM system). Such images are transported to *AIPS* by writing them on magnetic tape using the standard FITS format.

A listing of the software associated with TV displays can be found by listing the **HELP** files **CURSOR**, **TVGEN**, **TVINTER**, and **TVCOLOR**. These files are reproduced in § 13 of the *COOKBOOK*.

### 6.1. Loading an image from tape to disk

The task **IML** is used to load images from tape to disk. It can read images in either the standard FITS format (which has one image per tape file) or in the CV-IBM map format (which may have more than one image per file). **IML** assumes that the tape is positioned at the beginning of the image. To move the tape forward or backward by a specified number of file marks, enter:

- > **INTAPE** *n* **CR**                                   to specify the tape drive labeled *n*.
- > **NFILES** *nf* **CR**                                   to specify the number of file marks to move the tape.
- > **AVFILE** **CR**                                       to move the tape.

If *nf* > 0, **AVFILE** will advance the tape the specified number of file marks. If *nf* = 0, the tape is moved backward to the beginning of the current file. If *nf* < 0, the tape is moved backwards to the abs(*nf*) previous file. In all cases, the tape is left at the beginning of a file. With a CV-IBM format tape, the verb **AVMAP** may also be needed. Type **HELP AVMAP** for details.

To make sure that the tape is positioned correctly, type:

- > **TPHEAD** **CR**

This verb will display on the terminal information about the image header at which the tape is positioned. The tape position is not altered. Once the tape has been positioned at the desired image, enter:

- > **OUTNAME** '*your-chosen-name*' **CR**                   to specify the output disk file name in *AIPS*.
- > **GO IML** **CR**                                       to run the task.

The string *your-chosen-name* can be any (< 12-character) title that you want to use as the image name within *AIPS*. To cross-reference your bookkeeping on multiple computers, it is often useful to specify the name of the mapfile and its extension (e.g. **166unt.m11**) from the previous machine (e.g. the VLA 11/70 system). *AIPS* provides equivalents of the DEC-10 extensions, which it calls the image "Class". **IML** allows you to specify this 6-character parameter, **OUTCLASS** '*abcdef*' **CR**, as well.

If **OUTNAME** is unspecified, it defaults to the "name" of the map read from the FITS header—either the name previously used in *AIPS* or the source name. If **OUTCLASS** is unspecified, it defaults to the Class previously used in *AIPS* or to a compound name (e.g. **IMAP**, **IBEM**, **QMAP**, **ICLN**) which attempts to describe the image. You can edit the *AIPS* map and class names later using **RENAME** (see § 4.2.1 of the *COOKBOOK*).

To load further images from the same tape in sequence:

- > **OUTNAME** '*your-chosen-name*' **CR**                   to set a new name.
- > **GO IML** **CR**                                       to run the task again.

To load *n* consecutive images from tape using the system default **OUTNAME** (the name from the FITS header):

- > **OUTNAME** '' **CR**                                   to specify the system default is wanted.

---

> TASK 'IMLOD' C<sub>R</sub>                                   to set task name for GO verb.  
 > DOWAIT TRUE C<sub>R</sub>                                   to tell AIPS to continue only after the task has finished.  
 > FOR I = 1 TO *n* ; PRINT I ; GO ; END C<sub>R</sub>                                   to loop for *n* executions of IMLOD.  
 To rewind the tape before unloading:  
 > REWIND C<sub>R</sub>

### 6.2. Loading an image from disk to the TV display

To select an image file for display you must assign a number of input parameters to the values carried by the chosen file. If you know the disk and catalog numbers of the file from an MCAT listing, you can initialize all of the name parameters together with the instruction:

> INDI *n* ; GETN *ctn* C<sub>R</sub>                                   where *n* and *ctn* select the required map.

Alternatively, you may specify the parameters individually.

The verb TVALL incorporates most of the major display functions and will be described here. See § 13 for a complete listing of available display verbs.

You should use one of the following commands to specify the initial transfer function converting your image file intensities to display pixel intensities (the slopes and intercepts of the display transfer functions can be modified later, but you have an initial choice between linear, logarithmic, and negative linear displays and a choice of the range of intensities loaded):

> FUNC 'LN' C<sub>R</sub>                                   linear—this is the default.  
 > FUNC 'LG' C<sub>R</sub>                                   logarithmic.  
 > FUNC 'NE' C<sub>R</sub>                                   negative linear.  
 > PIXRA *x1*, *x2* C<sub>R</sub>                                   to load only image intensities *x1* to *x2* in the units of the image.  
   The default is to load the full range of intensities in the image.

Then

> TVALL C<sub>R</sub>                                   to load the selected image.

The image should appear on the TV screen in black-and-white. If you don't get what you expected, hit button D to get out of TVALL and review your inputs to the loading routine with:

> INP TVALL C<sub>R</sub>

If *no* image was loaded, check which image channel (TVCHAN) is specified by your inputs. There are several image channels available on most TV display systems (*e.g.* 4 on the Charlottesville I<sup>2</sup>S systems and 3 on the VLA I<sup>2</sup>S systems). Try:

> TVON C<sub>R</sub>                                   to turn on channel TVCHAN.

If this still does not display your image, call an AIPS Manager. If a labeled wedge does not appear on the display, you should also call the Manager, as the display may need adjustment.

After your image has been displayed on the TV by TVALL, the trackball and its buttons (labeled A, B, C, and D) can be used to modify the display. Pressing button A will enable black-and-white and color-contour coding of the image intensities, successively. Adjustment of the cursor position on the TV (using the trackball) will vary the slope and intercept of the display transfer function. TVALL will superpose a calibrated horizontal wedge on the image. This should help you choose the optimum cursor setting for the display.

The I<sup>2</sup>S display can show images up to 512 by 512 in size. If the image is larger than about 424 pixels in the *y*-direction, portions of the labeling of the wedge will be omitted (the units) or superposed on top of the wedge (the tick numeric values). Other television systems will have comparable, but not necessarily identical numeric limits. A useful technique for displaying large images is to load only alternate pixels. The command:

> TXINC 2 ; TYINC 2 C<sub>R</sub>                      to load every other *x* and *y* pixel.

before TVALL would do this. Also use:

> TBLC = *n1, n2, n3, ...* C<sub>R</sub>                      bottom left pixel to load.

> TTRC = *m1, m2, m3, ...* C<sub>R</sub>                      top right pixel to load.

inputs to limit the displayed field.

You will find that the TV display does not respond instantaneously to the changes in the transfer function requested by adjustment of the trackball, especially when the computer is being used heavily. It's best to turn the trackball slowly.

Pressing buttons **B** and **C** adjusts the zoom of the display: **B** to increase the magnification and **C** to decrease it. When these buttons are enabled, the cursor controls the position of the center of the zoomed field of view. Magnification factors of 1, 2, 4 and 8 are available on the I<sup>2</sup>S.

Pressing button **D** will exit from the TVALL display modification routine, leaving your display parameters as they were when button **D** was pressed. Note that your terminal issues instructions when buttons are pressed, but that it is in the death-like grip of TVALL otherwise until you press button **D** to exit from it.

You may resume the image modification without reloading the image by typing:

> TVFIDDLE C<sub>R</sub>

### 6.2.1. Alternative color coding

To obtain alternative color coding of the TV image:

> TVPSEUDO C<sub>R</sub>

and then follow the instructions listed on your terminal.

A rich zoo of color coding is available, but TVPSEUDO defaults to a coding similar to the "Spectrum Colors" on *IMPS*. First-time users should take a little time to experiment with the AIPS coloring options until they develop a feel for the effects of the trackball settings on the image appearance. The wedge displayed by TVALL will adjust to the alternative colorations selected with TVPSEUDO. Remember that the TV image does not respond instantaneously to trackball motions; move the ball slowly.

### 6.2.2. Transfer functions

To fiddle the (black-and-white) transfer function of the display without reusing TVALL:

> TVTRAN C<sub>R</sub>

and then follow the instructions listed on the terminal.

When using TVTRAN, notice that the TV display does not respond instantaneously to modifications you make to the transfer function using the trackball. It's best to turn the trackball slowly. Also, the transfer function is displayed in the top right corner of the image only when the zoom is used at lowest magnification.

**6.3. Reading out image data using the trackball/cursor**

There are several facilities for reading out intensity and position information from displayed images using the TV cursor and the trackball:

> IMPOS  $C_R$

reads out the two coordinate values (e.g. RA and Dec) from the cursor position when any trackball button is depressed.

> CURV  $C_R$

continuously reads out pixel coordinates and the pixel intensity in user-recognizable units at the position selected by the TV cursor. The pixel parameters are displayed in the upper left corner of the image.

> TVWIN  $C_R$

reads pixel coordinates from the next two cursor positions at which a trackball button is depressed. The TV graphics shows the current shape and position of the window. Button A allows you to switch to (re)-setting the other corner while the other buttons exit after both corners have been set. **TVWIN** uses the pixel coordinates to set up the bottom left (**BLC**) and top right (**TRC**) corners of an image subsection, e.g. for input to the contouring programs **CNTR** and **PCNTR**, to the mean/rms calculator **IMEAN**, and to many other tasks.

> SETXWIN ( $dx, dy$ )  $C_R$

reads pixel coordinate of the center of a  $dx$ -pixel by  $dy$ -pixel window and sets the adverbs **BLC** and **TRC**.

> TVBOX  $C_R$

works like **TVWIN** above except that it is used to set up pixel coordinates to define **CLEANing** areas for the **AIPS CLEAN** routine (**APCLN**). The adverb **NBOXES** must be entered before the **TVBOX** command and is the number of rectangular **CLEANing** boxes to be set with the TV cursor, trackball, and buttons.

## 7. TAKING HARD COPY OF IMAGES

A listing of available *AIPS* tasks for two-dimensional displays and hard copies of images can be obtained by typing `HELP PL2D`. The listing is also given in § 13 of the *COOKBOOK*.

### 7.1. Ordinary contour plots (CNTR)

In addition to the usual image selection parameters, you may specify the following parameters to the contouring task `CNTR`:

> `TASK 'CNTR' ; INP CR` to tell you what you may specify.

Examples of the command syntax (*not* a recommended combination):

> `BLC 250, 230 CR` to set the bottom left corner of plot at 250,230.

> `TRC 300, 330 CR` to set the top right corner of plot at 300,330 (in pixels with 1,1 at extreme bottom left of the image).

> `CLEV 0 ; PLEV 1 CR` to get contour levels at 1% of the peak map value.

> `PLEV 0 ; CLEV .003 CR` to get contour levels at 3 mJy.

> `LEVS -1, 1, 2, 4, 6 CR` to get actual contours at -1, 1, 2, 4, and 6 times the basic level set by `PLEV` or `CLEV`. `LEVS` need not be integers, but you should not use finer subdivisions than tenths (e.g. 2.5 is ok, but 2.47 will cause confusion).

*N.B.* If you request more than one negative level with the `LEVS` input, you *must* use commas between the negative levels. Otherwise the minus sign(s) will be treated as subtraction symbols and the desired levels will be combined into a single negative level by the *AIPS* language processor. `BLC` and `TRC` can be initialized conveniently from the TV display using the cursor with the `TVWIN` instruction (see § 6.3). Check:

> `INP CR` to review what you have specified.

When you're satisfied with the inputs, then:

> `GO CR`

generates a plot file as an extension to your image file, with the parameters you have just specified. Watch the *AIPS* monitor to see the progress of this task. If the number of blocks written in the plot file is over 200, the contour plot will be messy!

To look at the results, enter:

> `GO TKPL CR` to display the plot file produced by `CNTR` on the Tektronix 4012 (graphics) terminal. The `HARD COPY` switch on the terminal can be used to send the TEK image to the copier (often a printer/plotter).

> `ABORT TKPL CR` to stop an overly messy plot on the 4012.

> `GO TVPL CR` to display the plot file on a TV graphics plane.

> `GO PRTPL CR` to display the plot file to the printer/plotter. This task is a bit slow—be patient or do something else on *AIPS* while you wait.

Systems having other types of display devices may have other tasks to put the results on those devices. The above tasks will select the plot file with the largest version number if `INVER 0`. This is generally what is desired.

**7.2. Contour plots with polarization vectors (PCNTR)**

PCNTR plots polarization vectors on top of contours. You must first make a polarized-intensity map and a polarization position-angle map from the Q and U maps (see § 8.1.1 below). The inputs to PCNTR can be reviewed by:

> TASK 'PCNTR' ; INP CR

It is a long list, so you will need to use the page-scroll keys (see § 2.2) to review the start of these inputs.

Try:

> INDI 0 ; MCAT CR

to review the contents of your map catalog for image names and classes.

> INDI *n1* ; GETN *ctn1* CR

where *n1* and *ctn1* select the disk and catalog numbers of the map to be contoured.

> IN2DI *n2* ; GET2N *ctn2* CR

where *n2* and *ctn2* select the polarized intensity map.

> IN3DI *n3* ; GET3N *ctn3* CR

where *n3* and *ctn3* select the position-angle map.

> PCUT 0.*nn* CR

to blank out vectors less than *nn*% of the peak polarized intensity.

> ICUT 0.*mm* CR

to blank out vectors at pixels where the total (map 1) intensity is less than *mm*% of peak map 1 intensity.

> FACTOR *zz* CR

to set the length of the longest vector to *zz* cell widths.

> INP CR

to review your inputs and remind you of others. Most are similar to those in CNTR and sensibly defaulted.

> GO PCNTR CR

to generate the plot file, which can then be routed to output devices via TKPL, TVPL, or PRTPL.

**7.3. Deleting unwanted plot files (EXTDEST)**

Plot files generated by CNTR and PCNTR are archived in your disk catalog as PL extensions to the image file from which they were derived. Running CNTR or PCNTR again with new parameters does not overwrite the previous plot file, but makes another with a higher "version" number.

You can review the parameters of the plot files associated with a given mapfile by typing:

> INDI *n* ; GETN *ctn* CR

where *n* and *ctn* select the disk and catalog number of the desired map.

> INEXT 'PL' ; EXTLIST CR

to select a listing of plot file contents.

Plot files (and other "extension files") are automatically deleted when an image is deleted by ZAP. However, large plot files should be deleted as soon as they are no longer needed:

- > INP EXTDEST  $C_R$                       to review the inputs required.
- > INDI  $n$  ; GETN  $ctn$   $C_R$               where  $n$  and  $ctn$  select the disk and catalog numbers of the map file.
- > INEXT 'PL' ; INVERS  $m$   $C_R$               to set the type to PL (plot) and the version number to be deleted to  $m$ .
- > EXTDEST  $C_R$                       to do the deletion.
- > INVERS 0  $C_R$                       to reset the version number to its default — usually advisable.

#### 7.4. SLICE files (profile plots)

You can generate a one-dimensional slice (profile) through any two-dimensional mapfile using the AIPS task SLICE. The output file is appended to the mapfile as an SL extension file. Slices are computed along lines in the two-dimensional map joining any valid pair of points selected by BLC and TRC. The set of software dealing with slice file analysis and display can be obtained by typing HELP SL1D. The list is also given in § 13.

To generate a slice:

- > TASK 'SLICE' ; INP  $C_R$                       reviews the inputs to SLICE.
- Use **INDISK** and **GETNAME** to select the input image. The bottom left (**BLC**) and top right (**TRC**) end points for the slice can be specified conveniently using the TV cursor if the map to be sliced is first displayed on the TV with TVL0D or TVALL. To set these points with the TV, type:
- > TVWIN  $C_R$
- then set the TV cursor to the desired beginning (**BLC**) point for the slice, press any trackball button, and repeat for the ending (**TRC**) point for the slice. Note that, for slices, **BLC** need not be below or to the left of **TRC**. Finally:
- > GO  $C_R$                       to generate the slice file.

When SLICE has terminated (watch the AIPS monitor), the file may be plotted on the Tektronix 4012 (graphics) terminal using the verb **TKSLICE**:

- > INP TKSLICE  $C_R$                       to review what you can specify.
- > INEXT 'SL' ; EXTL  $C_R$                       to find the intensity range and number of points in the interpolated slice.

The default scales will plot all slice points on a vertical scale from the map minimum to the map maximum. You can alter the scales by specifying, for example:

- > BDROP 100 ; EDROP 225  $C_R$                       to drop 100 points from the beginning and 225 points from the end of the slice.
- > PIXRANGE -0.001 0.004  $C_R$                       to set the range of the vertical axis to be -1 to 4 mJy/beam.

Then:

- > TKSLICE  $C_R$                       to plot the slice on the TEK screen.

Note: several slices may be put on one TEK plot. Use **TKASLICE**  $C_R$  for the additional ones.



**Slice files may be converted into plot files by:**

> GO SL2PL C<sub>R</sub>

**The resulting plot files may then be output on the printer/plotter by:**

> GO PRTPL CR

to obtain a better-quality plot than can be obtained with the **HARD COPY** button on the Tektronix. In addition:

> GO TKPL C<sub>R</sub> will display the plot file on the TEK.

> GO TVPL C<sub>R</sub> will display the plot file on a TV graphics channel.

A one-dimensional fit of up to four Gaussian components to slice data can be calculated using the task **SLFIT** (see § 8.3.3). Relevant displays of the data, the fit, and the residuals are also available both as verbs (like **TKSLICE**) and as options in the task **SL2PL**. These are illustrated in § 8.3.3. See **HELP SL1D** or the listing in § 13 for complete lists.

Slice files are archived in your disk catalog as SL extensions to the image file from which they were derived. Running SLICE again with new parameters does not overwrite the slice file, but makes another with a higher "version" number. To review and/or delete slice files, follow the instructions for plot files in § 7.3 above, but use INEXT 'SL' in place of INEXT 'PL'.

### 7.5. Dicomed copies of images

At the time of this edition of the *COOKBOOK*, there is no *AIPS*-standard method for recording images on the Dicomed film recorders. However, ad hoc methods exist both at the VLA site and in Charlottesville for doing this. These methods require that the *AIPS* image be written on tape in the FITS format using task FITTP and then carried to another computer. Type `HELP FITTP CR` for an explanation of the *AIPS* task FITTP.

When your images are on tape *at the VLA*, take your tape to the Telex tape drive for the 11/44 (in the main computer room of the Control Building, near the Dicomed) and physically mount the tape as described in Appendix Z of this *COOKBOOK*. Go to the *DISPLAY (IMPS)* terminal and type RUN FITS. Then follow the instructions on the terminal. Ask for unlabeled tape processing and use the instruction RESTORE to load all tape images to the *IMPS* disk. The map coordinates in the 11/44 will be scrambled, due to different conventions for the reference pixel location in *AIPS* and in the 11/44. This is of little importance provided all you want to use within *IMPS* is the Dicomed output function.

In Charlottesville, a similarly inelegant route is still required to get from *AIPS* to the Dicomed. *FITS* tapes must be written by *FITTP*, read in to the IBM via *FITS2DEC*, converted to a picture tape via *DECPHX*, and then brought back up to the Modcomp computer. The Modcomp program *DCP* interprets picture tapes, recording images and (correct) labeling on the Dicomed. (Consult Kerry Hildrup in Charlottesville for details.)

## 8. ANALYZING IMAGES

In order to obtain useful astronomical information from the data, software is being developed for the analysis of images, combining of images, estimating errors, *et al.* Only a few of the programs are described here; the others should be self-explanatory using the HELP and INPUTS files for the tasks listed in § 13. A complete list of software in AIPS for the analysis of images may also be obtained by typing HELP ANALYSIS  $\mathcal{C}_R$ .

### 8.1. Combining images (COMB)

The task COMB is a general purpose program for combining two images, pixel by pixel, to obtain a third image. Many options are available and, as a first example, we illustrate inputs to derive the polarization intensity and angle from the Q and U Stokes parameter maps.

#### 8.1.1. Polarized intensity and p.a. maps

To compute a polarized intensity image, enter:

- > TASK 'COMB' ; INP  $\mathcal{C}_R$                       to review the required inputs.
- > INDI 0 ; MCAT  $\mathcal{C}_R$                             to help you find the catalog numbers of the Q and U maps that you want to combine.
- > INDI *n1* ; GETN *ctn1*  $\mathcal{C}_R$                     where *n1* and *ctn1* select the Q map.
- > IN2D *n2* ; GET2N *ctn2*  $\mathcal{C}_R$                     where *n2* and *ctn2* select the U map.
- > OUTN 'xxxx'  $\mathcal{C}_R$                             where *xxxx* is your choice of outfile name for the polarized intensity map, e.g. NGC315.PIL.
- > OUTC 'ccc'  $\mathcal{C}_R$                             where *ccc* is your choice of classname for the polarized intensity map, e.g. PMAP.
- > OPCODE 'POLI'  $\mathcal{C}_R$                             to select the  $\sqrt{Q^2 + U^2}$  algorithm.
- > GO  $\mathcal{C}_R$                                       to compute the polarized intensity map.

The AIPS monitor should note TASK COMB BEGINS followed by a listing of the POL. INTENSITY algorithm. While it is running, you can prepare the inputs to make a polarization position angle map. Type:

- > OUTN 'yyyy'  $\mathcal{C}_R$                             where *yyyy* is your choice of outfile name for the polarization angle map.
- > OUTC 'ddd'  $\mathcal{C}_R$                             where *ddd* is your choice of class name for the polarization angle map, e.g. PSIMAP, CHIMAP.
- > OPCODE 'POLA'  $\mathcal{C}_R$                             to select the  $\frac{1}{2} \tan^{-1} \left( \frac{U}{Q} \right)$  algorithm.

Once COMB has finished, enter:

- > GO  $\mathcal{C}_R$                                       to compute the polarization angle map.

The AIPS monitor should note again TASK COMB BEGINS followed by a listing of the POL. ANGLE algorithm. Once both COMB tasks have terminated with COMB: ENDS SUCCESSFULLY messages on the message monitor, you should find the requested maps in your catalog:

- > MCAT  $\mathcal{C}_R$                                       to list the maps in your catalog.

### 8.1.2. Other map combination options

**COMB** may also be used to add or subtract images, to rescale them, and to compute spectral indices, optical depths, *et al.*. Type:

- > **HELP COMB C<sub>R</sub>**                      to review the available options.
- > **HELP COMBCODE C<sub>R</sub>**                to see even more details.

At the time of writing the options include:

'SUM '	Addition	$a_1 M_1 + a_2 M_2 + a_3$
'MULT'	Multiplication	$a_1 M_1 M_2 + a_2$
'DIV '	Division	$a_1 \frac{M_1}{M_2} + a_2$
'SPIX'	Spectral Index	$a_1 \ln \left( \frac{M_1}{M_2} \right) / \ln \left( \frac{\nu_1}{\nu_2} \right) + a_2$
'OPTD'	Opacity	$a_1 \ln \left( \frac{M_1}{M_2} \right) + a_2$
'CLIP'	Clipping	$M_1$ <i>except blanked where</i> $a_1 > \frac{M_2}{Peak(M_2)} > a_2$
'POLI'	RMS sum	$a_1 \sqrt{M_1^2 + M_2^2} + a_2$
'POLA'	Arctangent	$a_1 \tan^{-1} \left( \frac{M_2}{M_1} \right) + a_2$
'REAL'	Real part	$a_1 M_1 \cos(a_2 M_2)$
'IMAG'	Imag part	$a_1 M_1 \sin(a_2 M_2)$

where the  $\alpha_i$  are user-adjustable parameters and  $M_1$  and  $M_2$  are the images selected by `INNAME`, *et al.* and by `IN2NAME`, *et al.*, respectively.

For some applications of **COMB**, undefined pixel values may occur. For example, if the spectral index is being calculated and the intensity level on either map is negative, the index is undefined. In this case, the pixel value is given a number which is interpreted as undefined or blanked. Blanking also arises naturally in operations of division, opacity, poalrization angle, and clipping. Using **APARM(8) = 1 C<sub>R</sub>**, the user may specify how blanking will be done on the output maps. **APARM(9)** and **APARM(10)** can be specified to blank the output map whenever either of the input maps are less than a specified amount. See **HELP COMBCODE**.

## 8.2. Map statistics and flux integration (IMEAN)

The task **IMEAN** is used to determine the statistics in the map over a specified rectangular area. It derives the minimum and maximum value and location, the rms, the average value and an approximate flux density within the area (if it is a CLEANed map). A typical run might be:

- |  |  |
|--|--|
| > TASK 'IMEAN' ; INP C <sub>R</sub>                    | to list the input parameters.  |
| > INDI <i>n</i> ; GETN <i>ctn</i> C <sub>R</sub>       | where <i>n</i> and <i>ctn</i> select the disk and catalog numbers of the relevant map file.                      |
| > BLC <i>n1, n2</i> ; TRC <i>m1, m2</i> C <sub>R</sub> | to set the window from ( <i>n1,n2</i> ) to ( <i>m1,m2</i> ) -- or use TVWIN or SETWIN with the cursor on the TV. |
| > DOHIST TRUE C <sub>R</sub>                           | to make a plot file of the pixel histogram.  |
| > PIXRANGE <i>x1, x2</i> C <sub>R</sub>                | to set the range of the histogram from <i>x1</i> to <i>x2</i> .  |
| > NBOXES <i>n</i> C <sub>R</sub>                       | to set the number of boxes in the histogram.   |
| > GO C <sub>R</sub>                                    | to run the task.   |

The statistics will appear on the *AIPS* monitor. For a hard copy type:

- > PRTMSG C<sub>R</sub> with PRIO < 5.

To see the histogram of the intensities, type one of:

- |                           |  |
|---------------------------|--|
| > GO TKPL C <sub>R</sub>  | to display histogram on the Tektronix.       |
| > GO PRTPL C <sub>R</sub> | to display histogram on the printer/plotter. |

### 8.3. Fitting of images

There are two programs which estimate the position and intensity of a component on a two-dimensional image. The simple and fast method is the verb **MAXFIT**. This fits a two-dimensional parabola to the maximum within a few pixels of a map position and gives the peak and its position. The task **IMFIT** fits an image subsection with up to four Gaussian components and makes error estimates. The task **SLFIT** fits Gaussian components to one-dimensional slice data.

### 8.3.1. Parabolic fit to maximum (MAXFIT)

**MAXFIT** is inexact, but its speed makes it useful for simple regions. Type:

- |                              |   |
|------------------------------|---|
| > HELP MAXFIT C <sub>R</sub> | to get a good explanation of the algorithm. |
| > INP MAXFIT C <sub>R</sub>  | to list the inputs.                         |

These should be self-explanatory. The **IMSIZE** parameter can be important in crowded fields. **MAXFIT** can be used conveniently by first displaying the image on the TV and then typing:

- > IMXY ; MAXFIT C<sub>R</sub>

First the cursor will appear on the TV. Move it close to a maximum and hit any of the buttons behind the trackball. The fit will then appear on your terminal.

### 8.3.2. Two-dimensional Gaussian fitting (IMFIT)

A more sophisticated fit of an image is obtained with the task **IMFIT**, which will fit an image with up to four Gaussian components and attempt to derive error estimates. A sample set-up is as follows:

- |  |  |
|--|--|
| > TASK 'IMFIT' ; INP C <sub>R</sub>                    | to list the input parameters.  |
| > INDI <i>n</i> ; GETN <i>ctn</i> C <sub>R</sub>       | where <i>n</i> and <i>ctn</i> select the disk and catalog numbers of the required map.   |
| > BLC <i>n1, n2</i> ; TRC <i>m1, m2</i> C <sub>R</sub> | to set the area to be fitted as ( <i>n1,n2</i> ) to ( <i>m1,m2</i> ) — or use TVWIN with the cursor on the TV.   |
| > NGAUSS 2 C <sub>R</sub>                              | to set the number of components to be fitted to 2.   |
| > GMAX 0.34 0.20 C <sub>R</sub>                        | to give estimates of peak intensity in Jy.   |
| > GPOS 200, 100, 210, 110 C <sub>R</sub>               | to give estimates of the pixel locations of each component.  |
| > GWID 6 4 20 6 4 20 C <sub>R</sub>                    | to give estimates of component sizes in pixels. In this case, each component has a FWHP of 6 by 4 pixels with the major axis at position angle 20 deg. |
| > DOWID FALSE C <sub>R</sub>                           | to hold all of the widths constant in the fitting process (if required).   |
| > INP C <sub>R</sub>                                   | to review inputs.  |
| > GO C <sub>R</sub>                                    | to run the task.   |

Use the **RUN** file **INPFIT** (See § 12.6) to obtain a procedure which will help to supply the needed input parameters to the task **IMFIT**.

The task **SLFIT** fits Gaussian components to one-dimensional data in slice files (see §7.4). Assuming that the usual **GETNAME** step has been done, a typical session would go like:

- This verb will prompt you to **POSITION CURSOR AT CENTER & HEIGHT OF GAUSSIAN COMP 1** and will turn on the Tektronix crosshairs. Move the crosshairs with the thumbwheels to the requested position and press any key *except*  $C_R$  on the Tektronix 4012 keyboard. The terminal then requests you to **POSITION CURSOR AT HALFWIDTH OF GAUSSIAN COMP 1** and turns the crosshairs back on. Move the crosshairs until they are at the half-intensity point of the component and press any key except  $C_R$ . The process continues until all components have been entered. Then type:

- If everything looks ok, then:

- > GO SLFIT C<sub>R</sub> to run the task.

> PRTMSG C<sub>R</sub> to print the message file.

- To get a higher quality plot of the results:**

- |   |   |
|---|---|
| > DORES TRUE ; DOMOD TRUE C <sub>R</sub>  | to request the model and the residuals.             |
| > DOSLICE FALSE C <sub>R</sub>            | to leave the slice data out of the plot.            |
| > TASK 'SL2PL' ; GO ; WAIT C <sub>R</sub> | to make a plot file and wait for it to be complete. |
| > GO PRTPPL C <sub>R</sub>                | to display it on the printer/plotter.               |

## 9. SPECTRAL LINE SOFTWARE

Spectral line software generally involves three-dimensional images where the third dimension is frequency or velocity. Special programs are needed to build and transpose these cubes and display them properly. Much of the continuum software will work on data cubes or on appropriate two-dimensional images from the data cube. Spectral line UV data are read into *AIPS* from EXPORT tapes containing many pseudo-continuum data bases. Spectral line maps made in the 11/70 mapping system can be read into *AIPS* from FITS tapes. When processing many spectral line maps at different frequencies, it is best to give them the same name and class, as the sequence number can be used to identify the different frequency channels.

### 9.1. Reading in multichannel UV data

The following inputs could be used to load  $n$  pseudo-continuum (usually  $2n$  frequency) channels of data for the source N5128:

```
> TASK 'UVLOD' CR
> SOURCE 'N5128'
> OUTC ' '; OUTN = SOURC CR          set output file name.
> OUTSEQ 0 ; OUTCL ' ' CR           to use the sequence number for channel identification.
> NPOINTS mm CR                   to load mm thousands of points.
> FOR I = 1 TO n ; PRINT I ; GO ; WAIT ; END CR
                                         to loop over n executions.
```

The  $n$  data sets will be read from the Export tape. Then:

```
> PRTMSG CR                        will produce a printer listing which includes the apparent frequency of each data set. This frequency is that of the data stored in the pseudo-continuum RR channel. The user will have to provide the frequency of the LL channel.
```

### 9.2. Making spectral line maps

The task **UVMAP** (§ 3.1) has certain features that are useful for spectral line mapping of pseudo-continuum data:

```
> STOKES 'RL' CR                  to make separate maps of the channels stored in the R and L IFs.
> XPARAM(9) = 1 CR                to request a different frequency for the L map and beam.
> XPARAM(10) = 3.91E5 CR          to have the L map made at a frequency +391 kHz higher than the UV data header frequency used for the R map.
```

The resulting R and L maps and beams will have the correct frequencies in their image headers. If **XPARAM(9)  $\neq$  1** or **XPARAM(10) = 0**, the same frequency is used for the R and L maps and only one beam (for R) is produced.

### 9.3. Building the cube

The task **MCUBE** is used to put a series of maps at different frequencies into a 3-dimensional image (cube) cataloged on disk. The main advantage of building a cube is that you may transpose it in order to create maps with velocity as one axis and RA or Dec as the other axis of each plane. As all maps in the cube must have the same name and class, you must use **RENAME** to change the class names and sequence numbers. For example, the following commands would convert a series of 16 **RMAPs** and 16 **LMAPs** to one series of 32 **MAPs**:

```
> INN 'N5128' ; INCL 'RMAP' CR      to set input names to the R-channel maps.
> OUTN = INN ; OUTSEQ 0 CR          to set output names.
> OUTCL 'MAPS' CR                  to set output class to MAPS.
> FOR I = 16 TO 1 BY -1 ; INSEQ = I ; OUTSEQ = 2 * I - 1 ; RENAME ; END CR
                                   to loop over all inputs. Note OUTSEQ.
> INCL 'LMAP' CR                  to repeat the above with new output sequence numbers for the
                                   L-channel maps.
> FOR I = 16 TO 1 BY -1 ; INSEQ = I ; OUTSEQ = 2 * I ; RENAME ; END CR
                                   note OUTSEQ computation.
```

The 32 **MAPs** can be put into a 3-dimensional image cube by:

```
> TASK 'MCUBE' ; INP CR            to review the inputs.
> INCL 'MAPS' CR                  input map class for all maps now.
> INSEQ 1 ; IN2SEQ 32 ; IN3SEQ 1 CR to set the first and last map and the step in the task's loop
                                   over input sequence number.
> BLC 64 ; TRC 192 CR            to put only the 64,64 to 192,192 pixel subimages into the cube
                                   (for a quick first look).
> AXREF 1 ; AX2REF 32 CR          pixel coordinates of first and last map on 3rd axis.
> NPOINTS 32 CR                  total number of points on 3rd axis.
> OUTN ' ' ; OUTCL 'LMFCUB' CR    to set outname to default with a special outclass.
> GO CR                          to run MCUBE.
```

It is also possible to insert or replace maps in an existing cube. For details:

```
> HELP MCUBE CR                  to review the HELP file.
```

**MCUBE** may complain that it cannot build a complete cube because the third axis does not increment uniformly in the input set of 2-D maps. Unfortunately, in producing spectral line maps of pseudo-continuum data from the VLA, this problem can occur. The frequency option in **UVMAP** is one attempt to avoid the problem. However, if **MCUBE** complains, use the verb **AXDEFINE** to place the correct third-axis coordinate values in the header of each image.

#### 9.4. Transposing the cube

The task **TRANS** will transpose the cube. Typical inputs are:

- |  |  |
|--|--|
| > TASK 'TRANS' ; INP C <sub>R</sub>        | to review the inputs.  |
| > INCL 'LMFCUBE' C <sub>R</sub>            | to point at the untransposed cube.   |
| > TRANSC '312' C <sub>R</sub>              | to make new order of axes 3,1,2 in terms of the old axis order (i.e. RA, Dec, freq becomes freq, RA, Dec).                         |
| > OUTN ' ' ; OUTCL 'FLMCUB' C <sub>R</sub> | to have an output name reflecting the new axis order.  |
| > BLC 0 ; TRC 0 C <sub>R</sub>             | to transpose the whole cube. Remember that AIPS adverbs are <i>global</i> and that we had set BLC and TRC for doing <b>MCUBE</b> . |
| > GO C <sub>R</sub>                        | to transpose the cube.   |

The images produced by **MCUBE** are written in real format. Some tasks, such as **TRANS**, can handle either real or integer format. However, since most tasks and verbs in *AIPS* require integer format, you should convert the cubes to this format with **CNVRT**. Type **HELP CNVRT C<sub>R</sub>** for instructions.

#### 9.5. Further display and processing

Transposed images can be displayed on the TV screen. You must specify a specific pixel for the third coordinate, unless you want to look at the first plane. For example:

- > TBLC 0 0 *n* ; TTRC 0 C<sub>R</sub> (the defaults handle all but **TBLC(3)**.)

displays a velocity-RA map at the *n*<sup>th</sup> declination. Velocity profiles could be produced from such a map using **SLICE**, then plotted using **TKSLICE** (see § 7.4). Task **PROFL** may also be of interest for displaying multiple velocity profiles.

A list of available spectral line software can be obtained by typing:

- > **HELP CUBE C<sub>R</sub>**

The list is also given in § 13 of this *COOKBOOK*. Most of the analysis and display programs can also access single planes of a data cube. Many also do quite sensible things with more than one plane at a time.

The task **MOMNT** will calculate a set of moment maps (from moment 0 to moment 4) from a data cube. The task **SMOTH** will convolve each image in a cube as desired in order to obtain better signal to noise ratios on extended emission. **CONVL** performs a similar function with an alternate set of inputs. Frequency-smoothing (actually *x*-axis smoothing) may be done more quickly with the task **HANSM**. Smoothed maps assist in determining the boundaries of sources as windows to be used in subsequent spectral line analysis. For example, the smoothed cube could be used to set the **CLIP** limits in task **COMB** to be applied to the unsmoothed cube. Note that **COMB** will accept a cube as the first input map and a single, matching plane as the second map. Thus, you can use **COMB** to subtract the continuum from the cube or convert the cube to optical depth (for example).

Several spectral line oriented displays are available. The task **RGBMP** creates a cube of three planes from an arbitrary sized cube which can then be used for a false color display using the verb **T3COLOR** (after appropriate **TVLODs**). The zero and first moments produced by **MOMNT** can be loaded into separate planes of the TV device with **TVLOD**. Then the verb **TVHUE** will produce a color display where color (hue) represents the first moment value and intensity represents the zero moment value. **TVHUE** allows interactive enhancements as well. The verb **TMOVIE** will load subimages of each plane of a cube into portions of the TV memory with labeling and then display them in sequence at a user-controlled rate. **REMOVIE** allows a previously loaded "movie" to be rerun.



## 10. TIDYING UP AND EXITING AIPS

Before exiting from *AIPS* there are several things to be done to tidy up various kinds of debris you may have left in the system.

### 10.1. Backups

While processing and particularly just before exiting *AIPS*, please back up and delete (or simply delete) as many of your own data sets as possible. Maps and UV data may be backed up on tape using the task **FITTP**. UV data may also be backed up on tape using the task **UVEXP**. If space is short, data sets which remain on the disk for more than a week or two will be ruthlessly deleted by designated *AIPS* gorillas. If you wish to leave data on disk for an extended period of time, please notify the System Manager and bring an appropriate "gift".

On some systems (e.g. VAX/VMS), a system-dependent global backup procedure may be available. This procedure will allow you to write all of your files on tape quickly, verify the tape, and then, optionally, delete the files. Type **HELP BCKAIPS** and **HELP RESAIPS** for information. Such procedures are very nice, but they produce a highly system-dependent tape. A tape written on one VAX may not be able to be read on another, for example. However, **BCKAIPS** should be used (where available) instead of leaving data on disk for prolonged periods.

### 10.2. Deleting your data

All users should delete redundant images and data as soon as possible to preserve disk space for others. It is tempting to work on many sets of data at the same time, but this generally takes a lot of disk space and the user is encouraged to limit the amount of data resident on disk to that which will be processed during the session.

A data set and all extension files can be deleted by:

```
> IND n ; GETN ctn CR           where n and ctn select the disk and catalog numbers of the
                                     data set to be deleted.
> ZAP CR                           to do the deletion.
```

To delete data in contiguous slots from *n* to *m* in a catalog, use the loop:

```
> FOR I = n TO m ; GETN I ; ZAP ; END CR
                                     after setting the INDISK.
```

For massive deletions — the kind we hope you will use when you depart — use:

```
> ALLDEST CR                       to destroy all data files which are consistent with the inputs to
                                     ALLDEST.
```

DO NOT DELETE OTHER USERS' DATA WITHOUT THE EXPLICIT PERMISSION EITHER OF THE OTHER USER OR OF THE SYSTEM MANAGER. Old data belonging to any user (including you) may be deleted by the verb **TIMDEST**. The definition of "old" is set by the local *AIPS* Manager, who must be consulted about the rules for invoking **TIMDEST**.

The software associated with deleting various files can be found by typing:

```
> HELP DELETE CR
```

This listing is also given in § 13.

### 10.3. Sending comments to the AIPS programmers

Gripes and comments about any facet of AIPS can be entered into text files by typing:

> GRIPE `CR` while in AIPS.

Simply follow the directions and record your gripe. Replies to the gripes are generated by the AIPS programming staff and are mailed to each individual "griper". Normally, an initial reply will be given in several weeks, but modifications and additions to AIPS, if warranted, may take a longer period of time to implement. In such cases, follow-up replies are also generated and sent to the gripers. The gripes and responses will be open to public display, so write them clearly and in good taste. At present, complete copies of the gripes and responses are kept in notebooks in the AIPS Caiges at Charlottesville and the VLA. Since they are often amusing and educational, the *Collected Gripes* may eventually be published more widely.

Gripes can be registered outside of AIPS in monitor level by typing (on VAX/VMS):

\$@GRIPE `CR`

and following directions. The program invoked by this procedure, GRIPR, implements the same code as AIPS, but its control structure is much simpler. Try:

> HELP `CR` after the @GRIPE.

for some information about the program's capabilities. The command to invoke this program may be different at some installations, particularly non-VAX/VMS ones.

### 10.4. Exiting

To exit from AIPS type:

> EXIT `CR`

Your message file will be printed automatically. If you do not want your messages, type:

> QE `CR`

instead. Please clean up any papers, tapes, etc. in the area around your terminal before you go.

## 11. PANIC SECTION

### D O N ' T      P A N I C      ! ! ! !

On all computer systems things go wrong due to user error, program error, or hardware failure. Unfortunately, *AIPS* is not immune to this. The Section below reviews several general problem areas and their generalized solutions. Refer to Appendix Z for the details appropriate to your particular computer system. Type `HELP PANIC CR` to review recent information on bugs, disasters and misfortunes within *AIPS* that have known solutions. Some well-known possibilities follow.

#### 11.1. My printout is semi-infinite!!

If you wish to abort a job which is currently printing out do the following:

1. Turn the power off on the printer to conserve paper.
2. Type `SPY CR` to see if the offending program is still executing. If so, `ABORT taskname` will kill it.
3. Get into monitor mode (prompt `$` on most machines) either on your terminal or another one.
4. Abort the current print job. See Appendix Z for the details for your machine.
5. Turn the power back on for the printer.

#### 11.2. The TEK screen will not print hard copy!!

1. Find the device which makes the hard copies. Is it turned on?
2. Check to see if there is paper in the device.
3. Power it down, then power it up again. Try the `HARD COPY` button on the TEK again.
4. If it still fails, call for hardware repair.

#### 11.3. My tape refuses to be read!!

1. Have you set the density correctly (check the density indicator lights on the tape drive)?
2. Have you specified the `INTAPE` number to correspond with the drive you mounted the tape on?
3. Did you actually mount the tape in software from the Monitor level (on those systems requiring it, e.g. VAXes)?

**11.4. My data catalog has vanished!!**

1. Are you logged in to the correct login area? Older *AIPS* systems, in particular, may have more than one distinct login area.
2. Have you set *INDISK et al.* correctly before running *CAT*? Type *INP CAT CR* to check. Is *USERID* not set to 0 or your user number?
3. Is a non-system disk mounted? Exit from *AIPS*, and have the operating system tell you which disks are currently running. See Appendix Z for details.

**11.5. My program crashed for lack of disk!!**

Read this *COOKBOOK*, § 4.4 and Appendix Z for your computer system.

**11.6. My terminal has hung itself up!!**

If your terminal is "dead", i.e. refuses to echo characters typed on the keyboard or otherwise to show signs of talking to your computer, you have a problem. There are numerous possible causes.

1. Are you executing a long verb, e.g. *DISKUSE*? If so, be patient.
2. Are you executing some interactive TV or TEK verb which is waiting on input from the cursor or buttons? If so, provide the input.
3. Have you started a task with *DOWAIT* set to *TRUE* (+1.0)? If so, wait for the task to finish. Most tasks report their progress on the message monitor.
4. Is *AIPS* waiting while a tape rewinds or skips files or waiting to open some disk file currently being used by one of your tasks? Be patient.
5. Have you stopped output to your terminal accidentally by hitting the appropriate *NO SCRL* or other *XOFF* control sequence? If so, hit the *XON* control sequence. (These are *CTRL S* and *CTRL Q*, resp. on *VAX/VMS*.)
6. Can you abort *AIPS* at your terminal using the appropriate system commands (i.e. *CTRL Y* on *VAX/VMS*)?
7. Do other terminals connected to the computer appear to be "alive"? If so, login on one of them and inquire about the status of your *AIPS* program and tasks. It might be necessary to stop them from your new terminal and then log back in on your old terminal. If this doesn't bring your terminal back to life, report the problem to the *AIPS* Manager.
8. If all terminals appear dead, then your computer has probably "crashed". Report the problem to your *AIPS* or System Manager. If you feel you must reboot the system, do so *only* after checking that all current users and the System Manager (if he is available) agree that that action is required.
9. If even a reboot fails, report the problem to the System Manager or hardware experts and go do something else. **UNDER NO CIRCUMSTANCES SHOULD YOU ATTEMPT TO REPAIR THE TERMINAL, TELEVISION, DISK, CPU, OR OTHER HARDWARE DEVICES.** Such repairs must be performed by trained personnel.

## 12. AIPS FOR THE MORE SOPHISTICATED USER

The program AIPS uses a computer language called POPS to communicate with the user. This language has numerous capabilities which have been mostly hidden in the previous Sections. However, to use the full power of the AIPS system, the user will need to know more. This Section of the COOKBOOK attempts to provide additional detail. There is also a four-volume AIPS Manual, the first volume of which is devoted to the information most pertinent to users. A copy should be available in your AIPS Cage.

### 12.1. AIPS syntax

Some niceties of using the syntax of AIPS are:

1. More than one expression can be put on a line. These expressions must be separated by a semicolon ( ; ). Exceptions are STORE, RESTORE, GET, and SAVE and a few other "pseudoverbs" which, with their arguments, must stand alone. For example, GET APMAP ; INDISK = 1 C<sub>R</sub> will ignore the INDISK = 1. When in doubt, see the HELP files for the pseudoverb to find the restrictions in its use.
2. As in numerous systems, recognized keywords in AIPS do not need to be typed in full. One must type only enough of the leading characters so that the symbol is not ambiguous. The first four characters usually suffice. This "minimum matching" has been exploited throughout this COOKBOOK.
3. The parameter variables in AIPS are called "adverbs". They are assigned values by the equals verb ( = ), e.g.: INTAPE = 2 C<sub>R</sub>. However, the equals sign may be replaced by a space in almost all cases. The exception arises when the variable on the left is a subscripted array element and the expression on the right involves a unary minus or other function reference (i.e. SIN).
4. Array adverbs are set to a constant value by putting a single value on the right hand side of the equals sign, e.g. CELLSIZE = 1.5 C<sub>R</sub>. A list of values may be put in the array by putting the list on the right hand side of the = sign separated by commas ( , ), e.g. LEVS = -1, 1, 2, 3, 6, 9 C<sub>R</sub>. The commas may be replaced by spaces in most cases. An exception occurs if an element is negative or some other arithmetic expression. Thus, SHIFT = -19 -2 C<sub>R</sub> will produce SHIFT = -21, -21.
5. Adverbs can be used in arithmetical expressions or set equal to other adverbs, e.g. OUTNAME = INNAME ; OUTSEQ = 2.5 \* INSEQ + 3.
6. Both upper and lower case letters may be used in AIPS. However, adverb character string values are often differentiated on the basis of upper and lower case.

These shortcuts permit the following AIPS command sequence, for example:

```
> INNAME '3C138' CR
> INCLASS 'IMAP' CR
> INSEQ 0 CR
> BLC 200,200 CR
```

> TRC 300,300 C<sub>R</sub>

> OUTNAME '3C138' C<sub>R</sub>

> OUTSEQ 5 C<sub>R</sub>

> GO PRTIM C<sub>R</sub>

to be shortened to:

> Inn '3C138'; INC 'IMAP'; blc 200; Trc 300 C<sub>R</sub>

note use of upper and lower case.

> OUTN INN; OUTS INS + 5 C<sub>R</sub>

> go prtI C<sub>R</sub>

task name can be in either case.

## 12.2. Datafile formats

The physical name of the data file is generated internally, is dependent on the type of computer, and need not concern the user. The user selects an image by specifying the disk number, the type of image ('MA' for maps, 'UV' for UV data), the user identification number, and the three parts of the user-assigned image designation. The last are:

1. Name — A string of up to 12 legal characters.
2. Class — A string of up to 8 legal characters.
3. Seq — A number between 0 and 9999.

Each of these parts corresponds to separate input adverbs called **INNAME**, **INCLASS**, and **INSEQ** (and their variations). The image name can be chosen arbitrarily by the user, but sensible choices of names can relieve a lot of bookkeeping. Many programs will choose a reasonable image name if not specified by the user.

Only the array data, in the form of 2-byte scaled integers or 4-byte floating-point numbers, are stored in the data file. The header information is stored in a special file, called the Catalog File. Each disk has such a file and it contains all of the map headers for all users on the disk pack. On some systems, each user will have his own catalog file on each of the disks. The Modcomp in Charlottesville has the former arrangement, while the VAXes at the VLA and Charlottesville have the latter.

Extension files may be associated with any image data file. Each image can have (in principle) up to ten types of extension files and up to 99 "versions" of each type. These subordinate files contain additional information associated with the image and are designated by a two-letter type code. 'HI' is a history file, 'CC' is a CLEAN components file, 'PL' is a plot file, 'AN' is an antennas file, 'SL' is a slice file. In *AIPS*, an extension file associated with an image is uniquely specified by the usual file-naming adverbs plus the extension file type (adverb **INEXT**) and the version number (adverb **INVERS**).

Array elements in an image are designated by their pixel values. If  $M(i, j, k, l, m, n, o)$  is a seven-dimensional array, the (1,1,1,1,1,1,1) pixel will be associated with the lower left hand corner of the image. The  $i^{\text{th}}$  (first) coordinate increments fastest and is associated with a column in each plane of the image. The  $j^{\text{th}}$  (second) coordinate is associated with a row in each plane. The other coordinates allow the image to be generalized to cover up to seven dimensions, i.e. "cubes" and the like. The two adverbs **BLC** for bottom left corner and **TRC** for top right corner allow the user to specify the desired subarray in up to seven dimensions. Whenever a sub-image is taken from an image, the pixel designation of any map element will usually change.

**12.3. AIPS language**

AIPS contains a basic set of symbols and keywords, which are needed for facility in the language, as well as the symbols needed by the application code. A list of the basic symbols is given in § 13 and may be listed on your terminal by typing:

> HELP POPSYM C<sub>R</sub>

Here are some simple examples of uses of the AIPS language:

> TYPE (2 + 5 \* 6) C<sub>R</sub>

32 is written on the terminal.

> TYPE 'X =', ATAN (1.0 / 3.14159 \* 180.) C<sub>R</sub>

X = 45 is written on the terminal.

> TYPE 'MAPNAME ', INNAME, INCLASS, INSEQ C<sub>R</sub>

MAPNAME 3C138 IMAP 1 is written on the terminal.

**12.4. Processing loops**

The do-loop capability in AIPS uses the pseudoverbs FOR, TO, and BY and allows repetitive operations. Such loops are intended for use in "procedures". However, in practice, if the loop can be typed fully on one input line, then it will work outside the procedure mode. The following example shows how to delete a series of maps with the same name and class, but with consecutive sequence numbers 1 through 10:

> INNA 'TEST'; INCL 'IMAP' C<sub>R</sub> to set (fixed) name parts.

> INDI 1 C<sub>R</sub> to set (fixed) disk number.

> FOR INSEQ = 1 TO 10 ; ZAP ; END C<sub>R</sub> to delete the files.

FOR loops must be terminated with END.

**12.5. Procedures**

Procedure building is a method of combining keywords in AIPS in any convenient way to obtain useful constructs. For complicated sequences, it will be easier to prepare procedures in RUN files (§ 12.6) than to prepare them in interactive AIPS. A procedure construct is given a name, with or without arguments, and then can be treated as a verb in AIPS. For an example, we will build a procedure which will allow the user to load an image on the TV, set the cursor, and fit for the maximum intensity. The following should be typed on the terminal:

> PROC MFIT(I) C<sub>R</sub>

defines procedure MFIT with one argument I. (I and J are two dummy adverbs which are already defined in AIPS.)

: GETNAME(I) C<sub>R</sub>

notice the prompt symbol : . This means that we are in the procedure building mode.

: TVLOD ; IMXY ; MAXFIT C<sub>R</sub>

the instructions to load the image, produce and read the cursor, and fit the maximum near the cursor position when a TV button is pressed.

: RETURN C<sub>R</sub>

to designate a return point in the procedure.

: FINISH C<sub>R</sub>

to designate the end of the procedure building mode and to get back into the normal (prompt >) mode.

>

When you are typing a procedure into AIPS, the code is compiled as it is typed. Most syntax errors are spotted immediately and the user is unceremoniously dumped out of procedure mode. However, all lines written before the detected error are kept and the procedure editor can be used to carry on.

Rather limited procedure editing capabilities are available. However, for making procedures which are longer than about five lines, the use of permanent storage files in the "RUN" login area is recommended and is discussed in § 12.6 below.

To list procedure **MFIT**, type:

> LIST MFIT CR

This will produce the following:

```

1  PROC MFIT(I)
2  GETNAME(I)
3  TVLOD ; IMXY ; MAXFIT
4  RETURN
5  FINISH

```

The procedure is identical to that typed, with the addition of line numbers.

Procedure editing is done on a line by line basis. To edit line 2 in the above procedure type:

> EDIT MFIT 2 CR	to enter Procedure editing mode.
; GETNAME(I) ; TVLOD CR	notice prompt symbol ; for procedure editing mode. This change replaces the old line 2 adding a TVLOD.
; IMXY ; MAXFIT CR	to add a line between the changed line 2 and old line 3.
; GETNAME(I+1) CR	to add yet another line after 2.
; ENDEDIT CR	to terminate procedure editing.
> LIST MFIT CR	

The listing of the modified procedure will give:

```

1  PROC MFIT(I)
2  GETNAME(I) ; TVLOD
3  IMXY ; MAXFIT
4  GETNAME(I+1)
5  TVLOD ; IMXY ; MAXFIT
6  RETURN
7  FINISH

```

To delete line *n* from a procedure, type:

> EDIT <i>xxxxxxx</i> <i>n</i> CR	where <i>xxxxxxx</i> is the name of the procedure.
; ENDEDIT CR	

To insert one or more lines between line 3 and 4 of a procedure, type:

```

> EDIT xxxxxxx 3.5 CR
; (type additional lines as needed.)
; ENDEDIT CR

```

Notice that the lines are renumbered after any EDIT. Use LIST to determine the new line numbers.

The pseudoverb **MODIFY** allows the user to modify characters within a line of a procedure in order to correct that line or to change its meaning. The grammar is:



> **MODIFY** *proc-name* *line-number*                where *proc-name* is the name of the procedure and *line-number* is the line number in the procedure as shown by **LIST**.

**MODIFY** begins by showing the existing line with a ? as a prefix. Then it prompts for input with a ? To keep the character of the original line immediately above the cursor, type a blank (space-bar). To delete that character, type a \$ (dollar-sign). To replace that character, type the new character (to get a new blank character type an @ sign). Insertions complicate things. To insert text prior to the character immediately above the cursor, type a \ followed by the desired text followed by another \. You may continue to **MODIFY** the remainder of the line, but you must remember that the current character position in the old line is to the left of the current cursor position by the number of inserted characters (including the 2 \s). **MODIFY** will display the resulting line of code after you hit a carriage return ( **CR**) and does not change the line number.

Example:

```
> MODIFY ED 2 CR
?TYPE 'THIS IS EDS PROC'
?           MYO\NEW\      OFOROEEXAMPLE' CR
TYPE 'THIS IS MY NEW PROC FOR EXAMPLE'
> MODIFY ED 2 CR
?TYPE 'THIS IS MY NEW PROC FOR EXAMPLE'
?           $$$$ \EDURE,\ CR
TYPE 'THIS IS MY PROCEDURE, FOR EXAMPLE'
```

More information about procedure building and editing can be found by typing:

> **HELP PROCEDURE** **CR**

The procedure **MFIT** can be executed by:

> **MFIT**(*n*) **CR**                                where *n* is the slot number of the appropriate image.

(It is assumed that the correct disk unit number has already been set.) This procedure can also be part of another procedure or put in a loop. For example:

```
> FOR I= 1 TO 10 BY 2; MFIT(I) ; END CR
```

will TV load and fit the maximum for the first ten images on the appropriate disk.

Procedures are lost when you exit from **AIPS**, but they can be saved in **SAVE** files. Procedures can also be stored more permanently in **RUN** files which are described below. To list which procedures are currently in **AIPS**, type:

> **HELP PROCS** **CR**

This will produce a listing of internal **AIPS** procedures as well as those produced by the user.

All of the syntax available in **AIPS** is available for use inside procedures *except for certain pseudoverbs*. These "prohibited" pseudoverbs include **SAVE**, **GET**, **STORE**, **RESTORE**, **PROCEDURE**, **EDIT**, **ENDEDIT**, **MODIFY**, **LIST**, **CORE**, and **SCRATCH**. Others do not make much sense in procedures, including **MSGKILL**, **DEBUG**, and **ABORTASK**. Other pseudoverbs are, however, particularly useful in procedures. These include **TGET**, **TPUT**, and **GO**. Once a procedure is written and edited, it can be stored in a **SAVE** file for later use. Procedures are lost whenever another **GET** file is obtained. To obtain a listing of all of the procedures which are currently in your **AIPS** environment, type:

> **HELP PROCS** **CR**

**12.6. RUN files**

The writing and editing of procedures more than about 10 lines long is cumbersome in AIPS because of its primitive editor. Long procedures are best written in text files at monitor level in the computer where there are good editing facilities. These files can then be transferred to AIPS very simply using the RUN file facility.

Any set of commands which can be typed on the terminal can be stored in a text file. At the present time, the computer's own text editor must be used to generate this file and is thus machine dependent. The text files are generally stored in an area designated by the word RUN. Thus, in VAX/VMS these text files are stored in a subdirectory called [AIPS.RUN] under the AIPS logon. To use the SOS editor to generate text in a RUN area with a file name of XXXXXX (for example), type the following (*on a VAX/VMS system*):

> STORE 2 CR	save AIPS environment.
> QE CR	no-message exit from AIPS.
\$ EDIT/SOS RUNFIL:XXXXXX.RUN CR	create text file XXXXXX.RUN in RUN area.
*00100 — Ed Fomalont 3/81 — CR	first line is ignored by AIPS. It must contain your name to identify the file owner to the System Manager.
*00200 TASK = 'CNVRT' CR	begin executable file.
*00300 INN 'TEST' ; INCL 'CNVRT' CR	
*00400 INSEQ = 0 ; INDISK = 1 CR	
*00500 GO CR	
*00600 ESCAPE	hit escape key.
* ET CR	catalog file and leave editor.
\$ @AIPS CR	get back into AIPS.
Enter AIPS version you want, NEW or OLD: NEW CR	select version.
? nnn CR	enter user number nnn.
> RESTORE 2 CR	get AIPS environment back.
> RUN XXXXXX CR	read and execute file XXXXXX into AIPS. The text will be listed as it is read.
>	continue AIPS in normal mode.

Note that this example applies to VAX/VMS systems only. Consult Appendix Z for advice on your non-VAX/VMS system.

RUN files are a convenient place to store procedures which are used commonly in AIPS since the code can be accessed in AIPS by typing RUN *whatever* CR. It is much easier to type a long procedure using the computer text editor than the AIPS text editor. RUN files are also useful for storing the commands associated with a convenient set of tasks, such as loading a map, sorting the data, making and cleaning the map. All facilities in AIPS such as GET, SAVE, and TGET can be used in RUN files.

**12.7. Batch jobs**

AIPS has a batch processor which can be used to run jobs outside interactive AIPS. The job consists of a set of AIPS instructions which do not need user interaction. This excludes the TV, the Tektronix, and the tape drive oriented tasks and verbs. The Array Processor can be used in batch although interactive users generally have AP priority.

The instructions to be executed in the batch processor are prepared in a workfile. The workfile can be made while in AIPS and detailed instructions are given by typing:

> HELP BATCHJOB CR

A simple example is given here:

> BATQUE = 2 ; BATCLEAR CR

select queue 2 and clear its workfile. There are generally two queues and, if so, jobs which use the array processor cannot be run in queue 1.

> BATCH CR

enter batch preparation mode.

< TASK = 'UVSRT' CR

(notice < prompt.) Begin typing as in AIPS.

< INN = '3C16' ; INCL = 'UVDATA' CR

< INSEQ 1 ; OUTN INN ; OUTCL INCL CR

< OUTSEQ = 0 ; SORT = 'XY' CR

< GO CR

batch AIPS always waits for a task to finish before continuing.

< RUN XXXXX CR

RUN files may be used in a batch job.

< GO CR

< ENDBATCH CR

to leave batch preparation mode type in ENDBATCH spelled out in full.

>

To list a batch file type:

> BATFLINE = 0 ; BATLIST CR

To edit line *n* in a batch file, type:

> BATEDIT *n* CR

< *put text here* CR

replaces old line *n*.

< *some more text* CR

inserted between old lines *n* and *n+1*.

< ENDBATCH CR

spelled out in full.

>

As with procedures, if *n* is an integer, the existing line *n* is overwritten with the line or lines typed before ENDBATCH. If *n* is not an integer, the new lines are simply inserted between lines *n* and *n+1*.

Finally, the workfile can be submitted to the batch processor by typing:

> SUBMIT CR

The instructions are sent to a checking program which checks that the code is free of obvious errors. All RUN files are expanded and checked. If Checker (the task AIPSC*m* where *m* is some hexadecimal number less than F) approves, the job goes into the AIPS job queue.

To submit a batch job outside of the AIPS interactive program, log in to the AIPS area and then type:

**\$ QBATER CR**

and generate the batch work file as shown above. It can be submitted without ever going into AIPS. Note that a different version of this command syntax may be required on your non-VAX/VMS computer.

Batch has several limitations. First, devices which require interactive use (TV device, Tektronix device, and the tape drives) cannot be used in batch. Also, batch uses a different set of TPUT and TGET files. Thus, a TGET in batch does not get the adverbs from your last interactive use of the specified task. However, the AIPS facilities GET and SAVE are particularly useful for batch. You can use interactive AIPS to set up and test set(s) of procedures and adverb values and SAVE them in named files. These files may then be recovered by batch for the routine processing of large sets of data.

At present, batch jobs are run after a delay of about 10 minutes, on a first-come-first-served basis. However, batch jobs which use the array processor are forbidden batch queue number 1 and are postponed until evening in the other queues. After your job has been submitted successfully, type:

> QUEUES CR to list jobs in the queue.

Note the SUBMIT TIME for your job. It will not start before that time.

**12.8 Writing your own programs with POPS**

Users may wish to write their own programs for AIPS. To generate a full, AIPS-standard FORTRAN program requires a considerable amount of additional information and labor. Fortunately, less elaborate programs may be able to be constructed as procedures using existing verbs and tasks.

Consider a more complicated procedure as an example. (This example is presented as if it were typed into an interactive AIPS. However, users will probably prefer to prepare such complicated procedures in RUN files.) We wish to determine the average value and rms scatter at any pixel location in a set of *n* maps. We shall demand that the *n* images all have the same INNAME and INCLASS with sequence numbers between 1 and *n*. The RENAME verb can be used to name the maps appropriately. If we call this procedure:

AVGRMS (PIXXY, N, AVG, RMS)

where

PIXXY	is the pixel location in the images,
N	is the number of images,
AVG	is the average value at the pixel location, and
RMS	is the rms value at the pixel location.

The array adverb PIXXY is a standard AIPS adverb, but the variables N, AVG, and RMS are unknown to AIPS. These must be defined before we can write the procedure AVGRMS. This is done by a short dummy procedure which we will call DAVGRMS:

> PROC DAVGRMS CR	define dummy procedure.
: SCALAR N, AVG, RMS CR	define scalar adverbs.
: FINISH CR	exit from dummy procedure.

Now begin the procedure AVGRMS:

> PROC AVGRMS (PIXXY, N, AVG, RMS) CR	enter procedure building mode.
: SCALAR SUM, SUM2 CR	define more variables.
: ARRAY VAL(20) CR	define an array.
: RMS = 0 ; SUM = 0 ; SUM = 0 CR	zero some variables.
: FOR INSEQ = 1 TO N CR	begin summing loop.

<pre> :   IMVAL C<sub>R</sub> :   VAL(INSEQ) = PIXVAL C<sub>R</sub> :   SUM = SUM + PIXVAL C<sub>R</sub> :   SUM2 = SUM2 + PIXVAL * PIXVAL C<sub>R</sub> :   END C<sub>R</sub> :   AVG = SUM / N C<sub>R</sub> :   IF N &gt; 1.5 THEN C<sub>R</sub> :     RMS = SQRT((SUM2 - N*AVG*AVG) / (N * (N-1))) C<sub>R</sub> :   END C<sub>R</sub> :   TYPE 'AVG=',AVG,'RMS=',RMS,'AT PIXEL',PIXXY C<sub>R</sub> :   TYPE ' # ',' VAL ',' ERROR ' C<sub>R</sub> :   FOR INSEQ = 1 TO N C<sub>R</sub> :     SUM = AVG - VAL(INSEQ) C<sub>R</sub> :     TYPE INSEQ, VAL(INSEQ), SUM C<sub>R</sub> :   END C<sub>R</sub> :   RETURN C<sub>R</sub> :   FINISH C<sub>R</sub> &gt; </pre>	<pre> verb to get pixel value at PIXXY in map INNAME INCLASS INSEQ. to save pixel value (placed in PIXVAL by IMVAL) in our array. sum for averaging. sum for rms. end of FOR loop. get average value. check if N &gt; 1 calculate rms if N &gt; 1. end of IF clause.  print a heading. begin another loop. get residual. print data and residual. end of FOR loop. return from Procedure. return to regular AIPS mode. </pre>
---	---

The above procedure could be run in the following way. First fill in the adverbs **INNAME**, **INCLASS** and **PIXXY** with the desired values. Then type:

<pre> &gt; AVGRMS (n, PIXXY, AVG, RMS) C<sub>R</sub> </pre> <p>The calculation of the average and rms will take place and be written on the terminal and on the message file.</p>	<pre> where n is the number of images to average. </pre>
---	--

This procedure could be used by another procedure. Suppose we wanted to determine the average and rms of the pixels within a rectangular area. If we set **BLC** and **TRC** in the usual way to define the rectangular boundary, then the procedure:

<pre> &gt; PROC AVGARRAY (BLC, TRC) C<sub>R</sub> :   FOR I = BLC(1) TO TRC(1) C<sub>R</sub> :     FOR J = BLC(2) TO TRC(2) C<sub>R</sub> :       PIXXY = I, J C<sub>R</sub> :       AVGRMS (PIXXY, N, AVG, RMS) C<sub>R</sub> :     END C<sub>R</sub> :   END C<sub>R</sub> :   RETURN C<sub>R</sub> :   FINISH C<sub>R</sub> &gt; </pre>	<pre> define new proc. loop over x-coordinate loop over y-coordinate. set pixel coordinates for AVGRMS.  end y loop. end x loop. return to calling program. end of proc. </pre>
--	---

will calculate the average value and rms at this array of pixel locations.

## 13. Current AIPS software

The complete list of software in *AIPS* is kept up-to-date in certain special **HELP** files. The list of all such special **HELP** files can be obtained by typing:

> **HELP** *C<sub>R</sub>*

Among these is a very long alphabetical list of all tasks, verbs, pseudoverbs and **AIPS** built-in procedures. This may be scanned at the terminal by typing:

> **HELP INDEX** *C<sub>R</sub>*

Shorter lists of the software, grouped according to usage, are also given in special **HELP** files. These lists are presented below, followed by the long **INDEX** file. The name of the special **HELP** file is shown in bold type above each group. Each line within a group lists a task (T), verb (V), pseudoverb (PV and pV), or procedure (P) with a very brief description of its function. Pseudoverbs come in two flavors: those that act roughly like verbs (pV) and those that must be treated specially (PV).

> **HELP** *name* *C<sub>R</sub>* where *name* is one of these entries

will give more useful information about that task, verb, pseudoverb, or procedure. References to Sections in the *AIPS COOKBOOK* are given.

### TAPU

Type: Reading from, writing on, and positioning tapes

#### General tape handling

<b>AVFILE</b>	V	Advance or backspace files on a tape	§ 6.1
<b>AVMAP</b>	V	Advance maps on a tape	§ 6.1
<b>REWIND</b>	V	Rewind a tape	§ 3.1.2, 3.2, 6.1
<b>AVEOT</b>	V	Advance tape to end-of-information	§
<b>PRTTP</b>	T	Print summary of a tape; any recognized format	§ 3.1.1
<b>AVTP</b>	T	Reposition tape	§

#### Reading and writing FITS-format tape of maps or UV data

<b>TPHEAD</b>	V	Give info on tape data at current position	§ 6.1
<b>IMLOD</b>	T	Copy an image to disk from FITS or IBM tape	§ 6.1
<b>UVLOD</b>	T	Copy UV data to disk from FITS tape	§ 3.1.2
<b>FITTP</b>	T	Copy a maps or UV data to FITS-tape from disk	§ 10.1
<b>IBMTTP</b>	T	Copy an image to an IBM-tape from disk	§

#### Reading and writing EXPORT-format tape of UV data

<b>EXIND</b>	T	List index from an EXPORT tape	§ 3.1.1
<b>EXPND</b>	T	Copy EXPORT tape to a tape with one source per file.	§
<b>UVLOD</b>	T	Copy UV data to disk from an EXPORT tape	§ 3.1.2, 9.1
<b>UVEXP</b>	T	Copy UV data to an EXPORT tape from disk	§ 10.1

#### Data analysis

<b>EXFND</b>	T	List UV data directly from an EXPORT tape	§
<b>APMAP</b>	T	Make map(s) directly from an EXPORT tape	§ 3.2

## UVPR

Type: UV processing

### Index, write and read EXPORT and uv-FITS tapes

EXIND	T	List index from an EXPORT tape	§ 3.1.1
EXFND	T	List data from an EXPORT tape	§
EXPND	T	Copy EXPORT tape, one source/file	§
FITTP	T	Copy uv data from disk to uv-FITS tape	§ 10.1
PRITP	T	Print summary of tapes, any recognized format	§ 3.1.1
UVEXP	T	Copy from disk uv data to an EXPORT tape	§ 10.1
UVLOD	T	Copy uv data from EXPORT or uv-FITS tape to disk	§ 3.1.2, 9.1
WSLOD	T	Load uv data from WSRT tape to AIPS disk format	§

### UV data manipulation

DBCON	T	Concatenate UV data bases	§
UVCOP	T	Copy part of a UV data base	§
AVER	T	Average UV data in time	§
DESCM	T	Reduce size of a UV file	§
UVSRT	T	Sort UV data on disk	§ 3.1.3
UVFLG	T	Flag selected UV data	§ 5.3
CLIP	T	Flag UV data above a given intensity	§ 5.3

### UV processing

APMAP	T	Make map(s) from EXPORT tape	§ 3.2
UVMAP	T	Make map(s) from disk file	§ 3.1.4, 9.2
UVSUB	T	Subtract point components from UV data	§ 5.3
ASCAL	T	Self-cal uv data	§ 5.2
ASCOR	T	Apply ASCAL results	§
VBFIT	T	Global fringe fitting for VLBI	§
VBCOR	T	Apply results of VBFIT	§
VBCAL	T	Apply antenna scale factors to data	§
FFT	T	Make Fourier transform of an image	§

### Display of UV data

PRTUV	T	Print selected UV data	§
UVFND	T	Print selected UV data (similar to EXFND)	§ 5.3
PRTAN	T	Print antenna files	§
UVPLT	T	Plot UV data in a variety of ways ⇒ plot file	§ 5.3

### Miscellaneous UV processing

CORER	T	Calculate UV correlator statistics	§
CORFQ	T	Correct UV data header for incorrect frequency	§
FUDGE	T	Template task to work on UV data	§

## MAPETC

Type: Mapping, cleaning and Self-calibration

### Mapping

AFMAP	T	Make a map directly from the EXPORT tape	§ 3.2
UVMAP	T	Make map from sorted data on disk	§ 3.1.4
H2MEM	T	Make a map using the maximum entropy method	§
FFT	T	Take the Fourier transform of an image	§
UVDIS	T	Display the Fourier transform on the TV	§

### Cleaning

APCLN	T	Clean an image using the Array Processor	§ 5.1
PHCLN	T	Clean an image (Cornwell modification)	§
CCMOD	T	Prepare a clean component file for planets	§
PRTCC	T	Print selected parts of clean components file	§ 5.1.2

### Self-calibration

ASCAL	T	Self-calibrate a UV data set using a clean image	§ 5.2
ASCOR	T	Apply ASCAL gain solutions to another file. Useful for spectral line	§
GNPLT	T	General plot routine for ASCAL gain solution	§
VBCOR	T	Apply VBFIT results to other data	§
VBFIT	T	Global fringe fitting for VLBI	§

## APTASKS

Type: Tasks which use the Array Processor

APCLN	T	Clean an image	§ 5.1
AFMAP	T	Make a map directly from the EXPORT tape	§ 3.2
ASCAL	T	Self-calibrate a UV data set using a clean image	§ 5.2
CONVL	T	Convolve with a Gaussian or another image	§ 9.5
FFT	T	Take the Fourier transform of an image	§
H2MEM	T	Make a map using the maximum entropy method	§
INTERP	T	Interpolate an image with splines	§
PHCLN	T	Clean an image (Cornwell modification)	§
UVDIS	T	Display FFT of map on TV	§
UVMAP	T	Make a map from sorted data on disk	§ 3.1.4
UVSUB	T	Subtract point components from UV data	§ 5.3
VBFIT	T	Global fringe fitting for VLBI	§



**GENERAL**

Type: General AIPS utilities

HELP	pV	Provide information for verbs, tasks, adverb.	§ 4.5
INPUTS	pV	Display current parameters for a task or verb	§ all
INP	pV	Same as INPUTS except not written on message file	§ all
DISKUSE	V	List all disk space used in AIPS	§ 4.4, 11.6
SAVE	PV	Save AIPS environment in a semi-permanent file	§ 4.3, 12.1
GET	PV	Retrieve a SAVE file	§ 4.3, 12.1
TGET	pV	Get adverb values used in previous task execution	§ 4.3
TPUT	pV	Save adverb values for later retrieval by TGET	§ 4.3
STORE	PV	Save AIPS environment in a temporary file	§ 2.5, 12.1
RESTORE	PV	Retrieve a STORE file	§ 2.5, 12.1
SGINDEX	V	List all SAVE / GET files	§
TGINDEX	V	List all TPUT / TGET entries	§
PRINT	V	Print the following keyword value(s)	§
TYPE	V	Print the following keyword value(s)	§ 12.3
READ	V	Read values of the following keyword(s)	§
PRTMSG	V	Print the user's messages in the message file	§ 3.1.2, 4.1, 9.1
MSGKIL	pV	Stop all messages from being saved	§
PRTHI	V	Print the history file of an image	§ 3.1.4, 5.1
RESTART	V	Restart AIPS and print message file	§
EXIT	V	Exit from AIPS and print the message file	§ 2.3, 10.4
QEXIT	P	Quick exit from AIPS with no print out	§ 2.3, 10.4
GO	pV	Starts a shed task and passes parameters to it	§ all
ABORTASK	pV	Aborts specified task, removes scratch files	§ 7.1, 11.1
SPY	V	Inquire which tasks are active	§ 4.4, 11.1
WAITTASK	V	Suspends AIPS until specified task is done	§ 9.1
PROC	PV	Begin building a procedure (See HELP PROC)	§ 12.5
PROCEDUR	PV	Begin building a procedure (See HELP PROC)	§ 12.5
RUN	PV	Execute a text file generated in the RUN area	§ 12.6
BATCH	V	Begin building a batch file (See HELP BATCHJOB)	§ 12.7
SUBMIT	V	Submits specified batch queue into execution	§ 12.7
GRIPE	V	Enter a complaint/suggestion in the Gripe file	§ 10.3
GRINDEX	V	Index of all entries in the current Gripe file.	§
GRLIST	V	List any one of the entries in the Gripe file.	§

## CATINFO

Type: Dealing with the image catalog

CATALOG	V	List all images in catalog consistent with inputs	§ 4.2
MCAT	V	List all map images in catalog	§ 3.1.4, 4.2
UCAT	V	List all UV images in catalog	§ 3.1.1, 4.2
CLRSTAT	V	Clear WRIT or READ status from catalog	§ 4.2.3
TPHEAD	V	List image header from tape	§ 6.1
IMHEADER	V	List the contents of an image header	§ 4.2.4
GETNAME	V	Fill INNAME <i>et al.</i> from slot number of INDISK	§ 4.2.2
GET2NAME	V	Fill IN2NAME <i>et al.</i> from slot number of IN2DISK	§ 4.2.2
GET3NAME	V	Fill IN3NAME <i>et al.</i> from slot number of IN3DISK	§ 4.2.2
CLRNAME	V	Fill INNAME <i>et al.</i> with null values	§ 3.1.4, 4.2
TVNAME	V	Set INNAME <i>et al.</i> to that under TV cursor	§
RENAME	V	Rename file	§ 4.2.1, 9.3
RESCALE	V	Rescale file using header scale factor and offset	§
SGINDEX	V	List all SAVE / GET files	§
TGINDEX	V	List all TGET files	§
EXTLIST	V	List extension file information	§ 7.3, 7.4
CORFQ	T	Correct UV data header for incorrect frequency	§
AXDEFINE	V	Redefine axis values in header	§ 9.3

## TVGEN

Type: General TV functions for loading and display

TVINIT	V	Initialize TV	§
TVCLEAR	V	Erase entire TV channel	§
TVLOD	V	Load image into TV channel	§ 9.5, 12.5
TVON	V	Turn on TV channel	§ 6.2
TVOFF	V	Turn off TV channel	§
IMERASE	V	Erase image on a portion of a TV channel	§
GRCLEAR	V	Erase entire graphic channel	§
GRON	V	Turn on specified graphic channel	§
GROFF	V	Turn off specified graphic channel	§
TVALL	V	General purpose interactive TV display	§ 6.2
TVFIDDLE	V	Same as TVALL with no TV load	§ 6.2
TVMOVIE	V	Load planes of cube, run movie	§ 9.5
UVDIS	T	Display FFT of an image	§

**TVINTER**

Type: Interactive use of the TV display

## Loading of TV image

TVINIT	V	Initialize TV	\$
TVCLEAR	V	Clear TV channel	\$
TVLOD	V	Load an image on a TV channel	\$ 9.5, 12.5
TVALL	V	General purpose interactive TV display	\$ 6.2
TVMovie	V	Load planes of cube, run movie	\$ 9.5

## Image transfer function (contrast) and wedges

TVTRANS	V	Modify transfer function for image	\$ 6.2.2
OFFTRANS	V	Reset normal transfer function for image	\$
IMWEDGE	V	Display image wedge from mapmin to mapmax	\$
TVWEDGE	V	Display image wedge, range as loaded	\$
IMERASE	V	Erase image (or wedge) from TV	\$
TVALL	V	Provides contrast and wedges	\$ 6.2
TVFIDDLE	V	Same as TVALL except no TV load	\$ 6.2
TVHUEINT	V	Use 2 images for hue/intensity with enhancement	\$ 9.5
OFFHUINT	P	Clear the hue/intensity display	\$
TVLUT	V	Set n-point linear transfer function	\$
TVMLUT	V	Set many point linear transfer function	\$

## Zooming

TVZOOM	V	Activate hardware zoom on TV	\$
OFFZOOM	V	Deactivate hardware zoom on TV	\$
TVALL	V	Provides zooming capability	\$ 6.2
TVFIDDLE	V	Same as TVALL except no TV load	\$ 6.2

## Scrolling

TVSCROL	V	Activate hardware scroll on TV	\$
OFFSCROL	V	Deactivate hardware scroll on TV	\$

## Blinking

TVBLINK	V	Automatic blink between two channels	\$
TVMBLINK	V	Manual blink between two channels	\$
TVMovie	V	Load planes of subimage, run movie	\$ 9.5
REMOVIE	V	Re-run preloaded movie	\$ 9.5

## Roaming--using several channels for one image

TVROAM	V	Load image on several channels, initiate scroll and zoom	\$
SETROAM	V	Fill in roam parameters, then roam	\$
REROAM	V	Same as TVROAM with no loading	\$
OFFROAM	P	Initialize TV after roaming	\$

## Other displays

---

CURVALUE	V	Display pixel position and value under cursor	§ 6.3
TVNAME	V	Print name of image under the cursor	§
TVLABEL	V	Label image or wedge with coordinate axes	§
GRCLEAR	V	Clear image or label coordinate axes	§

## CURSOR

Type: Interactive use of cursor on TV and TEK

IMPOS	V	Find absolute position of cursor on TV	§ 6.3
TKPOS	V	Find absolute position of cursor on TEK	§
IMXY	V	Find image pixel location of cursor on TV	§ 8.3.1
TKXY	V	Find image pixel location of cursor on TEK	§
IMXY; IMVAL	P	Find intensity at cursor location on TV	§
TKXY; IMVAL	P	Find intensity at cursor location on TEK	§
TVPOS	V	Find TV pixel location of cursor on TV	§
TVNAME	V	Display name of image under the cursor	§
TKVAL	V	Find intensity from slice plot using cursor	§
CURBLINK	V	Blink cursor or return to steady cursor	§
CURVALUE	V	Display location and intensity under cursor	§ 6.3
TVWINDOW	V	Set BLC and TRC using cursor, graphics display	§ 6.3
TVBOX	V	Set cleaning boxes with graphics display	§ 6.3
SETBOX(I)	P	Set I <sup>th</sup> cleaning box using cursor	§
SETNBOX(I)	P	Set I cleaning boxes using cursor	§
SETWIN	P	Set BLC and TRC using cursor	§
SETXWIN(I, J)	P	Set rectangular box $I \times J$ around cursor location	§ 6.3

## TVCOLOR

Type: Using color on the TV

TVINIT	V	Initialize TV	§
TVPSEUDO	V	Enable several type of pseudo coloring	§ 6.2.1
OFFPSEUD	V	Disable pseudo coloring	§
TVALL	V	Load,color,contrast and zoom TV display	§ 6.2
TVFIDDLE	V	Like TVALL except no load	§ 6.2
TV3COLOR	V	Produce false color display for 3 image planes	§ 9.5
TVHUEINT	V	Produce Hue-intensity display from 2 image planes	§ 9.5
OFFHUINT	P	Turn off the hue-intensity display	§
RGBMP	T	Create 3 planes for false color from data cube	§ 9.5
IMLHS	T	Produce Intensity/Hue/Saturation image from 3 maps	§

## PL2D

Type: Two-dimensional displays and plot files

CNTR	T	Contour diagram $\Rightarrow$ plot extension file	§ 7.1
PCNTR	T	Contour plus polarization line vector $\Rightarrow$ plot file output	§ 7.2
GREYS	T	Greyscale image plus contour $\Rightarrow$ plot file output	§
PROFL	T	Profile (perspective) diagram $\Rightarrow$ plot file output	§ 9.5
TKPL	T	Display plot file on the TEK	§ 7.1, 7.4, 8.2
PRTPL	T	Display plot file on the line printer	§ 7.1, 7.4, 8.2, 8.3.3
TVPL	T	Display plot file on the TV	§ 7.1, 7.4
EXTLIST	V	List extension files associated with an image	§ 7.3, 7.4
EXTDEST	V	Destroy extension files	§ 7.3
TVLOD	V	Load an image onto specified TV channel(s)	§ 9.5, 12.5
TVALL	V	Load an image onto TV and interact with display	§ 6.2
TVROAM	V	Load a large image onto several TV channels and roam	§
PRTIM	T	Print image intensities on the line printer	§ 5.1

## SL1D

Type: One-dimensional displays and analysis

### Create slice and display

SLICE	T	Store intensity along a line segment in extension file	§ 7.4
TKSLICE	V	Initialize TEK and display slice file	§ 7.4, 8.3.3
TKASLICE	V	Superpose slice file to existing display on TEK	§ 7.4
SL2PL	T	Convert slice file to a plot file for general display devices	§ 7.4, 8.3.3

### Fitting model to slices

SLFIT	T	Fit model guess to a slice file	§ 8.3.3
TKSET	V	Use TEK cursor to guess initial Gaussian model for slice	§ 8.3.3
TK1SET(I)	V	Use cursor on TEK to change guess of $I^{\text{th}}$ component	§
TKGUESS	V	Initialize TEK and display guess	§
TKAGUESS	V	Superpose guess to existing plot on TEK	§ 8.3.3
TKMODEL	V	Initialize TEK and display fitted model	§
TKAMODEL	V	Superpose fitted model to existing display on TEK	§ 8.3.3
TKRESID	V	Initialize TEK and display fitted model-slice	§
TKARESID	V	Superpose fitted model-slice to existing display on TEK	§ 8.3.3

### Utilities

EXTLIST	V	List extension files associated with an image	§ 7.3, 7.4
EXTDEST	V	Destroy extension files	§ 7.3

## ANALYSIS

Type: Image processing, analysis and combination

### Image processing

CONVL	T	Convolve with a gaussian or another image	§ 9.5
GEOM	T	Stretch, rotate and displace an image	§
INTERP	T	Interpolate an image to a new grid	§
PBCOR	T	Correct a map for the primary beam attenuation	§
SMOTH	T	Smooth an image of up to 7-dimensions	§ 9.5
SUBIM	T	Create a new image from subsection of an image	§
TRANS	T	Transpose an image	§ 9.4

### Image analysis

IMEAN	T	Intensity statistics, flux density and histogram	§ 8.2
IMFIT	T	Fit Gaussian components to a portion of an image	§ 8.3.2
IMVAL	V	Get image intensity at specified pixel	§
MAXFIT	V	Determine intensity and position of a maximum.	§ 8.3.1
MOMFT	T	Fit moments to a portion of an image	§
MOMNT	T	Finds maps of moments 0 to 4 from a data cube	§ 9.5
SLFIT	T	Fit Gaussian components to a 1-D slice	§ 8.3.3

### Image combination

COMB	T	Image combination with many algorithms	§ 8.1, 9.5
CORMS	T	Image combination with error analysis	§
RM	T	Rotation Measure and B field from set of images	§
SUMIM	T	Add together many images	§

## DELETE

Type: Deleting files

ZAP	V	Delete an image and all extension files	§ 3.1.3, 10.2, 12.4
ALLDEST	V	Delete all images consistent with the inputs	§ 10.2
TIMDEST	V	Delete all images older than time parameters	§ 4.4
SCRDEST	V	Delete a scratch file associated with a task	§ 4.4
EXTDEST	V	Delete an extension file	§ 7.3
SAVDEST	V	Delete all <b>SAVE</b> / <b>GET</b> files	§
SGDEST	V	Delete specified <b>SAVE</b> / <b>GET</b> file	§

## CUBE

Type: Data cube (spectral line) processing

Note that most other programs support data cube processing in a general way.

AXDEFINE	V	Redefine axis values in the header	§ 9.3
CNVRT	T	Convert between I*2 and R*4 images	§ 9.4
COMB	T	Image combination including cube with a plane	§ 8.1, 9.5
HANSM	T	Hanning smooth first axis of cubes	§ 9.5
MCUBE	T	Build an n-dim image from set of (n-1) dim images	§ 9.3
MOMNT	T	Calculate 4 moment maps from a cube	§ 9.5
REMOVE	V	Re-run preloaded sequence of planes on TV	§ 9.5
RGBMP	T	Create a cube with three planes for TV3COLOR use	§ 9.5
SMOTH	T	Smooth an image, up to 7-dimensions	§ 9.5
SUBIM	T	Create a new image from subsection of an image	§
SUMIM	T	Add together many images	§
TRANS	T	Transpose an image	§ 9.4
TV3COLOR	T	False color TV display from three image planes	§ 9.5
TVHUEINT	T	Hue-intensity TV display using two image planes	§ 9.5
TVMOVIE	V	Load planes to TV, display in sequence	§ 9.5

## VLBI

Type: Software dealing specifically with VLB data

AVER	T	Time averages data	§
CITCC	T	Writes CC file in CIT package format	§
FITIP	T	Write uv-FITS tape from AIPS uv data files	§ 10.1
PRDR	T	Prints contents of fringe fitting (DR) file	§
STRIP	T	Antenna scaling for VLB	§
TOAIP	T	Reads uv data and converts to AIPS data set	§
TOVLB	T	Converts AIPS UV data to CITMERGE data set	§
UVEXP	T	Write EXPORT tape from VLB and other disk data	§ 10.1
UVLOD	T	Write disk data from normal and VLB EXPORT tapes	§ 3.1.2, 9.1
VBCAL	T	Multiplies amplitudes by antenna scale factors	§
VBC	T	Writes CC file in CIT package format	§
VBCOR	T	Applies the result of VBFIT to other data	§
VBEXP	T	Write EXPORT tape from VLB disk data	§
VBFIT	T	Global fringe fitting task	§
VBLOD	T	Write VLB disk data from EXPORT tape	§
VBMRG	T	Merges a VLBI data set including polarization	§
VLDR	T	Estimates antenna delays and rates from NRAO-FRIDGE output and writes control table for FRIDGE	§

## POPSYM

Type: Symbols used in the POPS interpretive language

VERB	USE	COMMENTS	§
--- Arithmetic expressions ---			
+	A + B	Add the expression A to B	§ 12.1
-	A - B	Subtract the expression B from A	§
*	A * B	Multiply the expression A with B	§ 12.1
/	A / B	Divide the expression A by B	§ 12.3
**	A ** B	Calculate A to the power B	§
( )	(A+B)*C	Grouping expressions as desired	§
=	A = B	Store the value of B into A	§ 12.1
,	A = 3,5,4	Separator of elements in an array	§ 12.1
:	TO	Equivalent to the verb TO	§
;		Separator between POPS statements	§ 12.1
--- Logical expressions ---			
>	A > B	A greater than B	§ 12.8
<	A < B	A less than B	§
=	A = B	A equal B	§
>=	A >= B	A equal to or greater than B	§
<=	A <= B	A equal to or less than B	§
<>	A <> B	A not equal to B	§
!	A ! B	A or B	§
&	A & B	A and B	§
^	~ A	not A	§
--- Looping constructions ---			
FOR TO BY END	FOR I = A TO B BY C		§ 6.1
	<i>any valid set of POPS syntax</i>		§ 9.1, 9.3
	END		§ 12.4, 12.8
WHILE END	WHILE <i>any logical expression</i>		§
	<i>any valid set of POPS syntax</i>		§
	END		§
IF THEN ELSE END	IF <i>any logical expression</i>		§ 12.8
	THEN <i>any valid set of POPS syntax</i>		§
	ELSE <i>any valid set of POPS syntax</i>		§
	END		§
--- Built-in functions ---			
ATAN	X = ATAN(A)	Arctangent (result in degrees)	§ 12.3
ATAN2	X = ATAN2(B,A)	Arctangent (result in degrees)	§
COS	X = COS(A)	Cosine (A in degrees)	§
SIN	X = SIN(A)	Sine (A in degrees)	§
TAN	X = TAN(A)	Tangent (A in degrees)	§



EXP	X = EXP(B)	Exponential	§
LN	X = LN(B)	Log base <i>e</i>	§
LOG	X = LOG(B)	Log base 10	§
SQRT	X = SQRT(B)	Square-root	§
MODULUS	X = MODU(Q,U)	Root-square sum: $X = \sqrt{Q^2 + U^2}$	§
MOD	X = MOD(A,B)	A - floor (A/B) * B, i.e., remainder of A/B	§

— Procedure building verbs —

PROC	PV	Begin building a procedure	§ 12.5
PROCEDUR	PV	Begin building a procedure	§ 12.5
LIST	PV	List a procedure	§ 12.5
EDIT	PV	Edit a procedure	§ 12.5
ENDEDIT	PV	End editing a procedure	§ 12.5
MODIFY	PV	Modify a line in a procedure	§ 12.5
RETURN	V	Last statement in a procedure	§ 12.5
FINISH	PV	End procedure building	§ 12.5

— Variable declarations —

SCALAR	pV	Declare scalars	§ 12.8
ARRAY	pV	Declare arrays	§ 12.8
STRING	pV	Declare strings	§

— Input/Output functions —

PRINT	V	Print the following keyword value(s)	§
TYPE	V	Print the following keyword value(s)	§ 12 3, 12.8
READ	V	Read value(s) from terminal after # prompt	§

— Other information —

CORE	PV	Amount of core left in <i>POPS</i>	§
DEBUG	pV	Debug: turns on compiler debug information	§
DUMP	V	Dump K array on terminal screen	§
SCRATCH	PV	Remove a procedure in <i>POPS</i>	§

## INDEX

Type: Alphabetical listing of all verbs (V), tasks (T), Pseudoverbs (PV) and built-in procedures (P) currently in *AIPS*.

Use: All of the following keywords initiate action in *AIPS* by causing a program or set of programs to execute. A more complete description of each keyword can be obtained from the additional **HELP** files indicated in the last column below or from the **HELP** file associated with the keyword.

KEYWORD	TYPE	DESCRIPTION	§
— A —			
ABORTASK	pV	Abort a task	§ 7.1, 11.1
ALLDEST	V	Destroy many files	§ 10.2
APCLN	T	Clean a map	§ 5.1
APMAP	T	Make map from tape	§ 3.2
ASCAL	T	Self-cal UV data	§ 5.2
ASCOR	T	Apply <b>ASCAL</b> results	§
ARRAY	pV	Array declaration	§ 12.8
ATAN	V	Arctangent (half-circle, result degrees)	§ 12.3
ATAN2	V	Arctangent (full-circle, result degrees)	§
AVER	T	Average BT sorted UV data	§
AVEOT	V	Advance tape to end of data	§
AVFILE	V	Advance Files	§ 6.1
AVMAP	V	Advance maps on IBM tape	§ 6.1
AXDEFINE	V	Define axis header	§ 9.3
— B —			
BAMODIFY	V	Modify a line in work area	§
BATCH	V	Enter batch mode	§ 12.7
BATCLEAR	V	Clear batch work area	§ 12.7
BATEDIT	V	Edit batch work area	§ 12.7
BATLIST	V	List batch work area	§ 12.7
BY	V	<b>POPS</b> ( <b>FOR TO BY</b> ) loop	§ 12.4, 12.5
— C —			
CATALOG	V	List specific contents	§ 4.2
CCMOD	T	Planetary clean components	§
CITCC	T	Write <b>CC</b> in <b>CIT-VLB</b> format	§
CLIP	T	Flag UV data above value	§ 5.3
CLRNAME	V	Set <b>INNAME</b> etc. to null	§ 3.1.4, 4.2
CLRSTAT	V	Clear catalog status	§ 4.2.3
CNTR	T	Produce contour plot file	§ 7.1
CNVRT	T	Convert map between <b>I*2</b> and <b>R*4</b>	§ 9.4
COMB	T	Combining two maps	§ 8.1, 9.5
COMPRESS	PV	Not implemented	§
CONVL	T	Convolve two images	§ 9.5

CORER	T	UV correlator statistics	§
CORE	PV	Available space in <i>POPS</i>	§
CORMS	T	COMB with error analysis	§
COS	V	Cosine (arg in degrees)	§
CORFQ	T	Correct observing freq.	§
CURBLINK	V	Turn cursor blink on/off	§
CURVALUE	V	Display value under cursor	§ 6.3

— D —

DBCON	T	Concatenate two UV files	§
DEBUG	pV	Turn on compiler debug	§
DESCM	T	Reduce size of UV file	§
DISKUSE	V	List of disk use	§ 4.4, 11.6
DUMP	V	Dump <i>POPS</i> K array	§

— E —

EDIT	PV	Begin editing procedure	§ 12.5
END	V	End of <i>POPS</i> loop	§ 6.1, 9.3, 12.4
ENDEDIT	PV	End editing procedure	§ 12.5
ELSE	pV	<i>POPS</i> IF THEN ELSE	§
EXFND	T	List UV data from tape	§
EXIND	T	Index UV data from tape	§ 3.1.1
EXIT	V	Exit from AIPS	§ 2.3, 10.4
EXP	V	Exponentiation	§
EXPND	T	Copy Exp. tape, 1 source/file	§
EXTDEST	V	Destroy extension file	§ 7.3
EXTLIST	V	List extension files	§ 7.3, 7.4

— F —

FFT	T	Fourier transform of map	§
FINISH	PV	Leave procedure building	§ 12.5
FITTP	T	Write image on tape	§ 7.5, 10.1
FOR	V	<i>POPS</i> FOR TO BY loop	§ 6.1, 9.3, 12.4
FUDGE	T	UV data template task	§

— G —

GEOM	T	Map interpolation	§
GET	PV	Get <b>SAVE</b> file of AIPS envir.	§ 4.3, 12.1
GETNAME	V	Insert map name by cat. number	§ 4.2.2
GET2NAME	V	Insert map name by cat. number	§ 4.2.2
GET3NAME	V	Insert map name by cat. number	§ 4.2.2
GNPLT	T	Generate Self-cal plot file	§
GO	pV	Initiate a task	§ all
GRCLEAR	V	Clear graphic channel(s)	§
GREYS	T	Construct a grey-scale plot	§
GRINDEX	V	List Gripe file contents	§
GRIPE	V	Enter a complaint/suggestion	§ 10.3
GRLIST	V	Full list of one Gripe	§

---

GROFF	V	Turn off graphic channel(s)	§
GRON	V	Turn on graphic channel(s)	§
— H —			
H2MEM	T	Make MEM map from data	§
HANSM	T	Hanning smooth x-axis	§ 9.5
HELP	PV	List HELP file on terminal	§ 4.5
— I —			
IBMTP	T	Write map on tape for IBM	§
IF	pV	POPS IF THEN ELSE	§ 12.8
IMEAN	T	Obtain various map statistics	§ 8.2
IMERASE	V	Selective image erase from TV	§
IMFIT	T	§ 2-D Gaussian fitting of map	§ 8.3.2
IMHEADER	V	Display image header	§ 4.2.4
IMLHS	T	I-H-S TV display of 3 images	§
IMLOD	T	Load image from tape to disk	§ 6.1
IMPOS	V	Obtain cursor position	§ 6.3
IMVAL	V	Obtain intensity under cursor	§
IMWEDGE	V	Put wedge on TV (mapmin-max)	§
IMXY	V	Obtain cursor pixel position	§ 8.3.1
INP	pV	List inputs for verb or task	§ all
INPUTS	pV	List inputs for verb or task	§ all
— J —			
JOBLIST	V	List batch jobs in queue	§
— L —			
LIST	V	List a procedure	§ 12.5
LN	V	Natural log	§
LOG	V	Base 10 log	§
— M —			
MAXFIT	V	Parabolic fit of image peak	§ 8.3.1
MCAT	V	Index of maps in catalog	§ 3.1.4, 4.2
MCUBE	T	Build a data cube	§ 9.3
MOD	V	Mod function (i.e. remainder)	§
MODIFY	PV	Modify a line in a procedure	§ 12.5
MODULUS	V	Modulus ( $\sqrt{A^2 + B^2}$ )	§
MOMFT	T	Find moments of a component	§
MOMNT	T	Finds moments 0 to 4 of cube	§ 9.5
MSGKIL	pV	Stop messages from file	§
— N —			
NTERP	T	Map interpolation	§

— O —

OFFPSEUD	V	Turn off TV pseudo color	§
OFFROAM	P	Turn off TV roam mode	§
OFFSCROL	V	Turn off TV scrolling	§
OFFTRANS	V	Turn off TV transfer function	§
OFFZOOM	V	Turn off TV zooming	§

— P —

PBCOR	T	Correct for primary beam attenuation	§
PCNTR	T	Contour and pol line vector	§ 7.2
PHCLN	T	Modified CLEAN by Cornwell	§
PRINT	V	Print the following value	§
PROC	PV	Define procedure	§ 12.5
PROCEDUR	PV	Define procedure	§ 12.5
PROFL	T	Produce profile plot file	§ 9.5
PRAN	T	Print antenna file	§
PRICC	T	Print clean compl file	§ 5.1.2
PRIDR	T	Print VLB fringe fit file	§
PRTHI	V	Print history file	§ 3.1.4, 5.1
PRTIM	T	Print digital display of map	§ 5.1
PRMSG	V	Print message file	§ 3.1.2, 4.1, 9.1
PRTPL	T	Print plot file on printer	§ 7.1, 7.4, 8.2
PRTPP	T	Print summary of a tape	§ 3.1.1
PRUV	T	Print selected UV data	§

— Q —

QEXIT	P	Exit from AIPS with no messages	§ 2.3, 10.4
QUEUES	V	List batch job queue	§ 12.7

— R —

READ	V	Read terminal inputs	§
REMOVIE	V	Rerun previously loaded TV movie	§ 9.5
RENAME	V	Rename a file	§ 4.2.1, 9.3
REROAM	V	Reenter roam mode on TV	§
RESCALE	V	Rescale map values	§
RESTART	V	Restart AIPS from scratch	§
RESTORE	PV	Get STORE file of AIPS envir.	§ 2.5, 12.1
RETURN	V	Last statement in procedure	§ 12.5
REWIND	V	Rewind a tape	§ 3.1.2, 3.2, 6.1
RGBMP	T	Create RED, GREEN, BLUE cube	§ 9.5
RM	T	Determine Rotation Measure	§
RUN	PV	Execute text in RUN file	§ 12.6

— S —

SAVDEST	V	Destroy a save file	§
SAVE	PV	Permanent storage of AIPS	§ 4.3, 12.1

SCALAR	pV	Scalar declaration in <i>POPS</i>	§ 12.8
SCRATCH	PV	Remove procedures in <i>POPS</i>	§
SCRDEST	V	Scratch temporary AIPS files	§ 4.4
SETBOX	P	Fill any BOX using TV cursor	§
SETNBOX	P	Fill BOXES using TV cursor	§
SETROAM	V	Fill in ROAM parameters	§
SETWIN	P	Fill BLC, TRC using cursor	§
SETXWIN	P	Fill BLC, TRC using cursor and width	§ 6.3
SGDESTR	pV	Destroy SAVE / GET files	§
SGINDEX	V	List SAVE / GET files	§
SIN	V	Sine (in degrees)	§
SL2PL	T	Convert slice file to plot file	§ 7.4, 8.3.3
SLFIT	T	§ 1-D Gaussian fit to slice file	§ 8.3.3
SLICE	T	Make a slice of an image	§ 7.4
SMOTH	T	Smooth an image, up to 7 dim.	§ 9.5
SPY	V	Interrogation of system	§ 4.4, 11.1
SQRT	V	Square-root	§
STRING	pV	Declare string in <i>POPS</i>	§
STRIP	T	Antenna scaling for VLB	§
STORE	PV	Temporary storage of AIPS	§ 2.5, 12.1
SUBIM	T	Create a subimage	§
SUBMIT	V	Submit batch file to queue	§ 12.7
SUMIM	T	Sum or average many images	§
— T —			
T1VERB	V	Spare verb for development	§
T2VERB	V	Spare verb for development	§
T3VERB	V	Spare verb for development	§
T4VERB	V	Spare verb for development	§
TAN	V	Tangent (in degrees)	§
THEN	pV	<i>POPS</i> IF THEN ELSE	§ 12.8
TGET	pV	Get most recent task set-up	§ 4.3
TGINDEX	V	List all TGET files	§
TIMDEST	V	Destroy all files older than..	§ 4.4
TO	V	<i>POPS</i> FOR TO BY loop.	§ 6.1, 9.3, 12.4
TOAIP	T	Convert VLB data to AIPS	§
TOVLB	T	Convert AIPS data to VLBI	§
TPHEAD	V	List image header from tape	§ 6.1
TPUT	pV	Save a task or verb set-up	§ 4.3
TRANS	T	Transpose a data cube	§ 9.4
TYPE	V	Print the following value	§ 12.3, 12.8
— TK —			
TKGUESS	V	Initialize and display guess	§
TKAGUESS	V	Add guess to display	§ 8.3.3
TKMODEL	V	Initialize and display model	§
TKAMODEL	V	Add model fit to display	§ 8.3.3
TKPL	T	Display plot file on Tek	§ 7.1, 7.4, 8.2
TKPOS	V	Obtain position under cursor	§

TKRESID	V	Initialize and display resid.	§
TKARESID	V	Add fit resid. to display	§ 8.3.3
TKSET	V	Use cursor to fill initial guess	§ 8.3.3
TK1SET	V	Modify model guess	§
TKSLICE	V	Initialize and display Slice	§ 7.4, 8.3.3
TKASLICE	V	Add Slice to plot	§ 7.4
TKXY	V	Get pixel pos. under cursor	§
TKVAL	V	Obtain map value under cursor	§

--- TV ---

TV3COLOR	V	Make a false color display	§ 9.5
TVALL	V	General purpose TV display	§ 6.2
TVBLINK	V	Blink TV between two images	§
TVBOX	V	Set BOX with cursor, graphics	§ 6.3
TVCLEAR	V	Clear TV channels(s)	§
TVFIDDLE	V	Same as TVALL with no load	§ 6.2
TVHUEINT	V	Hue-intens. display 2 images	§ 9.5
TVINIT	V	Initialize TV	§
TVLABEL	V	Label image with coordinates	§
TVLOD	V	Load image on TV channel(s)	§ 9.5, 12.5
TVLUT	V	N-point transfer function	§
TVMBLINK	V	Blink TV between two images	§
TVMLUT	V	Many-point transfer function	§
TVMOVIE	V	Load planes of cube, run movie	§ 9.5
TVNAME	V	Print image name under cursor	§
TVOFF	V	Turn off image channel(s)	§
TVON	V	Turn on image channel(s)	§ 6.2
TVPL	T	Display plot file on the TV	§ 7.1, 7.4
TVPOS	V	Obtain position under cursor	§
TVPSEUDO	V	Initiate pseudo coloring	§ 6.2.1
TVROAM	V	Load image channels and roam	§
TVSCROLL	V	Scroll an image if necessary	§
TVTRANS	V	Change transfer function	§ 6.2.2
TVWEDGE	V	Put wedge on TV (as loaded)	§
TVWINDOW	V	Set BLC / TRC with graphics	§ 6.3
TVZOOM	V	Turn on TV zoom capability	§

--- U ---

UNQUE	V	Unque waiting batch files	§
UCAT	V	Index of UV files in catalog	§ 3.1.1, 4.2
UVCOP	T	Copy selected UV data	§
UVDIS	T	Display FFT of map on TV	§
UVFLG	T	Flag selected UV data	§ 5.3
UVFND	T	Print selected UV data	§ 5.3
UVEXP	T	Write UV data on tape	§ 10.1
UVLOD	T	Read UV data from tape	§ 3.1.2, 9.1
UVMAP	T	Make map from sorted UV data	§ 3.1.4, 9.2
UVPLT	T	Generate UV plots	§ 5.3
UVSRT	T	Sort UV data	§ 3.1.3

---

UVSUB	T	Subtract model from UV data	§ 5.3
— V —			
VBCAL	T	Apply VLB antenna factors	§
VBCC	T	Write CC files in C17 format	§
VBCOR	T	Apply fringe fit result	§
VBFIT	T	Global fringe fitting for VLB	§
VBMRG	T	Merge VLB data	§
VLBDR	T	Fringe, delay and antenna fit	§
— V —			
WAITTASK	V	Permits task sequencing	§ 9.1
WHILE	pV	POPS conditional expression	§
WSLOD	T	Load WSRT uv-data tape to disk	§
— Z —			
ZAP	V	Destroy an image	§ 3.1.3, 10.2, 12.4
— NON-ALPHA —			
+	V	Addition	§ 12.1
-	V	Subtraction or Negative	§
*	V	Multiplication	§ 12.1
/	V	Division	§ 12.3
**	V	Exponentiation	§
(	V	Begin grouping	§
)	V	End grouping	§
=	V	Equals or Store	§ 12.1
,	V	Separator of array elements	§ 12.1
:	V	Same as TO in loop	§
;	V	Separator of POPS statements	§ 12.1
>	V	Greater than	§ 12.8
<	V	Less than	§
>=	V	Greater or equal to	§
<=	V	Less than or equal to	§
<>	V	Not equal to	§
!	V	Logical or	§
&	V	Logical and	§
~	V	Logical not	§



## B. Glossary

**adverb** — see *POPS symbols*.

**AIPS monitor** — a computer terminal (perhaps lacking a keyboard) whose CRT screen is used in AIPS solely for the display of information related to the progress of the execution of the AIPS tasks. (Except—at those AIPS sites without a terminal dedicated to this use, the AIPS user's interactive terminal is used for dual purposes—i.e., to serve as the AIPS monitor as well.) Many of the messages which the AIPS tasks write to the monitor also are recorded in the *message file* (*q.v.*).

**aliased response** — in a radio interferometer map, a spurious feature due to a source—or to a sidelobe—that lies outside of the field of view. Consider the sampling of a visibility function  $V$  at the lattice points of a rectangular grid as multiplication of  $V$  by the comb-like distribution  $R(u, v) = \sum_k \sum_l \delta(u - k\Delta u, v - l\Delta v)$ . The Fourier transform  $\hat{R}V$  of  $RV$  is given by the convolution  $\hat{R} * \hat{V}$ . Since  $\hat{R}$  is again a comb-like distribution, with peaks, or teeth, separated by  $\frac{1}{\Delta u}$  in one direction and by  $\frac{1}{\Delta v}$  in the perpendicular direction,  $\hat{R}V$  is periodic, and, about the position of each tooth in the comb, it looks like an infinite summation of rectangular pieces of  $\hat{V}$ , each of size  $\frac{1}{\Delta u} \times \frac{1}{\Delta v}$ , taken from all over the plane. Aliased responses can be suppressed, very effectively, by judicious choice of the *gridding convolution function* (*q.v.*).

For a more complete discussion, see Dick Sramek's Lecture No. 2 in the 1982 Summer Workshop Proceedings. Also see VLA Scientific Memoranda Nos. 129 and 131.

**ALU** — (Arithmetic Logic Unit) an (optional) micro-computer CPU unit within the I<sup>2</sup>S TV display device which allows simple arithmetic operations, such as sums, products, and convolutions, to be performed on the data recorded in the I<sup>2</sup>S image planes. At present, AIPS makes little use of the ALU, since many of its features are unique to the I<sup>2</sup>S display unit. See *I<sup>2</sup>S*.

**antenna file** — in AIPS, an *extension file*, associated with a *u-v* data file, in which a list of the interferometer antenna positions is stored.

**antenna/l.f. gain** —

**antenna/l.f. phase** —

**AP** — see *array processor*.

**AP-120B array processor** — an array processor manufactured by Floating Point Systems, Inc., and used at a number of AIPS sites. Its floating-point word length is 38-bits. Typically it is equipped with a main data memory of 32-64 kilowords and a program source memory of 2048 words. With both a pipeline multiplier and a pipeline adder, and a memory cycle time of 167 ns., when programmed at top efficiency it can perform at an arithmetic rate of 12 million floating-point operations per second.

**array processor** —

**array processor microcode** —

**associated file** — In AIPS, any two or more files among a collection consisting of a *primary data file* and all of its *extension files* are termed *associated*.

**auto re-boot** — a boot initiated by the computer itself, of its own volition. See *boot*.

**back-up** — The act of copying the contents of a computer file to some permanent storage medium such as magnetic tape or punched cards, for the purpose of protecting against accidental loss or in order to liberate storage space (e.g., disk space), is termed *backing-up*. The new copy of the file is termed a *back-up* copy, or simply a *back-up*. See *scratch*.

**baseline-time order** — An ordered set of visibility measurements  $\{V_{ij}(t_k) \mid 1 \leq i < j \leq n, k = 1, \dots, l\}$  recorded with an  $n$  element interferometer at times  $t_k$  is said to be in *baseline-time order* if the ordering is such that all of the data for the 1-2 baseline, sorted by time, occur first, followed by the data for the 1-3 baseline, again sorted by time, etc., etc. (This canonical ordering by baseline is the order  $V_{12}, V_{13}, \dots, V_{1n}, V_{23}, \dots, \dots, V_{n-1,n}$ .) Compare *time-baseline order*.

Baseline-time ordering of a *u-v* data file is convenient for purposes of data display.

**batch job** —

**batch queue** —

**batch processor** —

**"battery-powered" CLEAN algorithm** — a modified version of the Clark CLEAN algorithm, devised by Fred Schwab and Bill Cotton. At each major cycle of the algorithm, or perhaps less frequently, the residual map is computed not by convolving the current iterate with the dirty beam map, but rather by computing the visibility residuals, and then re-gridding and re-mapping. By this means, the edge effects are compensated, and hence one can search the full dirty map field of view for clean components. Simultaneously, instrumental effects (finite bandwidth and finite integration time) and sky curvature (the *wz* term) can be compensated for (i.e., the algorithm solves a more general equation than a convolution equation). See *Clark CLEAN algorithm*.

This algorithm has undergone preliminary testing, but it is not yet fully implemented in AIPS.

**beam** — 1. in radio interferometry, the Fourier transform (FT) of the *u-v* sampling distribution, or the FT of a weighted *u-v* sampling distribution, possibly convolved with a gridding convolution function—the idealized response to a point, or unresolved, radio source. 2. also, a numerical approximation to 1. 3. a digitized version of the latter, sampled on a regular grid (usually regarded as a map or image). 4.  $\approx$  *point spread function*, *q.v.* 5. (occasionally) as above, but taking into account instrumental effects, so that the beam depends on position in the sky. See *dirty map*.

Occasionally, any one of the above, other than 5, is termed the *synthesized beam*.

**beam patch** — in the Clark CLEAN algorithm, that portion of the central part of the beam which is used in the inner iterations, or the minor cycles. In the AIPS implementation, the beam patch size typically is set at 101 pixels  $\times$  101 pixels. See *Clark CLEAN algorithm*.

**blanked pixel** — in a digital image, a pixel whose value is undefined. In computer storage of quantized digital images, some special numeric value is assigned to the blanked pixels, so that they may be recognized as undefined and given whatever special treatment is required. See *pixel*.

**BLC** — *bottom left corner* (of an image). See *m  $\times$  n map*.

**blink** — see *TV blink*.

**boot** — A full computer system is restarted by means of a bootstrapping procedure, whereby the operating system and the data management facilities are re-initialized in a succession of steps, some of which necessarily must precede others. This sequence of restarting procedures, which may be highly involved, is termed the *boot*. A boot ( $\approx$  *re-boot*) is required after any system crash (e.g., after a power failure). Usually the sequence of steps required to accomplish the boot is posted in a notice located close to the system operating console, or on the CPU panel. On modern computers, such as the Vax, the boot procedure is highly automated. In fact, there may be an abbreviated boot procedure, termed a *quick boot*, to follow after a "soft" system crash. (On such systems, a quick boot should be attempted before resorting to a full boot.) Indeed, some systems (the Vax included) re-boot on their own initiative following a soft system crash—this is termed an *auto re-boot*.

**BOT marker** — (Beginning-Of-Tape marker) a short strip of metal foil attached near the front, or beginning, end of a computer magnetic tape. The tape drive uses the BOT marker in order to position the tape at its starting position.

**bpi** — (bits per inch) the basic unit of measurement used to specify the density at which information is recorded on a computer magnetic tape: the effective number of bits per inch per track. The standard recording densities are 800, 1600, and 6250 bpi. Modern computer tapes are nine-track tapes: eight recording tracks are used for the data, and the ninth track is used to record "parity bits" for error-checking. See *tape blocking efficiency*.

**b-t order** — same as *baseline-time order*.

**bug** — an actual or a perceived programming error or program deficiency. The bug may be in the eye of the beholder since the program user may fancy an application similar to, but differing from, the one for which the program is intended. In AIPS there is a formal mechanism for reporting program bugs; see *gripe file* for a description.

**byte** — a unit of eight bits of computer storage. See *units of computer storage*.

**carriage-return key** — One of the most used keys on any computer terminal keyboard is the carriage-return key ( $\text{CR}$ ). This is the key which must be depressed whenever one has finished typing a command to the computer, in order for the computer to accept or acknowledge the command.

**catalog file** — In AIPS, each user has his own *catalog file*, a repository for the header records of each of his primary data files. (Except—at some AIPS sites there is only a shared, public catalog file.) See *header record*.

**catalog entry** — a (header) record stored within an AIPS *catalog file* (q.v.).

**catalog slot** — in AIPS, a numbered space reserved in a *catalog file* for the insertion of a *catalog entry*.

**chromaticity diagram** —

**chrominance**

**Clark CLEAN algorithm** — a modified version of the Högbom CLEAN algorithm, devised by Barry Clark in order to accomplish an efficient array processor implementation of CLEAN (see [B. G. Clark, An Efficient Implementation of the Algorithm CLEAN, *Astron. Astrophys.*, **89** (1980) 377-378]). To operate on, say, an  $n \times n$  map, the original CLEAN algorithm requires on the order of  $n^2$  arithmetic operations at each iteration, and typically there may be hundreds or

thousands of iterations. The Clark algorithm proceeds by operating not on the full residual map, but rather by picking out only the largest residual points, iterating on these for a while (during its *minor cycles* or inner iterations) and only occasionally (at the *major cycles*) computing the full  $n \times n$  residual map, by means of the FFT algorithm. After each major cycle, it again picks out the largest residuals and goes into more minor cycles. And, for further economy, during these inner iterations the dirty beam is assumed to be identically zero outside of a relatively small box (termed the *beam patch*) which is centered about the origin. See *Högbom CLEAN algorithm*.

**CLEAN** — see *Högbom CLEAN algorithm*.

**clean beam** — in the Högbom CLEAN algorithm, an elliptical Gaussian function  $h$  with which the final iterate is convolved, in order to diminish any spurious high spatial frequency features—also termed *restoring beam*.  $h$  is specified by its major axis (usually the FWHM), its minor axis, and the position angle on the plane of the sky of its major axis. Usually these parameters are set by fitting to the central lobe of the dirty beam. See *Högbom CLEAN algorithm*.

**clean box** — a rectangular subregion of a *clean window* (q.v.).

**clean component** — in the Högbom CLEAN algorithm, a  $\delta$ -function component which is added to the  $(n-1)$ st iterate in order to obtain the  $n$ th iterate. Its location is the location of the peak residual after the  $(n-1)$ st iteration, and its amplitude is a fraction  $\mu$  (the *loop gain*) of the largest residual. See *Högbom CLEAN algorithm*.

The AIPS task implementing the (Clark) CLEAN algorithm stores a list of the clean components in an extension file which is termed a *components file*.

**clean map** — an approximate deconvolution of the *dirty beam* from the *dirty map*, derived by an application of the Högbom CLEAN algorithm or one of its derivatives. See *Högbom CLEAN algorithm*.

**clean speed-up factor** — in the Clark CLEAN algorithm, a number  $\alpha$  in the range  $[-1, 1]$  used in determining when to end a major cycle. Smaller  $\alpha$  causes a larger number of major cycles to occur (at greater computational expense) but yields a result closer to that of the classical Högbom CLEAN algorithm.

**clean window** — in the Högbom CLEAN algorithm, the region  $A$  of the residual map which is searched in order to locate the *clean components* comprising the successive approximants to the radio source brightness distribution. In the AIPS implementation,  $A$  is a union of rectangles, called *clean boxes*, which may be specified by the user. When  $A$  is not explicitly specified, the algorithm searches over the rectangular central one-quarter region of the residual map. See *window clean* and *Högbom CLEAN algorithm*.

**clipping** —

**closure amplitude** —

**closure phase** —

**color contour display** —

**color triangle** —

**compact support** — see *support*.

**components file** — in AIPS, an extension file, associated with an image file containing a *clean map*, whose content is a list of the positions and amplitudes of the *clean components* included in that clean map, as determined by the CLEAN algorithm. The source model specified by this list of components often is used in *self-calibration*.

**Comtal display unit** — a computer-controlled TV display device with capabilities similar to those of the I<sup>2</sup>S. A Comtal unit has been in use at the VLA for several years, as part of the IMPS image analysis system. At present this model of display unit is not used in any of the AIPS systems at the NRAO. See *IMPS* and *I<sup>2</sup>S*.

**conjugate symmetry** — that property which characterizes an *Hermitian function* (*q.v.*). Generally an assumption of conjugate symmetry is implicit whenever one speaks of the *u-v coverage* corresponding to some VLA observation.

**Conrac monitor** —

**convolution function** — see *gridding convolution function*.

**coordinate reference pixel** —

**correlator offset** —

**crash** — the abrupt failure of a computer system or program. More specifically, a *system crash* is the abrupt failure of a computer— or of a computer's operating system— causing the computer to halt the execution of programs; and a *program crash* is the abrupt failure of a computer program resulting either from a flaw in the logic of the program itself, or from some peculiar interaction with the operating system, the storage management facility, another program, or the user—or from an act of God. A *hardware crash* is a crash which results from the failure of the computer electronics or electro-mechanics, and a *software crash* is one which results from a flaw or an inadequacy in program logic, or in operating system program logic. A *soft crash* is a crash from which it is easy to recover—i.e., easy to restart the computer and resume work—, and a *hard crash* is the opposite.

**crosshair** — 1. a marker on the *TEK screen*, or *green screen*, which may be moved about through the use of thumbwheel knobs which are located on the terminal keyboard panel. The position of the crosshair may be sensed by the computer program, and thus the user may point out to the program features that are of interest in the graphical display on the CRT screen. 2. a marker with the same function as just described, but on a TV display device, and more likely controlled by a *trackball* than by thumbwheels. Same as *TV cursor*; and see *trackball*.

**cube** see *data cube*.

**cursor** — 1. a marker on an interactive computer terminal indicating the position on the CRT screen where the next character is to be typed. 2. *TV cursor*—on a TV display device, a marker whose manually controlled position may be sensed by the computer. See *crosshair*.

**data cube** — 1. in VLA spectral line data analysis, a three-dimensional map or "image" representing a function of three real variables—two spatial variables representative of position in the sky, and one variable related to frequency or velocity. 2. any *n*-dimensional *image*,  $n \geq 3$ .

Computer access of a multi-dimensional data array, residing in any standard type of storage medium such as disk or magnetic tape, is sequential, as if the data were one-dimensional. Spectral line data cubes are stored plane-by-plane, row-by-row, column-by-column. Permutation of the

correspondence between plane, row, and column, and the coordinate axis numbering, is referred to as *transposition* of the data cube.

**database** — a computer filing system, or file structure system. For example, the AIPS database consists not only of the data themselves, but also of the directories and the cross-reference lists of all the AIPS data files (including extension files), the data formats, etc., as well as the rules and principles governing the use thereof.

**data file** — on a computer storage medium, such as disk or magnetic tape, the concrete, or physically present representation of a logically distinct grouping of data in a manner permitting repeated access by computer programs.

**DCP** — (Dicomed Control Program) a program (which runs only on the Modcomp Classic Computer in Charlottesville) to control the Dicomed Image Recorder, for purposes of photographic display of image data recorded on magnetic tape. The DCP, not a part of AIPS, is described in [T. R. Cram and E. W. Greisen, *The NRAO Image Recording System*, NRAO Computer Division Internal Report No. 21, May 1975]. At present (January 1983) there is no path, other than through the DCP, for Charlottesville AIPS users to use the Dicomed unit.

**DFT** — see *discrete Fourier transform*.

**Dicomed Image Recorder (Model D47)** — a computer-controlled image display device intended for photographic reproduction of digital images. The film is exposed by a cathode ray tube. The device is capable of 4096 pixel  $\times$  4096 pixel resolution, and of both black-and-white and color reproduction. The digital exposure control and eight-bit pixel input allow 256 discrete exposure levels. The CRT has a single electron gun and a screen with a white phosphor; color reproduction is accomplished by means of multiple exposures, with the insertion of red, green, and blue filters. There is a Dicomed recorder at the NRAO in Charlottesville, and another at the VLA. See *DCP*.

**direct Fourier transform** — a term used imprecisely to mean either: 1) the *discrete Fourier transform* (*q.v.*) of irregularly-spaced data, or 2) the naïve, or brute-force, evaluation of the discrete Fourier transform of regularly-spaced data —see *naïve discrete Fourier transform evaluation*.

**dirty beam** — in radio interferometry, simply a *beam*, but computed with precisely the same operations as those used to compute some companion *dirty map* (i.e., with the same *u-v coverage*, the same manner of gridding convolution, the same *u-v weight function* and taper, etc.). In cleaning a dirty map, no other beam than the companion dirty beam should be used.

**dirty map** — 1. ignoring instrumental effects, the Fourier transform (FT) of the product of the visibility function *V* of the radio source and the (possibly *weighted* and/or *tapered*) *u-v sampling distribution* *S*; i.e., the FT of the *u-v measurement distribution*. 2. a discrete approximation to 1; in this case, the product *SV* is convolved with some function *C*, of *compact support*, and a discrete Fourier transform of samples of  $C * (SV)$  taken over a regular grid yields the *dirty map*. 3. as in 2, but corrected for the taper ( $\hat{C}$ , the FT of *C*) induced by the convolution. 4. any of the above, but now taking into account various instrumental effects (receiver noise, non-monochromaticity or finite bandwidth, finite integration time, sky curvature, etc.).

If it is assumed that  $V \equiv 1$ , then the map, or point source response, so obtained is termed the *beam* (*q.v.*). Also

see *gridding convolution function*, *u-v taper function*, *u-v weight function*, and *dirty beam*.

**discrete Fourier transform —**

**disk hog** — a derogatory term, used to connote a computer user whose disk data files are excessively voluminous or numerous, therefore putting other computer users at a relative disadvantage. Unneeded data files should be *scratched*, or destroyed, in order to free up disk space. Large disk files which will not be needed for a time should be *backed-up* on magnetic tape and then deleted from disk.

**dynamic range —**

**EDT** — a sophisticated text editor used on the Vaxes. It makes use of the "keypad" feature of the fancier terminals. EDT can be run only on certain model terminals: on the DEC (Digital Equipment Corp.) Models VT-52 and VT-100, and on terminals such as the Visual-50's and the Visual-100's which are capable of emulating the DEC terminals. See *text editor*.

**EMACS** — a sophisticated text editor used on the Vaxes, as well as on many computers which run under the UNIX operating system. EMACS is the text editor which is favored by most of the AIPS programmers. On the terminals with the "keypad" feature, the keypad keys can be programmed by the user to perform many useful editing tasks. See *text editor*.

**EXPORT format** — a visibility data magnetic tape format for transport of VLA data from the DEC-10 computer or the on-line computer at the VLA.

**EXPORT tape** — a magnetic tape containing data recorded in the *EXPORT format*.

**exp  $\times$  sinc function** — a useful gridding convolution function: same as the *Gaussian-tapered sinc function* (q.v.), except that the argument of the exponential function may be raised to a power other than 2.

**extension file** — in AIPS, a data file containing data supplemental to those contained in a primary data file (either a *u-v data file* or an *image file*). Whenever a primary data file is deleted by the standard mechanism within AIPS for file destruction, all extension files associated with that primary data file also are destroyed. Extension files, however, may be deleted without deleting the the associated primary data file.

Extension files are grouped into categories of named types. Examples: *plot files*, *history files*, *slice files*, *gain files*, etc.

When an AIPS task creates a new primary data file from an old one, generally it attaches, to the new file, clones of any extension files associated with the old file that remain relevant to the new one.

**false color display —**

**FFT** — see *fast Fourier transform algorithm*.

**fast Fourier transform algorithm —**

**FITS format** — (Flexible Image Transport System) a magnetic tape data format well-tailored for the transport of image data among observatories. See [D. C. Wells, E. W. Greisen, and R. H. Harten, *Astron. Astrophys. Suppl. Ser.*, **44** (1981) 363-370]. Also see *u-v FITS format* and *FITS tape*.

**FITS tape** — a magnetic tape containing data recorded in the *FITS format*. FITS format data blocks are 2880 bytes in length. The resultant *tape blocking efficiency* is 83%, 75%, and 61% at recording densities of 800, 1600, and 6250 bpi, respectively.

**flag —**

**gain file —**

**Gaussian-tapered sinc function** — A useful *gridding convolution function* (q.v.), of support width equal to the width  $m\Delta u$  of  $m$  *u-v* grid cells, is given by the separable product of two Gaussian-tapered sinc functions, each of the form

$$C(u) = \begin{cases} e^{-\left(\frac{\pi u}{a\Delta u}\right)^2} \frac{\sin\left(\frac{\pi u}{b\Delta u}\right)}{\frac{\pi u}{b\Delta u}}, & |u| < \frac{m\Delta u}{2}, \\ 0, & \text{otherwise.} \end{cases}$$

The choice  $m = 6$ ,  $a \simeq 2.52$ , and  $b \simeq 1.55$ , yields what is, in a certain natural sense, an optimal gridding convolution function of this particular parametric form (see [F. R. Schwab, *Optimal Gridding*, VLA Scientific Memo. No. 132]). Also see *spheroidal function*.

**Gibbs' phenomenon** — in the neighborhood of a discontinuity of a periodic function  $f$ , the overshoot and oscillation (or *ringing*) of the partial sums  $S_n$  of the Fourier series for  $f$ . In the vicinity of a simple jump discontinuity,  $S_n$  always overshoots the mark by about 9%, regardless how large  $n$ . See [H. S. Carslaw, *Introduction to the Theory of Fourier's Series and Integrals*, Dover, N. Y., 1930, ch. IX].

In harmonic analysis, often the Fourier coefficients are multiplied by a weight function tending smoothly to zero at the boundaries of its support, in order to smooth out the discontinuities and thereby reduce the ringing in the synthesized spectrum. (This degrades the spectral resolution, however.) See *Hanning smoothing*. For a discussion of Gibbs' phenomenon in the context of VLA cross correlation analysis, see § 4 of Larry D'Addario's Lecture No. 7 in the 1982 Summer Workshop Proceedings.

**GPSY** — a data reduction system, similar in scope to AIPS, used in the Netherlands for analysis of Westerbork Synthesis Radio Telescope (WSRT) data.

**graphics overlay plane** — same as *graphics plane*.

**graphics plane** — a storage area within a TV display device, such as the I<sup>2</sup>S, in which a full screen load of one-bit graphics information (labeling, plotting, axis lines, etc.) is stored. A typical I<sup>2</sup>S unit is equipped with four graphics planes, each 512 pixels  $\times$  512 pixels in area. Compare *image plane*.

**gray-scale memory plane** — same as *image plane*.

**green screen** — same as *TEK screen*.

**gridding convolution function** — in radio interferometer mapping, a function  $C$ —usually supported on a square the width of, say, six *u-v* grid cells—with which the *u-v* measurement distribution is convolved. The purpose is twofold: 1) to interpolate and smooth the data, so that samples may be taken over the lattice points of a rectangular grid (in order that the fast Fourier transform algorithm may be applied), and 2) to reduce aliasing (the convolution in the *u-v* plane induces a taper in the map plane). See *aliased response*, *gridding correction function*, and *dirty map*.

With judicious choice of  $C$ , a high degree of aliasing suppression is possible. A high degree of suppression is desirable, even when there are no "confusing" radio sources very near the field of interest, because the effect is not only to reduce the spurious responses due to sources lying outside of the field of view, but also to reduce the response to sidelobes of the source of interest, which too are aliased into the map from outside the field of view. See *spheroidal function*.

**gridding correction function** — in radio interferometry, the reciprocal  $1/\hat{C}$  of the Fourier transform (FT) of the *gridding convolution function*  $C$ . Since the map plane taper induced by the gridding convolution usually is very severe, the dirty map normally is corrected by pointwise division by the FT of the convolution function. Obviously  $C$  should be chosen such that  $\hat{C}$  has no zeros within the region that is mapped. See *dirty map*.

**gripe** — in AIPS, an entry in the *gripe file* (*q.v.*).

**gripe file** — in AIPS, a disk file repository for formal reports of program *bugs*, and for formal complaints and suggestions of a more general nature. A mechanism by which the user may enter gripes into the gripe file is activated by the issuance of the AIPS verb: GRIPE.

**Hanning smoothing function** — in the analysis of power spectra, a weight function  $w$  by which the measured correlation function is multiplied, in order to reduce that oscillation (*Gibbs' phenomenon*) in the computed spectrum which is due to having sampled at only a finite number of lags.  $w$ , as a function of lag, is given by

$$w(\tau) = \begin{cases} \frac{1}{2} \left( 1 + \cos \frac{\pi \tau}{\tau_{\max}} \right), & |\tau| < \tau_{\max}, \\ 0, & \text{otherwise.} \end{cases}$$

Hanning smoothing sometimes is applied to the cross correlation measurements obtained in VLA spectral line observing, in order to reduce the effect of sharp bandpass filter cutoffs. It also is used frequently in radio astronomical autocorrelation spectroscopy. See *Gibbs' phenomenon*.

**hard copy** — computer output printed on paper (rather than, say, written on magnetic tape). E.g., a printed contour plot or gray scale display, or a listing of a catalog file.

**hardware mount** — the combined acts of installing a computer external storage module, such as a disk pack or a reel of magnetic tape, in some electro-mechanical unit (e.g., a disk drive or a tape drive) that provides computer access to this data storage medium, and placing that unit in readiness to be operated under computer control (e.g., positioning a magnetic tape at the *BOT marker*). Compare *software mount*.

**header record** — a distinguished record within a data file—generally the first record—which serves to define the contents of the other records in the file by supplying relevant parameters, units of measurement, etc.; also termed simply *header*.

In AIPS, however, the header records of the *primary data files* are stored in a special *catalog file*. AIPS extension file headers are stored within the extension files themselves.

**help file** — in AIPS, a *text file*, whose contents may be displayed on the terminal screen of the interactive user, giving a brief explanation of a particular AIPS verb, adverb, pseudoverb, task, or miscellaneous general feature.

**Hermitian function** — A complex-valued function, of one or more real variables, whose real part is an even function and whose imaginary part is odd is called an *Hermitian function*. The Fourier transform (FT) of a real-valued function is Hermitian, and the inverse FT of an Hermitian function is real.

Since each of the radio brightness distributions  $I(x, y)$ ,  $Q(x, y)$ ,  $U(x, y)$ , and  $V(x, y)$  representing *Stokes' parameters* is real-valued, *Stokes' visibility functions* have the property of *conjugate symmetry*:  $V_I(-u, -v) = \bar{V}_I(u, v)$ ,

$V_Q(-u, -v) = \bar{V}_Q(u, v)$ ,  $V_U(-u, -v) = \bar{V}_U(u, v)$ , and  $V_V(-u, -v) = \bar{V}_V(u, v)$ . (Here,  $V_I = \hat{I}$ ,  $V_Q = \hat{Q}$ , etc., where  $\hat{\cdot}$  denotes FT.)

**history file** — in AIPS, an *extension file* containing a summary of all, or most of the processing, by AIPS tasks, of the data recorded in all associated files.

**Högbom CLEAN algorithm** — a deconvolution algorithm, for use in radio interferometry, devised by Jan Högbom [J. A. Högbom, *Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines*, *Astron. Astrophys. Suppl. Ser.*, 15 (1974) 417-426]. Denoting (the discrete representations of) the dirty map by  $g$  and the dirty beam by  $b$ , the algorithm iteratively constructs discrete approximants  $f_n$  to a solution  $f$  of the equation  $b * f = g$ . The initial approximant  $f_0 \equiv 0$ . At the  $n$ th iteration, one searches for the peak in the residual map  $g - b * f_{n-1}$ . A  $\delta$ -function component, centered at the location of the largest residual, and of amplitude  $\mu$  (the *loop gain*) times the largest residual, is added to  $f_{n-1}$  to yield  $f_n$ . The search over the residual map is restricted to a region  $A$  termed the *clean window*. The iteration terminates with an approximate solution  $f_N$  either when  $N$  equals some iteration limit  $N_{\max}$ , or when the peak residual (in absolute value) or the r.m.s. residual decreases to some given level.

To diminish any spurious high spatial frequency features in the solution,  $f_N$  is convolved with a narrow elliptical Gaussian function  $h$ , termed the *clean beam*. Generally  $h$  is chosen by fitting to the central lobe of the dirty beam. Also, one generally adds the final residual map  $g - b * f_N$  to the approximate solution  $f_N * h$ , in order to produce a final result, termed the *clean map*, with a realistic-appearing level of noise.

For further details, see Tim Cornwell's Lecture No. 9 in the 1982 Summer Workshop Proceedings.

**hue** —

**hybrid mapping algorithm** —

**IIS** — see *I<sup>2</sup>S*.

**image** — in the context of AIPS, any finite-volume, completely filled, linear, rectangular, or hyper-rectangular array of pixels (except see *blanked pixel*). E.g., a digitized photograph, or a radio map. The term also is used (less technically) to refer to the *display* of data—e.g., a television picture of a radio map.

**image file** — in AIPS, a *primary data file* whose content is an *image*.

**image plane** — a storage area within a TV display device, such as the *I<sup>2</sup>S*, in which a full screen load of single word pixels is stored. A typical *I<sup>2</sup>S* unit is equipped with four image planes, each 512 pixels  $\times$  512 pixels in area (each pixel is represented by eight bits). Often several image planes are used at one time—either for black-and-white or *pseudo-color* display of a large image, sections of which may occupy different image planes—or for *false color* or *true color* display of a smaller image, now using, say, three image planes one to control each of the three electron guns (for red, green, or blue phosphor) in the TV display. Compare *graphics plane*.

**IMPS** — (Interactive Map Processing System) an image analysis system in use at the VLA site, configured around a DEC PDP-11/44 computer and a Comtal TV display device. IMPS' capabilities are similar to the image display and image manipulation capabilities of AIPS.

**inputs file** — in AIPS, a *text file*, whose contents may be displayed on the terminal screen of the interactive user, giving a summary of the *adverbs* relevant to a given *verb* or a given AIPS *task*.

**intensity** —

**invisible distribution** —

**IPL** — (Initial Program Load) same as *boot*.

**isoplanaticity** —

**I<sup>2</sup>S** — (International Imaging Systems Model 70) a TV display device, capable of both black-and-white and color display, manufactured by the Stanford Technology Corporation. At an AIPS site typically it is equipped with four 512 pixel × 512 pixel eight-bit image planes, four one-bit graphics planes, a trackball, and sometimes an ALU. The eight-bit pixel representation (in the image planes) allows the intensity of each of the three electron gun beams to be set at any of 256 discrete levels. (Actually, 1024 levels can be used, because of an extra two bits of capability provided in the transfer function tables and the internal arithmetic unit.) An I<sup>2</sup>S is attached to each of the NRAO's computers on which the AIPS system runs (except for the IBM-4341 in Charlottesville).

**loop gain** — in the Högbom CLEAN algorithm, the fraction  $\mu$  of the largest residual, used in determining the amplitude, or flux, of a *clean component*. Convergence can be achieved for  $\mu$  in the range (0, 2), but generally a small value, say  $\mu = \frac{1}{10}$ , is recommended, especially in dealing with extended sources. See Högbom CLEAN algorithm.

**l<sub>1</sub> solution algorithm** — see *self-calibration gain solution algorithm*.

**l<sub>2</sub> solution algorithm** — see *self-calibration gain solution algorithm*.

**major cycle** — In the Clark CLEAN algorithm (*q.v.*), a number of minor cycles, or inner iterations, followed by the computation by the FFT algorithm of the full residual map, comprise a major cycle.

**map** — an *image*, one or more of whose coordinate axes represents some spatial coordinate.

**MAX IV** — the operating system of the Modcomp Classic computer which is used for AIPS reduction in Charlottesville. See *operating system*.

**memory page** — see *virtual memory page*.

**memory paging** — same as *virtual memory page swapping*.

**memory thrashing** — an excessive amount of *virtual memory page swapping* (*q.v.*) on a computer (such as the Vax) with a virtual memory operating system. A condition of memory thrashing is likely to occur whenever too many programs with large memory requirements are active (a single program with excessive memory requirements can cause memory thrashing).

**message file** —

**message terminal** — same as *AIPS monitor*.

**minor cycle** — in the Clark CLEAN algorithm (*q.v.*), an inner iteration, in which the peak residual over a subregion (the *clean window*) of the full residual map is found and is used to obtain the next successive iterate. Compare *major cycle*.

**microcode** — see *array processor microcode*.

**monitor** — see *AIPS monitor* or *Conrac monitor*.

**m × n map** — The convention adopted for AIPS is opposite the standard matrix algebra terminology: whereas an  $m \times n$  matrix is comprised of  $m$  rows and  $n$  columns, an  $m \times n$  map or image in AIPS has, in the usual display format,  $m$  pixels along the horizontal axis (usually termed the *x-axis*) and  $n$  pixels along the vertical axis (usually termed the *y-axis*). Moreover, pixels of a two-dimensional map in the usual display format are numbered from the bottom left-hand corner: the pixel location specified by the ordered pair ( $i, j$ ) is in column number  $i$  and row number  $j$ , counting from the bottom left. In other than two-dimensional "images", the (1, ..., 1) pixel is also said to be at the "bottom left corner" (BLC), just as in the two-dimensional case. See *data cube*.

**naïve discrete Fourier transform evaluation** —

**natural weighting** —

**operating system** —

**page** — see *virtual memory page* and *terminal page*.

**page swapping** — see *virtual memory page swapping*.

**physical memory** — core or semiconductor memory within a computer (as opposed to slower memory—virtual memory, disk storage, magnetic tape footage, etc.). A typical Vax is equipped with a physical memory 3–4 megabytes in size. The Modcomp Classic computer in Charlottesville has  $\frac{1}{2}$  megabyte.

**pixel** — (*picture element*) an element of a digitized image (or of a map). A pixel is characterized by its *location* and its *value*.

**plot file** — an AIPS extension file containing plotting information, in the form of the commands which are necessary in order for a line drawing peripheral device, such as a Calcomp or other pen plotter, a green screen, or an electrostatic printer/plotter, to generate a plot.

**point source response** — same as *point spread function*.

**point spread function** — (PSF) 1. the response of a system or an instrument to an impulsive, or point source, input. 2. in radio interferometry, the response of the instrument to a point, or unresolved, radio source—a fancy term for *beam*. Ignoring instrumental effects, such as finite bandwidth and finite integration time, the response does not depend upon the displacement of the source away from the visibility phase tracking center—hence the term *space-invariant PSF* (SIVPSF), and the contrary term *space-variant PSF* (SVPSF).

A so-called linear space invariant measurement system (i.e., a linear system with an SIVPSF) is equivalently described as a system which can be modeled by a convolution equation; a linear space-variant measurement system can be modeled by a more general linear Fredholm integral equation of the first kind.

**POPS** — (People-Oriented Parsing System) the parser, or command interpreter, embedded within the AIPS program; that part of the AIPS program which attempts to interpret the user's commands (*POPS symbols*) and then initiate the appropriate reaction. POPS is used in other astronomical data reduction programs at the NRAO, such as the Condare, TPOWER/SPOWER, and Tucson 12 m single-dish packages.

**POPS symbols** — The AIPS user's primary means of communicating his wishes to AIPS is by typing commands, termed *POPS symbols*, at the keyboard of a computer terminal. There are four classes of POPS symbols: *adverb*, *verb*, *pseudoverb*, and *procedure*. An *adverb* is a *symbol* representing the storage area for a datum or for data that are used to control the action of verbs, tasks, and procedures; that is to say that the adverb symbols are used to set *control parameters*. A *verb* is a symbol which causes POPS (or AIPS) to initiate some action after POPS has finished interpreting, or compiling, the command line typed at the computer terminal. A *pseudoverb* is a symbol which suspends, temporarily, the normal parsing of an input line and which causes some action to take place while the line is being compiled, and, possibly, after compilation. A *procedure* is a symbol representing a pre-compiled sequence of POPS symbols. Also see *task*.

**POPS procedure** — see *POPS symbols*.

**primary data file** — in AIPS, either a *u-v data file*, containing measurements of the visibility function of a radio source, or an *image file*, containing a digitized image or a radio map. Compare *extension file*.

**principle solution** —

**procedure** — see *POPS symbols*.

**prolate spheroidal wave function** — an eigenfunction of the finite, or truncated, Fourier transform—more precisely, for given  $c$ , one of the countably many solutions of the integral equation

$$(*) \quad \nu f(\eta) = \int_{-1}^1 e^{ic\eta t} f(t) dt;$$

equivalently, a solution of the differential equation  $(1 - \eta^2)f'' - 2\eta f' + (b - c^2\eta^2)f = 0$ ; or, equivalently, a solution of the wave equation in a system of prolate spheroidal coordinates. The eigenfunction of  $(*)$  associated with the largest eigenvalue  $\nu$  is termed the 0-order solution.

If we want a gridding convolution function  $C$ , of *support width* equal to the width of  $m$  grid cells, that is optimal in the sense that its Fourier transform  $\hat{C}$  has the property that the concentration ratio

$$\frac{\iint_{\text{map}} |\hat{C}(x, y)|^2 dx dy}{\iint_{-\infty}^{\infty} \iint_{-\infty}^{\infty} |\hat{C}(x, y)|^2 dx dy}$$

is maximized, then  $C$  is the separable product of two 0-order prolate spheroidal wave functions, with  $c = \pi m/2$ . See *gridding convolution function* and *spheroidal function*.

**prompt character** — a character (often the dollar sign "\$" or the greater-than sign ">") which the computer program or the operating system prints on the terminal screen of the interactive user in order to prompt, or invite, a typed response from the user. The AIPS program's standard prompt character is the greater-than sign, and on the Vax and Modcomp systems at the NRAO the operating system's prompt character is the dollar sign. Thus, most commands (or *POPS symbols*) peculiar to AIPS must be typed on a line beginning with the >-character, and any command to the operating system, such as the command to mount a tape, must be typed on a line beginning with the \$-character.

When operating in some lesser-used, special modes, AIPS employs other prompt characters: ":" for procedure building, ";" for procedure editing, "I" for entry of gipes, "<" for batch file preparation, and "#" for parameter reading.

**Prussian helmet CLEAN algorithm (PHCLN)** — a modified version of the Högbom CLEAN algorithm, devised by Tim Cornwell. The idea is to drive the CLEAN algorithm toward an approximate solution  $f$  of minimal Euclidean norm—i.e., to find an  $f$  consistent with the data, band-limited to the clean window, comprised of a small number of point components, and such that  $\iint_{\text{window}}^{\text{clean}} |f(x, y)|^2 dx dy$  is minimized. This is accomplished by adding a  $\delta$ -function of amplitude  $w$ , centered at the origin, to the dirty beam, and then just proceeding as normal with the CLEAN algorithm. Proper choice of  $w$  depends on the level of noise contaminating the data. See [T. J. Cornwell, Can CLEAN be improved?, VLA Scientific Memo. No. 141]. Sometimes this algorithm is called by the name, PHCLN, of the AIPS program in which it is implemented.

**pseudo-AP** — see *pseudo-array processor*.

**pseudo-array processor** — in AIPS, the term which is applied to a collection of Fortran subroutines which may be used to emulate the operation of an FPS Model AP-120B array processor. At those AIPS sites which do not have an array processor (generally because of its high cost), the AIPS tasks which normally would make use of an array processor use the pseudo-array processor subroutines instead. See *array processor*.

**pseudo-color display** —

**pseudo-continuum u-v data file** — in VLA spectral line data reduction, a *u-v data file* containing the visibility measurements from a small number of spectral line channels, recorded in the same format as continuum visibility data. The purpose is to enable the use, for spectral line data analysis, of programs originally intended only to handle continuum data reduction.

**pseudoverb** — see *POPS symbols*.

**PSF** — see *point spread function*.

**quick boot** — an abbreviated boot procedure. See *boot*.

**RANCID** — (Real (or Radio) Astronomical Numerical Computation and Imaging Device) the name by which the AIPS data reduction system formerly was known.

**re-boot** — Having booted once already, one *re-boots*. See *boot*.

**re-IPL** — same as *re-boot*.

**restoring beam** — same as *clean beam*.

**roam** — see *TV roam*.

**run file** —

**saturation** —

**scratch** — 1. The act of deleting a data file—i.e., surrendering the storage medium space which that file occupies—is termed *scratching* the data file. Use of the term *delete* may be preferable, but *scratch* is more common among AIPS users. One who is about to delete a data file may wish first to create a back-up copy. See *back-up*. 2. an adjective meaning *temporary*, as in *scratch file*.

In AIPS a primary data file and all of its associated extension files can be deleted by means of the verb: ZAP.

**scratch file** — a data file intended for temporary storage (esp., of data which represent intermediate results—i.e., *scratchwork*). Many of the AIPS tasks use scratch files; the necessary scratch files are created and destroyed automatically by the tasks. However, when an AIPS task *crashes*, sometimes a scratch file remains.



**scroll** — see *terminal scroll* and *TV scroll*.

**SEDIT** — an unsophisticated text editor available on the Modcomp computers. See *text editor*.

**self-calibration algorithm** —

**self-calibration gain solution algorithm** —

**shed** — see *sub-task*.

**SIVPSF** — see *point spread function*.

**slice** —

**slice file** —

**software mount** — the issuance of a command to a computer informing it that the *hardware mount* of some external storage module, such as a disk pack or a reel of magnetic tape, has occurred, and that the computer should open the channel of access to this module. See *hardware mount*.

**sort order** — the ordering of visibility measurements within a *u-v data file*. *Time-baseline order* is convenient for purposes of calibration, *baseline-time order* for data display, and *x-y order* for gridding and subsequent mapping.

**SOS** — a text editor available on the NRAO's Vaxes, as well as on the PDP-11's and the DEC-10 at the VLA and on any Vax operating under control of the VMS operating system. SOS is a product provided by the computer manufacturer. See *text editor*.

**source editor** — same as *text editor*. (Formerly, computers were used mainly for numerical computations and text editors primarily for the editing of program source code—hence the name *source editor*).

**spawn** — see *sub-task*.

**spheroidal function** — an eigenfunction  $\psi_{\alpha n}$  of a finite, weighted-kernel Fourier transform—more precisely, for given  $c$  and given  $\alpha > -1$ , one of the countably many solutions of the integral equation

$$(*) \quad \nu f(\eta) = \int_{-1}^1 e^{ic\eta t} (1-t^2)^\alpha f(t) dt;$$

equivalently, a solution of the differential equation  $(1-\eta^2)f'' - 2(\alpha+1)\eta f' + (b-c^2\eta^2)f = 0$ . The eigenfunction  $\psi_{\alpha 0}$  of  $(*)$  associated with the largest eigenvalue  $\nu$  is termed the 0-order solution. The choice  $\alpha = 0$  of weighting exponent yields the family  $\{\psi_{0n} | n = 0, 1, 2, \dots\}$  of prolate spheroidal wave functions.

Weighted 0-order spheroidal functions  $(1-\eta^2)^\alpha \psi_{\alpha 0}$  are optimal gridding convolution functions in the same sense that the *prolate spheroidal wave functions* (q.v.) are optimal, except that now the weighted concentration ratio

$$\frac{\int \int_{\text{map}} |\hat{C}(x, y)|^2 (1 - (2x\Delta u)^2)^\alpha (1 - (2y\Delta v)^2)^\alpha dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |\hat{C}(x, y)|^2 |1 - (2x\Delta u)^2|^\alpha |1 - (2y\Delta v)^2|^\alpha dx dy}$$

is maximized (see [F. R. Schwab, Optimal Gridding, VLA Scientific Memo. No. 132]). The weighting exponent  $\alpha$  is used to trade off the effectiveness of the aliasing suppression at the edge of the field of view, against that in the central region of the map. The choice  $\alpha = 1$ , with a *support width* of six *u-v* grid cells, yields an effective gridding convolution function, emphasizing aliasing suppression in the central region of the map; this function,  $\psi_{10}$ , with  $c = 3\pi/2$ , is the default function used in the AIPS mapping program. See *gridding convolution function*.

**Stokes' parameters** —

**Stokes' visibility functions** — Stokes' visibility functions,  $V_I$ ,  $V_Q$ ,  $V_U$ , and  $V_V$ , are the Fourier transforms (FT's) of the radio brightness distributions of *Stokes' parameters*,  $I(x, y)$ ,  $Q(x, y)$ ,  $U(x, y)$ , and  $V(x, y)$ . (Here,  $V_I = \hat{I}$ ,  $V_Q = \hat{Q}$ , etc., where  $\hat{\cdot}$  denotes FT.)

For a radio interferometer with ideal circularly polarized feeds, the relations between Stokes' visibility functions and the visibilities,  $V_{RR}$ ,  $V_{LL}$ ,  $V_{RL}$ , and  $V_{LR}$ , derived by correlating right circular response with right, left with left, etc., are  $V_I = \frac{1}{2}(V_{RR} + V_{LL})$ ,  $V_Q = \frac{1}{2}(V_{LR} + V_{RL})$ ,  $V_U = \frac{1}{2}(V_{LR} - V_{RL})$ ,  $V_V = \frac{1}{2}(V_{RR} - V_{LL})$ . Note that each of Stokes' visibility functions is *Hermitian*. On the assumption that circular polarization is absent (i.e., that  $V(x, y) \equiv 0$ ),  $V_{RR}$  is equal to  $V_{LL}$ , and both are Hermitian.

Components of the systematic errors affecting visibility measurements are i.f.-dependent; hence VLA *u-v* data files usually do not contain Stokes' visibilities, but rather  $V_{RR}$ ,  $V_{LL}$ ,  $V_{RL}$ , and  $V_{LR}$ —as these are what is required for calibration purposes. Stokes' visibility functions generally are constructed only within the mapping programs. (But the AIPS visibility data format is designed to accommodate either type of visibility function, and the mapping task is able to recognize the form of its input data and deal with it appropriately.)

**subimage** — any completely filled linear, rectangular, or hyper-rectangular portion of an *image*.

**sub-task** — a task, or computer program, whose execution is initiated by the action of another program. The act of initiating the execution of the sub-task is called *task shedding* or *task spawning*. See *task*.

**support** — The closure of that subset of the domain of definition of a function  $f$  (or of a generalized function, or distribution) on which the function assumes a nonzero value is called the *support* or the *support set* of the function, and is denoted by  $\text{supp}(f)$ . I.e.,  $\text{supp}(f) = \{x | f(x) \neq 0\}$ .

For example, the support of the function  $f(x) = x$  is the whole real line, even though  $f(0) = 0$ . And the support of

$$f(x, y) = \begin{cases} 1, & x^2 + y^2 < 1, \\ 0, & \text{otherwise,} \end{cases}$$

is the *closed* unit disk,  $\{(x, y) | x^2 + y^2 \leq 1\}$ .

In Euclidean space, a function  $f$  whose *support set* is bounded—i.e., such that  $f \equiv 0$  “far-out”—is said to be of *compact support*. The Fourier transform of a nontrivial function of compact support (such as a *u-v measurement distribution* or a *gridding convolution function*) cannot itself be of c.s.; i.e., it has “sidelobes” extending to infinity.

**support width** — of a function whose *support* is a rectangle or a hyper-rectangle (e.g., a band-limited function), the linear measure of one of the edges of its *support set*.

**SVPSF** — see *point spread function*.

**synthesized beam** — in radio interferometry, the *beam*—but always ignoring instrumental effects. Hence, the synthesized beam is fully determined by the *u-v* sampling distribution, the *u-v* weight function, the *u-v* taper function, and the gridding convolution function. See *beam*.

**tape blocking efficiency** — Data are stored on magnetic tape in units of *blocks*. An *inter-record gap*—essentially



wasted space—separates the blocks. The *tape blocking efficiency*, or the fraction of unwasted space, is the ratio

$$\frac{\frac{\text{block length}}{\text{recording density}}}{\frac{\text{block length}}{\text{recording density}} + \text{length of an inter-record gap}}$$

The length of an inter-record gap is about  $\frac{3}{4}$ ,  $\frac{3}{5}$ , and  $\frac{3}{10}$  inch at recording densities of 800, 1600, and 6250 bpi, respectively.

**tape label** —

**taper** — see *u-v taper function*.

**task** — used in two senses: 1) the execution of a computer program; and 2), the program itself. Thus, if two computer users are (independently) running the same program at the same time, it may be said either that two tasks are running, or that two incarnations of the same task are in existence. A *sub-task* (*q.v.*) is a task whose execution is initiated by the action of another program. Many of the more complicated and the more specialized functions of AIPS are accomplished by the action of sub-tasks shed by the AIPS program. (Simpler functions are invoked by the issuance of *verb* commands—see *POPS symbols*.)

**t-b order** — same as *time-baseline order*.

**TEK screen** — a cathode ray tube (CRT) terminal and display device appropriate for pictorial display of data, in the form of contour plots, graphs, etc., as well as for display of textual data. The Tektronix company's Model 4012 terminal (with a green P4 phosphor, hence the synonymous term *green screen*) is the canonical device of this type. The "make copy" button on this device can be used to produce a copy, on paper, of the image shown on the CRT screen. Each of the NRAO's AIPS data reduction computers is outfitted with a *TEK screen*.

**TEK4012** — same as *TEK screen*.

**Telex 6250 tape drive** — a model of tape drive used on the VLA Vaxes, capable of operation at 1600 and 6250 bpi.

**terminal page** — Many modern computer terminals contain a semiconductor memory with a capacity of several CRT screen loads ( $\approx 24$  lines) of character data. A *terminal page* is a unit of one screen load of such data. Certain terminal keys allow one to cause data which previously appeared on the CRT screen to reappear—this feature is called *terminal scroll* (*q.v.*). A typical terminal at the NRAO has three terminal pages of memory.

**terminal scroll** — that feature present on certain models of computer terminals which allows data which previously appeared on the CRT screen to be made to reappear. Often, depressing one key on the terminal will cause earlier information to reappear line-by-line (this is termed *line scroll*), while the action of another key will cause a whole earlier screen load to reappear (this is termed *page scroll*).

**text editor** — a computer program designed for the creation, manipulation, and modification of computer files containing textual data such as reports, documentation, alphanumeric command lines, and program source code. Generally, one or more text editors is supplied by the computer manufacturer. Three text editors are in widespread use on the Vax— *SOS*, *EMACS*, and *EDT*—, and *SEdit* is used on Modcomp computers.

**text file** — a computer data file containing only textual data, as might be written by a *text editor* (*q.v.*). Programs such as the AIPS tasks often write messages, especially progress report messages, into a text file—see *message file*.

**thrashing** — see *memory thrashing*.

**time-baseline order** — An ordered set of visibility measurements  $\{V_{ij}(t_k) \mid 1 \leq i < j \leq n, k = 1, \dots, l\}$  recorded with an  $n$  element interferometer at times  $t_1 < t_2 < \dots < t_l$  is said to be in *time-baseline order* if the ordering is such that all of the data obtained at time  $t_1$ , sorted into the canonical ordering by baseline, occur first, followed by the data obtained at time  $t_2$ , again ordered canonically, etc., etc. (The canonical ordering by baseline is  $V_{12}, V_{13}, \dots, V_{1n}, V_{23}, \dots, \dots, V_{n-1,n}$ .) Compare *baseline-time order*.

Time-baseline ordering of a *u-v* data file is convenient for calibration purposes. The AIPS task for self-calibration requires that its input *u-v* data file be time-baseline ordered.

**trackball** — a spherical ball mechanism, about the size (10 cm., or so, in diameter) of a tennis ball, which may be oriented manually by the interactive user of a television display device such as the I<sup>2</sup>S. The ball can be rotated about any axis, and its orientation, which is sensed by the computer, typically is used to control the enhancement or the coloration of the displayed data (i.e., to control the TV *transfer function(s)*), or to position the TV cursor, in order to point out to a program features in the displayed image which are of particular interest.

**trackball button** — On the unit which houses the trackball for the I<sup>2</sup>S Model 70 TV display device are the four *trackball buttons*, labeled A, B, C, and D. These are switches that are used, in conjunction with the display routines, to exert additional control over the TV display. Occasionally these buttons are put to other use in AIPS, such as stopping the CLEAN deconvolution program.

**transfer function** —

**TRC** — *top right corner*, the corner of an image diagonally opposite the BLC. See *m × n map*.

**true color display** —

**TU77 tape drive** — a model of tape drive used on the NRAO's Vaxes, capable of operation at 800 and 1600 bpi.

**TU78 tape drive** — a model of tape drive used on the VLA Vaxes, capable of operation at 1600 and 6250 bpi.

**TV blink** —

**TV cursor** — see *crosshair*.

**TV look-up table** — a memory within the control unit of a TV display device which is used for storage of the *transfer functions* controlling the intensity of the display, as a function of pixel value. Within AIPS, the transfer functions may be altered through the use of interactive verbs and manipulation of the *trackball*.

**TV roam** — a feature of a computer-controlled TV display device such as the I<sup>2</sup>S which allows contiguous parts of a single large image, stored on more than one image plane, to be displayed as if the image were stored on a single, larger image plane. See *image plane*.

**TV scroll** —

**TV zoom** —

**uniform weighting** —

**units of computer storage** —

**UNIX** — a “universal” computer operating system developed at the Bell Telephone Laboratories. Its virtue is that systems such as AIPS—once having been made to run under one UNIX-based operating system—ought to be able to run on any other such system, even on a computer of different manufacture, with no alterations. Many Vaxes operate under UNIX, though not the NRAO's. A UNIX operating system is run on the IBM-4341 in Charlottesville. See *operating system*.

**user-coded task** — an AIPS task written by a user, rather than by a professional programmer or a member of the AIPS programming group. One of the design goals for AIPS, not yet fully realized, has been that it should be as easy as is practicable for a user who is not an experienced programmer to write an AIPS task suited to his own needs—i.e., that it should be fairly simple for him to make some sense of the AIPS database, and to get at his data and manipulate it as he sees fit. There is an AIPS task called FUDGE which is designed as a paradigm for modification of a *u-v* data file.

**u-v coverage** — the support of the *u-v sampling distribution* (*q.v.*). Also see *conjugate symmetry*.

**u-v data file** — in AIPS, a *primary data file* designed to accommodate the measurements of the visibility function of a radio source.

**u-v FITS format** — an extension of the *FITS format* (originally designed for the interchange of image data) to accommodate radio interferometer visibility data [E. W. Greisen and R. H. Harten, *Astron. Astrophys. Suppl. Ser.*, **44** (1981) 371–374]. See *FITS format*.

**u-v measurement distribution** — in radio interferometry, a linear combination of shifted Dirac  $\delta$ -functions, one located at the position in the *u-v* plane of each visibility measurement, and each weighted by the visibility measurement obtained at that location. Denoting the *u-v* coverage by  $\{(u_i, v_i)\}_{i=1}^N$ , the visibility function by  $V$ , and the measured visibility by  $\tilde{V}$ , the (two-dimensional) *u-v* measurement distribution  $S$  is given by  $S(u, v) = \sum_{i=1}^N \tilde{V}(u_i, v_i) \delta(u - u_i, v - v_i)$ . Compare *u-v sampling distribution*.

This definition may be modified to incorporate two types of weight function, yielding a *weighted* and/or *tapered* measurement distribution—see *u-v taper function* and *u-v weight function*.

The visibility measurements  $\{\tilde{V}(u_i, v_i)\}$  are not actual samples of  $V$ , but rather are error-corrupted samples of a function which represents some sort of local average of the visibility—this is a distinction which it is worthwhile to note, and then to ignore. Generally one assumes the relation between the visibility measurement obtained on the *i-j* baseline at time  $t$  and the true visibility to be given by  $\tilde{V}(u_{ij}(t), v_{ij}(t)) = g_i(t) \bar{g}_j(t) V(u_{ij}(t), v_{ij}(t)) + \epsilon_{ij}(t)$ , where the spatial frequency coordinates  $(u, v)$  have been parametrized by time, where the multiplicative errors  $g$  are comprised of a component of systematic “calibration” error and a random component, and where the additive error  $\epsilon$  is random and well-behaved.

**u-v sampling distribution** — in radio interferometry, a linear combination of shifted Dirac  $\delta$ -functions, one located at the position in the *u-v* plane of each visibility measurement. Sometimes termed a *u-v transfer function*. See *beam*.

If  $\{(u_i, v_i)\}_{i=1}^N$  (the *u-v coverage*) is the set of spatial frequency coordinates at which the source visibility has been

sampled, then the (two-dimensional) *u-v* sampling distribution  $S$  is given by  $S(u, v) = \sum_{i=1}^N \delta(u - u_i, v - v_i)$ .

Occasionally the term *u-v sampling distribution* is used in the same sense as the term *u-v measurement distribution* (*q.v.*).

**u-v taper function** — an even, real-valued weight function (typically, an elliptical Gaussian), smooth and peaked at the origin, which may be incorporated into the definition of *u-v measurement distribution* or *u-v sampling distribution*, above, serving to control the spatial resolution of the radio map or the beam; i.e., to enhance the response to extended features in the radio source brightness distribution by giving relatively higher weight to the measurements at short *u-v* spacings. Compare *u-v weight function*.

**u-v transfer function** — same as *u-v sampling distribution*, but always explicitly incorporating any *u-v* weight function or *u-v taper function*.

**u-v weight function** — a real-valued function which may be incorporated in the definition, above, of *u-v measurement distribution* or *u-v sampling distribution*, serving to weight each measurement either according to an estimate of the statistical measurement error, or according to the local density of sampling, or both. Compare *u-v taper function*.

**Varian printer** — an electrostatic printer/plotter manufactured by the Varian Corp., and attached to the Modcomp Classic computer used to run AIPS in Charlottesville.

**verb** — see *POPS symbols*.

**Versatec printer** — an electrostatic printer/plotter manufactured by the Versatec Corp., and used on the NRAO's AIPS computer systems.

**virtual memory page** — on a computer running under a virtual memory operating system, one unit of *virtual memory storage*. At a typical Vax installation, the size of a virtual memory page is 512 bytes.

**virtual memory page swapping** — on a computer running under a virtual memory operating system, the action (initiated automatically by the operating system) of reading new virtual memory pages into the physical memory, and storing on disk (i.e., in the *virtual memory*) the data which thus have been displaced. Each occurrence of the displacement of a memory page is referred to as a *page fault*. See *memory thrashing*.

**virtual memory storage** —

**VMS** — (Virtual Memory System) the operating system used on the NRAO's Vax computers. See *virtual memory storage* and *operating system*.

**wedge** —

**window clean** — an application of the Högbom CLEAN algorithm, with an explicit specification, by the user, of the clean window. Generally it is recommended that the user specify a clean window whenever he can make a reasonably valid and restrictive estimate of the support of the true radio source brightness distribution. See *clean window*.

**working set size** — on a computer running under a virtual memory operating system, the amount of *physical memory* allocated to a task. Any program memory requirement in excess of the working set size is relegated to *virtual memory storage*. At a typical Vax installation, the working set size is set at  $\frac{1}{4}$  or  $\frac{1}{2}$  megabyte.

**x-y order** — An ordered set of visibility samples  $\{V(u_i, v_i, w_i)\}_{i=1}^n$  arranged according to descending absolute value of the spatial frequency coordinate  $u$  — i.e., with  $|u_1| \geq |u_2| \geq \dots \geq |u_n|$  — is said to be in *x-y order*.

*x-y order* is a convenient ordering for the operation of gridding convolution; hence the AIPS mapping task requires that its input *u-v data file* be sorted accordingly. See *sort order*.

**zoom** — see *TV zoom*.

**4012** — see *TEK screen*.

## Appendix Z. SYSTEM-DEPENDENT AIPS TIPS

Although *AIPS* attempts to be system independent, some aspects of its use inevitably depend on the specific site. These vary from procedural matters (e.g. location of sign up sheets, terminals, and tape drives) to the hardware (e.g. names and numbers of tape and disk drives, the parameters of television and array processor devices) to the peculiar (e.g. the response of the computer to specific keys on the terminal, the presence of useful job control procedures). This Appendix contains information specific to NRAO's individual *AIPS* installations.

### Z.1. VLA Site VAXes

This Section contains information specific to the *AIPS* hardware and software at the VLA. The computers are two VAX 11/780s running the VMS operating system each equipped with an I<sup>2</sup>S display and a Floating-Point Systems AP-120B array processor. Each has three (logical) disks and two or three tape drives. The *AIPS* Caige (VAX room) is located at the East end of the Library-Office Building. There are two *AIPS* terminals (Tektronix 4025s) for each computer. To the left as you enter the door are those for the older VAX known as VAX1 (or as *AIPS*:: to Decnet). To the right are the terminals for VAX3. Both I<sup>2</sup>S devices are equipped with three gray-scale memory planes and four graphics overlay planes. A Tektronix 4012 is available for TK graphics outputs on each. A Versatec is used for plotting and hard copy of the TK screens on each and is also used, on VAX1, for printed output. On VAX3, a separate line printer is used for print outs.

#### Z.1.1. Signing up for *AIPS* time at the VLA

There is a sign-up sheet for *AIPS* on the noticeboard just to the left of the entrance to the *AIPS* Caige in the Library-Office Building. As *AIPS* is used by a majority of VLA observers, it is usually essential to sign up for your *AIPS* time in advance. To promote fair and efficient use of the system, there are strong restrictions on the amounts of time that any one user or user group may reserve — see the *AIPS* Caige noticeboard for details.

*AIPS* supports up to three simultaneous users on each computer at the VLA. The user signed up for *AIPS*1 has priority for the use of the displays, array processor, and tape drives. In many cases however, the user signed up for *AIPS*2 may carry out similar processing tasks, by verbal negotiation with the *AIPS*1 user over the use of these resources. The facilities available through *AIPS*3 are very limited and most users will require either *AIPS*1 or *AIPS*2.

#### Z.1.2. Using the Tektronix 4025 terminals at the VLA

As you attempt to correct mistyped characters, you will note that CTRL H does not delete on VAX/VMS terminals as it does on DEC-10 terminals. Use the RUB OUT key at the upper right of the keyboard instead of CTRL H.

During execution, paging can be halted by typing CTRL S and resumed by typing CTRL Q. To abort any execution, type CTRL Y or CTRL C. Using CTRL Y while in *AIPS* will unceremoniously eject the user to

monitor level (prompt \$). If CTRL Y was entered accidentally, AIPS may be resumed by typing CONT C<sub>R</sub>. If the abort was intentional, then the user will have to reinitialize AIPS with all the input parameters having been lost.

The Tektronix 4025 terminals have three pages of memory. To scroll upwards press the 1 key on the numeric pad to the right of the main keyboard. The 2 key on the numeric pad will scroll downwards.

### Z.1.3. Logging in at the VLA

Go to the AIPS Cage at the East end of the Library-Office Building and find the terminal connected to the VAX for which you have signed up. Each Tektronix 4025 is labeled to tell you which computer and which AIPS number it is for. Typing C<sub>R</sub> on the terminal will produce a prompt which will tell you the current state of that terminal.

If the prompt symbol is > the terminal is already being used for AIPS. Check with other possible users before proceeding. If it's OK to use the system, type either

EXIT C<sub>R</sub>

or

QE C<sub>R</sub>

depending on whether the previous user wants or does not want, respectively, his message log.

If the prompt is \$ (due to the initial C<sub>R</sub> or to following the step above), the computer is at system level, but not in use. Check nearby users to make sure and then type

LO C<sub>R</sub>

C<sub>R</sub>

to log off the previous user and to begin the log in process.

Either the above or the initial C<sub>R</sub> should give a prompt of Username:, which means that you can log in. Proceed as follows:

Username: AIPS C<sub>R</sub>

The message **Welcome to VAX/VMS Version ...** should appear. After any system messages the question **Enter AIPS version you want to run. NEW, OLD, or NONE:** will appear. Type in your choice. The OLD version is likely to be relatively free of bugs, but the NEW version will contain the latest improvements. (Note that this choice affects only the version of the AIPS program itself. You may choose NEW or OLD versions of the AIPS reduction programs at a later time.) Following this is a message about tape mounting. This is the simplest time to mount a tape if you will be reading in images from a FITS- or IBM-format tape or UV data from an EXPORT- or FITS-format tape. The tape drive(s) may already be assigned to another terminal, however. Check with other active VAX users if in doubt at this point, then proceed with steps 1 and 2 below if you indeed want to mount a tape. If not, skip to step 2.

1. Physically mount the tape as described in § Z.1.4 below. You must mount the tape before typing in the tape drive number (the next step).
2. Type C<sub>R</sub> if you do not want to mount a tape.  
Type n C<sub>R</sub> to software-mount the tape drive labeled n.

You should then see the message:

**BEGIN THE ONE TRUE AIPS NUMBER *n***

where *n* is a number between 1 and 3. Check that this "AIPS number" is indeed the one for which you signed up. If not, you are on the wrong terminal: AIPS1 and AIPS2 have specific terminals, labeled as such, while all other AIPS numbers are available on a first-come-first-served basis. AIPS will now ask you for your VLA user number:

**AIPS: ENTER USER ID NUMBER**

**? *uuu* CR**

where *uuu* is your VLA user number. The AIPS prompt **>** should now appear.

#### **Z.1.4. Hardware tape mount at the VLA**

Go into the "Inner Sanctum" of the VAX room containing the VAXes themselves. Three drives are available on VAX1: two Digital TU78s operable at 1600 or 6250 BPI and a Telex 6250 operable at 1600 or 6250 BPI. VAX3 also has a Telex 6250 plus a Digital TU77 which operates at 800 or 1600 BPI. FITS and EXPORT tapes written on the DEC-10 are normally at 1600 BPI density, while FITS tapes written by the 11/70 MAPPER or 11/70 SORTER may be at either 1600 or 6250 BPI density.

##### **Z.1.4.1. Mounting tapes on TU77s and TU78s**

Open the door by pulling from the right hand edge. The upper spool is ready to receive or release a tape reel if a red band shows inside the spool hub. Mount your tape on this upper spool so that the protrusions in its protective rim engage the slots to the right hand side of the tape enclosure. Press the central lock in the hub until it snaps into position with no red band showing. Shut the door and press the white LOAD button at the left of the button row between the spools. The tape should thread itself and advance to the first file after which the BOT signal should light up. Press the ON LINE button — the ON LINE signal should light up.

If this sequence fails repeatedly, consult a VAX advisor (see the list posted on the noticeboard beside the entrance to the main VAX room).

##### **Z.1.4.2. Mounting tapes on Telex 6250s**

Open the door by pulling from the left hand edge. Mount the tape on the upper spool so that the protrusions on its protective rim engage the slots on the upper side of the tape enclosure. There is no manual lock for the tape spool on this drive. Close the door, then press the white LOAD REWIND button, followed by the ON LINE button. The tape should load and go to the BOT point for the first tape file. If you have difficulty, consult a VAX advisor (see the list on the notice board beside the entrance to the main VAX room).

**Z.1.5. Software tape mount at the VLA**

When you have the tape mounted on the tape drive, VAX/VMS must also mount the tape in software. At the moment, this must be done at system job control level. If you are logging in to AIPS, go to step 2 of the login procedure described in § Z.1.3 above. If you have already logged in to AIPS without mounting a tape:

```
> STORE 1 CR          to save your AIPS inputs.
> EXIT CR             to exit to monitor level.
$ @MOUNTn CR          to mount the tape in software on the drive labeled n.
$ @AIPS CR            to reactivate AIPS.
```

The message **Enter the AIPS version you want, NEW or OLD** will appear. Repeat the answer you gave when first starting AIPS. Then the message **AIPS: ENTER USER ID NUMBER** will appear, followed by a ? prompt. Now type your VLA User Number followed by CR. Then:

```
> RESTORE 1 CR        to recover your AIPS inputs.
```

**Z.1.6. Monitoring disk space at the VLA**

See § 4.4 for a general discussion of the problem of disk space. Since disk space is often very tight, there are, at the VLA, a number of system utilities and procedures to assist the user to monitor how much space is available and to clear additional space as required. These capabilities are available only at monitor (system job control) level. To use them:

```
> STORE 1 CR          to save your AIPS inputs.
> EXIT CR             to exit from AIPS.
```

The job control command:

```
$ SHOW DEV MX CR
```

will list the vacant space on each of the disks. When the available space drops to less than about 30,000 blocks some programs will begin to fail. On VAX1, disks **MXA0** and **MXA1** together correspond to **INDISK 1**; **MXA2** corresponds to **INDISK 2**; and **MXA3** corresponds to **INDISK 3**. On VAX3, **MXA0** and **MXA1** are combined as logical disk 1, **MXA2** is disk 2, and **MXA3** is disk 3.

Sometimes there are lots of scratch files which are no longer in use and can be deleted. To check this, exit to monitor level (prompt \$) and type:

```
$ @SCRATCH CR
```

to list all known scratch files and their creation times. If there are any old files listed, make sure that nobody else is executing any programs in AIPS by:

```
$ SHOW SYS CR          to list all VAX users.
```

If no AIPS tasks are running, then use:

```
$ @AJAX CR
```

to delete all scratch files generated by the AIPS system. Be *very* careful about this since you can delete a scratch file which is currently being used by another AIPS user (or batch task). Don't worry about all the error messages which come out while AJAX is running. The system is simply reporting that some of the possible scratch files did not in fact exist. Type:

```
$ SHOW DEV MX CR
```

to find out if you have freed up enough disk space.

If destroying scratch files is not enough, you will need to find the worst disk space hogs and apply appropriate peer pressure. At monitor level, type:

**\$ @SPACE CR**

which spawns the process "SYSYPHUS" to find out who has how much disk space and when they last used it. After about 1-2 minutes a list will appear on the line printer of the disk usage of each user. You must then persuade one or more of these recalcitrants to relinquish some space. DO NOT delete files without the express permission of either the user or the System Manager.

### **Z.1.7. Solving problems at the VLA**

Below are the details specific to the VLA VAX/VMS systems for handling some of the problems which may arise in *AIPS*.

#### **Z.1.7.1. Stopping excess printout at the VLA**

If you wish to abort a job which is currently printing out do the following:

1. Turn the power off on the printer to conserve paper.
2. Get into monitor mode (prompt \$). See a suggested procedure for this in § Z.1.5 above.
3. Type **SHOW/QUE LPA0: CR**. On VAX3, substitute **LPB0:** for **LPA0:** .
4. Find the job number, *nnn*, associated with the current print job.
5. Type **DELETE/ENTRY = nnn LPA0: CR**. On VAX3, substitute **LPB0:** for **LPA0:** .
6. Turn the power back on for the printer.

#### **Z.1.7.2. Hard copy device**

The Versatec printer/plotter is used as the hard copy device. Try the instructions in § 11.2. Paper for the Versatecs is kept on the metal cabinet at the back of the VAX computer room ("Inner Sanctum").



### Z.1.7.3. Booting the VLA VAX systems

Large VAX systems, such as those at the VLA, should not need to be booted manually except under special circumstances. Consult a staff member before booting. Detailed instructions for booting are posted on the front panel of each VAX.

To reboot the VAX, go to the console of your VAX at the right of the "Inner Sanctum" and then type:

CTRL P	to get its attention and the prompt >>>.
>>> HALT CR	to bring its activities to an orderly stop.
>>> BOO CR	to reboot it.

Eventually (about 3 minutes later, following a torrent of messages about disks being dismounted and mounted, DECNET and other esoterica), the message **Welcome to VAX/VMS** ... (or a simple variant, depending on which VAX you are using) should appear on the console and on all of the terminals connected to that VAX. The hangups should be cured by this. If they are not, summon a VAX Advisor, or go to bed, depending on the time of day.

### Z.1.7.4 Other problems at the VLA

Sometimes tasks under development or investigation are compiled and linked in a "debug" mode. If they have been left this way, they will wake up (after the GO command) and issue a prompt of **DBG>**. To run the program, type

DBG> GO CR	to start it.
------------	--------------

and, after it has finished and issued another prompt:

DBG> EXIT CR	to finish it.
--------------	---------------

Tell the System Manager so that the program may be relinked in normal mode.

### Z.1.8. Editing RUN files at the VLA

RUN files are located in a directory called **MXAO:[AIPS.RUN]** and must have the three-letter extension **.RUN** in their names. They are edited from monitor level using either **EDT** or **SOS**. For instructions on these editors consult Volume 3 of the VAX/VMS Manuals. An example appropriate to the VAX is given in § 12.6.

## Z.2. Charlottesville VAX

This Section contains information specific to the *AIPS* VAX hardware and software in Charlottesville. The computer is a VAX 11/780 running the VMS operating system and equipped with an I<sup>2</sup>S television display device and a Floating-Point Systems AP-120B array processor. It has three disks and two tape drives. The *AIPS* Caige is located on the second floor in Room 215 of the Edgemont Road Office Building. The primary *AIPS* terminal is a Tektronix 4025 located at the back left (as you enter) next to the television monitor. Three other terminals (2 Visual 400s and a Visual 100) are located along the left side of the room. Additional VAX terminals are located in Room 214. The I<sup>2</sup>S device is equipped with four gray-scale memory planes and four graphics overlay planes. A Tektronix 4012 is available in the *AIPS* Caige for TK graphics output. A Versatec is used for plotting and printing.

### Z.2.1. Signing up for *AIPS* time on the CV VAX

There is a sign-up sheet for *AIPS* on the noticeboard just to the left of the entrance to the *AIPS* Caige, Room 215. If you wish to be certain of priority on the VAX, it is usually advisable to sign up for your *AIPS* time in advance. To promote fair and efficient use of the system, users are asked to restrict the amount of time that they reserve. Formal rules are not now in effect, but may be imposed should the need arise. The Charlottesville VAX also supports significant VLBI data reduction and most of the *AIPS* software development. These projects must receive their fair share of the computer resources.

*AIPS* allows up to six simultaneous users on CVAX:: (as the Charlottesville VAX is called by Decnet). The user signed up for AIPS1 has priority for the use of the displays, array processor, and tape drives. He is expected to be reasonable about sharing these resources with the programming staff and other users, however. Normally, the facilities available through the higher numbered AIPS*n* are sufficient for most users.

### Z.2.2. Using the terminals on the CV VAX

To correct characters which you have typed, do not use the BACK SPACE key on VAX/VMS systems. Instead, on the Tektronix 4025, use the RUB OUT key at the upper right of the keyboard. On the Visual terminals, press the SHIFT key and hit the DEL key as often as required. (Visual 100s do not require the SHIFT part.)

During execution, paging can be halted by typing CTRL S and resumed by typing CTRL Q. To abort any execution, type CTRL Y or CTRL C. Using CTRL Y while in *AIPS* will unceremoniously eject the user to monitor level (prompt \$). If CTRL Y was entered accidentally, *AIPS* may be resumed by typing CONT Q<sub>R</sub>. If the abort was intentional, then the user will have to reinitialize *AIPS* with all the input parameters having been lost.

The Tektronix 4025 terminal has three pages of memory. To scroll upwards press the 1 key on the numeric pad to the right of the main keyboard. The 2 key on the numeric pad will scroll downwards. The Visual 400s also have three pages of memory. Hold the SHIFT key down and press the "up-arrow" key to scroll upwards and the "down-arrow" key to scroll back.

**Z.2.3. Logging in to AIPS on the CV VAX**

Go to the *AIPS* Cage in Room 215 and find a terminal connected to the VAX. The Tektronix 4025, located next to the television monitor, is reserved for the user who has signed up to get AIPS1. All other terminals are available on a first-come-first-served basis. We recommend the Visual 400s for use in *AIPS* since they have a multi-page memory. Typing `CR` on the terminal will produce a prompt which will tell you the current state of that terminal.

If the prompt symbol is `>` the terminal is already being used for *AIPS*. Check with other possible users before proceeding. If it's OK to use the system, type either

`EXIT CR`

or

`QE CR`

depending on whether the previous user wants or does not want, respectively, his message log.

If the prompt is `$` (due to the initial `CR` or to following the step above), the computer is at system level, but not in use. Check nearby users to make sure and then type

`LO CR`

`CR`

to log off the previous user and to begin the log in process.

Either the above or the initial `CR` should give a prompt of `Username:`, which means that you can log in. Proceed as follows:

`Username: AIPS CR`

The message **Welcome to VAX/VMS Version ...** should appear. After any system messages the question **Enter AIPS version you want to run. NEW, OLD, or NONE:** will appear. Type in your choice. The OLD version is likely to be relatively free of bugs, but the NEW version will contain the latest improvements. (Note that this choice affects only the version of the *AIPS* program itself. You may choose NEW or OLD versions of the *AIPS* reduction programs at a later time.) Following this is a message about tape mounting. This is the simplest time to mount a tape if you will be reading in images from a FITS- or IBM-format tape or UV data from an EXPORT- or FITS-format tape. The tape drive(s) may already be assigned to another terminal, however. Check with other active VAX users if in doubt at this point, then proceed with steps 1 and 2 below if you indeed want to mount a tape. If not, skip to step 2.

1. Physically mount the tape as described in § Z.2.4 below. You must mount the tape before typing in the tape drive number (the next step).
2. Type `CR` if you do not want to mount a tape.  
Type `n CR` to software-mount the tape drive labeled `n`.

You should then see the message:

`BEGIN THE ONE TRUE AIPS NUMBER n`

where `n` is a number between 1 and 6. AIPS will now ask you for your user number:

`AIPS: ENTER USER ID NUMBER`

`? uuu CR`

where `uuu` is your Charlottesville user number. The AIPS prompt `>` should now appear.

### Z.2.4. Hardware tape mount on the CV VAX

The VAX has two tape drives. Drive number 1 is a TU77 drive located in the VAX (and IBM) machine room on the first floor (Room 112). The VAX is the computer all the way to the left (as you enter the room) next to the windows and the drive is adjacent to the CPU. Densities of 800 and 1600 BPI are supported. Tape drive number 2 is located in the AIPS Caige just behind the primary AIPS user's chair. It is a Systems Industries drive and supports 800, 1600, and 6250 BPI.

#### Z.2.4.1. Mounting tapes on the TU77

Open the door by pulling from the right hand edge. The upper spool is ready to receive or release a tape reel if a red band shows inside the spool hub. Mount your tape on this upper spool so that the protrusions in its protective rim engage the slots to the right hand side of the tape enclosure. Press the central lock in the hub until it snaps into position with no red band showing. Shut the door and press the white **LOAD** button at the left of the button row between the spools. The tape should thread itself and advance to the first file after which the **BOT** signal should light up. Press the **ON LINE** button — the **ON LINE** signal should light up. If this sequence fails repeatedly, consult a VAX advisor.

#### Z.2.4.2. Mounting tapes on the Systems Industries drive

Open the door by removing the metal "night door latch", if needed. The door will slide down. Pull out (toward you) the central lock in the hub. Mount the tape on the spool so that the protrusions on its protective rim engage the slots on the left side (slightly below center) of the tape enclosure. Then press the **LOAD REWIND** button, followed by the **ON LINE** button. The door will close and the tape should load and go to the **BOT** point for the first tape file. If you have difficulty, consult a VAX advisor.

### Z.2.5. Software tape mount on the CV VAX

When you have the tape mounted on the tape drive, VAX/VMS must also mount the tape in software. At the moment, this must be done at system job control level. If you are logging in to AIPS, go to step 2 of the login procedure described in § Z.2.3 above. If you have already logged in to AIPS without mounting a tape:

> STORE 1 CR	to save your AIPS inputs.
> EXIT CR	to exit to monitor level.
\$ @MOUNTn CR	to mount the tape in software on the drive labeled n.
\$ @AIPS CR	to reactivate AIPS.

The message **Enter the AIPS version you want, NEW or OLD** will appear. Repeat the answer you gave when first starting AIPS. Then the message **AIPS: ENTER USER ID NUMBER** will appear, followed by a ? prompt. Now type your VLA User Number followed by CR. Then:

> RESTORE 1 CR	to recover your AIPS inputs.
----------------	------------------------------

### Z.2.6. Monitoring disk space on the CV VAX

See § 4.4 for a general discussion of the problem of disk space. The verb **TIMDEST** is useful, but we prefer that users ask the System Manager before invoking it. There are four disk packs for disk drive number 3. The packs are called **VLAV2**, **ALBERT**, **BRYAN**, and **CALVIN**. Users with large disk space requirements should inquire about having one of these temporarily assigned to them and should force their output files to disk 3. Disk packs are to be changed only by the IBM operators. When you are the primary *AIPS* user, you may ask them to mount the pack assigned to you. At other times, you must negotiate with the other users of the VAX before asking the IBM operators to change packs.

Since disk space is often very tight, there are, on the Charlottesville VAX, a number of system utilities and procedures to assist the user to monitor how much space is available and to clear additional space as required. These capabilities are available only at monitor (system job control) level. To use them:

> STORE 1 CR                      to save your **AIPS** inputs.  
> EXIT CR                        to exit from **AIPS**.

The job control command:

\$ SHOW DEV UM CR

will list the vacant space on each of the disks. When the available space drops to less than about 30,000 blocks some programs will begin to fail. On CVAX, the disk **UMAO** corresponds to **INDISK 1**; **UMA1** corresponds to **INDISK 2**; and **UMA2** corresponds to **INDISK 3**. This command is useful for determining if your assigned disk pack is currently mounted.

Sometimes there are lots of scratch files which are no longer in use and can be deleted. To check this, exit to monitor level (prompt \$) and type:

\$ @SCRATCH CR

to list all known scratch files and their creation times. If there are any old files listed, make sure that nobody else is executing any programs in *AIPS* by:

\$ SHOW SYS CR                      to list all VAX users.

If no *AIPS* tasks are running, then use:

\$ @AJAX CR

to delete all scratch files generated by the *AIPS* system. Be *very* careful about this since you can delete a scratch file which is currently being used by another *AIPS* user (or batch task). Don't worry about all the error messages which come out while **AJAX** is running. The system is simply reporting that some of the possible scratch files did not in fact exist. Type:

\$ SHOW DEV UM CR

to find out if you have freed up enough disk space.

If destroying scratch files is not enough, you will need to find the worst disk space hogs and apply appropriate peer pressure. At monitor level, type:

\$ @SPACE CR

which spawns the process "**SYSPHUS**" to find out who has how much disk space and when they last used it. After about 1-2 minutes a list will appear on the line printer of the disk usage of each user. You must then persuade one or more of these recalcitrants to relinquish some space. **DO NOT** delete files without the express permission of either the user or the System Manager (other than with the *AIPS* verb **TIMDEST**).

### Z.2.7. Solving problems on the CV VAX

Below are the details specific to the Charlottesville VAX/VMS system for handling some of the problems which may arise in *AIPS*.

#### Z.2.7.1. Stopping excess printout on the CV VAX

If you wish to abort a job which is currently printing out do the following:

1. Turn the power off on the printer to conserve paper.
2. Get into monitor mode (prompt \$). See a suggested procedure for this in § Z.2.5 above.
3. Type SHOW/QUE LPA0: C<sub>R</sub>.
4. Find the job number, *nnn*, associated with the current print job.
5. Type DELETE/ENTRY = *nnn* LPA0: C<sub>R</sub>.
6. Turn the power back on for the printer.

#### Z.2.7.2. Hard copy device

The (Modcomp) Varian printer/plotter is used as the hard copy device. Try the instructions in § 11.2. Paper for the Varian and the Versatec is kept in the Modcomp computer room (Room 216) on the floor at the back right. Be sure to get the correct type for the printer you are filling.

#### Z.2.7.3. Booting the CV VAX system

Large VAX systems should not need to be booted manually except under special circumstances. The Charlottesville VAX supports terminals all over the building and remote entry terminals from Ivy Road, Green Bank, and the phone system. Be sure that the VAX is truly dead and consult a staff member before booting. Detailed instructions for booting are posted on the front panel of the VAX (Room 112).

To reboot the VAX, go to the console of the VAX at the back left of the IBM/VAX machine room (Room 112) and type:

CTRL P	to get its attention and the prompt >>>.
>>> HALT C <sub>R</sub>	to bring its activities to an orderly stop.
>>> BOO C <sub>R</sub>	to reboot it.

Eventually (about 3 minutes later, following a torrent of messages about disks being dismounted and mounted, DECNET and other esoterica), the message **Welcome to VAX/VMS** ... (or some simple variant) should appear on the console and on all of the terminals connected to the VAX. The hangups should be cured by this. If they are not, summon a VAX Advisor, or go to bed, depending on the time of day.

#### Z.2.7.4 Other problems on the CV VAX

Sometimes tasks under development or investigation are compiled and linked in a "debug" mode. If they have been left this way, they will wake up (after the GO command) and issue a prompt of DBG>. To run the program, type

DBG> GO CR to start it.

and, after it has finished and issued another prompt:

DBG> EXIT CR to finish it.

Tell the System Manager so that the program may be relinked in normal mode.

#### Z.2.8. Editing RUN files on the CV VAX

RUN files are located in a directory called UMAO:[AIPS.RUN] and must have the three-letter extension .RUN in their names. They are edited from monitor level using either EDT or SOS. For instructions on these editors consult Volume 3 of the VAX/VMS Manuals. An example appropriate to the VAX is given in § 12.6. The EMACS editor is also available and Don Wells can provide an abridged instruction manual.

### Z.3. Charlottesville MODCOMP

This Section contains information specific to the *AIPS* MODCOMP hardware and software in Charlottesville. The computer is a MODCOMP Classic equipped with an I<sup>2</sup>S television display device and a Floating-Point Systems AP-120B array processor. It has three disks and two tape drives. The *AIPS* Caige is located on the second floor Room 215 of the Edgemont Road Office Building. There are four Modcomp terminals: 2 Tektronix 4025 text terminals and two Tektronix 4012 graphics terminals. They are labeled to show their various functions. The primary *AIPS* text terminal is located at the back right (as you enter) next to the television monitor. The 4012 to its right (labeled A02) is used by AIPS1 for its graphics output. The other 4025 is primarily an Operator terminal, but may be used for AIPS2. The other green screen (labeled TY1) is then used for its graphic output. The I<sup>2</sup>S device is equipped with four gray-scale memory planes and four graphics overlay planes. A Varian is used for plotting and printing.

#### Z.3.1. Signing up for *AIPS* time on the MODCOMP

There is a sign-up sheet for *AIPS* on the noticeboard just to the left of the entrance to the *AIPS* Caige, Room 215. If you wish to be certain of the availability of the MODCOMP, it is usually advisable to sign up for your *AIPS* time in advance. To promote fair and efficient use of the system, users are asked to restrict the amount of time that they reserve. Formal rules are not now in effect, but may be imposed should the need arise.

*AIPS* supports up to two simultaneous interactive users, plus batch queues, on the MODCOMP. The user signed up for AIPS1 has priority for the use of the displays, array processor, and tape drives. He is expected to be reasonable about sharing these resources, however. Normally, the facilities available through AIPS2 are *not* sufficient for most users. The MODCOMP simply cannot handle the loads which even one experienced *AIPS* user can readily generate.

#### Z.3.2. Using the Tektronix terminals on the MODCOMP

To correct characters which you have typed, use the BACK SPACE key on the MODCOMP. The RUB OUT key (at the upper right of the keyboard) is interpreted as deleting the entire line.

During execution, paging *cannot* be halted, nor can the program be aborted (by normal means). Fortunately, the Tektronix 4025 terminal has three pages of memory. To scroll upwards press the 1 key on the numeric pad to the right of the main keyboard. The 2 key on the numeric pad will scroll downwards. If a program must be aborted, go to the Operator terminal (4025 labeled TY), press the CTRL key and hit the A key. (This signals your desire to talk to the OC (Operator Communications) program.) Then type:

```
/AIPS1/A Cr                                for example
```

to abort the AIPS1 program.



### **Z.3.3. Logging In to AIPS on the MODCOMP**

Go to the *AIPS* Cage in Room 215 and find a terminal connected to the MODCOMP. The Tektronix 4025, located next to the television monitor, is reserved for the user who has signed up to get *AIPS1*. Typing `CR` on the terminal will produce a prompt which will tell you the current state of that terminal.

If the prompt symbol is `>` the terminal is already being used for *AIPS*. Check with other possible users before proceeding. If it's OK to use the system, type either

`EXIT CR`

or

`QE CR`

depending on whether the previous user wants or does not want, respectively, his message log.

If the prompt is `$` (due to the initial `CR` or to following the step above), the computer is at system level, but not in use. Check nearby users to make sure and then type

`$ JOB CR` to init the job control processor.

`$ AIPS CR` to make an *AIPS* run.

No matter which terminal the above commands were entered upon, they will make *AIPS1* begin on the terminal next to the television monitor or, if *AIPS1* was already running, they will make *AIPS2* begin on the Operator terminal. These commands will also terminate any jobs running on the green screen appropriate to the *AIPS* being started. If someone is using the relevant green screen, follow the instructions in the next paragraph.

On the MODCOMP, it is possible to detach all job control processors from a terminal. Thus, your initial `CR` may produce no reaction at all. In this case, try some other terminal, particularly the Operator terminal. If a `CR` still produces no reaction, try on the Operator terminal:

`CTRL A` to get *OC*'s attention.

`//I CR` to see if the system is working at all.

`/AIPS1/E,,LMV CR` to start *AIPS1*.

If the `CTRL A` produces no response, then the MODCOMP has probably halted. See the booting instructions in § Z.3.7.3.

You should then see the message:

`BEGIN THE ONE TRUE AIPS NUMBER n`

where *n* is 1 or 2. *AIPS* will now ask you for your user number:

`AIPS: ENTER USER ID NUMBER`

`? uuu CR`

where *uuu* is your Charlottesville user number. The *AIPS* prompt `>` should now appear.

#### Z.3.4. Hardware tape mount on the MODCOMP

The MODCOMP has two identical tape drives located in the *AIPS* Cage. They are labeled **MT1** and **MT2**, corresponding to the adverb **INTAPE** values of 1 and 2, respectively. Densities of 800 and 1600 BPI are supported.

To mount a tape on one of these drives, open the door by pulling from the left hand edge. Push in the spot marked **PRESS** in the hub of the upper spool. Remove any protective rims from the tape and slide it over the upper spool. Then engage the tape lock by pressing on a spot opposite that marked **PRESS** in the hub of the upper spool. The tape must be threaded manually. Follow the black lines drawn on the unit around the various capstans and onto the take up spool at the bottom. Wind the tape onto the take up spool for several turns. Then close the door and press the white button marked **LOAD** (2<sup>nd</sup> from the top). The tape should advance to the **BOT** point. Press the **ON LINE** button. The **PE/NRZI** lighted button should be on for 1600 BPI tapes and off for 800 BPI tapes. Press the button if needed to get this state.

When you are done with a tape, press the **RESET** button and then the **REWIND** button. This will cause the tape to rewind to the **BOT** point. Press the **REWIND** button again to dismount the tape.

#### Z.3.5. Software tape mount on the MODCOMP

The MODCOMP is happy to take your word (or anyone else's) that the hardware-mounted tape is yours. Therefore, no software mount is required and MODCOMP users do not have to exit from *AIPS* to mount and dismount tapes. However, there is some danger that another user might accidentally read or write your tape. Therefore, cautious users do not leave their tapes **ON LINE** longer than they have to and do not insert write rings unless they are about to write on their tapes.

#### Z.3.6. Monitoring disk space on the MODCOMP

See §4.4 for a general discussion of the problem of disk space. The verb **TIMDEST** is useful, but we prefer that users ask the System Manager before invoking it. On the MODCOMP, the *AIPS* catalog files are "public" — all users' images are cataloged in the same file. Type:

> <b>USER = 3200</b> <i>C<sub>R</sub></i>	to see all users.
> <b>INDISK = 0</b> <i>C<sub>R</sub></i>	to see all disks.
> <b>MCAT</b> <i>C<sub>R</sub></i>	to list all map files.
> <b>UCAT</b> <i>C<sub>R</sub></i>	to list all uv files.

The users who have large numbers of files, particularly uv files, are the users on whom you wish to apply pressure in order to obtain free disk space. Typing:

> **DISK** *C<sub>R</sub>*

provides more exact and detailed information, but takes a very long time to run.

Since disk space is often very tight, there is, even on the Charlottesville MODCOMP, a utility to assist the user to monitor how much space is available. This capability is available only at monitor (system job control) level. To use it:

> STORE 1 CR	to save your AIPS inputs.
> EXIT CR	to exit from AIPS.
\$ JOB CR	to init the job control processor.
\$ ASS PL XQL 6 LO CR	to make assignments.
\$ EXEC SPACE,PL CR	to run the utility.
\$ AIPS CR	to restart your AIPS.
? uuu CR	to enter your user number.
> RESTORE 1 CR	to recover your AIPS inputs.

### **Z.3.7. Solving problems on the CV MODCOMP**

Below are the details specific to the Charlottesville MODCOMP system for handling some of the problems which may arise in *AIPS*.

#### **Z.3.7.1. Stopping excess printout on the MODCOMP**

If you wish to abort a job which is currently printing out do the following:

1. Turn the power off on the printer to conserve paper.
2. Make sure that the *AIPS* task doing the printing has stopped or **ABORT** it.  
(See §11.1.)
3. Go to the Operator terminal and get OC's attention with a **CTRL A**. Type **/S/A CR** to stop the print out.
4. Turn the power back on for the printer.

#### **Z.3.7.2. Hard copy device**

The Modcomp Varian printer/plotter is used as the hard copy device. Try the instructions in §11.2. Paper for the Varian and the Versatec is kept in the Modcomp computer room (Room 216) on the floor at the back right. Be sure to get the correct type for the printer you are filling.

### Z.3.7.3. Booting the MODCOMP system

The MODCOMP can get hung up fairly easily. The easiest cure, if all (any) current users agree, is to boot the system. Go to the MODCOMP machine room (Room 216). The CPU is at your left as you enter the door — it has flashing lights (like a *real* computer). Turn the RUN / HALT toggle switch to the HALT (upper) side. Make sure that the numbered switches are all in the horizontal position except switch 15 which should be up. Then, depress the lower side of the toggle switches labeled **M CLEAR**, **FILL**, and **RUN** in that order leaving the last down in the **RUN** position. The lights should do a lot of blinking. After a couple of minutes, you will be asked on the Operator terminal to provide the date and time. Please do these accurately using a 24-hour clock. The system should then provide the \$ prompt on the Operator terminal.

### Z.3.7.4 Other problems on the MODCOMP

The MODCOMP has a wide bag of tricks to play on experienced and inexperienced users alike. We will add more hints in this Section as they arise.

Inside the MODCOMP operating system lives a gremlin called TMP. For cognoscenti, TMP can perform wondrous tricks. However, TMP can take control of a terminal without warning (thinking some static electricity discharge called him from his bottle). When a terminal appears dead or TKPL asserts that it can't access the green screen now, one thing to try is to type, on the Operator terminal:

CTRL A	to get OC's attention.
/TMP/A CR	to abort TMP.

This incantation may resurrect your terminal.

The MODCOMP reacts to devices which are not ready by issuing an error message to the Operator terminal and placing the device in a software "Off Line" status. To bring the device back on line, first mount the required tape, add the needed paper, or whatever. Then turn the power on and type on the Operator terminal:

CTRL A	to get OC's attention.
/ON/LPP CR	to turn on the line printer.
/ON/MT1 CR	to turn on tape drive 1.
/ON/MT2 CR	to turn on tape drive 2.
/ON/SP CR	to turn on the plotter part of the Varian.

The MODCOMP normally gives AIPS high priority, AIPS tasks medium priority, and other users low priority. Such an arrangement is not always needed. To change the priority of any program, type on the Operator terminal:

CTRL A	to get OC's attention.
/pgmname/CHA pn CR	where <i>pgmname</i> is the full name of the program to be changed and <i>pn</i> is the priority it is to assume.

The normal values of *pn* are 128 (high), 200 (medium), and 255 (low). Since TPOWER users don't require much CPU time or real memory, it is friendly to change them to 128 or so. Since the MODCOMP is not a virtual memory machine, it can get clogged with too many equal priority tasks all rolling in and out and competing for the available resources. Changing AIPS1 to low priority often helps and, in bad situations, changing some of the tasks to low priority may also be needed.

There is a useful tape copy routine on the MODCOMP. Get to monitor level (prompt \$) and type:

\$ JOB CR	to init the job control processor.
\$ ASS PL DQL CR	
\$ ASS BI MT1 CR	the input tape drive name.
\$ ASS BO MT2 CR	the output tape drive name.
\$ EXEC TAPTAP,PL CR	to copy the first file.
\$ WEOF BO CR	to write an EOF (if desired) on the output tape.

This may be repeated for as many files as needed. Note that the output files are concatenated if you do not type the WEOF BO CR command.

### Z.3.8. Editing RUN files on the MODCOMP

RUN files are located in a special directory on the MODCOMP and may have any name of  $\leq 8$  characters. They are edited from monitor level using SEDIT. For instructions on this editor consult the MAX II/III/IV System Processors Manual. To get into SEDIT in the appropriate directory, type on any terminal at monitor level:

\$ JOB CR	to init the job control processor.
\$ RUN CR	to make appropriate assignments and start SEDIT.
A session in which you create a new file might go as:	
RAN REC CR	to go to random mode.
D 1 1000 CR	to clear work area.
A CR	to start typing new lines.
<i>any lines you want of POPS text</i>	
.CAT filename CR	to catalog the new RUN file.
E CR	to exit from SEDIT.
\$ JOB CR	to init the job control processor again.
\$ AIPS CR	to resume AIPS.

