NATIONAL RADIO ASTRONOMY OBSERVATORY
Arizona Operations

CONDAR
CONTINUUM ANALYSIS SYSTEM
for the
NRAO SINGLE DISH TELESCOPES

May, 1984

FOREWORD

This manual describes the single dish continuum data reduction program currently running in Charlottesville and Green Bank known as CONDAR and in Tucson known as CONDR1 and CONDR2. This analysis program is based on the POPS language originally written by T. R. Cram and J. A. Hudson. Although the programs at each site are not exactly the same, they appear and function alike to the user. Verbs unique to each site are flagged accordingly in the text.

The object of the analysis program is to furnish the observer with a complete data reduction package. Standard reduction procedures are built in to the program for the convenience of the observer. In Tucson, additional popular procedures exist as text files outside the program that can be INSTALLed for use. Also, tools are provided for the observer to create his own procedures to tailor the data reduction to his own needs.

The NRAO single dish programmers have combined the best of the NRAO single dish continuum programs in one program for all sites. This has proved to be an effective use of manpower as well as a convenience for the observer. The programs grow as we (the programmers) think of new ideas or improvements but also as you (the observer) give us "constructive" criticism of the system. Our doors are always open for your comments.

Many thanks to Werner Scharlach for proofing this manual and producing its index.

Lorrie Morgan Harry Payne Chris Salter Betty Stobie Bob Vance

Table of Contents

POPS: The Interactive Base

POPS (which stands for "People-Oriented Parsing Service") is an interactive, interpretive compiler with various sets of built-in functions that may be extended by the programmer before the program is put into operation, and by the user, afterwards. In addition to supporting continuum data reduction, POPS is also used to support spectral line and off-line VLA data reduction at most NRAO sites.

On-site single dish continuum data reduction program (POPS/CONDAR) consists of an interactive base program and a set of continuum application routines. Although the compiler portion of the continuum data reduction program is POPS, the program will hereafter be referred to as CONDAR, its program name.

When the user types "RUN $CONDAR"[1], the operating system prints a "prompt" character on the CRT screen of the interactive graphics terminal after CONDAR has finished loading and initializing memory. The user may enter a command by typing it on the keyboard (it will be echoed on the CRT screen). The typed message is completed by striking the RETURN key on the keyboard. Once this key has been struck, CONDAR examines the typed line for errors. If an error is found, a message is printed and a new prompt character is printed. If no errors are found, the line is processed. How the line is processed depends on the resident CONDAR "mode of operation".

[1] In Tucson, there are two identical continuum analysis programs, CONDR1 and CONDR2. The user types "RUN CONDR1" or "RUN CONDR2". No "$" is needed in Tucson. Also at the 12-meter telescope, the user must wait for a second prompt before conversing with the program.

Modes of Operation

CONDAR has three basic modes of operation: EXECUTE, INPUT and COMPILE. These modes are distinguished by how a line of input is treated and by different prompt characters.

The resident mode is usually the EXECUTE mode. The program enters this mode when loaded. Entry to each of the other modes is from this mode. Each of the other modes exits to this mode, and an error in a statement in any mode returns the program to this mode.

EXECUTE mode prompts with a caret (>) and as its name implies, immediately executes the commands entered by the user. A session with CONDAR in EXECUTE mode might look like this:

```
>PRINT 'RADIO ASTRONOMY'
 RADIO ASTRONOMY
>PRINT SQRT (2)
 1.4142
```

INPUT mode is a special purpose mode. When in this mode CONDAR prompts with a number sign (#). This mode is entered by means of the READ command, which has the form READ list , list is a list of variable names. Once this command is entered, CONDAR prompts with the number sign until the user has input enough values to satisfy the list. For example, a session might look like this:

```
>READ BMARK
#1
```

(BMARK is a single-valued application variable.) In INPUT mode, input is treated as a value to be assigned to a variable. Note that once the argument list is satisfied, CONDAR automatically returns to EXECUTE

mode. The reader is referred to the Verb and Operator Dictionary entry for READ.

COMPILE mode is really two modes -- DEFINE and EDIT. Both of these use the colon (:) as the prompt character. DEFINE mode is entered by means of the PROCEDURE statement and is used to create a new user-defined operator. The word PROCEDURE must be followed by an unused name, which becomes the new operator. The program responds to the PROCEDURE statement by prompting with a colon. All statements entered by the user after a colon, constitute the definition of the new operator. The definition is terminated by the command FINISH (entered immediately after a colon) which also returns the program to EXECUTE mode. An additional flexibility of procedures is that their definitions may include logical constructions which cannot be used in EXECUTE mode. These constructions are the FOR, IF THEN ELSE, and WHILE constructions. A session with CONDAR using the DEFINE mode might look like:

```
>PROCEDURE EXAMPLE
:FOR I = 1 TO 10
:    PRINT I, SQRT(I)
:    END
:FINISH
```

Note that END completes the FOR construction; FINISH completes the procedure definition. Input entered in the DEFINE mode is treated as part of a definition of a new operator. Note that this code is not executed until the user returns to EXECUTE mode and invokes the name of the procedure he has defined. The reader is referred to the Verb and Operator Dictionary entries for PROCEDURE, FOR, IF and WHILE.

EDIT mode is similiar in some respects to DEFINE mode. Statements entered in EDIT mode are treated as part of the definition of a procedure--however, in this case the procedure already exists. EDIT

mode is used to change the definition of a user-defined operator. A session using EDIT mode might look like:

```
>EDIT EXAMPLE 3
:     PRINT SQRT(I), I, I*I
:ENDEDIT
```

Note that 3 is the line in the procedure EXAMPLE that is to be changed. The first statement entered will replace statement 3; if further statements were entered, they would be inserted between the new statement 3 and 4. ENDEDIT signals the end of the corrections at statement 3 and returns the program to EXECUTE mode. Note also that the editing is of entire lines. To change anything in a line, you must replace the entire line. To insert a new line in the above procedure between lines 3 and 4, the user types EDIT EXAMPLE 3.5, the line to insert and ENDEDIT. The reader is referred to the Verb and Operator Dictionary entry for EDIT.

In Green Bank and in Charlottesville only, there is a second edit word, MODIFY. With MODIFY, the user can selectively edit within a procedure line.

Operators, Variables, Constants and Expressions

CONDAR recognizes two basic types of symbols. These are operators and variables. Operators stand for an operation. The program decodes these as instructions to do something. Operators may be arithmetic (+ - * / ** SQRT), inclusion (), logical (= < > ¬ | &), pseudo (IF THEN ELSE FOR WHILE END), or symbolic (PRINT PROCEDURE EDIT LIST STORE).

Symbolic operators may also be application operators, or user-defined operators. User-defined operators are defined in terms of CONDAR and application operators.

Since an operator is a command to do something, various operators have restrictions on the context or format in which they may be properly used. For example, "=" doesn't mean much without something on both sides of it, and PROCEDURE needs an "Operand", which is a procedure name. Some errors of these types are checked for by CONDAR, but the user should understand how an operator is properly used.

A variable is an entity that has a value. Variables, like symbolic operators, may also be CONDAR, application, or user-defined. User variables may be defined only in COMPILE mode. Variables may be either scalars or arrays.

Variables are maintained as single-precision floating point numbers (32 bits). When needed for an array index, the floating point number is rounded down to the next integer value.

Names of variables may be up to 10 alphanumeric characters. The leading character must not be numeric. No imbedded blanks are permitted. No imbedded non-alphanumeric characters are permitted

(A*B will be three symbols.) These rules also apply to procedure names (user-defined operators). All names, whether operator or variable, should be distinct. CONDAR or application names cannot be redefined by the user.

User scalars are defined by a statement of the form A=B, appearing in a procedure. B may be an already defined symbol or an expression consisting of constants and already defined variables and operators. User arrays are defined by a statement of the form ARRAY A(N), appearing in a procedure. The reader is referred to the Dictionary entry for ARRAY for further information.

CONDAR recognizes two types of constants; decimal numbers written with or without a decimal and fraction, and Hollerith strings, enclosed in single quotes. Decimal constants may appear in expressions, and as operands of operators. Hollerith constants are used as operands of the verb PRINT.

CONDAR expressions are combinations of a few operators: symbolic, arithmetic, logical and inclusion and variables of any type.

CONDAR expressions are of two kinds: arithmetic and logical. Arithmetic expressions make use of the operators:

= + - * / **

(shown in order of ascending precedence; those of equal precedence are grouped together). Function calls and subscript evaluation are of the highest precedence (and of the same precedence). The logical operators are:

& %] < = >

(again grouped according to precedence where & is AND, % is OR and] is

NOT. The operations carried out by the = symbol varies with the context in which it is found. It is used as a "store" operator when it appears the first time in any executable CONDAR statement, except when that statement begins with the 'IF' keyword. The second instance of = in an arithmetic expression, or any instance within an IF statement is taken to be a comparison operation. The group, < = >, is on the same precedence level as the = (store) operation.

The same rules of precedence apply as in common mathematical notation wherever symbols of inclusion ( ) are encountered.

Operands of arithmetic or logical operators may be real scalars, constant or variable, and also elements of arrays. Arithmetic operations on Hollerith constants or entire (unsubscripted) arrays is not defined.

The special values "TRUE" and "FALSE" are defined in the language for assigning Boolean states to variables. (Their values are +1.0 and -1.0, respectively.) Any variable may be used as a logical variable; no special declaration is required.

Examples of arithmetic and logical expressions:

```
Y(I)+3*X+ALPHA(M,N)
LOGICAL=(P>Q1BETA(2*J))
IF 3*X<40 THEN . . .
```

Generally, an expression may be substituted for a variable or constant in CONDAR statements; the exceptions to this rule are PROCEDURE and ARRAY declarations, FINISH, EDIT, LIST, MODIFY and ENDEDIT. Formats for these statements are necessarily rigid.

Communication with CONDAR

As mentioned in the introduction to CONDAR, commands are entered to the program by typing a message immediately after a prompt character and striking the RETURN key. A message must consist of recognizable symbols to be accepted. (A message containing an unrecognizable symbol will generate the message SYMBOL?, but no other action.) In Tucson, a message may be corrected by backspacing over the error and retyping the message beginning with the last character backspaced over. To backspace strike the RUB OUT key. The response will be a \ (back slash) for each time the key is struck. To see the line retyped with the corrections, strike the CTRL and R keys simultaneously. A message may also be cancelled by striking the CTRL and U keys simultaneously. In Green Bank and Charlottesville, a message may be corrected by backspacing over the error with the BACKSPACE key and retyping the message. To cancel a message, hit the RUBOUT key.

CONDAR is a linear communicator. It accepts one message at a time. Each time the RETURN key is hit, the input is examined and acted upon. While the program is processing the message, it cannot accept another message. Another message may be entered when the program prints another prompt character. A message is processed in the exact order given. SHOW BASELINE is not equivalent to BASELINE SHOW.

Once "RUN CONDAR" is typed from the TEKTRONIX, CONDAR is loaded and running ("ready and waiting"). The program will remain ready and waiting until a command is input. Once the command has been completed,

CONDAR returns to the ready and waiting state. No "initialization" is needed to get CONDAR's attention, unless the CRT screen is blank, in which case it is a good idea to hit the return key. If the program is loaded and running, it will print a caret (>) on the screen.

A message (one line of input begun after a prompt character and terminated by striking the RETURN key) to the program may contain two basic types of statements. Assignment statements are of the form A=EXPRESSION. Examples are BMARK=1 and ZETA=ALPHA*BETA**2. Command statements are of the form OPERATOR operand(s) ; the operator may or may not require one or more operands. Examples are RESTART, LIST EXAMPLE and EDIT EXAMPLE 3.

In general, each statement (or complete thought) of either type must be entered entirely on one line. (The exceptions to this rule are the logical constructions.)

More than one statement may be entered on one line if the following rules are followed:

(a) An assignment statement must be separated from any other statement by a semicolon (;).
(b) A command statement whose operator is listed in the operator synopsis must be separated from any other statement by a semicolon.
(c) The following operators should always appear immediately after a prompt character: PROCEDURE, EDIT, ENDEDIT, RESTART, STORE and RESTORE.

Examples are:

```
BMARK=1; PRINT BMARK
BMARK=1; ZLINE=0
PROCEDURE EG(A); PRINT SQRT(A); FINISH
```

We note here that application operators (verbs) need not be separated by semicolons, but user-defined operators should be treated as CONDAR operators.

Operator Synopsis

INFORMATION EXCHANGE

*	Begins a comment statement.
EXPLAIN	Lists documentation on resident CONDAR procedures (Tuc).
HELP	Lists the CONDAR and application operators, and/or their descriptions. HELP SCALAR and HELP ARRAY list all scalars and arrays, respectively. HELP PROC lists all procedures defined within the program.
PRINT field	Prints the requested field on the CRT. Field may be a literal, or an expression or adverb whose value is then printed.
READ field	Reads the values of the adverbs or variables named from the CRT.
? field	An alias for PRINT.

MEMORY CONTROL

RESTART	Returns the program memory to its original, or default, condition.
RESTORE page#	Copies the indicated disk page into the program memory space.
STORE page#	Stores a copy of the current state of the program memory in the indicated disk page save area.

PROCEDURE

DEFINITION

FINISH	Terminates the definition of a procedure.
INSTALL	Causes the designated procedure to be loaded into CONDAR from disk (Tuc).
PROCEDURE procname	Initiates the definition of a procedure with the given name.

RETURN	Is required in procedures called by other procedures.
SCRATCH procname	Deletes the specified procedure from the program, but does not return that procedure space.

LISTING

HELP PROCEDURE	Lists the names of the defined procedures.
LIST procname	Lists the entire named procedure.

EDITING

EDIT procname line#	Initiates editing of the named procedure at the specified line.
ENDEDIT	Terminates editing of a procedure.
MODIFY procname line#	Modifies the specified line of the named procedure (GB).

LOGICAL CONSTRUCTIONS

FOR x = a TO B BY c	Begins a logical construction like the PL/I interactive DO loop.
IF (test condition)	Begins a logical construction similiar to the PL/I IF-THEN-ELSE.
WHILE (test condition)	Begins a logical construction similiar to the PL/I DO WHILE loop.
END	Terminates the logical constructions begun by FOR, IF and WHILE.

SPACE ALLOCATION

ARRAY arrayname( )	Allocates the requested space for an array with the specified name.
CORE	Lists the beginning and last used locations of the procedure and variable definition spaces.

Applications

The continuum data reduction package is designed to manipulate scans. It therefore provides the user with various operators which do things to scans. These operators are the continuum verbs, each of which is associated with a distinct data reduction routine. The routines perform four types of operations:

(1) movement of scans,
(2) modification of data values,
(3) calculation and
(4) display of information or data values.

Some verbs perform only one of these functions; others combine two or three.

A scan consists of a set of data values and the identifying information associated with those values. It is this entity, or parts of it, which is manipulated by most of the continuum routines. In the continuum programs, each receiver channel is written to disk as a separate scan. Since there are differences in the recording of data between Green Bank and Tucson, each will be described separately.

In Green Bank, the maximum number of receiver channels is eight. Each receiver is written as a single scan with its scan number equal to the original scan number times ten plus the channel number. To retrieve the second receiver of scan 602, the user types GET 6022. Continuum and spectral line data are stored in separate files so the user cannot inadvertently retrieve spectral line data. Some adverbs are dimensioned by 8, the number of receiver channels. Although each receiver channel

is stored as a separate scan and is processed independently, these adverbs store separate values for each receiver channel.

In Tucson, there is a maximum of two receiver channels. If two channel continuum is being observed and scan 500 is the first receiver channel, then scan 501 is the second receiver channel. Both continuum and spectral line scans are stored in the same data file. If the user retrieves a spectral line scan in CONDAR, he will get the error message "NOT CONT DATA". Receiver dependent adverbs are dimensioned by 2, one value for each receiver channel.

Basic Algorithm/Program Structure

The program structure, like the continuum language (discussed below), reflects its function as a manipulator of scans. In order for a scan to be displayed, used or modified, it must first be copied from the disk, where it was stored by the on-line link program, to the CONDAR program memory. This memory contains three locations where scans may be kept. These three locations are the WORK, TEMP and HOLD arrays.

Once a scan has been put into one of these three arrays, it can be displayed, changed, moved to another array, or stored in one of the save bins on disk. These subsequent operations are influenced by the array pointer, PTWH. This pointer always has the value 1, 2 or 3, indicating that the "currently referenced array" is the TEMP, WORK, or HOLD array. Since most verbs are defined as operating on the scan in the currently referenced array, it is sometimes necessary and often useful to know which array is current. However, for many simple data reduction schemes, the user will find that the pointer takes care of itself. Briefly, those operations which move data change the pointer to the new location of the data. Most other operations use, but do not change, the pointer.

The user who is designing a data reduction scheme will be most successful if he keeps track of the following: the contents of each program array, the value of the array pointer and the definitions of the verbs. Careful attention to these points will prevent accidental destruction of much-reduced data and enable the user to utilize the program most effectively.

Verbs, Objects, and Adverbs

The continuum data reduction package has a language of its own. This language is very similiar to the CONDAR command language; most of what has been said about the latter applies to the former. We add here additional qualifying remarks.

Verbs are operators; when encountered by the program, the action which is the definition of the verb is performed. Verbs may be said to be of two kinds: those that require an object, and those which have implied objects. Examples of verbs of the first type are

GET scan#

TELL DISK

Each of these verbs requires an object which further specifies the action the verb is to perform. In the first case, scan# specifies which scan is to be copied. In the second case, DISK specifies that a listing of scans in the disk index is desired (the only other permitted objects of TELL are CSTACK, KSCANS, DATA and GPARMS).

Most verbs are of the second type. Examples are

BASELINE

MOMENT

TRH

ACCUM

The definition of verbs of the second type specify upon what the verbs act. BASELINE calculates a baseline for and removes it from the scan in the "currently referenced array" (see "Basic Algorithm/Program Structure"). MOMENT calculates a moment for the scan in the currently

referenced array. TRH copies the scan in the currently referenced array into the HOLD array. ACCUM adds the scan in the WORK array to the HOLD array. To use the verbs of the second type successfully, the user should understand what are the implied objects of the verbs. This information is given in the Verb Dictionary. For verbs of the first type, the Verb Dictionary entries describe the required object.

An adverb is simply a variable that is assigned a value which is used by a verb. Examples are the adverbs ZLINE and NBOX. If ZLINE = 1, the verb SHOW will draw a horizontal line at zero Kelvins. BOXCAR smoothes the data by averaging NBOX data points together.

The values of adverbs are set by the program when it is restarted. However, many are set to zero, and the user must specify a meaningful value before using the verb which uses that adverb. New adverb values may be specified at any time and remain the same until the program is restarted or the user changes them again.

Many adverbs are vectors. Examples are CENTER, CLIPMIN and NREGION. CENTER has six values, one for each of up to six gaussian components. CLIPMIN has two values in Tucson and eight in Green Bank, one for each channel. NREGION is doubly dimensioned; it has eight values for each receiver channel.

A single valued adverb is set by a statement of the form:

ADVERB=value

Similarly, all values of a multi-valued adverb may be set equal by a statement of the same form. To set the values of a multi-valued adverb separately, use statements of the form

ADVERB(1)=value1

ADVERB(2)=value2

and so on. Also for convenience, a multi-valued adverb may be set in the following manner:

ADVERB=value,value2,value3,value4,. . . etc.

All subsequent values of the adverb array will be set to zero.

CONDAR also uses a few variables which are not strictly adverbs, such as SIZE and VRMS; they are used by the verbs MOMENT and RMS to store the results of calculations. They are most often used as operands of the operator PRINT.

Verb Synopsis

BASELINE COMPUTATION AND REMOVAL

BASELINE	Computes and subtracts a polynomial baseline from the currently referenced array.
BMODEL	Constructs a model of the previously computed polynomial baseline in the currently referenced array.
BSHAPE	Computes a polynomial baseline for the currently referenced array.
DCBASE	Computes and subtracts an average data value from the currently referenced array.
MDBASE	Performs a point by point baselining in which the median intensity for the NBOX points centered on a position is subtracted from the value at that position (GB).
PCBASE	Computes and subtracts a constant value from the data in the currently referenced array such that the specified percentage of points are negative.

DATA TRANSFER

ARRAY-ARRAY

TRH	Copies the contents of the currently referenced array into the HOLD array. (Transfer to HOLD array.)
TRT	Copies the contents of the currently referenced array into the TEMP array. (Transfer to TEMP array.)
TRW	Copies the contents of the currently referenced array into the WORK array (Transfer to WORK array.)

ARRAY-DISK

SAVE	Copies the currently referenced array into a save bin on the disk.
KEEP	Copies the currently referenced array into the KEEP file to be transferred to tape later.

DISK-ARRAY

CGET #	Retrieves the specified receiver (feed) of the most recently completed scan.
FETCH scan#	Retrieves the specified scan number into the WORK array and retrieves scan number +1 into the TEMP array (GB).
GET scan#	Copies the indicated scan to the WORK array.
OFF scan#	Copies the indicated scan into the TEMP array.
ON scan#	Copies the indicated scan into the WORK array.
RECALL	Copies the contents of a disk save bin into the currently referenced array.

TAPE-DISK

LOAD	Reads scans from 9-track KEEP tape and writes them to the raw data file for processing by CONDAR (GB).

DISPLAY OF DATA

AUTO	Alias for FREEY.
AZ	Sets the y-coordinate of maps to azimuth and the x-coordinate to elevation (Tuc).
CCUR	Returns the point value at the current x crosshair position.

CROSSHAIR	Prints the point number and temperature at the current crosshair positions.
DEC	Sets the y-coordinate of maps to declination and x-coordinate to right ascension (Tuc).
EL	Sets the y-coordinate of maps to elevation and the x-coordinate to azimuth (Tuc).
FIX	Alias for HOLDY.
FREEY	Returns determination of the y-scaling to SHOW.
FULLGRID	Draws a rectangular grid on the screen.
HISTOGRAM	Sets the SHOW display mode to histogram.
HOLDY	Fixes the scaling used by SHOW.
LABEL	Labels a contour plot.
LIMIT (#,#)	Limits the SHOW display to only those points between the minimum and maximum values specified.
LINE	Sets the SHOW display mode to a continuous line.
MAKECOPY	Causes a hardcopy to be made of the CRT screen.
MAPSHOW	Creates a position vs. position contour map of a series of scans.
ONS	Flag all "on" samples with a + (Tuc).
PAGE	Erases the CRT.
POINTS	Sets the SHOW display mode to points.
RA	Sets the y-coordinate of maps to right ascension and the x-coordinate to declination (Tuc).
RANGE(#,#,)	Sets the y scaling for SHOW according to MIN and MAX values entered with it.
RESHOW	Graphs the currently referenced array on the CRT with no labels or borders, using the scaling last used by SHOW.

RHIST	Sets the RESHOW display mode to histogram.
RLINE	Sets the RESHOW display mode to a continuous line.
RPOINTS	Sets the RESHOW display mode to points.
SHOW	Graphs the currently referenced array on CRT, with labels and borders.
TCUR	Returns the temperature value at the current y-crosshair position.

DISK INDEX

SUMMARY	Produces a listing of the scans on disk belonging to the specified user (GB).
TELL DISK	Lists the scans currently in the disk index.
TELL DATA	Lists all continuum scans on disk in Tucson.
WIPE	Lists all numbers of users who have data in raw data file and prompts observer for the user number of the observer whose data is is to be wiped from disk (GB).

DOCUMENTATION

DOC	Lists TEMP, RMS, and S/N of scan in the currently referenced array (Tuc).
HEADER	Lists the header information for the scan in the currently referenced array on the CRT.
PDOC	Writes documentation information about a Five-point on the CRT and the results of the fit. Is used by the F procedure (Tuc).
SNAP	Lists the header information for the scan in the currently referenced array on the printer.

TABLE	Lists the data values for the scan in the currently referenced array on the CRT.
TITLE	Prints one line of header documentation on the CRT (Tuc).

EDITING

CLIP	Limits the data values of the scan in the currently referenced array to within the range CLIPMIN to CLIPMAX.
GAINCRV	Scales the data points of the scan in the currently referenced array, taking into account the changes of telescope gain with Declination (GB) or Elevation (Tuc).
INVERT	Flips the scan end for end in the currently referenced array.
SEDIT#	Eliminates the specified pair from the on-off scan in the currently referenced array and recomputes the MEAN and RMS (Tuc).
SEDITS (#,#)	Eliminates the specified pairs from the scan in the currently referenced array and recomputes the MEAN and RMS (Tuc).

GAUSSIAN FITTING

GAUSS	Fits the requested number of gaussians (1 to 6) to the data in the currently referenced array.
GMODEL	Constructs a model of the (sum of the) gaussian(s) fit by GAUSS in the currently referenced array.
GPARTS	Evaluates the parameters and separately displays NGAUSS components of a gaussian.
PEAK	Finds the CENTER, HWIDTH, and HEIGHT of a single gaussian.

REGAUSS	Fits the requested number of gaussians without resetting the first guess of the HEIGHTS to the value at the CENTERS.
RESIDUAL	Subtracts the gaussian(s) fit by GAUSS from the data in the currently referenced array.

MATHEMATICAL

BASIC FUNCTIONS

AVG	Computes the mean temperature of the on-off pairs and the RMS of the mean of the scan in the currently referenced array. Prints out the results on the CRT (Tuc).
BIAS	Adds a constant to the scan in the currently referenced array.
DIFF	Subtracts the scan in the TEMP array from the scan in the currently referenced array.
DIVIDE	Divides the scan in the currently referenced array by the scan in the TEMP array.
EQTOGAL	Converts equatorial coordinates (1950 RA and Dec) into galactic coordinates and prints the results on the Tektronix screen.
GALTOEQ	Converts galactic coordinates into equatorial coordinates and prints the results on the Tektronix screen.
PB	Computes the MEAN and RMS of the scan in the currently referenced array. Prints out the results on the CRT (Tuc).
FOFFSET	Corrects the positions stored in the header of the scan in the currently referenced array for the offset position of the receiver feed relative to the indicated positions of receiver box center (GB).

SCALE	Multiplies the scan in the currently referenced array by the constant, FACT.

SPECIAL FUNCTIONS

FFT	Replaces the data in the currently referenced array with its Fast Fourier Transform (GB).
MOMENT	Computes the moment requested by the value of the adverb NMOMENT over the points specified by BMOMENT and EMOMENT of the scan in the currently referenced array.
PVLS	Computes the pointing corrections needed by the control computer to point the telescope accurately (GB).
RMS	Computes the root-mean-squared noise over the regions specified for the scan in the currently referenced array.
SOLVETIP	Computes a non-linear least squares fit to the data of an extinction scan.

POINTER CHANGE

PMH	Sets the array pointer to the HOLD array. Points to HOLD array.
PMT	Sets the array pointer to the TEMP array. Points to TEMP array.
PMW	Sets the array pointer to the WORK array. Points to WORK array.

SMOOTHING

BOXCAR	Smoothes the scan in the currently referenced array by averaging NBOX points together and placing the result in the central point.

HANNING	Smoothes the scan in the currently referenced array by averaging three points together, giving the central point twice the weight of either flanking point.
SMOOTH	Smoothes the scan in the currently referenced array with the SMWGT weighting function.

STACKING

A #	Inserts a single scan into the STACK.
ACCUM	Adds the scan in the WORK array to the contents of the HOLD array with no weighting.
ADD (#,#)	Inserts the specified scans into the STACK.
AVE	Divides the HOLD array by the number of scans that have been accumulated.
DELETE #	Deletes the specified scan from the STACK.
EMPTY	Empties the STACK.
SCLEAR	Clears the accumulator (HOLD) array before stacking data.
STACK	Lists all the scans entered in the STACK.
TELL CSTACK	Lists the number of scans accumulated in the HOLD array, as well as the scan numbers of the first 17 and the last scan that were accumulated.

TWO-CHANNEL (Tucson)

ONE	Selects the increment between scan numbers processed to one. (For single channel continuum).
TWO	Selects the increment between scan numbers processed to two. (For dual channel continuum).

Adverb Synopsis

ACOUNT	The STACK counter.
ASTACK(100)	The 100 element STACK array.
BBASE(CH)	The number of data points at the left side of the data which will be used to compute a baseline or RMS.
BDROP(CH)	The number of data points at the left side of the data which will not be used.
BEAMHW	The half-width (in arcmin) of the double gaussian model of the beam-switched response constructed by BMSWITCH (GB).
BEAMSEP	The separation of the two feeds (in arcmin) used in constructing the beam-switched model (GB).
BGAUSS	The point at which GAUSS is to begin to fit.
BMARK	If >=1, SHOW will draw boxes to indicate the regions last used to compute a baseline or RMS.
BMOMENT (CH)	The data point at which MOMENT will begin to compute.
BRANGE	Minimum 1950 Right Ascension (in hours) of scans to be loaded from a KEEP tape (GB).
BSCAN	Minimum scan number of scans to be loaded from a KEEP tape (GB).
CENTER(6)	The points at which up to six gaussians have their centers.
CLIPMAX(CH)	The maximum data value desired.
CLIPMIN(CH)	The minimum data value desired.
CMARK(6)	The data points at which up to six vertical lines are to be drawn by SHOW.
DCPCT	The percentage of data points to be made negative in the verb PCBASE.

EBASE(CH)	The number of data points at the right end of the data which are to be used to compute a baseline or RMS.
EDROP(CH)	The number of data points at the right end of the data which are not to be used.
EGAUSS	The point at which GAUSS is to stop fitting.
EMOMENT(CH)	The data point at which MOMENT is to stop computing.
ERANGE	Maximum 1950 Right Ascenscion (in hours) of scans to be loaded from a KEEP tape (GB).
ESCAN	Maximum scan number of scans to be loaded from a KEEP tape (GB).
ETA	The coupling efficiency of the telescope. Used by the SOLVETIP verb for a total power data (Tuc).
FACT(CH)	A number of Kelvins to be added to a scan, or a numerical factor to be multiplied with a scan.
FIXC	If <0, the values of CENTER will not be iterated by GAUSS.
FIXHW	If<0, the values of HWIDTH will not be iterated by GAUSS.
FTM	The fraction, Mean Temperature of Atmosphere/Ambient Temperature. Used by SOLVETIP to compute Tm for total power data (Tuc).
FTSBR	The fraction, Spillover, Background and Loss Temperature/Ambient Temperature. Used by EXT to compute Tsbr.
GCRV (12)	Table of gain calibration values spaced at equal values of Declination (GB) or Elevation (Tuc).
GGRID	The spacing between adjacent entries in the GCRV table in decimal degrees.
GMIN	The minimum position of the scale factors in the gain table in decimal degrees.
HEIGHT(6)	The height, in Kelvins, of up to six gaussians.
HWIDTH(6)	The half-width, in number of points, of up to six gaussians.

HP	Half-power beamwidth of telescope. Used by PROCEDURE F (Tuc).
LEVS(20)	Up to twenty levels at which contours may be drawn by MAPSHOW.
MRATIO	The ratio between the horizontal and vertical scaling of a contour map.
NBOX	The number of data points which are to be averaged together by BOXCAR.
NFIT	The order of a polynomial which is to be fit by BASELINE or BSHAPE.
NGAUSS	The number of gaussians which are to be fit by GAUSS.
NITER	The number of iterations which GAUSS will execute trying to compute a fit, before giving up.
NMOMENT	Denotes the type of moment to be calculated. Currently only the zero order (area) is supported.
NREGION(CH,8)	The data points, in sequential order, at which up to four regions per channel are to begin and end.
NPOINTS	The number of data points of the scan in the currently referenced array. Set by verb SLENGTH.
NSAVE	Names a disk save bin.
OFMAP	If<1, denotes a map of ON-OFF scans to MAPSHOW and LABEL (Tuc).
SHIFT	The offset between scans used by the SHOW1 and SHOW2 procedures.
SINCR	The increment between scans. SINCR=1 for single channel continuum and SINCR=2 for dual channel continuum in Tucson. SINCR=10 in GB.
SLABEL	A flag to determine where SHOW writes certain documentation information on the display.
SMWGT(12)	Smoothing function used by SMOOTH.

SQMAP	If <1 causes the size of the x-axis to be equal to the size of the y-axis (Tuc).
TMARK(6)	The temperatures at which up to six horizontal lines are to be drawn by SHOW.
TRX	The receiver temperature, used by SOLVETIP for total power data (Tuc).
XWIND(2)	Contains the minimum and maximum x-coordinate used to window scans.
YINCR(CH)	The distance between Y tic marks that is to be used by SHOW.
YMIN(CH)	The value of Y which is to be the minimum value to appear on a graph by SHOW.
ZLINE	If >=1, SHOW will draw a horizontal line at zero Kelvins.

Quick Reference

Common continuum quick look data reduction recipes are described below:

1. Recall scan n of various types

```
n CAL              CALIBRATE
n D                Drift Scan
n F                FIVE POINT
n FOCALIZE         FOCALIZE
n S                SEQUENCE
n TPTIP            Total power EXTINCTION
n SPTIP            Switched power EXTINCTION
```

2. Put ten consecutive scans (single channel) in the STACK and average (Tuc).

```
EMPTY
500 509 ADD
C1
```

3. Put ten scans of the second receiver channel in the STACK and average (GB).

```
EMPTY
5000 5090 ADD
C2
```

4. Put ten scans of dual channel data in STACK and average both channels (Tuc).

```
TWO*
EMPTY
500 518 ADD
CB
```

5. Print out average temperature and its error for on-off scan n (Tuc).

```
n LOG
```

6. Print out average temperature and its error for on-off scans n through m (Tuc).

```
n m LOGS
```

*The increment between scan numbers is always ten in GB. In Tucson it is either one (set by the verb ONE) or two for dual channel (set by the verb TWO).

7. Print out average temperature and its error for on-off scans in the STACK (Tuc).

```
EMPTY
n1 A; n2 A; n3 A
SLOG
```

8. Eliminate the pair, m, from on-off scan n, redisplay data and print out recomputed MEAN and RMS (Tuc).

```
n S
m ELIM
```

9. Eliminate the set of pairs, m to n, from on-off scan s, redisplay data and print out recomputed MEAN and RMS (Tuc).

```
s S
m n ELIMS
```

10. Display on-offscan n and print out the individual pair temperatures (Tuc).

```
n S
TEMPS
```

11. Expand a portion of the plot.

```
m1  m2  LIMIT          m1 and m2 are the minimum and maximum
                       point numbers of data to be
                       displayed.
XX
```

12. Remove a baseline and fit a single component gaussian to a drift scan.

```
BASELINE PAGE SHOW PEAK GAUSS
```

13. Remove a baseline and fit a single component gaussian to a drift scan, and display model on top of data.

```
BASELINE PAGE SHOW PEAK GAUSS
TRT GMODEL RESHOW PMW
```

14. Get a listing of header values of scan n on the printer.

```
n SNAPS                 (Tuc)
GET n PRINTER HEADER    (GB)
```

15. Define a procedure to use the crosshairs to describe baseline regions for a single channel scan.

```
PROCEDURE BASESET
   PRINT 'ENTER NO. BL SEGS'
   READ NSEG
   FOR I= 1 TO NSEG*2
          X1=CCUR;  NREGION(1,I)=X1;  END
   IF NSEG <4 THEN NREGION(1,NSEG*2+1)=0;  END;
   RETURN
   FINISH
```

BASESET is a typical user defined procedure which makes use of the READ verb, do loops and conditional execute statements. In Tucson, BASESET is available on disk to be installed by typing

INSTALL BASESET

THE STACK

The STACK is an array of scan numbers, called ASTACK, that is used by the resident procedures C1, C2, CB, SHOW1 and SHOW2. The STACK can also be used by the observer in his own procedures. The adverb, ACOUNT, is used with ASTACK to denote the number of entries currently in the STACK. The STACK has a limit of 100 scans.

Several verbs alter the STACK. They are listed below.

To empty the STACK, type

```
EMPTY
```

To list the scan numbers currently in the STACK, type

```
STACK
```

To add a scan to the STACK, type

```
592 A    or    A 592
```

To add a contiguous group of scans to the STACK, type

```
606 672 ADD    or    ADD (606,672)
```

To delete an individual scan from the STACK, type

```
648 DELETE    or    DELETE 648
```

The STACK is not used by any verbs except those listed above which define its contents. The purpose of the STACK is to give the observer a means to process several scans once or more in a procedure with minimal effort.

The observer can use the STACK for many purposes other than the current resident procedures. For example, to display the last twenty scans observed, one would define the following procedure:

```
PROCEDURE DISPLAY
     FOR I = 1 TO ACOUNT
          GET ASTACK(I) PAGE SHOW END; RETURN
     FINISH
```

All that is now needed is to define the STACK.

```
6000  6190 ADD
```

When DISPLAY is invoked the specified scans will be displayed on the CRT screen one at a time.

Resident Procedures in Tucson

Common data reduction procedures are built into the program for the convenience of the observer. They are listed below for reference.

Data Retrieval and Display

```
PROCEDURE S (XSCAN)
   GET XSCAN PAGE SHOW RETURN
   FINISH

PROCEDURE S1 (XSCAN)
   GET XSCAN PAGE SHOW RETURN
   FINISH

PROCEDURE S2 (XSCAN)
   GET (XSCAN+1) PAGE SHOW RETURN
   FINISH

PROCEDURE D(XSCAN)
   GET XSCAN SLABEL=4; PAGE SHOW RETURN
   FINISH
```

S is a procedure which retrieves and displays the desired scan. S1 and S2 are procedures which retrieve and display the first and second channels respectively of the desired scan. Each procedure takes one argument, the desired scan number, which may be entered before or after the procedure.

```
PROCEDURE CAL (XSCAN)
   SLABEL=3;   GET XSCAN PAGE SHOW TABLE
   PRINT '          TC=',TWH(49,2),' TS=',TWH(50,2)
   PRINT '          #C=',TWH(45,2),'#CP=',TWH(46,2)
   SLABEL=0;   RETURN
   FINISH
```

CAL will retrieve and display a calibration scan, printing out the individual data points, TC, TS, #C and #CP. CAL takes one argument, the desired scan number, which may be entered before or after the procedure.

```
PROCEDURE XX
   PAGE SHOW RETURN
   FINISH
```

XX will clear the screen and display the currently referenced array. It takes no arguments.

Stacking Data

```
PROCEDURE C1
   SCLEAR SLABEL=1; PSTACK(0); AVE TRW PAGE SHOW SLABEL=0; AVG
   PRINT '          C1:'; STACK RETURN
   FINISH

PROCEDURE C2
   SCLEAR SLABEL=1; PSTACK(1); AVE TRW PAGE SHOW SLABEL=0; AVG
   PRINT '          C2:'; STACK RETURN
   FINISH

PROCEDURE CB
   SCLEAR SLABEL=1; PSTACK(0); PSTACK(1); AVE TRW PAGE SHOW SLABEL=0; AVG
   PRINT '          CB:'; STACK RETURN
   FINISH

PROCEDURE PSTACK (OSET)
   FOR I = 1 TO ACOUNT
         GET (ASTACK(I)+OSET) ACCUM END; RETURN
   FINISH
```

C1, C2 and CB are procedures that use the STACK to display the average of a group of on-off scans. The user fills the STACK with the appropriate scans. C1 will only process scans of the first channel, C2 processes scans of the second channel and CB processes both channels. All three procedures retrieve and add subsequent scans to the first scan in the STACK. The final sum is averaged, the page is cleared and the result is displayed. Only positive scan numbers may be entered in the STACK.

```
PROCEDURE C
     SCLEAR SLABEL=4; PSTACK(0); AVE TRW PAGE SHOW SLABEL=0; AVG
     PRINT ' '; PRINT ' '; AVG SUMM=0; RSUM=0;
     FOR I = 1 TO ACOUNT
          GET ASTACK(I) DOC SUMM=SUMM+TWH(51,2); RSUM=RSUM+TWH(52,2);
          END
     AV=SUMM/ACOUNT;  AVR=RSUM/(ACOUNT*SQRT(ACOUNT))
     PRINT '         AV=',AV,'AVR=',AVR; RETURN
     FINISH
```

C is a procedure requested by Dr. William Dent for averaging channel one of a group of scans in the STACK, displaying the averaged data, printing the newly calculated MEAN and RMS from the averaged data and calculating the average of the MEANS of the scans in the STACK.

Editing Data

```
PROCEDURE ELIM (V1)
   V1 SEDIT XX RETURN
   FINISH

PROCEDURE ELIMS (V1,V2)
   V1 V2 SEDITS XX RETURN
   FINISH
```

ELIM is an editing procedure that eliminates the specified pair (one pair = four samples; OFF, ON, ON, OFF) from the currently referenced array, recomputes TAV and TAV-RMS with the selected pair omitted and displays the data with the selected pair omitted. ELIM takes one argument, the desired pair to eliminate, which may be entered before or after the procedure. ELIMS takes two arguments, the beginning and ending number of the desired pairs to be eliminated, which may be entered before or after the procedure.

Documentation

```
PROCEDURE HEAD (XSCAN)
   GET XSCAN HEADER RETURN
   FINISH

PROCEDURE SNAPS (XSCAN)
   GET XSCAN SNAP RETURN
   FINISH
```

HEAD will generate a HEADER on the CRT screen of the specified scan. SNAPS will generate a HEADER on the printer of the specified scan. Both procedures take one argument, the desired scan number, which may be entered immediately before or after the procedure.

```
PROCEDURE LOG (XSCAN)
     GET XSCAN DOC RETURN
     FINISH

PROCEDURE LOGS (BSCAN, ESCAN)
     PAGE FOR I = BSCAN TO ESCAN BY SINCR
               GET I DOC END; RETURN
     FINISH

PROCEDURE SLOG
     PAGE FOR I = 1 TO ACOUNT
               GET ASTACK(I) DOC END; RETURN
     FINISH
```

The LOG procedure prints out TAV and TAV-RMS for the specified scan in degrees K and the same quantities in flux units (Jansky) along with the signal-to-noise ratio. LOG takes one argument, the desired scan number, which may be entered immediately before or after the procedure. The LOGS procedure performs a LOG for a specified series of scans. LOGS takes two arguments, the beginning and ending scan numbers, which may be entered immediately before or after the procedure. SLOG prints out TAV and TAV-RMS for all scans in the STACK. SLOG takes no arguments.

The mean temperature, TAV, is determined from the formula:

$$TAV = \frac{\Sigma_{i=1,5\ldots} T_{i+1}+T_{i+2}-T_i-T_{i+3}}{RPT*2}$$

where T_i is the temperature value associated with the i^{th} datum point and NS is the number of samples in the scan and RPT is the number of OFF-ON-ON-OFF data groups. The first and last points of a RPT will be OFF's and the two intermediate points will be ON's.

The associated error, TAV-RMS, is computed by

$$TAV\text{-}RMS = \left[\frac{\Sigma_{i=1,5\ldots} (X_i)^2}{RPT(RPT-1)}\right]^{\frac{1}{2}}$$

where RPT equals the number of pairs observed and X_i is the residual of the i^{th} pair associated with the computation of the mean temperature, TAV.

$$X_i = (T_{i+1} + T_{i+2} - T_i - T_{i+3}) - TAV$$

Multiple Scan Display

```
PROCEDURE SHOW1
   HISTOGRAM RHIST SHOWS(0); RLINE RETURN
   FINISH


PROCEDURE SHOW2
   HISTOGRAM RHIST SHOWS(1); RLINE RETURN
   FINISH


PROCEDURE SHOWS(OSET)
   SLABEL=2; XSCAN=ASTACK(1)+OSET;  GET XSCAN PAGE SHOW
   FOR I = 2 TO ACOUNT
        XSCAN=ASTACK(I); GET XSCAN
        FACT=(I-1)*SHIFT; BIAS RESHOW END
   SLABEL=0; RETURN
   FINISH
```

SHOW1 and SHOW2 are procedures that use the STACK to create a profile of several scans on the CRT in histogram form. SHOW1 displays the first channel and SHOW2 displays the second channel of all scans in the STACK. The user specifies the bias that is to be added to all scans in the STACK except the first to displace each vertically from its neighbor with the adverb, SHIFT, in units of degrees Kelvin. The user must also set the Y-scaling with RANGE to allow sufficient room for the additional scans.

Special Purpose

```
PROCEDURE F (XSCAN)
PAGE OFF XSCAN; FOR I = 1 TO 5; GET (XSCAN + (I-3) * SINCR)
TWH (128 + I, 1) = TWH (51, PTWH)
TWH (133 + I, 1) = TWH (52, PTWH); END PMT PDOC
RETURN; FINISH
```

F is a procedure which fits gaussians to the N, C and S elevation scans and the W, C and E azimuth scans of a five point observation. The flux units and associated RMS are labelled with each point in a cross pattern. The main and reference offsets, the gaussian parameters for the fit with no restrictions, the gaussian parameters for a fit with the beamwidths fixed at HP*2 and the corrected offsets are printed. F takes one argument, the center scan of the FIVE, which may be entered before or after the procedure.

```
PROCEDURE FOCALIZE (XSCAN)
   SLABEL=2; GET XSCAN POINTS PAGE SHOW RLINE RESHOW
     NAX=0; MAX=0; FO=TWH(32,2); WL=TWH(126,2)
     PRINT '        FO(MM)=',FO,'WL(MM)='WL
     FOR I = 1 TO 7
          IF ABS(TWH(I+128,2)) > MAX THEN MAX=ABS(TWH(I+128,2)); NAX=I;
          END; END
     OAX=NAX-1; IF 7-NAX < OAX THEN OAX=7-NAX; END
     BGAUSS=NAX-OAX; EGAUSS=NAX+OAX; CENTER(1)=NAX; HWIDTH(1)=OAX*2
     GAUSS FOC=(CENTER(1)-4.)*WL+FO; MPT=OAX*2+1
     PRINT '        ',FOC,'BEST FO TO',MPT,' PT GAUSSIAN FIT'
     HISTOGRAM SLABEL=0; RETURN
     FINISH
```

FOCALIZE is a procedure that recalls a FOCALIZE scan, displays the data labelling the bottom axis in mm offset from FO, determines the maximum data value and fits a gaussian symmetrically about the maximum to determine the best FO. FOCALIZE takes one argument, the desired scan number, which may appear before or after the procedure.

```
PROCEDURE TPTIP (XSCAN)
     SLABEL=3; GET XSCAN POINTS PAGE SHOW TABLE HISTOGRAM
     SOLVETIP RHIST RESHOW RLINE SLABEL=0; RETURN
     FINISH
```

TPTIP is a procedure which recalls an EXTINCTION scan, displays the data as individual points (+) and fits an equation of the form $Y=ae^{-bx}$ to the data where

X = SEC(Z)

Y = 1 - (Tsky(secz)-(1-ETA)*Tsbr) / (ETA*Tm))

Tsky(secz) = Tant(secz) - Trx

Tant(secz) = Observed temperature values as a function of secant(Z)

TRX = Receiver temperature. Must be input by the user.

Tsbr = Temperature at which the spillover, blockage and radiative losses are terminated. Is computed as some fraction (FTSBR) of the ambient temperature. FTSBR may be set by the user and has a default value of 0.95.

Tm = Mean temperature of the atmosphere. Is computed as some fraction (FTM) of the ambient temperature. FTM may be set by the user and has a default value of 0.95.

Tamb = Ambient temperature recorded as the scan is being observed and is stored in the scan header.

ETA = Coupling efficiency of the telescope to the sky. May be set by the user and has a default value of 0.8.

The fit returns the values of a, b and FCF where

a = Constant where a=1.0 implies a perfect fit.

b = Zenith optical depth, ATTN

FCF = The coefficient of determination of the fit where FCF=1.0 implies a perfect fit.

The values of ATTN, a, FCF and Tamb are printed on the bottom left edge of the plot and the input adverbs: ETA, FTM, FTSBR and TRX are displayed at the lower right edge of the plot. The fitted Tant values are calculated and plotted over the original data in histogram mode.

TPTIP takes one argument, the desired scan number, which may appear before or after the procedure. It is very important that the extinction data be well-calibrated to insure a good fit.

```
PROCEDURE SPTIP (XSCAN)
SETA = ETA; ETA = -1; SLABEL = 3; POINTS PAGE GET XSCAN
FOR I = 1 TO 11
     TWH (I + 128, PTWH) = LN (TWH (I = 128, 2); END
TABLE HISTOGRAM SOLVETIP RHIST RESHOW RLINE
SLABEL = 0; ETA = SETA; RETURN: FINISH
```

SPTIP is a procedure which recalls a switched power EXTINCTION scan, displays the data as individual points (+) and fits an equation of the form

$$LOG_e\ \Delta = LOG_e\ T_{amb} - T/SIN\ EL$$

where Δ = the temperature values at each elevation angle. SOLVETIP performs a linear fit to $LOG_e\ \Delta$ as a function of secant Z by linear regression. The resulting slope = -T.

V E R B

A N D

O P E R A T O R

D I C T I O N A R Y

A

A will insert one scan into the STACK.

PTWH — Is not used or changed.

ADVERBS

ACOUNT	The STACK counter is incremented by one.
ASTACK	The specified scan is added to the STACK array.

OPERANDS

scan#	The scan number to be inserted into the STACK.

RELATED VERBS

ADD (#,#)	Inserts a group of scans into the STACK.
DELETE #	Deletes a scan from the STACK.
EMPTY	Empties the STACK.
STACK	Lists all scans in the STACK.

REMARKS

Positive scan numbers may be entered in the STACK. The operand may precede or follow A.

Multiple references to A on a single line must be separated by semicolons.

EXAMPLES

To add 650 to the scans already in the STACK specify:

650 A or A 650

ABS

A built-in absolute value function. Functions the same as the Fortran ABS.

PTWH — Neither used nor changed.

ADVERBS — None.

OBJECT

argument — The argument may be a constant, a variable, or an arithmetic expression, enclosed in parentheses.

EXAMPLE

```
PRINT ABS( -4 )
 4.0000
```

ACCUM

A routine that is used to stack scans in the HOLD array. The first use of ACCUM after an AVE, SCLEAR or a program restart copies the header information and the data values in the WORK array to the HOLD array. Subsequent uses of ACCUM add the data values contained in the WORK array to the contents of the HOLD array. Subsequent uses of ACCUM do not change the header stored in the HOLD array, except for the integration time, which is summed each time a scan is accumulated. ACCUM increments the internal stack counter each time a scan is accumulated, and stores the numbers of the first 17 and the last scan accumulated.

PTWH	Must be pointing to WORK array. Is set to the HOLD array.
ADVERBS	None

RELATED VERBS

AVE	Divides the accumulated scans by the number of scans accumulated. Sets the internal stack counter to zero. (See ** on the next page.)
TELL CSTACK	Prints the internal stack counter and the numbers of scans accumulated.
SCLEAR	Sets the accumulator flag and internal stack counter to zero.
SUM	Accumulates scans with user supplied weighting.

-continued-

EXAMPLE

To accumulate scans 602, 604, and 606 first specify:

```
SCLEAR
```

to empty the stack. (This is not necessary if no scans have been accumulated since the last AVE.) Then, to accumulate the scans, specify:

```
GET 602 ACCUM
GET 604 ACCUM
GET 606 ACCUM
```

If you now specify TELL CSTACK you will get:

```
3
602  604  606
```

To average the accumulated scans, specify:

```
AVE
```

** The internal stack and counter are used by ACCUM and AVE for bookkeeping purposes and are stored in the header of the accumulated scan. They are not to be confused with ASTACK and ACOUNT, the POPS adverbs which are defined by the observer and are used by procedures C1, C2, and CB (Tucson only) to stack scans.

ADD

A routine which inserts a series of scan numbers into the STACK.

PTWH — Is not used or changed.

ADVERBS

ACOUNT — The STACK counter is incremented by the number of scan numbers inserted, this by the call to ADD.

ASTACK — The scan numbers are inserted into the adverb array ASTACK.

SINCR — The increment between scan numbers. It may be set to one or two in Tucson (see ONE and TWO), but is fixed at ten in Green Bank.

OPERANDS — Beginning and end scan number.

RELATED VERBS

A # — Inserts one scan into the STACK.

DELETE # — Deletes one scan from the STACK.

EMPTY — Empties the STACK.

STACK — Lists the STACK.

REMARKS

The operands may precede or follow ADD. If they follow ADD, they must be enclosed in parentheses. Multiple references to ADD on a single line must be separated by semicolons.

EXAMPLE

To accumulate the first channel of scans between 500 and 600 type:

```
500  600  ADD      or      ADD (500,600)
C1
```

The first channel of scans 500 to 600 will be accumulated, averaged and displayed. (C1 is currently implemented in Tucson only.)

ARRAY

This operator is used in a procedure to declare an array. It requires as its object the name of the array that is to be constructed and the dimensions of the array. An array may have any number of dimensions, limited only by the amount of space available for variable declaration. The size of the array in any dimension may be specified in two ways:

(SCALAR) or (SCALAR TO SCALAR)

SCALAR may be either a constant or a variable which has been assigned a value, but may not be an expression.

PTWH Is not used or changed by ARRAY.

ADVERBS None.

RELATED VERBS None

REMARKS

ARRAY can be used to set up an array in which results of operations can be stored. It can only be used in procedures. Arrays defined in procedures cannot be edited.

EXAMPLES

(1) valid definitions:

```
ARRAY ALPHA(N)
ARRAY BETA(-1 TO +1), GAMMA(3,5)
ARRAY DELTA(6 TO LEVEL, 8 TO 10)
```

(2) You want to store a number of results and print them all out at one time. For RMS calculations, for example:

```
PROCEDURE RMSCALC(FSCAN, LSCAN,N)
ARRAY RESULTS (N)
J = LSCAN - FSCAN + 1
FOR I = 1 TO J
   GET FSCAN + I - 1; RMS
   RESULTS (I) = VRMS(1)
END
PRINT RESULTS
FINISH
```

ATAN

A built-in arctangent function. Functions the same as the Fortran ATAN.

PTWH — Neither used nor changed.

ADVERBS — None.

OBJECT

argument — The result of ATAN is expected to be in radians. The argument may be a constant, a variable, or an arithmetic expression, enclosed in parentheses.

RELATED VERBS

see COS, SIN, and TAN

EXAMPLE

```
PRINT ATAN( 1.0~)
 0.7854
```

AUTO

AUTO returns the control of the y-axis scaling to the verb SHOW. Automatic scaling is the default, so AUTO is used to restore automatic y-axis scaling after the user has overridden it by means of the verb HOLDY, or by setting the values of the adverbs YMIN and YINCR by using the verb RANGE or with assignment statements.

PTWH — Is not used or changed.

ADVERBS

YMIN — The minimum temperature to appear on the graph. AUTO sets YMIN to -90000. SHOW sets its own y-axis scaling if YMIN is less than -9999 but greater than -9.E9.

RELATED VERBS

FIX — Alias for HOLDY (below).

FREEY — Same as AUTO.

HOLDY — Causes SHOW to retain the last determined y-axis scaling.

RANGE — Defines the values of YMIN and YINCR according to the specified max and min values.

AVE

AVE divides the accumulated scans in the HOLD array by the accumulated weight. The result is left in the HOLD array. AVE also zeros the stack counter.

PTWH	Current value is not used. Is set to the HOLD array.
ADVERBS	None.

RELATED VERBS

ACCUM	Adds a scan to the HOLD array.
TELL CSTACK	Prints the stack counter and the scan numbers of the scans accumulated.

ERRORS

If the HOLD (accumulator) array is empty when AVE is specified, the message AVE EMPTY? or UNAVAILABLE! will appear. This tells the user that there is nothing to average.

-continued-

EXAMPLES

(1) see example with ACCUM

(2) to stack a number of scans, or two groups of scans, you might define the following procedure:

```
PROCEDURE STACKSCANS (FSCAN, LSCAN)
FOR I = FSCAN TO LSCAN BY 2
   GET (I) ACCUM
END
RETURN
FINISH
```

Then you could use the procedure to stack a few scans in the HOLD array:

```
STACKSCANS (700, 720)
STACKSCANS (760, 782)
```

When you have stacked all the scans you want, you can obtain the average and display it by

```
AVE PAGE SHOW
```

AZ (Tucson only)

AZ sets an internal flag that causes LABEL to identify azimuth as the y-coordinate and elevation as the x-coordinate for maps. Would be used if the telescope were scanned along lines of constant azimuth.

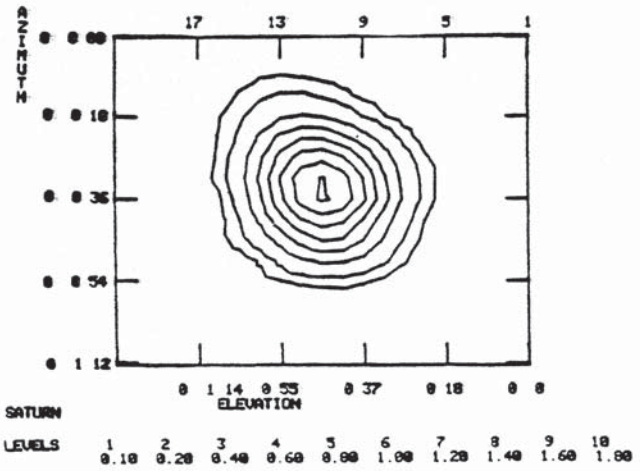
PTWH — Is neither used nor changed.

ADVERBS — None.

RELATED VERBS

DEC	Identifies declination as the Y-coordinate for maps (Tucson only).
EL	Identifies elevation as the y-coordinate for maps (Tucson only).
LABEL	Labels the map drawn by MAPSHOW.
RA	Identifies right ascension as the y-coordinate for maps (Tucson only).

EXAMPLE



BASELINE

This routine computes by a least squares fit the coefficients of a Chebyshev polynomial of a specified order for a specified region, or regions, of the scan in the currently referenced array. The polynomial is then evaluated for each point and subtracted from the scan.

PTWH — Determines which scan has a baseline removed. It is not changed.

ADVERBS

NFIT — The order of the polynomial baseline to be removed. NFIT must be between 1 and 12.

BDROP(CH)
EDROP(CH) — The number of points at each end of each receiver channel that will be ignored by the routine if NREGION(CH,1) = 0.

NREGION(CH,N) — Specifies the edges of a region or regions of the scan to be used to compute the baseline coefficients. Up to four regions may be specified for each receiver channel. N is an integer between 1 and 8.

BBASE(CH)
EBASE(CH) — The number of points at each end of each receiver channel (not including the points dropped by BDROP and EDROP) which will be used to compute the baseline coefficients if NREGION(CH,1) = 0.

RELATED VERBS

BSHAPE — Computes the coefficients of the Chebyshev polynomial.

BMODEL — Computes the point by point values of the baseline using the last computed coefficients of the Chebyshev polynomial.

DCBASE — Computes the average value of the data for a specified range of points and subtracts this from all points in the scan.

MDBASE — Computes the median values over a set of data points centered on each point in a scan, and subtracts the medians from the data.

PCBASE — Computes and subtracts from the data that constant which makes a specified percentage of the data points negative.

-continued-

REMARKS

The regions used to compute the baseline should not include a known or suspected source structure.

ERRORS

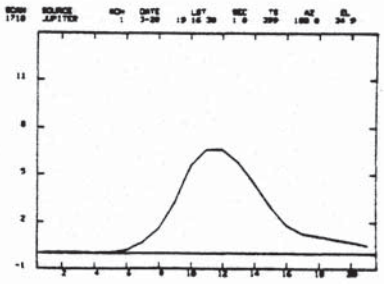
If NREGION(CH,N), where N is even, is less than NREGION(CH,N-1), the message

NREGION?

will appear. (The endpoint of a baseline fitting region must be greater than the startpoint.)

EXAMPLE

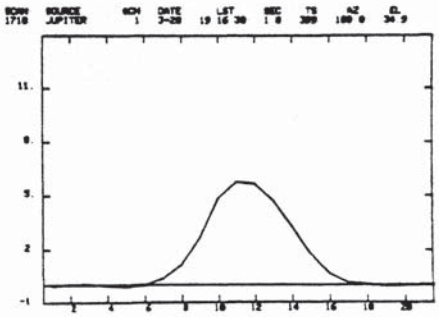
Your data looks like:



You want to remove a third order baseline. Specify:

```
NREGION (1,1) = 0
BBASE = 4
EBASE = 4
NFIT = 3
BASELINE PAGE SHOW
```

You will get:



BESTVIEW (Green Bank only)

BESTVIEW is used to find the maximum range in the x-coordinate shared by a group of scans. Minimum and maximum x-coordinates, in radians, are stored in XWIND(1) and XWIND(2), respectively. BESTVIEW assigns to XWIND(1) and XWIND(2) the more restrictive of (1) the current value of XWIND, or (2) the end point of the scan in the currently referenced array. Before using BESTVIEW for a group of scans, set XWIND(1) and XWIND(2) to values outside the expected range.

PTWH	Determines which array endpoints are compared to the current values of XWIND. It is not changed.

ADVERBS

XWIND(1) XWIND(2)	The minimum and maximum x-coordinates found by BESTVIEW. NOTE: the units for XWIND are radians.

RELATED VERBS

WINDOW	Sets the starting and stopping points of a scan to the values in XWIND.

REMARKS

BESTVIEW can only process a single scan at a time. It may be more convenient to use it in a procedure which processes several receivers or loops through a set of scans.

If you have multi-feed data, you should use FOFFSET to correct for the feed offsets from the center of the receiver box before using BESTVIEW.

-continued-

EXAMPLE

Suppose scans 8701 to 8901 are right ascension scans of a map, but that each has a different starting and stopping right ascension. To determine the least restrictive windowing of the data before mapping:

```
XWIND(1)=0;XWIND(2)=8
GET 8701 BESTVIEW
GET 8711 BESTVIEW
     ...
GET 8901 BESTVIEW
```

Then construct a map:

```
GET 8701 WINDOW MAPSHOW
GET 8711 WINDOW MAPSHOW
     ...
GET 8901 WINDOW MAPSHOW LABEL
```

Both of these steps are obviously facilitated by writing procedures for looping through the scans.

BIAS

BIAS adds the values of FACT(CH) to the scan in the currently referenced array. The array index is the receiver number.

PTWH — Determines to which scan FACT is added. It is not changed.

ADVERBS

FACT(CH) — The number of degrees Kelvin which is to be added to a scan. FACT is also used by the verb SCALE.

RELATED VERBS

SCALE — Also uses the adverb FACT. SCALE multiplies the data in the currently referenced array by the appropriate FACT(CH).

EXAMPLE

BIAS can be used to plot more than one scan on the same graph, separated for visibility. Specify

```
GET 1718 PAGE SHOW
```

-continued-

Set the Y-scaling so that the first scan will be plotted at the bottom of the graph. For the graph below,

```
-1 14 RANGE
```

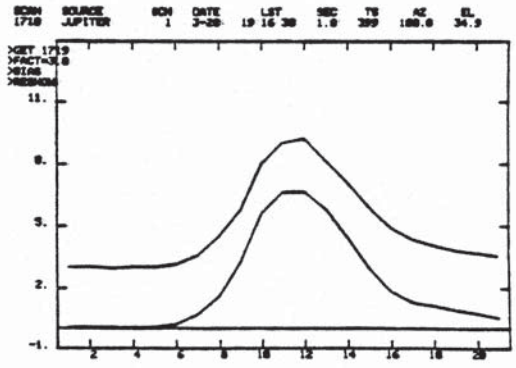
was used. Replot. Then set FACT to separate the scans:

```
PAGE SHOW
GET 1719
FACT = 3.0
```

and specify

```
BIAS RESHOW
```

The results look like:



BMODEL

BMODEL evaluates the last computed coefficients for a polynomial baseline for each point and replaces the currently referenced array with the evaluated polynomial.

PTWH — Determines which array is replaced by the baseline model. Is not changed.

ADVERBS — None used directly -- BMODEL uses the results of the previous BASELINE or BSHAPE.

RELATED VERBS

BASELINE — Computes the coefficients for the polynomial, evaluates the polynomial and subtracts it from the currently referenced array.

BSHAPE — Computes the coefficients for the polynomial using the currently referenced array.

REMARKS

Most often used in conjunction with BSHAPE to compute a baseline without subtracting it from the data. It is often useful to plot the baseline model on top of the data for comparison purposes.

-continued-

EXAMPLE

You have twenty scans of the same source and you want to remove exactly the same baseline from each of them. You can do this by

```
GET 1710 TRT BSHAPE BMODEL PMW DIFF SHOW
GET 1712 DIFF PAGE SHOW
GET 1714 DIFF PAGE SHOW
     .
     .
     .
GET 1748 DIFF PAGE SHOW
```

The first set of commands above create a model of the baseline in the TEMP array and subtract the result from the scan in the WORK array. The result of the subtraction is left in the WORK array and the TEMP array is not changed. Thus the model can be subtracted from scan after scan.

BMSWITCH (Green Bank only)

BMSWITCH constructs a double gaussian model of a beam-switched response in the TEMP array. The positive gaussian is in the left side of the array and the negative gaussian to the right. The gaussian half-width and the beam separation are input using the adverbs BEAMHW and BEAMSEP. The header is copied from the currently referenced array.

PTWH — This determines the header copied into the TEMP array. It is not changed.

ADVERBS

BDROP(CH)
EDROP(CH) — The number of points at each end of the data which are not used in defining the model. At present set these to zero.

BEAMHW — The half-width (in arcmin) of the gaussian used in constructing the model.

BEAMSEP — The separation of the two feeds (in arcmin) used in constructing the model.

RELATED VERBS

CROSSFCN — Used to obtain the cross-correlation function of the beam-switched data with the double gaussian model in the TEMP array.

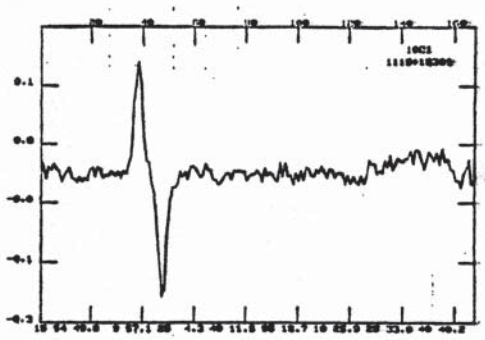
REMARKS

BMSWITCH overwrites the data in the TEMP array. If BDROP and EDROP are used, be careful that their values are the same if you will be using CROSSFCN.

-continued-

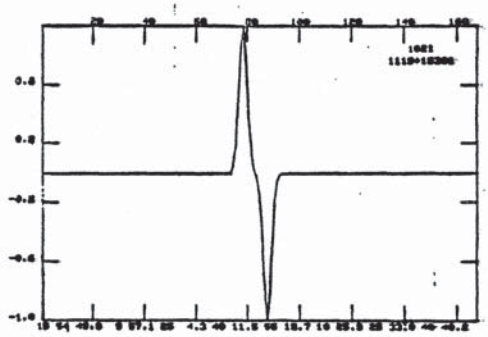
EXAMPLE

Suppose you have made the drift scan illustrated, across a source with a beam switched receiver, and you wish to determine the right ascension and antenna temperature of the source.



First, use BMSWITCH to create a model of the beam switched response. The beam halfwidth, assumed the same for both beams, and the beam separation must be specified beforehand:

```
BEAMHW=2.9;BEAMSEP=7.05
BMSWITCH PMT XX
```

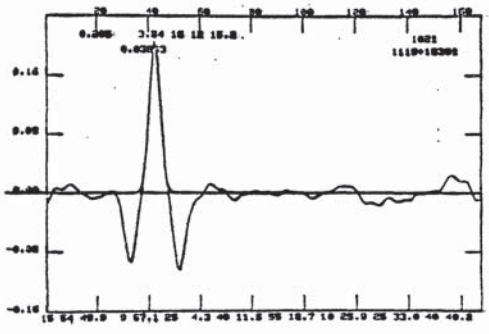


Next, use CROSSCFN to form the cross-correlation function of the model in the TEMP array with the data in the WORK array:

```
PMW CROSSFCN PAGE SHOW
```

Finally, fit a gaussian to the cross-correlation function to find the position and antenna temperature of the source:

```
PEAK GAUSS GMODEL RESHOW
```



BOXCAR

This routine smoothes a scan by averaging an odd number of consecutive points together and placing the result in the center point.

PTWH	Determines which scan is smoothed. Is not changed.

ADVERBS

BDROP(CH) EDROP(CH)	The number of points at each end of each receiver channe which will be ignored by the routine.
NBOX	The number of points which will be averaged together to smooth the scan.

RELATED VERBS

HANNING	A smoothing routine which averages three points toget with the center point getting twice as much weight as either side point.
SMOOTH	A smoothing routine which averages data according to user specified weighting.

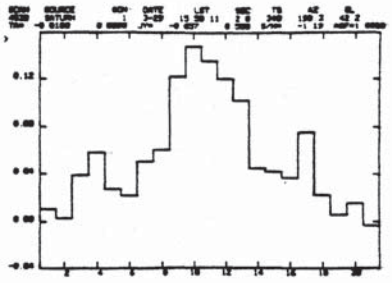
ERRORS

If NBOX is not odd, the error message "NBOX ODD?" is printed, and the BOXCAR operation is not performed.

-continued-

EXAMPLE

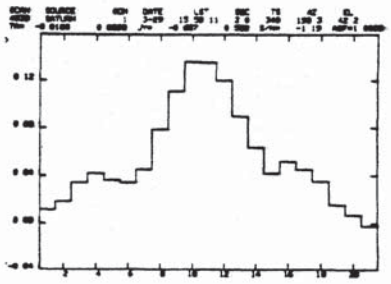
Your data looks like the scan on the right and you want to smooth it.



If you specify:

```
NBOX = 3
BOXCAR PAGE SHOW
```

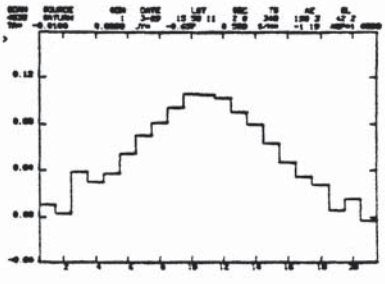
you will get the display to the right.



If you specify:

```
NBOX = 7
BOXCAR PAGE SHOW
```

you will get the display to the right.



BSHAPE

BSHAPE calculates the coefficients of a Chebyshev polynomial of a given order by a least squares fit for a specified region or region of spectral values. BSHAPE does NOT evaluate the polynomial. (BASELINE and BMODEL do.)

PTWH — Determines the scan to which the baseline is fit. It is not changed.

ADVERBS

NFIT — The order of the polynomial baseline to be removed. NFIT must be between 1 and 12.

BDROP(CH)
EDROP(CH) — The number of points at each end of each receiver channel that will be ignored by the routine if NREGION(CH,1) = 0.

NREGION(CH,N) — Specifies the region or regions of the scan to be used to compute the baseline coefficients. N is an integer between 1 and 8.

BBASE(CH)
EBASE(CH) — The number of points at each end of each receiver channel (not including the points dropped by BDROP and EDROP) which will be used to compute the baseline coefficients if NREGION(CH,1) = 0.

RELATED VERBS

BASELINE — Not only computes the coefficients but evaluates them for each point and subtracts the evaluated polynomial from the data.

BMODEL — Evaluates a polynomial using the last computed coefficients and replaces the data with the evaluated polynomial.

-continued-

REMARKS

BSHAPE is most frequently used in conjunction with BMODEL to construct a model of a particular baseline. The set of commands

TRT BSHAPE BMODEL PMW DIFF

is equivalent to BASELINE.

EXAMPLE

You want to remove a parabolic baseline from scan 1718, which is in the WORK array, but before you subtract the parabola from the scan, you want to see how well it fits. To copy the scan to the TEMP array, compute the polynomial coefficients, and replace your copy of the scan with the computed polynomial, specify:

TRT BSHAPE BMODEL

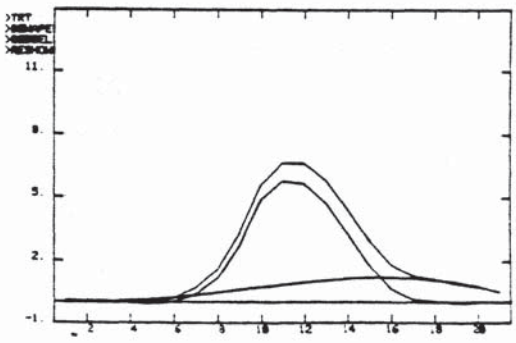
To plot the baseline model over the scan already plotted on the screen:

RESHOW

To subtract the model from the data in the WORK array and plot the results over the original scan and the baseline model:

PMW DIFF RESHOW

When you are done, you will have something like this:



CCUR

This routine activates the vertical crosshair and returns the sample number for that crosshair position on the current display when any key except RETURN is struck.

PTWH — Is not used or changed by CCUR.

ADVERBS — None.

RELATED VERBS

CROSSHAIR — Activates the horizontal and vertical crosshairs and prints the point number, position, and temperature values at crosshair positions.

TCUR — Returns the temperature value at the current horizontal crosshair position.

REMARKS

CCUR is best used in a procedure, generally for defining baseline regions or for entering initial guesses for fitting gaussians.

EXAMPLE

A procedure for setting gaussian parameters with the crosshairs is defined below:

```
PROCEDURE GSET
PRINT 'ENTER # OF GAUSSIANS'
READ NGAUSS
PRINT 'SET CENTERS'
FOR I = 1 TO NGAUSS
    X1 = CCUR;  CENTER(I) = X1;  END
PRINT 'SET HWIDTHS'
FOR I = 1 TO NGAUSS
    X1 = CCUR;  X2 = CCUR;  HWIDTH (I) = X2 - X1;  END;
PRINT     'SET BGAUSS & EGAUSS'
X1 = CCUR;  BGAUSS = X1;  X1 = CCUR;  EGAUSS = X1
RETURN
FINISH
```

CGET (Green Bank only)

CGET is used to retrieve from disk the indicated feed of the most recently completed scan. It requires the receiver (i.e. feed) number as its object. CGET copies the scan from disk into the WORK array.

PTWH — Not used but is set to the WORK array.

ADVERBS — None

OBJECT

Receiver # — The most recently completed scan for this receiver (feed) is copied into the WORK array.

RELATED VERBS

FETCH scan # — Copies the first feed of the specified scan into the WORK array and the second feed into the TEMP array.

GET scan # — retrieves the specified scan from disk and places it in the WORK array.

ON scan # — retrieves the specified scan from disk and places it in the WORK array.

OFF scan # — retrieves the specified scan from disk and places it in the TEMP array.

ERRORS

The receiver (feed) number must be in the range $1 \leq CH \leq 4$ or the message UNKNOWN? will be generated.

EXAMPLE

You wish to retrieve from disk the most recently completed scan from the second receiver and display it on the screen; specify:

```
CGET  2  PAGE  SHOW
```

CLIP

CLIP resets any data points in the currently referenced array with values greater than CLIPMAX to CLIPMAX and any data points with values less than CLIPMIN to CLIPMIN.

PTWH	Determines which scan is clipped. Is not changed.

ADVERBS

CLIPMAX	The maximum data value that will be found in a scan after it is clipped.
CLIPMIN	The minimum data value that will be found in a scan after it is clipped.

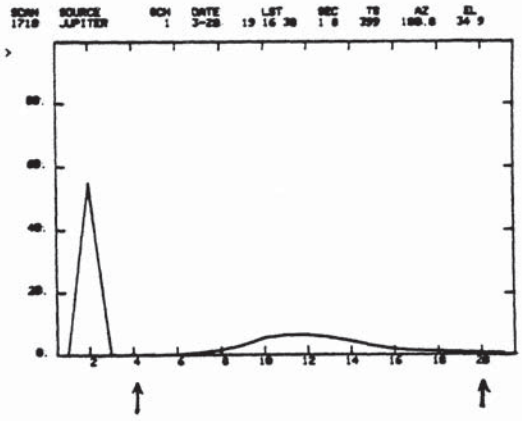
REMARKS

CLIP is commonly used to limit the range of data values in a scan.

-continued-

EXAMPLE

Your data look like the display to the right.

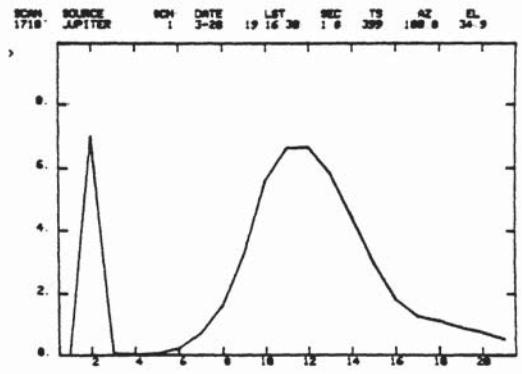


You are interested in the structure between the arrows.

Specify:

```
CLIPMIN = 0.0
CLIPMAX = 7.0
CLIP PAGE SHOW
```

and you will get the display to the right.



CORE

CORE prints the amount of memory available for procedure and variable definition.

PTWH — Is not used or changed by CORE.

ADVERBS — None.

RELATED VERBS

RESTART	Empties the procedure and variable space.
STORE page #	Stores the current procedures in disk page page#. page# must be between 1 and 6.
RESTORE page#	Retrieves disk page page#.

REMARKS

CORE is purely an informational verb. The amounts of space allocated for procedure source code, for compiled procedure code, and for variable definition are fixed. Filling up any of these spaces while you are defining a new procedure will give the error message BLEW CORE.

If you see the BLEW CORE error message, then you are stuck. You may continue with the procedures that you have defined already, but you will not be able to edit them. The only cure is to do a RESTART, which will delete all of the procedures that you have defined, and then start typing in your procedures again. Since editing a procedure uses additional memory space, the procedures may all fit if they can be typed in without editing, and if they were edited the first time around. Alternatively, you may decide which procedures may be grouped into two or more memory pages, stored separately on disk.

You can avoid getting the message BLEW CORE by not trying to define a procedure when there is very little space left in memory. You can protect yourself from the BLEW CORE condition to a limited extent by STOREing the procedure definition space after each new procedure is defined. If you then get the BLEW CORE message while defining a new procedure, you may RESTORE all your other procedures or RESTART and then put your new procedure on a new memory page.

-continued-

EXAMPLE

You have defined three procedures and you want to know if there is room for any more. You specify

```
CORE
```

and get the output

```
                    SPACE AVAILABLE FOR
PROGRAM             VARIABLES           SOURCE
4021 5027           3729 4189           1635 2560
```

This tells you that you have 1006 words of program space, 460 words of variable space and 825 words of procedure source space left.

COS

A built-in cosine function. Functions the same as the Fortran COS.

PTWH — Neither used nor changed.

ADVERBS — None.

OBJECT

argument — The argument of COS is assumed to be in radians. The argument may be a constant, a variable, or an arithmetic expression, enclosed in parentheses.

RELATED VERBS

see SIN, TAN, and ATAN

EXAMPLE

```
PRINT COS(0.7854)
 0.7071
```

CROSSFCN (Green Bank only)

CROSSFCN computes a cross-correlation function of a model in the TEMP array with the data in the WORK array. The data in the WORK array are replaced by the cross-correlation function. The original scan in the WORK array is copied to the HOLD array on exit.

PTWH — Must point to the WORK array. It is not changed.

ADVERBS

BDROP(CH)
EDROP(CH) — The number of points at each end of the data which are not used in computing the cross-correlation function. At present, set these to 0.

RELATED VERBS

BMSWITCH — Constructs a double gaussian beam-switching model in the TEMP array.

GMODEL — Constructs a gaussian model in the currently referenced array.

REMARKS

The model in the TEMP array must have the same number of points as the data in the WORK array. To assure this, after you have loaded the data into the WORK array, copy the WORK array to the TEMP array with:

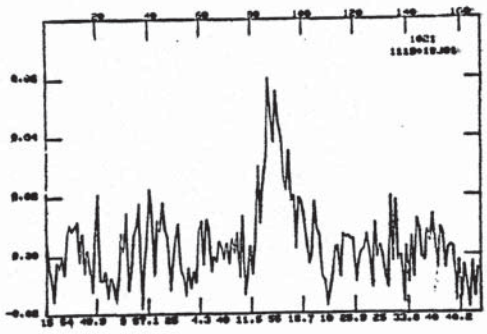
```
.PMW TRT
```

When dealing with beam-switched data, CROSSFCN, is often used with the verbs BMSWITCH and GAUSS to obtain the best estimate of the response of a single feed at the centre of the feed box to determine the peak flux at the constant coordinate of the scan and its location in the "scanning" coordinate. If BMSWITCH is used to construct the model for cross-correlation, it is assumed that the response of both feeds is equal and gaussian. Do not apply the feed offset to the coordinates of the feed before using CROSSFCN.

-continued-

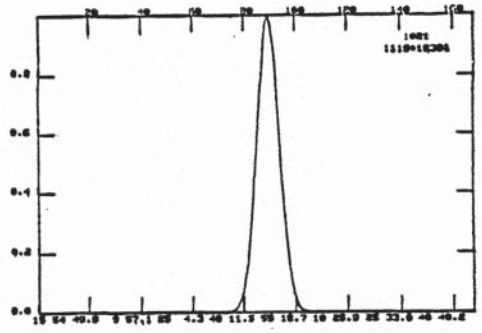
EXAMPLE

Suppose you have made the drift scan illustrated at right across a weak source. If you know the beamwidth of the feed, perhaps from a scan across a strong source, then you can optimally filter the data by cross-correlating the scan with a gaussian of the proper width.



First, specify the halfwidth of the gaussian in channels (use CROSSHAIR) and construct a gaussian in the TEMP array:

```
HWIDTH = 10; HEIGHT = 1
TRT GMODEL PAGE SHOW
```

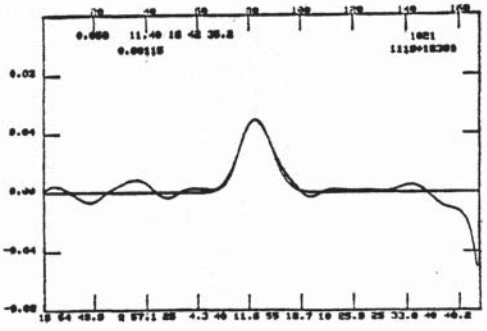


Use CROSSFCN to cross-correlate the gaussian model with the data:

```
PMW CROSSFCN PAGE SHOW
```

to produce the display at right. Finally, perform a gaussian fit to the cross-correlation function to determine the antenna temperature of the source:

```
PEAK GAUSS GMODEL RESHOW
```



CROSSHAIR

This routine activates the horizontal and vertical crosshairs.

In Green Bank:
Position the crosshairs with the thumbwheels located at the lower right edge of the terminal. Striking the space bar causes the intersection of the crosshairs to be marked with a '+' and the point number, position, and temperature at that point to be labeled. After the information is printed, the crosshairs are prompted again. Striking any character other than the space bar will deactivate the crosshairs. Do not strike the RETURN key while using crosshair.

In Tucson:
Position the crosshairs with the thumbwheels located at the lower right edge of the terminal. Striking any key except the ALTMODE or RETURN keys causes the intersection of the crosshairs to be marked with a '+' and the point number and temperature at that point to be labeled. After the information is printed, the crosshairs are prompted again. Striking the ALTMODE key will deactivate the crosshairs. Do not stike the RETURN key when using CROSSHAIR.

PTWH — Is not used or changed by CROSSHAIR.

ADVERBS — None.

RELATED VERBS

CCUR — Returns the point value at the current vertical crosshair crosshair position.

TCUR — Returns the temperature value at the current horizontal crosshair position.

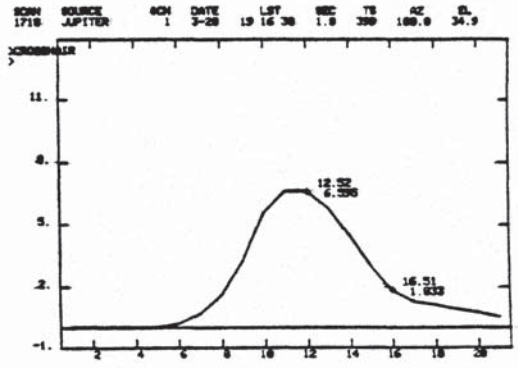
REMARKS

CROSSHAIR prints information concerning the crosshair positions on the CRT screen. CCUR and TCUR derive a value from the appropriate crosshair and stores the value in the program to be used in setting adverb values.

-continued-

EXAMPLE

A Tucson example of the output from CROSSHAIR is shown below.



CRT

Used to direct the output of certain print routines to the CRT screen. Those verbs affected are HELP, PRINT, SUMMARY, LIST, and TELL DISK.

PTWH — Neither used nor changed.

ADVERBS — None.

RELATED VERBS

PRINTER — Directs print output to the line- printer.

REMARKS

CRT is the default print output direction. Using PRINTER may result in more readable copy than printing on the CRT screen and making a copy of the screen.

DCBASE

DCBASE computes the average data value over a specified range of data values and subtracts the average from all the data values in the currently referenced array.

PTWH — Determines which scan has a baseline removed. Is not changed.

ADVERBS

BDROP(CH)
EDROP(CH) — The number of points at each end of each receiver channel which are ignored by the routine if NREGION(CH,1) = 0.

NREGION(CH,N) — Specifies the edges of a region or regions of the data which are used to compute the average value. Up to four regions may be specified for each receiver channel. N is an integer between 1 and 8.

BBASE(CH)
EBASE(CH) — Specifies the number of points at each end of each receiver channel (not including the points dropped by EDROP and BDROP) that will be used to compute the average data value if NREGION(CH,1) = 0.

RELATED VERBS

See BASELINE
BIAS

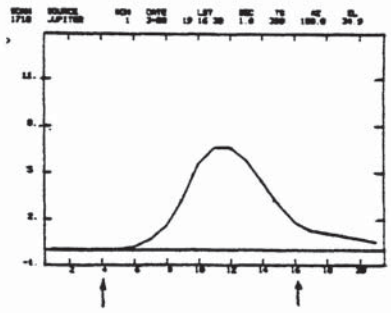
REMARKS

A conservative means of baseline removal since baseline slope is retained.

-continued-

EXAMPLE

Your data looks like

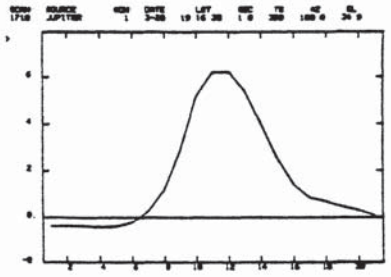


You want to remove an average computed over these regions

So you specify

```
NREGION (1,1) = 0
BBASE = 4
EBASE = 4
ZLINE = 1
DCBASE
PAGE SHOW
```

And you will get



DEC (Tucson only)

DEC sets an internal flag that causes LABEL to identify declination as the Y-coordinate and right ascension as the X-coordinate for maps. This is the default condition on entry to CONDAR. Would be used for drift scans or if the telescope were scanned along lines of constant declination.

PTWH — Is not used or changed.

ADVERBS — None.

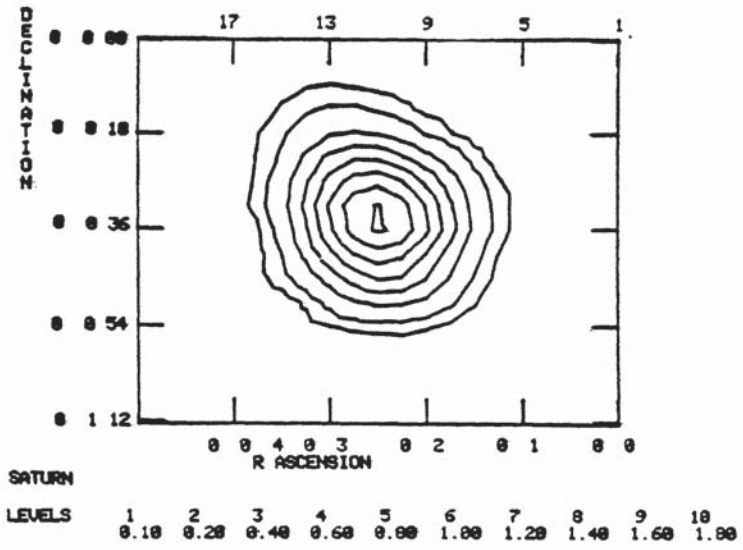
RELATED VERBS

RA	Identifies right ascension as the Y-coordinate and declination as the X-coordinate for maps (Tucson only).
AZ	Identifies azimuth as the Y-coordinate and elevation as the X-coordinate for maps (Tucson only).
EL	Identifies elevation as the Y-coordinate and azimuth as the X-coordinate for maps (Tucson only).
LABEL	Labels the map drawn by MAPSHOW.
MAPSHOW	Draws a map with the levels specified by LEVS.

-continued-

EXAMPLE

An example of a display is shown below:



DELETE

A routine which deletes one scan number from the STACK.

PTWH — Is not changed or used.

ADVERBS

ACOUNT — The STACK counter, which is decremented by one.

ASTACK — The specified scan number is deleted from the STACK array contained in the ASTACK array.

OPERANDS

scan# — The number of the scan to be deleted from the STACK.

RELATED VERBS

A scan# — Inserts one scan number into the STACK.

ADD (#,#) — Inserts a series of scan numbers into the STACK.

EMPTY — Empties the STACK.

STACK — Lists the STACK.

REMARKS

The scan number may be typed immediately proceeding or following the verb DELETE.

EXAMPLE

The STACK currently has scans 512-536 in it, but you want to delete scan 524. Type

```
524  DELETE
```

DIFF

This routine subtracts the scan in the TEMP array from the currently referenced array.

PTWH — Determines from which array the TEMP array is subtracted. It is not changed.

ADVERBS — None.

REMARKS

This verb can be used to remove baselines by subtracting an off-source scan.

EXAMPLE

To difference a scan 4588 and an off-source scan 4575, and display the result

```
OFF 4575 ON 4588 DIFF PAGE SHOW
```

DIVIDE

This routine divides the scan in the currently referenced array by the scan in the TEMP array. The result is left in the currently referenced array.

PTWH — Determines which array is divided by the TEMP array. It is not changed.

ADVERBS — None.

EXAMPLE

To divide scan 548 by scan 550:

put the dividend scan in the WORK array:	ON 548
put the divisor scan in the TEMP array:	OFF 550
point to the dividend scan:	PMW
form the quotient:	DIVIDE
display the result:	PAGE SHOW

The same example using GET rather than ON and OFF is:

```
GET 550 TRT
GET 548 DIVIDE PAGE SHOW
```

EDIT

EDIT is used to change an already defined procedure. EDIT puts the program into EDIT mode and signals this to the user by prompting with a colon instead of a caret. EDIT requires as its object the name of the procedure to be edited and the number of the line in the procedure which is to be changed. If the specified line number matches a line in the procedure, then that line is replaced by the first line entered in the EDIT mode and subsequent lines are inserted at that point. If the line number is an interpolation between two existing lines, then no line in the procedure is replaced, and lines entered in the EDIT mode are inserted at that point.

PTWH — Is not used or changed by EDIT.

ADVERBS — None.

OBJECTS

Procedurename	The name of the procedure to be edited.
Line#	The line number of the line in the procedure which is to be changed, if an integer. Subsequent lines inserted after that line. If line# is not an integer, lines entered are inserted between the lines implied. See the third example.

RELATED VERBS

LIST procedurename	Lists the procedure with the given name, and supplies line numbers.
ENDEDIT	Gets the program out of EDIT mode.
MODIFY	Changes characters within a procedure line.

See also PROCEDURE.

-continued-

EXAMPLE

To edit the already defined procedure PRCDR, use LIST to see line numbers:

```
LIST PRCDR
1  PROCEDURE PRCDR(N)
2  GET(N)
3  BASELINE
4  HANNING
5  PAGE SHOW
6  ACCUM
7  RETURN
8  FINISH
```

Suppose you want to change HANNING to BOXCAR:

```
 EDIT PRCDR 4
:BOXCAR
:ENDEDIT
```

Now suppose you wanted to add a line to your procedure:

```
 EDIT PRCDR 1.5
:NBOX=3
:ENDEDIT
 LIST PRCDR
1  PROCEDURE PRCDR(N)
2  NBOX=3
3  GET(N)
4  BASELINE
5  BOXCAR
6  PAGE SHOW
7  ACCUM
8  RETURN
9  FINISH
```

You may also delete a line with:

```
 EDIT PRCDR 7     or       EDIT PRCDR 7
:*                         :ENDEDIT
:ENDEDIT
```

EL (Tucson only)

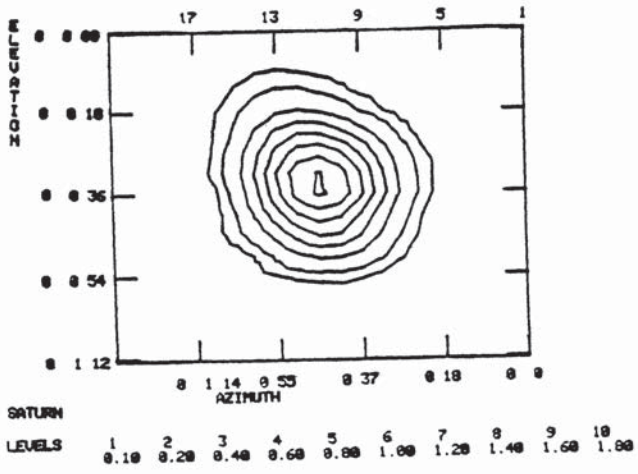
EL sets an internal flag that causes LABEL to identify elevation as the Y-coordinate and azimuth as the X-coordinate for maps. Would be used if the telescope were scanned along lines of constant elevation.

PTWH — Is neither used nor changed.

ADVERBS — None.

RELATED VERBS

AZ	Identified azimuth as the Y-coordinate for maps (Tucson only).
DEC	Identifies declination as the Y-coordinate for maps (Tucson only).
LABEL	Labels the map drawn by MAPSHOW.
MAPSHOW	Draws a map with the levels specified by LEVS.
RA	Identifies right ascension as the Y-coordinate for maps (Tucson only).



EMPTY

This routine will delete all scan numbers from the STACK, ASTACK, and set the STACK counter, ACOUNT, to zero.

PTWH — Is not used or changed.

ADVERBS

ACOUNT	The STACK counter, which is set to zero by EMPTY.
ASTACK	The STACK array, which is zeroed by EMPTY.

RELATED VERBS

A scan#	Inserts a single scan number into the STACK.
ADD (#,#)	Inserts a group of scan numbers into the STACK.
DELETE scan#	Deletes a scan number from the STACK.
STACK	Lists all scan numbers in the STACK.

EXAMPLE

You have just stacked a group of scans and now wish to clear the STACK. Type:

```
EMPTY
```

END

END is used to end a logical construction. It is used in conjunction with the verbs IF, FOR, and WHILE.

PTWH — It is not used or changed.

ADVERBS — None.

RELATED VERBS

See FOR
IF
WHILE

ENDEDIT

ENDEDIT is used in conjunction with EDIT. EDIT gets the program into EDIT mode; ENDEDIT returns the program to EXECUTE mode. The program signals that it is in EXECUTE mode by prompting with a caret.

PTWH — Is not used or changed.

ADVERBS — None.

RELATED VERBS

See EDIT

EQTOGAL

EQTOGAL will convert equatorial coordinates (epoch 1950.0) into galactic coordinates and print the results on the CRT screen.

PWTH — Is neither used nor changed.

ADVERBS — None.

OPERANDS

right ascension — Epoch 1950.0 right ascension in format HHMMSS.S.

declination — Epoch 1950.0 declination in format ±DDMMSS.S.

RELATED VERBS

GALTOEQ — Converts galactic coordinates to equatorial coordinates (epoch 1950.0) and prints the results on the screen.

REMARKS

The operands should follow EQTOGAL and be enclosed in parentheses, as in

EQTOGAL (HHMMSS.S, ±DDMMSS.S).

EXAMPLE

To convert the position of SS433, RA = $19^{h}09^{m}21.^{s}3$, DEC = $4^{o}53'53.1$, from equatorial to galactic coordinates, specify:

EQTOGAL (190921.3,45353.1)

the galactic coordinates are printed as,

39.694 -2.245

EXIT

EXIT is the routine that terminates the CONDAR program.

PTWH Is not used or changed.

ADVERBS None.

RELATED VERBS None.

REMARKS

Before you EXIT, take inventory of the items that will be lost when you EXIT. The contents of the three scan registers and the procedure definition space will be lost. To preserve any of the scan registers, save their contents in SAVE/RECALL bins. To preserve your procedure definitions and adverb values, use the STORE verb to make a copy of the procedure definition space in one of the six disk pages.

EXAMPLE

You have just finished your observing run and wish to take home a KEEP tape of reduced data. You must exit CONDAR in order to be able to run the KEEP program. To do so simply type:

```
EXIT
```

EXP

A built-in exponential function. Functions the same as the Fortran EXP.

PTWH — Neither used nor changed.

ADVERBS — None.

OBJECT

argument — The argument of EXP may be a constant, a variable, or an arithmetic expression, enclosed in parentheses.

RELATED VERB

LN — A built-in logarithm function.

EXAMPLE

```
PRINT EXP( 1.0 )
2.7183
```

EXPLAIN (Tucson only)

A routine which writes a page of documentation for the specified "resident procedure."

PTWH — Is neither used nor changed.

RELATED VERBS

HELP — A routine which prints documentation on verbs and adverbs.

REMARKS

Documentation only exits for resident procedures.

EXAMPLE

To find out what the procedure S does, type

```
EXPLAIN S
```

and you will get

S,S1 - - Procedures to retrieve the first (or only) receiver of the specified sequence from disk, erase the CRT screen and display the sequence on the CRT. S or S1 take one argument -- the desired sequence number which may be specified before or after S or S1.

Example: 500 S or S(500)
500 S1 or S1(500)

FETCH (Green Bank only)

FETCH is used to retrieve from disk a specified scan and place it into the WORK array, while also retrieving the scan whose scan number is that specified plus one, placing it into the TEMP array. FETCH requires one object, which is the scan number of the first scan to be retrieved.

PTWH — Not used, but is set to the TEMP array on exit.

ADVERBS — None.

OBJECT

Scan # — The number of the scan to be copied into the WORK array, where scan # +1 will be copied into the TEMP array.

RELATED VERBS

CGET feed # — gets the specifed feed of the most recently completed scan and places it in the WORK array.

GET scan # — gets the specified scan and places it in the WORK array.

ON scan # — gets the specified scan and places it in the WORK array.

OFF scan # — gets the specified scan and places it in the TEMP array.

EXAMPLE

To get scan 29301 into the WORK array and 29302 into the TEMP array and display them consecutively, you specify,

```
FETCH 29301 PMW PAGE SHOW          displays   29301
PMT  PAGE  SHOW                    displays   29302
```

FINISH

This verb is used to end the definition of a procedure. PROCEDURE puts the program into the COMPILE mode; FINISH puts the program back into EXECUTE mode. FINISH signals that the program has returned to EXECUTE mode by prompting with a caret.

PTWH — Is not used or changed.

ADVERBS — None.

RELATED VERBS

PROCEDURE — Begins the definition of a procedure.

FIX

FIX causes the verb SHOW to use the y-axis scaling that was last used before FIX was specified. Once FIX has been used, the y-axis scaling will remain fixed until FREEY or AUTO is specified.

PTWH — Is neither used nor changed.

ADVERBS

YMIN — Is set to -9.E10 by FIX. SHOW does not update its y-axis scaling if YMIN is less than -9.E09.

RELATED VERBS

AUTO — Alias for FREEY

FREEY — Sets YMIN to -90000. SHOW will automatically calculate the y-axis scaling if YMIN is between -9.E9 and -9999.

HOLDY — Alias for FIX.

RANGE — Sets YMIN and YINCR to span a range of y-coordinates specified as the arguments of RANGE.

EXAMPLE

See HOLDY.

FOFFSET (Green Bank only)

FOFFSET corrects the positions stored in the header of the currently referenced array for feed offsets with respect to the center of the receiver box. The feed offsets must be specified in the telescope control system at the time of the observations.

PTWH — Determines the currently referenced array. It is not changed.

ADVERBS — None

RELATED VERBS — None

REMARKS

FOFFSET should be executed only once after getting a scan since its effects are cumulative. Use this verb immediately after GET.

EXAMPLE

You are about to reduce scan 29301, specify

```
GET 29301 FOFFSET
```

if an offset feed was used, and feed offsets RHO and THETA for that feed were specified in the data taking program.

FOR

FOR begins a logical construction of the form

```
FOR (VARIABLE) = (ALPHA) TO (BETA) BY (GAMMA)
    (statements to be repeated for each value of (VARIABLE))
END
```

FOR can be used only in procedures. The FOR logical construction is a looping device similar to the iterative DO found in such languages as Fortran and PL/I.

(VARIABLE) can be any unused variable; it functions as an index for the loop. (ALPHA), (BETA), and (GAMMA) can be constants, variables, or arithmetic expressions. END is required to complete the construction. FOR loops may be nested and arranged on one or more lines; if arranged on one line, a semicolon must separate the statements to be iterated from the logical statements. BY (GAMMA) need not be included if (GAMMA) is to equal 1.

PTWH — Is not used or changed.

ADVERBS — None.

RELATED VERBS

END — Completes a logical construction.

WHILE — Begins a logical construction of the form:

```
WHILE (test condition)
     (statements to be iterated)
END
```

IF — Begins a logical construction of the form:

```
IF (test condition)
     THEN (do this)
     ELSE (do this)
END
```

See PROCEDURE

-continued-

EXAMPLES

(1) FOR construction on several lines in a procedure

```
PROCEDURE EG
READ N
FOR I=2 TO N
     PRINT SQRT(N)
END
FINISH
```

(2) FOR construction on one line in a procedure

```
PROCEDURE EGALT
READ N
FOR I=2 TO N; PRINT SQRT(N); END
FINISH
```

(3) FOR constructions nested in a procedure

```
PROCEDURE EG2
READ N, M
SUM=0
FOR I=1 TO N
    FOR J=1 TO M
        SUM=SUM+J
        END
    END
FINISH
```

FREEY

FREEY returns the control of the y-axis scaling to the verb SHOW. Automatic scaling is the default, so FREEY is used to restore automatic y-axis scaling after the user has overridden it by means of the verb HOLDY, or by setting the values of the adverbs YMIN and YINCR with the verb RANGE or with assignment statements.

PTWH — Is not used or changed.

ADVERBS

YMIN — The minimum temperature to appear on the graph. FREEY sets YMIN to -90000. SHOW sets its own y-axis scaling if YMIN is less than -9999 but greater than -9.E9.

RELATED VERBS

AUTO — Alias for FREEY.

FIX — Alias for HOLDY.

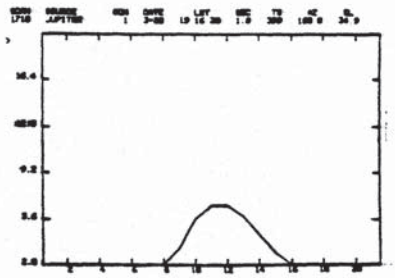
HOLDY — Causes SHOW to retain the last determined y-axis scaling.

RANGE — Defines the values of YMIN and YINCR according to the specified max and min values.

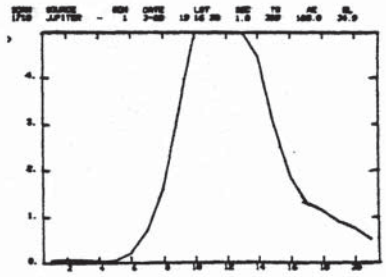
-continued-

EXAMPLE

Your last graph came out like this



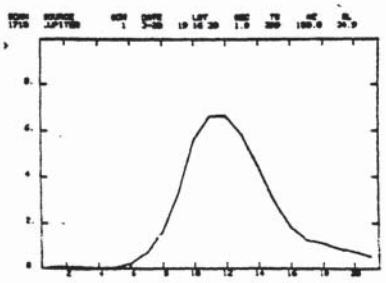
or this



To get a better graph, specify

FREEY PAGE SHOW

You will get



FULLGRID

This routine draws a rectangular grid over the current display on the CRT screen.

PTWH Is not used or changed by FULLGRID.

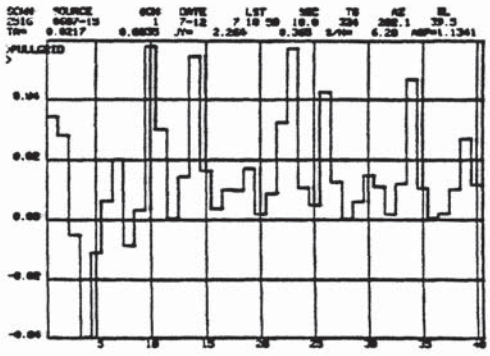
ADVERBS None.

EXAMPLE

In order to draw a rectangular grid on your current display specify

FULLGRID

and you will get



GAINCRV

GAINCRV can be used to scale the data points in a scan to take account of the changes of telescope gain with declination (Green Bank) or elevation (Tucson). The user must provide GAINCRV with a gain-declination, (gain-elevation), curve via the adverbs GCRV, GGRID and GMIN.

PTWH — Determines which array will be calibrated. It is not changed.

ADVERBS

GCRV(N) — A table of the gain calibration values spaced at equal steps of declination or elevation. The values could be given in units of flux density per Kelvin, relative gain, etc. and can contain up to twelve entries.

GGRID — The declination (elevation) spacing in decimal degrees between adjacent entries in the GCRV table.

GMIN — The minimum (elevation) declination of the scale factors in the gain table in decimal degrees.

RELATED VERBS

BIAS — Adds a constant to the data.

SCALE — Multiplies the data by a constant.

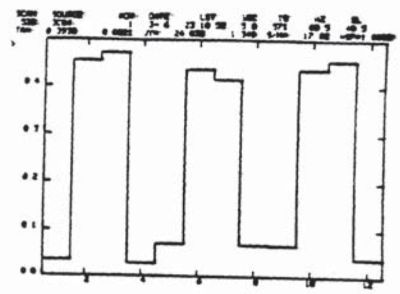
REMARKS

GAINCRV finds the scaling value via linear interpretation between points in GCRV adjacent to the observing declination (elevation). If the observing declination (elevation) is outside the range of GCRV, the data will not be scaled.

-continued-

EXAMPLE

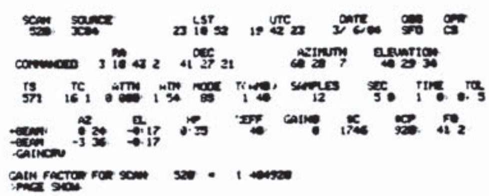
Your data looks like this:



You specify

```
GMIN = 10
GGRID = 5
GCRV = 1.1,1.15,1.2,1.25,1.3,1.35,1.4,1.45,1.6,1.65
HEADER
GAINCRV
```

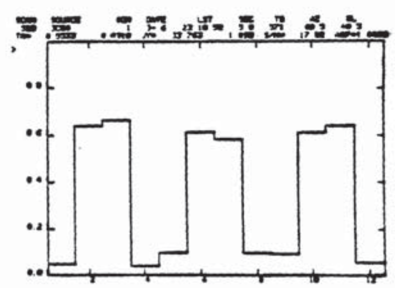
You will get



If you specify

```
PAGE SHOW
```

you will get



GALTOEQ

GALTOEQ will convert galactic coordinates into equatorial coordinates (epoch 1950.0) and print the results on the TEKTRONIX screen.

PTWH — Is neither used nor changed.

ADVERBS — None.

OPERANDS

longitude — Galactic longitude in decimal degrees.

latitude — Galactic latitude in decimal degrees.

RELATED VERBS

EQTOGAL — Converts equatorial coordinates (epoch 1950.0) to galactic coordinates and prints the results on the CRT screen.

REMARKS

The operands should follow GALTOEQ and be enclosed in parentheses, as in:

```
GALTOEQ(longitude,latitude)
```

EXAMPLE

To convert the position of SS433, l = 39°.694, b = -2°.245 from galactic to equatorial coordinates, specify

```
GALTOEQ (39.694, -2.245)
```

the equatorial coordinates are printed as

```
19  9  21.4    4 53 53.0
```

GAUSS

This routine fits up to six gaussian functions of the form

$$f(x) = HEIGHT * EXP (-2.772 * (x-CENTER)**2 / HWIDTH**2)$$

over a specified interval of a scan. The routine requires initial guesses of the center (explicit point number) and width (in points) at half the peak height of NGAUSS gaussians. The routine calculates initial guesses of the heights (in Kelvins) of the gaussians. Then the routine iterates the heights, center locations, and half-widths of the gaussians until a satisfactory fit is obtained. If the fit is successful, a caret is printed. If the fit is unsuccessful, the message FIT FAILED is printed.

PTWH — Determines for which array the fit is calculated. Is not changed.

ADVERBS

NGAUSS — The number of gaussians to be fit. NGAUSS must be between 1 and 6.

BGAUSS
EGAUSS — The explicit sample numbers where the fit is to begin and end.

CENTER(N) — The sample number where gaussian N has its center. Iterated by the routine.

HEIGHT(N) — The height in degrees Kelvin of gaussian N. First guess is set, and then iterated by the routine.

HWIDTH(N) — The width in number of points of gaussian N at half its peak height. Iterated by the routine.

NITER — The number of iterations which will be made to try to find a fit. The default value of NITER is 8.

FIXC — If FIXC is less than zero, GAUSS will not iterate CENTER.

FIXHW — If FIXHW is less than zero, GAUSS will not iterate HWIDTH.

-continued-

RELATED VERBS

GMODEL	Prints the gaussian parameters and the RMS error of the fit. Replaces the data in the currently referenced array with a model of the specified gaussian(s).
GPARTS	Displays each specified gaussian superimposed on the current plot.
RESIDUAL	Subtracts the sum of the specified gaussian(s) from the data in the currently referenced array.
PEAK	Finds a peak value and sets all the parameters for a single gaussian in preparation for GAUSS.

REMARKS

Any baseline offset should be removed before GAUSS is called. GAUSS is used to fit gaussians to data. With the exception of REGAUSS, the other gaussian verbs do not refine the parameters of the gaussian(s). GAUSS does not change the contents of any of the three scan arrays.

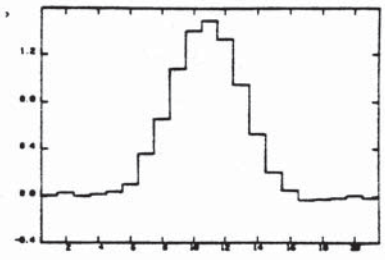
ERRORS

If the fit is unsuccessful after NITER iterations, the message FIT FAILED is printed.

EXAMPLE

Your data look like the display on the right. You specify:

```
NGAUSS=1; BGAUSS=4; EGAUSS=18;
CENTER=11; HWIDTH=6; GAUSS
```



All you get is a caret. To see the gaussian parameters, specify:

```
PRINT CENTER(1) HEIGHT(1) HWIDTH(1)
```

to get

```
10.82        1.521        5.141
```

(These refined values are in points and Kelvins.)

GET

GET is used to retrieve data from disk. It requires as its object a scan number. GET copies the indicated scan from the disk to the WORK array.

PTWH — Is not used, but is set to the WORK array.

ADVERBS — None.

OBJECT

scan # — The number of the scan to be copied from disk.

RELATED VERBS

CGET rcvr# — Copies the most recently completed scan for the specified receiver (Green Bank only).

ON scan# — Copies the requested scan into the WORK array.

OFF scan# — Copies the requested scan into the TEMP array.

TELL DATA — Lists the numbers of the continuum scans in the disk index (Tucson only).

TELL DISK — Lists the numbers of the scans in the disk index.

ERRORS

If scan # does not appear in the disk index, the message SCAN ? will be generated.

EXAMPLE

To bring a scan from the disk and display it on the screen, specify

```
GET 550 PAGE SHOW
```

where 550 is the number of the scan you want to see.

GMODEL

This routine evaluates the parameters of NGAUSS gaussians, and constructs the sum of the gaussians. The sum replaces the data in the currently referenced array. GMODEL also prints the value of the RMS error of the fit.

PTWH	Determines which array is replaced with the model. It is not changed.

ADVERBS

NGAUSS	The number of gaussians.
CENTER(N)	The center points of the gaussians.
HEIGHT(N)	The heights of the gaussians.
HWIDTH(N)	The half widths of the gaussians.

RELATED VERBS

GAUSS	Refines the user's first guesses of CENTER(N) and HWIDTH(N) for N gaussians. Determines HEIGHT for each gaussian.
GPARTS	Display each specified gaussian superimposed on the current plot.
RESIDUAL	Subtracts the sum of the specified gaussian(s) from the data in the currently referenced array.
PEAK	Finds a peak value and sets all the parameters for a single gaussian in preparation for GAUSS.

-continued-

REMARKS

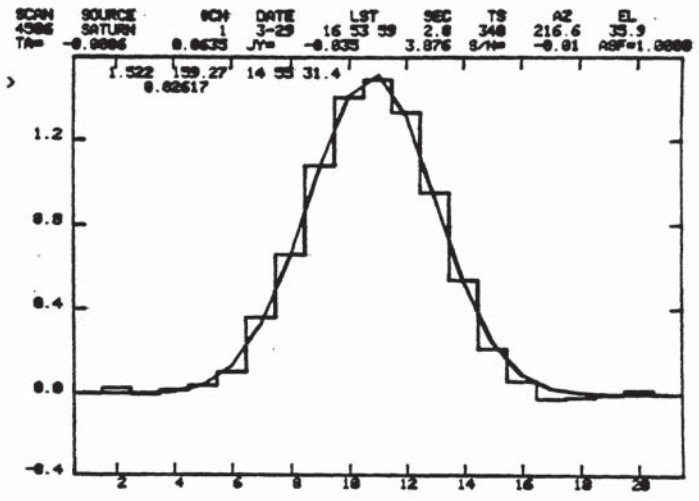
GMODEL is used after GAUSS to create a model of the fit, or by itself to create a particular gaussian or set of gaussians.

EXAMPLE

You fit a gaussian to your data. To display the model on top of the already displayed data, specify

TRT GMODEL RESHOW PMW

You will get



0.02617 is the RMS error of the fit in Kelvins

GPARTS

GPARTS evaluates the parameters of NGAUSS gaussians, and constructs and displays them separately. The data in the three main arrays are neither used nor affected.

PTWH — Is neither used nor changed.

ADVERBS

NGAUSS	The number of gaussians.
BGAUSS EGAUSS	The first and last points in the region within which the gaussians will be constructed and displayed.
CENTER(N)	The center points of the gaussians.
HEIGHT(N)	The heights of the gaussians.
HWIDTH(N)	The half widths of the gaussians.

RELATED VERBS

GAUSS	Refines the user's first guesses of CENTER(N) and HWIDTH(N) for N gaussians. Determines HEIGHT for each gaussian.
GMODEL	Evaluates the parameters of the gaussian(s), constructs a model of the gaussian (or sum of the gaussians), and replaces the data in the currently referenced array with the model. Also prints the parameters of the gaussians and the RMS error of the fit.
GPARTS	Displays each specified gaussian superimposed on the current plot.
RESIDUAL	Subtracts the sum of the specified gaussian(s) from the data in the currently referenced array.
PEAK	Finds a peak value and sets all the parameters for a single gaussian in preparation for GAUSS.

-continued-

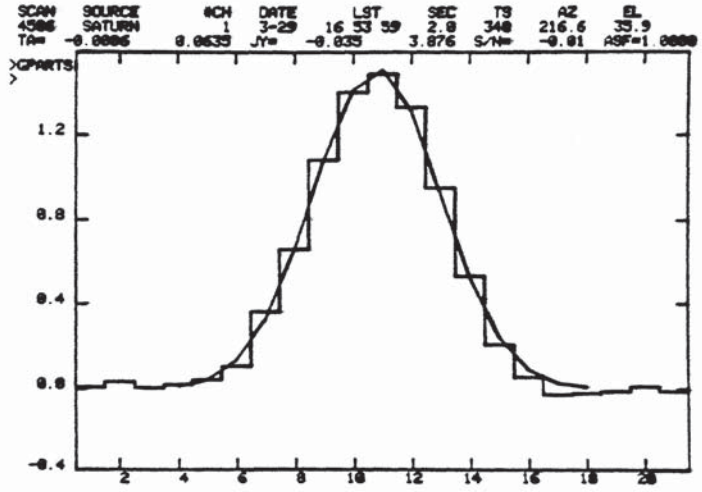
REMARKS

GPARTS is either used after GAUSS to create and display the fitted gaussians separately or by itself to create and display a particular gaussian or set of gaussians.

EXAMPLE

You have fitted a single gaussian to scan 4586. To display the gaussian within the area over which it was fitted specify,

GPARTS



GPUNCH (Green Bank only)

This routine, currently implemented only at the 300-foot telescope, copies the parameters of a single gaussian fit to a special disk file for copying to tape by a separate program. Scan number and source name accompany the source height in Kelvins, position in radians, half width in arcminutes, and rms of the fit. Data wind up on tape as 80 character card images, hence the reference to punched cards.

PTWH — Not used or changed.

ADVERBS — None.

RELATED VERBS

GAUSS — The gaussian fitting verb used to determine source height, position, and half width.

REMARKS

This routine is intended as a convenient method for transporting the results of continuum programs which observe a large number of compact sources, where the gaussian parameters completely describe the observations.

Along with the scan number, source name, and rms of the fit, the routine writes the values of HEIGHT(1), CENTER(1), and HWIDTH(1). If you would rather record some other result of the observation available in the analysis procedure, such as the level reading or the pointing offset, use assignment statements to put those values into HEIGHT, CENTER, or HWIDTH.

The external program for copying the disk file to tape can be run by EXITing CONDAR, mounting a tape, and entering:

```
$JOB
$GPUNCH
```

This program does not rewind the tape, so it is possible to put more than one file on a GPUNCH tape if you advance a number of file marks before invoking $GPUNCH.

Ask a programmer or the friend of the telescope for help in clearing the GPUNCH file before you begin to use it.

HANNING

HANNING smoothes the scan in the currently referenced array by the algorithm

$$x_i = \tfrac{1}{4} x_{i-1} + \tfrac{1}{2} x_i + \tfrac{1}{4} x_{i+1}$$

PTWH — Determines which scan is smoothed. Is not changed.

ADVERBS

BDROP(CH)
EDROP(CH) — The number of points at each end of each receiver channel which will be ignored by the routine.

RELATED VERBS

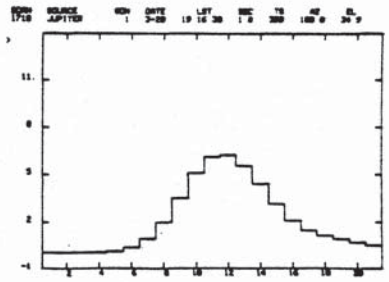
BOXCAR — Smoothes a scan by averaging a specified number of points together.

SMOOTH — Smoothes a scan by convolution with a user specified weighting function.

-continued-

EXAMPLE

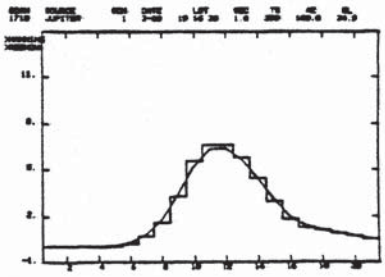
Your data looks like



To smooth it, specify

HANNING PAGE RESHOW

You will get



HEADER

HEADER is an informational verb which prints the header information for the scan in the currently referenced array.

PTWH — Determines which header is printed. It is not changed.

ADVERBS — None.

RELATED VERBS

TABLE — Prints the data values for the currently referenced array.

SNAP — Prints the header information for the scan in the currently referenced array on the line printer (Tucson only).

ERRORS

If there is no scan in the currently referenced array when HEADER is specified, the message LOAD SCAN! will be generated in Tucson, while in Green Bank the header information displayed will be all zeros.

-continued-

EXAMPLE

To print the header information for the 12-meter telescope scan 2516, specify

```
ON 2516
HEADER
```

You will get

```
  SCAN   SOURCE          LST        UTC        DATE     OBS   OPR
  2516   0607-15       7 18 50   19 23 50    7/12/82   HAD   CS

                 RA          DEC          AZIMUTH     ELEVATION
COMMANDED     6  8 51.6  -15 42 25       282  6 16      39 28 13

  TS     TC    ATTN   ATM  MODE  T(AMB)  SAMPLES    SEC    TIME   TOL
  324   12.0  0.000  1.57   BS   28.85       40     10.0    6:40   0: 5

            AZ      EL      HP     %EFF   GAIN@    @C     @CP     F@
+BEAM:      0:20   -4: 5   0:38     28       0    3165    3560   45.5
-BEAM:     -3:40   -4: 5
>
```

Similarly to print the header information for the 140-foot telescope scan 20751, specify:

```
GET 20751 HEADER
```

to produce the following:

```
 SCAN 20751.       198      LOCKMAN/KMJ  L188C   PMAP1

 DATE          EST          LST          IND POSITION        ROTAT FOCUS  CAL
 4 14 84    3 47  8.  16 58 39.  19 45 39.6     23 30  6.     90.0 450.8   NO

MPX    INT        CAL    CAL       NOISE     H RATE     DIR      NO. OF     SUB
CHN               VAL    FACT      TUBE      '/MIN      CODE     POINTS     SCAN
 8      0.40    306.0  0.01212     3.71     480.00       -1        675        1

          EPOCH (1950.00)          GALACTIC            DESCRIPTIVE          ZEN
          RA          DEC          L          B          H          V       DIST
    19 44 14.6   23 24 40.     59.97      -0.67      5 17      0.41      32 5

>
```

HELP

HELP is an informational verb with a number of uses:

1) when used by itself, it lists the various HELP options that are available (Tucson), or the verbs and symbols recognized by the program (Green Bank);
2) when used with the object VERB, it lists the verbs and symbols recognized by the program (adverbs not included);
3) when used with the object PROCEDURE, or PROC, it lists the names of the procedures currently defined and in memory;
4) when used with the objects ARRAY and SCALAR, it lists the arrays and scalars (adverbs) used by the program;
5) when used with a verb name as its object, it lists an informative description of the verb and lists the applicable adverbs with a description of each and its default value.

PTWH — Is neither used or changed.

ADVERBS — None.

OBJECTS

ARRAY	When used with HELP, generates a list of the arrays (most are adverbs) used by the program.
PROCEDURE or PROC	When used with HELP, generates a list of names of defined procedures.
SCALAR	When used with HELP, generates a list of the scalars (adverbs and variables) used by the program.

RELATED VERBS

EXPLAIN procedurename	Lists informative description of specified procedure (Tucson only).
LIST procedurename	Lists the procedure whose name is specified.
PRINT adverbname	Lists the value of the adverb whose name is specified. If the adverb does not exist, the message SYMBOL ? will appear.
? adverbname	Alias for PRINT.

-continued-

REMARKS

HELP generates lists of the symbols reserved by the program. These symbols cannot be used as procedure or variable names.

EXAMPLES

If you type HELP, the following output will appear on the screen:

```
HELP
 HELP -- DOCUMENTS ALL VERBS AND LISTS ALL PROCEDURES, ARRAY
         VARIABLES, SCALAR VARIABLES AND VERBS.
OBJECT                    DESCRIPTION
ARRAY          LISTS PROGRAM USER ARRAYS (MOST ARE ADVERBS).
PROCEDURE      LISTS NAMES OF THE DEFINED PROCEDURES.
SCALAR         LISTS PROGRAM USER SCALARS-ADVERBS & PROC VARIABLES
VERB           LISTS PROGRAM VERB NAMES.
VERBNAME       WHEN A VERBNAME IS SPECIFIED, DOCUMENTATION (WITH
               APPROPRIATE ADVERBS) IS PRINTED.
```

If you type HELP PROCEDURE before you have defined any of your own procedures, the following information will appear on the screen:

```
HELP PROC
S1          S2          S           ST          D           XX
HEAD        SNAPS       ELIM        ELIMS       PSTACK      C1
C2          CB          C           D1          SHOWS       SHOW1
SHOW2       LOG         LOGS        SLOG        CAL         F
FOCALIZE    TPTIP       M
FOR AN EXPLANATION OF A PROCEDURE TYPE  EXPLAIN PROCEDURENAME
FOR A LISTING OF A PROCEDURE      TYPE  LIST PROCEDURENAME
FOR A LISTING OF ADDITONAL PROCEDURES AVAILABLE TYPE  HELP INSTALL
```

The output from HELP array is:

```
HELP ARRAY
TWH         BDROP       EDROP       FACT        YMIN        YINCR
CLIP_MIN    CLIP_MAX    SIZE        VRMS        BMOMENT     EMOMENT
NREGION     BBASE       EBASE       XDATA       WORK        BREG
LEVS        CENTER      HEIGHT      HWIDTH      CMARK       SMWGT
WEIGHT      ASTACK      HG          CG          WG          TMARK
GCRV
```

-continued-

The output from HELP SCALAR is:

```
>HELP SCALAR
PTWH        NBOX        NFIT        NSAVE       Z_LINE      NMOMENT
SLABEL      SINCR       HP          SQUASH      BMARK       MRATIO
NGAUSS      BGAUSS      EGAUSS      NITER       FIXC        FIXHW
DISK        CSTACK      DATA        KSCANS      SCANS       SKIP
ETA         FTM         FTSBR       ACOUNT      SHIFT       TRX
OFFSET      SDMAP       DCPCT       SUMT        SUMT2       NPOINTS
GMIN        GGRID       XSCAN       V1          V2          OSET
I           SM          RM          SUMM        RSUM        BSCAN
ESCAN       XS          NAX         MAX         F0          WL
TX          OAX         FOC         MPT
```

To find out what the verb GAUSS does, type HELP GAUSS and get this output:

```
>HELP GAUSS
 GAUSS — FITS UP TO 6 GAUSSIAN FUNCTIONS OVER A SPECIFIED
         INTERVAL OF A SCAN. GAUSS REQUIRES INITIAL GUESSES
         OF CENTER(PT#) & WIDTH(POINTS) AT HALF THE PEAK HEIGHT
         OF NGAUSS GAUSSIANS. GAUSS CALCULATES THE INITIAL
         GUESSES OF THE HEIGHTS(KELVINS). IT THEN ITERATES THE
         HEIGHTS, CENTER LOCATIONS AND HALF-WIDTHS UNTIL A FIT
         IS OBTAINED. A CARET IS PRINTED IF IT FITS.
IV  ADVERBS          DESCRIPTION
 1  NGAUSS          # OF GAUSSIANS (1-6) TO BE FIT.
 0  BGAUSS          EXPLICIT POINT #'S WHERE FIT IS TO BEGIN
 0  EGAUSS          & END.
 0  CENTER(N)       POINT # WHERE GAUSSIAN N HAS ITS CENTER.
 0  HEIGHT(N)       HEIGHT IN DEGREES KELVIN OF GAUSSIAN N.
 0  HWIDTH(N)       HALFWIDTH IN # OF POINTS OF GAUSSIAN N.
 8  NITER           # ITERATIONS MADE TO TRY TO FIND A FIT.
 1  FIXC            IF <0 GAUSS WILL NOT ITERATE CENTER.
 1  FIXHW           IF <0 GAUSS WILL NOT ITERATE HWIDTH.
>
```

HISTOGRAM

A routine which sets the plotting mode for the SHOW display to histogram.

PTWH Is not changed or used.

ADVERBS None.

RELATED VERBS

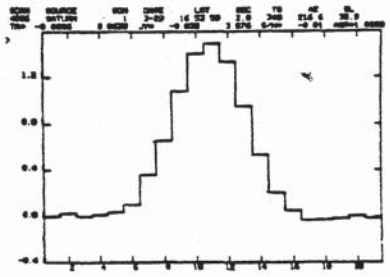
LINE	Causes SHOW to display data as a continuous line.
POINTS	Sets the SHOW display mode to points.
SHOW	Plots the contents of the currently referenced array on the CRT screen.

EXAMPLE

Your last display was in LINE mode. In order to redisplay the data in HISTOGRAM mode, specify

```
HISTOGRAM PAGE SHOW
```

and you will get



HOLDY

HOLDY causes the verb SHOW to use the y-axis scaling that was last used before HOLDY was specified. Once HOLDY has been used, the y-axis scaling will remain fixed until FREEY or AUTO is specified or the adverb YMIN is set to a number less than or equal to -9999, which restores automatic scaling, or until a new scaling is specified with the verb RANGE or by setting YMIN and YINCR with assignment statements.

PTWH — Is not used or changed.

ADVERBS

YMIN — The minimum temperature to appear on the graph. HOLDY sets YMIN to -9.E10. SHOW sets its own y-axis scaling if YMIN is less than -9999 but greater than -9.E9.

RELATED VERBS

FREEY — Sets YMIN to -90000. SHOW sets its own y-axis scaling if YMIN is less than -9999 but greater than -9.E9.

AUTO — Alias for FREEY.

RANGE — Defines the values of YMIN and YINCR according to the specified max and min values.

-continued-

EXAMPLE

You want to compare four scans of different sources. To do this most easily, you want to get all the scans plotted to the same scale. You can accomplish this by

```
GET 520 PAGE SHOW
HOLDY
GET 536 PAGE SHOW
GET 552 PAGE SHOW
GET 510 PAGE SHOW
FREEY
```

IF

IF begins a logical construction of the form

```
IF (test condition)
     THEN (do this)
     ELSE (do this)
END
```

IF can be used only in procedures. The IF logical construction is a conditional device similar to the IF-THEN-ELSE construction found in PL/I.

(test condition) can be any expression which has a single true or false result. The (do this) parts may each consist of many statements. The first (do this) will be executed if the test condition is true; the second (do this) will be executed if the test condition is false. END is required to complete the construction. IF statements may be nested and arranged on one or more lines. If they appear on the same line, the IF, THEN, ELSE, and END clauses must be separated by semicolons. ELSE (do this) need not be included when no action is desired if (test condition) is false.

PTWH — Is not used or changed.

ADVERBS — None.

RELATED VERBS

END — Completes a logical construction.

FOR — Begins a logical construction of the form:

```
FOR (VARIABLE) = (ALPHA) TO (BETA) BY (GAMMA)
     (statements to be repeated)
END
```

WHILE — Begins a logical construction of the form:

```
WHILE (test condition)
     (statements to be iterated)
END
```

see PROCEDURE

-continued-

EXAMPLES

(1) IF statement on several lines in a procedure

```
PROCEDURE EG
READ X1,X2
IF X1>X2
   THEN PRINT X1
   ELSE PRINT X2
   END
FINISH
```

(2) IF statement on one line in a procedure

```
PROCEDURE EGALT
READ X1,X2
IF X1>X2; THEN PRINTX1; ELSE PRINT X2; END
FINISH
```

(3) IF statements nested in a procedure

```
PROCEDURE EG2
READ X1, X2, X3
IF X1>X2; THEN PRINT X1
        ELSE IF X2>X3; THEN PRINT X2
                       ELSE PRINT X3
             END
        END
FINISH
```

INSTALL (Tucson only)

A routine which reads procedure files (name.PRC) from disk into CONDR1 or CONDR2. The files are written with the text editor and remain in the system indefinitely.

PTWH Is neither used nor changed.

ADVERBS None.

RELATED VERBS None.

EXAMPLE

You have defined the procedure file DISPLAY.PRC with the text editor. To load it into the program, type

```
INSTALL DISPLAY
```

INVERT

INVERT flips the data end-for-end in the currently referenced array. The data on the disk is not flipped. It may be necessary to use INVERT if you have scans across the source in a "negative" direction (i.e., scan rate negative). If you are mapping, you take some scans in the opposite direction, so you will have to use INVERT to flip the negative direction scans before using MAPSHOW (not in Green Bank, however).

PTWH — Determines the scan to be flipped end-for-end. It is not changed.

ADVERBS — None.

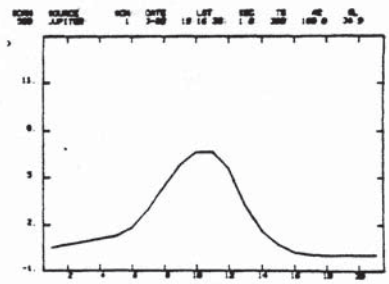
RELATED VERBS — None.

REMARKS

If you decide INVERT should not have been invoked, a second INVERT will restore the status quo!

EXAMPLE

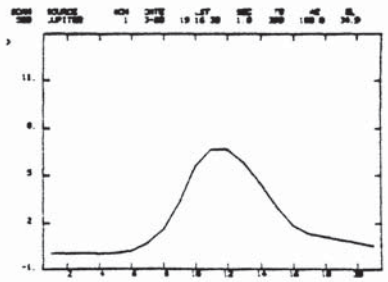
The scan in the currently referenced array looks like the display to the right. You have decided that the scan crossed the source the "wrong" way, and want to see how the scan would look if the scanning direction were reversed.



Specify:

```
INVERT PAGE SHOW
```

to produce the display to the right.



KEEP

KEEP copies the scan in the currently referenced array into a special file on disk. A separate program, external to CONDAR, will then copy data in this file to tape. This is useful in taking home reduced data. The external programs in Tucson are called KC1RIT and KC2RIT, while the one in Green Bank is $KEEP.

PTWH — Determines which array will be copied to disk. Is not changed.

ADVERBS — None.

RELATED VERBS

RECALL — Copies the scan in the bin NSAVE into the currently referenced array.

SAVE — Copies the currently referenced array into the disk bin NSAVE.

REMARKS

Please see the programmer for information on the data tape format and how to read the tape.

EXAMPLE

You have stacked sixty scans and wish to carry home the result to do further processing at your home institution. Type

```
KEEP
```

At the end of your run, exit CONDAR and ask the operator to make you a KEEP tape. The standard format of Tucson KEEP tapes is ASCII FITS. The standard format of Green Bank KEEP tapes is the same as the analysis disk format, in MODCOMP binary representation.

LABEL

LABEL is used to complete a contour map begun by MAPSHOW. LABEL draws and labels the axes of the contour map and labels the contour levels.

PTWH	Determines which array's contents are mapped. It is not changed.

ADVERBS

BDROP(CH) EDROP(CH)	The number of points at each end of each receiver channel which are to be ignored by the routine.
LEVS(M)	The levels at which the contours were drawn by MAPSHOW. Up to 20 levels may be specified.
MRATIO	is used to vary the relative X-Y scaling of the map made by MAPSHOW.
SQMAP	(Tuc only). If set to 1, a map with equal number of X and Y points will not overflow the screen. Default value is 0.

RELATED VERBS

MAPSHOW	Draws a contour map with the labels specified by LEVS.
RA	Identifies right ascension as the Y-coordinate for the map.
DEC	Identifies declination as the Y-coordinate for the map.
AZ	Identifies azimuth as the Y-coordinate for the map (Tuc only).
EL	Identifies elevation as the Y-coordinate for the map (Tuc only).

REMARKS

LABEL should always be called after MAPSHOW.

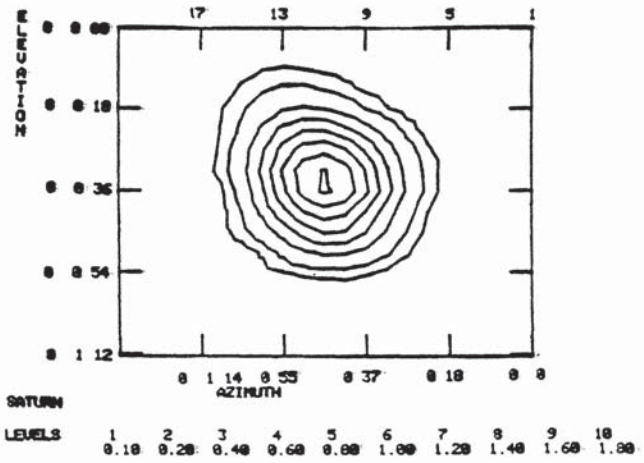
-continued-

EXAMPLE

Contour map before labeling (see MAPSHOW):

Same map after LABEL has been specified:

(point numbers)



LIMIT

This routine will restrict data processing to only those points within the minimum and maximum sample numbers specified.

PTWH — Is not used or changed by LIMIT.

ADVERBS — None.

OBJECTS — The minimum and maximum point numbers desired.

RELATED VERBS — None.

REMARKS

LIMIT restricts the processing of data by setting the appropriate values of BDROP and EDROP. To return to the original limits, reset BDROP and EDROP to zero.

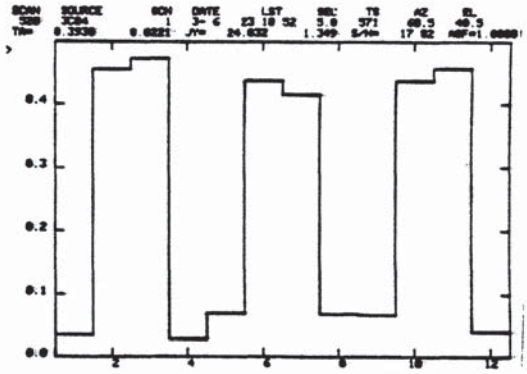
ERRORS

If LIMIT is specified without minimum and maximum values, the message "ARG LIST?" will be printed.

-continued-

EXAMPLE

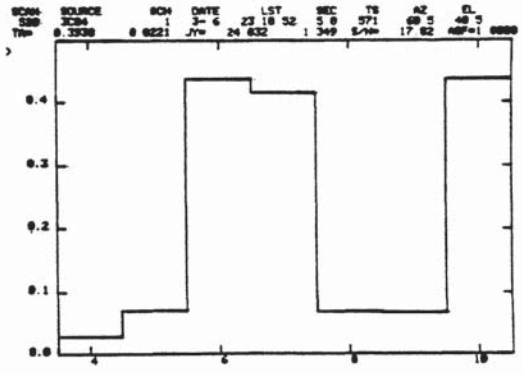
Your current display looks like the scan to the right. You would like to expand the horizontal scale by looking only at points 4 through 10.



Use LIMIT to restrict the plotting to the desired points:

```
4 10 LIMIT
PAGE SHOW
```

to produce the display to the right.



LINE

A routine which sets the plotting mode for the SHOW display to a continuous line.

PTWH — Is not changed or used.

ADVERBS — None.

RELATED VERBS

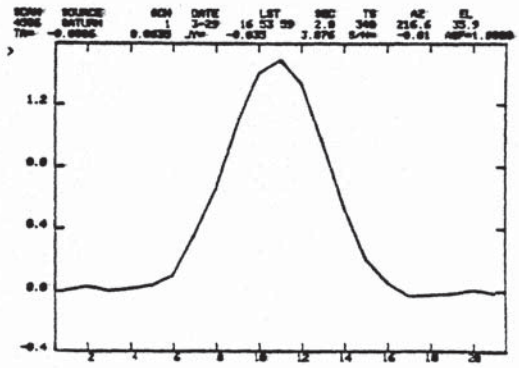
HISTOGRAM	Sets the SHOW display mode to a histogram.
POINTS	Sets the SHOW display mode to points.
SHOW	Plots the contents of the currently referenced array on the CRT screen.

EXAMPLE

Your last display was a HISTOGRAM. You want to change it to LINE. Specify

```
LINE PAGE SHOW
```

and you will get



LIST

LIST is an informational verb used to produce a listing of a defined procedure. LIST requires as its object the name of the procedure to be listed.

PTWH — Is not used or changed.

ADVERB — None.

OBJECTS

Procedurename	The name of the procedure to be listed.

RELATED VERBS

HELP PROCEDURE	Lists the names of the defined procedures.
PROCEDURE	Defines a procedure.

REMARKS

LIST numbers the lines of the procedures it lists. These line numbers should be used when editing the procedures.

EXAMPLE

You have defined the procedure DUMMY. To list it, specify

```
LIST DUMMY
```

You will get

```
1    PROCEDURE DUMMY
2    READ AA
3    PRINT AA
4    FINISH
```

LN

A built-in natural logarithmic function. Functions the same as the Fortran LOG.

PTWH — Neither used nor changed.

ADVERBS — None.

OBJECT

argument — The argument of LN may be a constant, a variable, or an arithmetic expression, enclosed in parentheses.

RELATED VERB

EXP — A built-in exponential function.

EXAMPLE

```
PRINT LN( 2.7183 )
1.0000
```

LOAD (Green Bank only)

This routine will read scans from a 9-track KEEP tape and write them to the analysis disk file for later processing.

PTWH — It is not used or changed.

ADVERBS

BRANGE, ERANGE — The minimum and maximum right ascension in hours (1950) of scans to be loaded.

BSCAN, ESCAN — The minimum and maximum scan numbers to be loaded.

TFMT — Tape format indicator. Use TFMT=1024 for MODCOMP produced KEEP tapes, and TFMT = 360 for IBM produced tapes.

RELATED VERBS

KEEP — Writes a scan to a disk keep file which can be copied to tape.

WIPE — Removes all of the scans belonging to a specified user from the disk index (GB only).

REMARKS

LOAD should not be used at the telescope when the telescope foreground task is running (normal case). To use LOAD at the telescope, first suspend data taking and abort the foreground task:

```
console interrupt
/TSC/A
```

Then use LOAD to read your KEEP tape. Before resuming data taking, reactivate the foreground task:

```
console interrupt
/TSC/ACT
```

-continued-

LOAD searches the entire tape for scans in the range of scan numbers from BSCAN to ESCAN, and the range of right ascensions from BRANGE to ERANGE. BRANGE must be less than ERANGE, so to load sources in the range 20 hours to 04 hours, first LOAD 20 to 24 hours and then LOAD 0 to 4 hours.

LOAD begins loading scans onto disk at a point immediately following the last scan previously written to disk. If the scan table or the disk file becomes full, LOAD will begin over-writing scans from the beginning of the scan table until the last scan on the tape has been loaded.

EXAMPLE

You have made a keep tape of partially reduced scans and would like to process them further on the Jansky Lab MODCOMP while someone else is observing at the telescope. After starting the program and threading the tape on the tape drive, type:

```
TFMT=1024;LOAD
```

and all of the scans on the tape will be loaded.

An alternative format allows the Charlottesville system to read KEEP tapes produced by the IBM program:

```
TFMT=360;LOAD
```

MAKECOPY

This command causes a hard copy to be made from the Tektronix CRT screen.

PTWH Is not used or changed by MAKECOPY.

ADVERBS None.

RELATED VERBS None.

REMARKS

This verb can be used to make automatic hardcopies within a procedure.

EXAMPLE

You would like a hard copy of the data on the screen. You can type MAKECOPY or strike the MAKECOPY button on the terminal.

MAPSHOW

This routine draws a position vs. position contour map of a set of scans (more than one). The first use of MAPSHOW after a LABEL (or program restart) copies the scan in the currently referenced array into the HOLD array. Subsequent MAPSHOW's compare the scan in the currently referenced array to the scan in the HOLD array, draw the contours, and move the scan in the currently referenced array into the HOLD array.

PTWH	Determines which scan is contoured. Is not changed.

ADVERBS

BDROP(CH) EDROP(CH)	The number of points at each end of each receiver channel that are ignored by the routine.
LEVS(N)	The levels at which a contour line is to be drawn. Up to 20 levels may be specified. If less than twenty levels are specified, the first level beyond the desired levels should be set equal to -9999.
MRATIO	Is used to vary the relative X-Y scaling of the map made by MAPSHOW.
SQMAP	(Tuc only). If set to 1, guarantees that a map with equal number X and Y points will not overflow the screen. Default value is zero.

RELATED VERBS

LABEL	Draws and labels the axes and labels the contours the map made by MAPSHOW. Used after all the scans have been mapped.
RA	Selects right ascension as the Y-coordinate of the map (Tuc only).
DEC	Selects declination as the Y-coordinate of the map (Tuc only).
AZ	Labels the X-coordinate of the map as azimuth (Tuc only).
EL	Labels the Y-coordinate of the map as the elevation (Tuc only).

-continued-

REMARKS

MAPSHOW contours only one scan at a time, so it is often used in a procedure. Note that you cannot map by recalling data into the HOLD array!

EXAMPLE

We need to produce a contour map of 17 scans, each 21 points long, that already have had baselines removed. The scans are stored in NSAVE bin no. 1 to 17. We wish to draw contour levels at 0.1, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8 Kelvin, specify

```
LEVS = 0.1, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, -9999
```

to set the levels, and

```
SQMAP = 1       (Tucson only; in Green Bank, see what you get and
MRATIO = 1                adjust MRATIO accordingly)
```

Then enter

```
 PROCEDURE MAP (NN)
:PMW
:FOR I = 1 TO NN
:NSAVE = I;  RECALL MAPSHOW
:END
:FINISH
 PAGE MAP (17)
```

and you will have something like this:

Use LABEL verb to label the completed contour map.

MDBASE (Green Bank only)

MDBASE performs a nonlinear baseline removal in which the value at each position is reduced by the median intensity over the set of NBOX points centered on that position. The result is a high pass filter since the median value depends weakly or not at all on the presence of infrequent spikes in the data.

PTWH — Determines which array contains the scan to be processed. It is not changed.

ADVERB

NBOX — The total number of points whose intensities are sorted to find a median value. The median value is then subtracted from the value at the position in the center of the box.

NBOX must be odd. If it is entered as even, the error message "NBOX ODD?" is printed, and no baselining is performed.

RELATED VERBS

BASELINE — Computes the coefficients for a polynomial, evaluates the polynomial and subtracts it from the currently referenced array. See the dictionary entry for BASELINE for additional related verbs.

DCBASE — Computes the average value of the data for a specified range of points and subtracts this from all points in the scan.

PCBASE — Creates a zero baseline such that a given percentage of the data is negative.

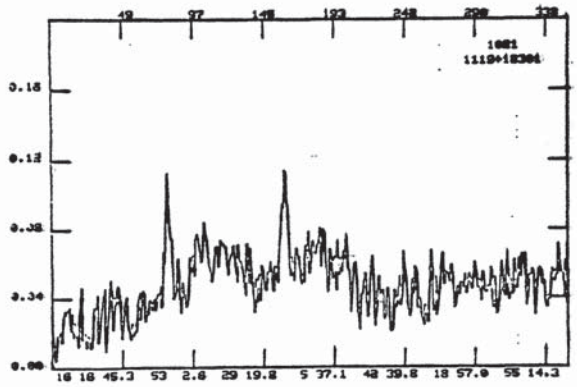
-continued-

REMARKS

MDBASE can be used when one is studying small scale structure against large scale background variations, such as discrete sources observed against the galactic plane. Alternatively, MDBASE can be used to isolate and remove interference spikes due to radar or lightning. One should experiment with values for NBOX to obtain the best results.

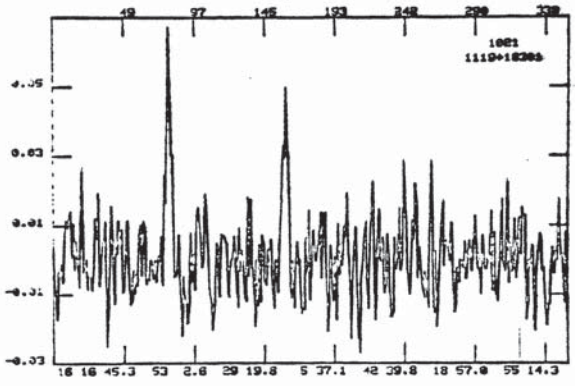
EXAMPLE

Suppose you had observed the scan illustrated, and were interested only in the two narrow sources, and not in the extended emission.



Set NBOX to a number that is wide in comparison with the structure of interest:

```
NBOX=41
MDBASE PAGE SHOW
```



MODIFY (Green Bank only)

MODIFY is an editing routine which allows you to change a line of a procedure by typing underneath the characters you want to change, the new characters you want. The cursor is positioned with the space bar and the backspace key. MODIFY prompts you for input with a question mark rather than the colon of the EDIT command.

OBJECTS

procedurename	The name of the procedure to be modified.
linenumber	The number of the line to be change.

RELATED COMMANDS

EDIT procedurename linenumber	Used to change lines, insert lines, and delete lines in a procedure.
LIST procedurename	Used to list a procedure.

REMARKS

Modify prints the line you wish to change preceded by a prompt character. The first character position to the right of the prompt character is the first character of the line. This could be a blank character. Next MODIFY does a carriage return, line feed, and prints the prompt character again. The cursor is left under the first character of the line above.

The spacebar and the backspace key do not affect the line being modified. They are only used to position the cursor underneath a character to be changed. When you backspace to a character, any changes you may have made to the right of the cursor are deleted. You will have to re-type them.

MODIFY uses two special characters. The @ symbol is used to replace the character above it with a space. The $ used to delete the character above it.

You can "escape" from MODIFY by typing a blank line. A blank line is a line with at least one space or blank and no other type of character on the line, including the special characters. If you type a null line (nothing on it), you will be prompted again.

-continued-

You cannot use MODIFY to insert a new line into a procedure. If you use a line number with a fractional part, the fraction will be truncated. You should not use MODIFY to delete a line.

Most errors detected by MODIFY will cause you to be re-prompted to change the line again.

EXAMPLE

To change a line of a procedure, first list it

```
 LIST PRCDR
1    PROCEDURE PRCDR(S)
2    GET(S) BASELINE NBOX=3; BOXCAR
3    PAGE SHOW ACCUM
4    RETURN
5    FINISH
```

to change NBOX=3 to NBOX=5, type

```
                          MODIFY PRCDR 2
MODIFY types             ?GET(S) BASELINE NBOX=3; BOXCAR
you type (after "?")     ?                        5
MODIFY then types         GET(S) BASELINE NBOX=5; BOXCAR
```

and changes the procedure in the procedure definition space.

To change BOXCAR to HANNING

```
                          MODIFY PRCDR 2
MODIFY types             ?GET(S) BASELINE NBOX=5; BOXCAR
you type (after "?")     ?                HANNING$$$$$$$
MODIFY then types         GET(S) BASELINE HANNING
```

If you try to delete a line with MODIFY:

```
                          MODIFY PRCDR 4
MODIFY types             ?RETURN
you type (after "?")     ?$$$$$$
MODIFY then types   (blank line)
```

```
you end up with           LIST PRCDR
                         1    PRCDR(S)
                         2    GET(S) BASELINE HANNING
                         3    PAGE SHOW ACCUM
                         4
                         5    FINISH
```

MOMENT

A routine which calculates the moment specified by the adverb NMOMENT over a specified range of points and stores the result in the variable SIZE. Currently the verb will only compute the answer for NMOMENT = 0.

PTWH	Determines for which scan the moment is calculated. Is not changed.

ADVERBS

NMOMENT	The order of the moment to be calculated, as follows: 0 gives AREA (Kelvins)
BMOMENT(CH) EMOMENT(CH)	The first and last points of the region over which the moment is to be calculated.

REMARKS

The moment calculated by MOMENT is stored in the variable SIZE, which can be printed by the command

```
PRINT SIZE
```

One value is printed for each receiver channel.

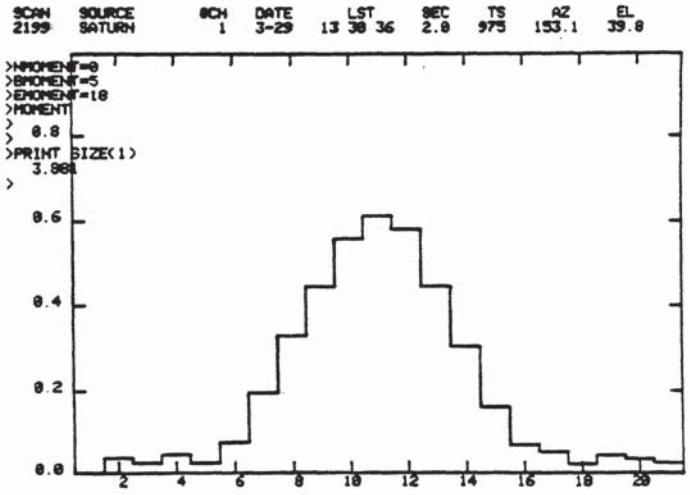
-continued-

EXAMPLE

You want to calculate the area under your data.

You specify:

```
NMOMENT = 0
EMOMENT = 18
BMOMENT = 5
MOMENT
PRINT SIZE(1)
```



OFF

OFF copies the indicated scan from the disk to the TEMP array. OFF requires a scan number as its object.

PTWH — Is not used, but is set to the TEMP array.

ADVERBS — None.

OBJECTS

scan # — The number of the scan to be copied into the TEMP array.

RELATED VERBS

ON scan # — Copies the indicated scan into the WORK array.

REMARKS

OFF is commonly used in conjunction with ON to pair up off and on scans so that they can be differenced to remove baselines.

ERRORS

If the scan # indicated is not in the disk index, the message SCAN ? will be generated.

EXAMPLE

To difference 4588 and offscan 4575 and display the result, specify

```
OFF 4575; ON 4588;  DIFF PAGE SHOW
```

ON

ON copies the indicated scan from the disk to the WORK array. ON requires a scan number as its object.

PTWH — Is not used, but is set to the WORK array.

ADVERBS — None.

OBJECTS

scan# — The number of the scan to be copied into the WORK array.

RELATED VERBS

OFF scan# — Copies the indicated scan into the TEMP array.

REMARKS

ON is commonly used in conjunction with OFF to pair up on and off scans so that they can be differenced to remove a baseline.

ERRORS

If the scan # indicated is not in the disk index, the message SCAN ? will be generated.

EXAMPLE

To difference onscan 4588 and offscan 4575 and display the result, specify

```
OFF 4575;  ON 4588;  DIFF PAGE SHOW
```

ONE (Tucson only)

ONE sets the CONDAR processing mode to single channel.

PTWH — Is not used or changed.

ADVERBS

SINCR — The scan increment is set to one.

RELATED VERBS

ADD (#,#) — Inserts a group of scans into the STACK with SINCR as the increment between scans.

TWO — Sets the processing mode to dual channel (SINCR = 2).

REMARKS

The five point fitting procedure, F, also uses SINCR to determine what scans are to be included in the STACK for the five point. The LOGS procedure uses SINCR as the increment between scans listed.

EXAMPLE

You have just switched to single channel continuum and want to fit the five point (center scan = 702) that just finished. You specify

```
ONE
702 F
```

ONS (Tucson only)

ONS marks every on sample of an on-off scan with a '+".

PTWH Is not used or changed.

ADVERBS None.

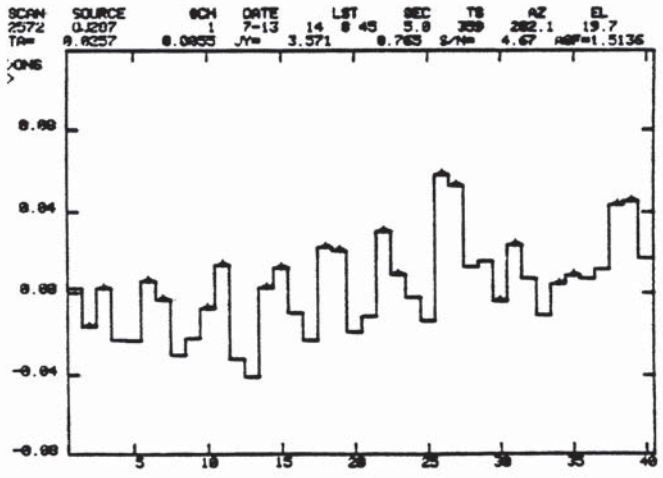
RELATED VERBS None.

EXAMPLE

You want to mark your on samples for the current scan. You specify

ONS

and get



PAGE

PAGE erases the CRT screen and returns a caret at the top of the now blank screen. PAGE is equivalent to striking the PAGE ERASE button except that the PAGE ERASE button does not respond with a caret.

PTWH — Is not used or changed.

ADVERBS — None.

REMARKS

If you hit the PAGE ERASE button instead of entering the command PAGE, you will not get a caret. To be certain that the program is ready to accept a command, you can strike RETURN to get a caret.

N.B. The verb PAGE is preferred over striking the PAGE ERASE button because PAGE also resets the relative origin of the CRT screen.

PCBASE

PCBASE subtracts a constant baseline from the data of the currently referenced array so that a specified percentage of data points become negative.

PTWH — Determines from which array the baseline will be subtracted. It is not changed.

ADVERBS

DCPCT — The percentage of data points that will be negative after baselining.

RELATED VERBS

BASELINE — Computes the coefficients for a polynomial baseline, evaluates the polynomial and subtracts it from the currently referenced array. See the dictionary entry for BASELINE for additional related verbs.

DCBASE — Computes an average value in a specified range and subtracts it from the currently referenced array.

MDBASE — Computes median values over a set of data points centered on each point in a scan and subtracts the medians from the data (Green Bank only).

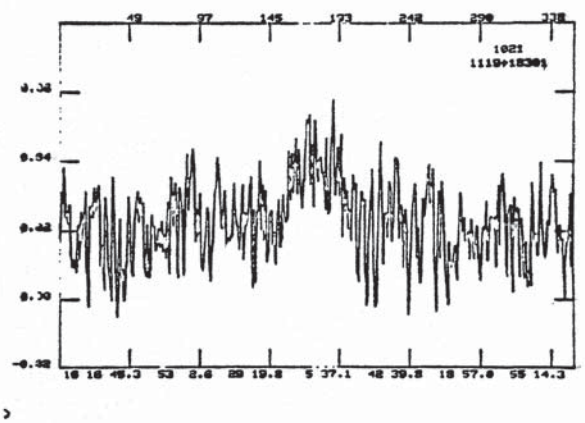
REMARKS

PCBASE searches for the value which is greater than or equal to DCPCT percent of the data points. This value is subtracted from all of the data points. There is no windowing in the routine.

-continued-

EXAMPLE

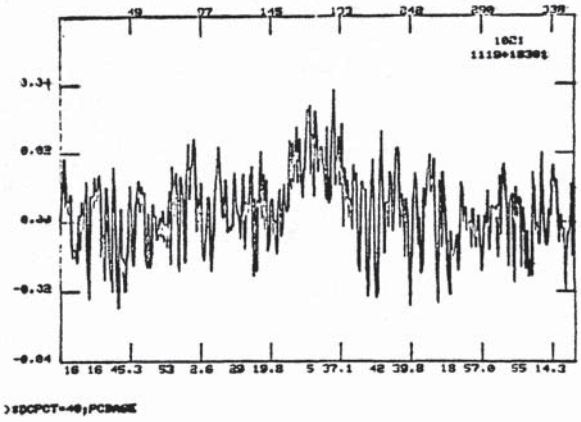
Suppose your data looks like the scan on the right, where there does not appear to be a slope in the baseline, only an offset.



>

You want to subtract a baseline so that 40 percent of the data points become negative. Specify:

```
DCPCT = 40
PCBASE
PAGE SHOW
```



PDOC (Tucson only)

PDOC is a verb used by the five point procedure, F, to display the five points, the gaussian fit, updated pointing corrections and peak flux densities.

PTWH — Determines which array is used for the documentation information.

ADVERBS — None.

RELATED VERBS — None.

REMARKS

This verb is used by the F procedure and should not be used outside of F.

EXAMPLE

A typical five point display is shown below

```
SCAN   #CH    DATE      LST        SOURCE        SEC   TIME   TS
X:29    1   ;-13-82   10 43 11   JUPITER        5.0   0:40 1020.

                          AZ          EL
TELESCOPE COMMANDED:  124 37 44   27 10 19      TAMB=31.21
          MAIN OFF:     -4: 0       0: 0       ELAM=26.47
          REF OFF:       4: 0       0: 0       HP =105.00

                                     +
                                   87.68
                                    1.73

                                     A

                          +          +          +
                        52.86     118.63     69.36
                         0.48       1.46      0.27


                                     +
                                   30.94
                                    1.59

FIT WITH AZBM=ELBM=210   (SYMBOL=F)                CORRECTED OFFSETS
DAZ= 11.0  PEAK(JY)= 120.83                            AZ     EL
DEL= 30.1  PEAK(JY)= 125.85                  MAIN:- 3:49 + 0:30
                                              REF:+ 4:11 + 0:30
FIT WITH NO RESTRICTIONS  (SYMBOL=M)
DAZ= 11.0  PEAK(JY)= 119.52  BM=212.8
DEL= 30.0  PEAK(JY)= 125.63  BM=287.8
```

PEAK

PEAK finds the CENTER, HWIDTH and HEIGHT of a single gaussian. Prints CENTER point, HWIDTH and PEAK temperature. Sets BGAUSS and EGAUSS for the GAUSS verb.

PTWH — Determines which array is to be used.

ADVERBS — Sets CENTER(1), HWIDTH(1), HEIGHT(1), BGAUSS and EGAUSS.
BGAUSS is set to (CENTER(1) - HWIDTH(1))
EGAUSS is set to (CENTER(1) + HWIDTH(1))

RELATED VERBS

GAUSS — Refines the first guesses of CENTER(N), HWIDTH(N), and HEIGHT(N) for N gaussians.

GMODEL — Evaluates the parameters of the gaussian(s), constructs a model of the gaussian (or the sum of the gaussians), and replaces the data in the currently referenced array with the model. Also prints the value of the RMS error of the fit.

-continued-

EXAMPLE

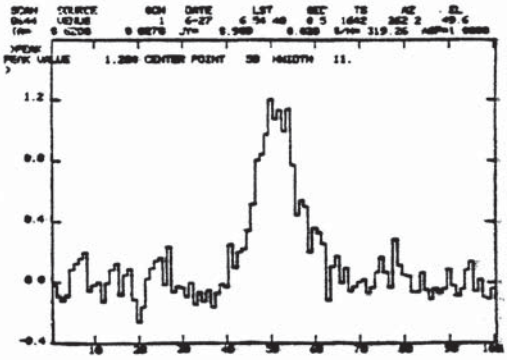
You have just removed a baseline from scan 8644 and produced the display at right. Now you want to fit a gaussian to the primary peak. Type:

PEAK

to set all the initial guesses for the gaussian fit. The output is seen on the display at right.

You are now ready to attempt a gaussian fitting by specifying

GAUSS



PMH PMT PMW

These verbs set the value of PTWH so that it points at one of the three arrays.

PMT sets PTWH = 1 (which means TEMP array)

PMW sets PTWH = 2 (which means WORK array)

PMH sets PTWH = 3 (which means HOLD array)

PTWH Is set as described above.

ADVERBS None.

RELATED VERBS None.

REMARKS

These three verbs are used to change the pointer to the desired array prior to performing operations which work on "the currently referenced array".

EXAMPLE

To display the contents of all three arrays on the same graph:

```
PMW PAGE SHOW
PMT RESHOW
PMH RESHOW
```

POINTS

A routine which sets the plotting mode for the SHOW display to points.

PTWH — Is not used or changed by POINTS.

ADVERBS — None.

RELATED VERBS

HISTOGRAM — Sets the SHOW display mode to a histogram.

LINE — Sets the SHOW display mode to a continuous line.

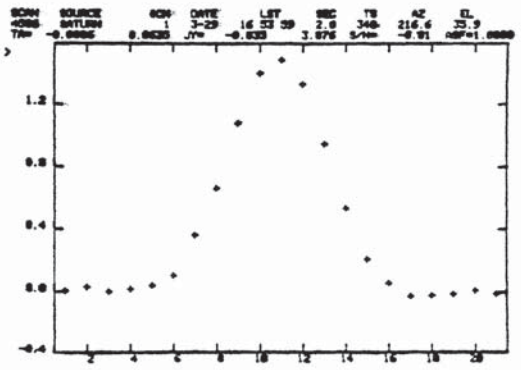
SHOW — Displays a scan with user specified or default scaling.

EXAMPLE

You wish to display the current scan in points mode. Specify:

```
POINTS PAGE SHOW
```

and you will get



PRINT

PRINT <field> prints the field requested on the CRT screen. <field> may be one or more variable or adverb names, literals, or arithmetic expressions. If <field> is an adverb or variable (a variable is an array or scalar defined in a procedure) name, the value(s) of the adverb or variable will be printed. Arrays of more than one dimension are printed with the first index varying most rapidly, as in Fortran. If <field> is a literal, it must be enclosed in single quotation marks, and it will be printed without the quotation marks. If <field> is an arithmetic expression, the result of the expression will be printed.

PTWH — Is not used or changed.

ADVERBS — None.

OBJECTS

<field> — As described above.

RELATED VERBS

READ <field> — Reads the values requested by field from the CRT. In this case field may be one or more variable or adverb names, and may not include expressions or literals.

? <field>
<field> ? — Alias for PRINT.

PRINTER — Directs the PRINT output to the line printer.

CRT — Directs the PRINT output to the CRT screen. CRT is the default assignment, but must be invoked after PRINTER to return output to the CRT.

REMARKS

PRINT can be used to monitor the values of adverbs, PTWH, SIZE and VRMS.

-continued-

EXAMPLES

```
You type        PRINT CENTER
Response is      25.0000  33.0000   78.0000   0.0000    0.0000    0.0000

You type        PRINT 'YOU SHOULD USE THE 300-FOOT'
Response is     YOU SHOULD USE THE 300-FOOT

You type        PRINT SQRT(144)
Response is      12.0000

You type        PRINT HWIDTH(2)
Response is      5.0000

You type        X = 1.25
                PRINT 'x =  'X
Response is     x =    1.2500

You type        PRINT BMARK ZLINE YMIN
Response is      1.0000   0.0000    -9999.9999     -9999.9999
```

PRINTER

Used to direct the output of certain print routines to the line printer. Those verbs affected are HELP, PRINT, SUMMARY, LIST, and TELL DISK.

PTWH — Neither used nor changed.

ADVERBS — None.

RELATED VERBS

CRT — Directs print output to the CRT screen.

REMARKS

CRT is the default print output direction. Using PRINTER may result in more readable copy than printing on the CRT screen and making a copy of the screen.

PROCEDURE

PROCEDURE is used to define a routine which can then be executed by entering its name on the terminal. (Procedurenames are essentially user-defined verbs.) The PROCEDURE statement is of the form

```
PROCEDURE procedurename  (optional arguments)
```

and begins the definition of the procedure. The complete definition of a procedure has the form:

```
PROCEDURE procedurename  (optional arguments)
statements the procedure is to execute
FINISH
```

The PROCEDURE statement puts the program into the COMPILE mode. When in the COMPILE mode, the program prompts with a colon (:) instead of a caret (>). The FINISH statement is required to complete the definition of the procedure and return the program to EXECUTE mode.

PTWH — Is not used or changed.

ADVERBS — None.

OBJECTS

procedurename — The name of the procedure which is being defined. A procedurename may be any alpha-numeric name up to 10 characters long. The first character of the name must be an alphabetic character. Procedure names are defined globally, and so must be unique.

arguments — Variables which are to be passed to the procedure when it is called by the user. Arguments may be any alpha-numeric name up to 10 characters long. The first character must be an alphabetic character. New names are implicitly declared. There is no restriction on the number of arguments permitted, but they must be encased in parentheses and separated by commas.

-continued-

RELATED VERBS

FINISH	Completes the definition of a procedure.
EDIT procedurename line #	Initiates editing of a procedure.
ENDEDIT	Terminates editing of a procedure.
RETURN	Is used in a procedure that is called by another procedure.
HELP PROCEDURE	Lists the names of the already defined procedures.
LIST procedurename	Lists the requested procedure.
CORE	Tells the user how much core is left for definition of procedures.
RESTART	Empties core of all user defined procedures (and returns adverbs to their default values).

EXAMPLE

To define a procedure to add two scans together:

```
PROCEDURE ADDSCANS (SCAN1,SCAN2)
GET (SCAN1)
FACT=-1;SCALE TRT
GET (SCAN2)
DIFF
PAGE SHOW
FINISH
```

Now to use ADDSCANS, you specify

```
ADDSCANS (710,712)
```

PVLS (Green Bank only)

PVLS computes corrections to the pointing equations used at the telescopes based on measured pointing offsets. The routine is intended to correct those terms in the equations affected by small feed offsets. PVLS solves for a constant term and a cos(dec) term in the hour angle pointing equation, and a constant term in the declination pointing equation. These corrections may then be entered into the control computer.

PTWH — Neither used nor changed.

ADVERBS — None.

REMARKS

The routine solves for the corrections by the method of least squares. Although a single pair of sources, well separated in declination, are sufficient for determining a solution, it is suggested that you use as many as practical. Use the template printed on the screen as a guide for your input.

EXAMPLE

A sample of a session with PVLS is:

```
>
.PVLS
THE VALUES OF P1, P2, AND P3 TO BE APPLIED

   TO THE 140-FT. POINTING CORRECTIONS FOR N SOURCE PAIRS

ENTER NUMBER OF SOURCE PAIRS (I2 FORMAT)
01
ENTER CORRECTIONS
SOURCE          DEC.              DELTA RA          DELTA DEC.
 PAIR        SDD MM SS.S       SMM SS.SS(TIME)   SMM SS.SS(ARC)
   1
              15 22 15.3         00 07.50          00 10.00
              55 26 22.0         00 04.60         -00 04.20


P1 = -1.00 P2 =   2.91 P3 =   0.05
>
```

RA (Tucson only)

RA sets an internal flag that causes LABEL to identify right ascension as the Y-coordinate and declination as the X-coordinate for maps. Would be used if the telescope was scanned along a line of constant right ascension.

PTWH — Is not used or changed.

ADVERBS — None.

RELATED VERBS

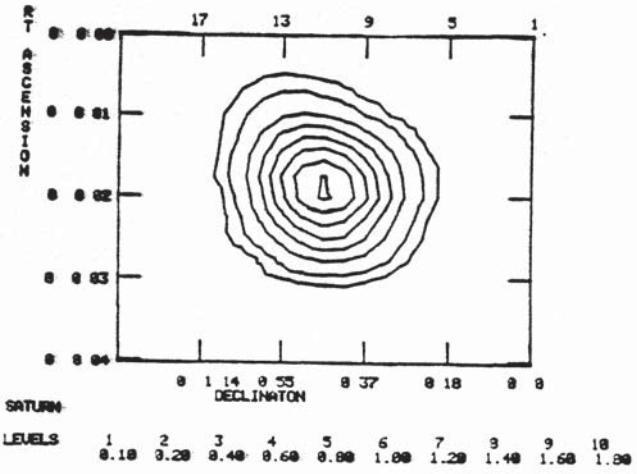
AZ — Identifies azimuth as the Y-coordinate for maps (Tuc only).

DEC — Sets declination as the Y-coordinate for maps (Tuc only).

EL — Identifies elevation as the Y-coordinate for maps (Tuc only).

LABEL — Labels the map drawn by MAPSHOW.

MAPSHOW — Draws a map with the levels specified by LEVS.



RANGE

RANGE allows the user to specify minimum and maximum values for the plotted y-values in SHOW. RANGE then sets the values of YMIN and YINCR.

PTWH — Is not used or changed.

ADVERBS

YMIN — The minimum y-coordinate to be plotted.

YINCR — The increment between each of the six tic marks on the y-axis.

OPERANDS

ymin — The minimum y-coordinate to be plotted.

ymax — The maximum y-coordinate to be plotted.

RELATED VERBS

HOLDY — Causes SHOW to reuse the previous y-scaling. Done by setting YMIN to -9E10.

FREEY
AUTO — Cause SHOW to do automatic scaling for the y-axis.

REMARKS

The operands may precede or follow RANGE. If they follow RANGE, they must be enclosed in parentheses.

EXAMPLE

You want to display 10 scans with the same y-axis scaling (-2_k to 8_k). You specify:

```
-2  8 RANGE
```

or

```
RANGE (-2,8)
```

READ

READ <field> reads from the CRT the values of the variables named in <field>. <field> may be one or more adverb or variable names (a variable is an array or scalar defined in a procedure). READ will respond with '#' until the user has entered enough values to satisfy field , at which time a caret (>) is returned.

PTWH — Is not used or changed.

ADVERBS — None.

OBJECTS

<field>	One or more adverb or variable names.

RELATED VERBS

PRINT <field>	Prints the values, or literals, requested by <field> on the CRT.
? <field>	Alias for PRINT

REMARKS

READ can be used to diminish the typing required to input the values of adverb arrays.

EXAMPLE

To input the values of CENTER, which has six elements, you can specify

```
READ CENTER
```

the program will respond with

-continued-

and then you input the values of CENTER -- either all at once:

```
#25 70 65 0 0 0
```

or one at a time:

```
#25
#70
#65
#0
#0
#0
```

When you have input six values, the program will give you a caret again.

The READ command can be used as a pause command in a procedure. For example, the procedure

```
PROCEDURE WAIT(FSCAN,LSCAN)
FOR I = FSCAN TO LSCAN
    GET (I) PAGE SHOW
    READ N
    END
FINISH
```

will graph FSCAN and then wait for the user to input something before it goes on to draw the next graph. The user can input either a number or a letter.

RECALL

RECALL copies the scan in the disk save bin named by the adverb NSAVE into the currently referenced array. Both the scan and its header information are copied from the disk bin into the array.

PTWH — Determines into which array the scan is copied. Is not changed.

ADVERBS

NSAVE — The number of the disk bin whose contents are copied into the currently referenced array.

RELATED VERBS

SAVE — Copies the currently referenced array into the disk bin NSAVE.

ERRORS

If NSAVE is greater than 80 (Green Bank) or 154 (Tucson), the message

```
NSAVE ?
```

will be generated.

EXAMPLE

To recall two scans into two different arrays, you can retrieve them by

```
NSAVE = 65;  PMW RECALL
NSAVE = 66;  PMT RECALL
```

REGAUSS

REGAUSS is used to fit up to six gaussian functions. REGAUSS is identical to GAUSS except that REGAUSS does not calculate first guesses for the gaussian heights. REGAUSS can be used to continue iterating the results of a previous GAUSS, or in the case where you can guess good first approximations to the gaussian heights.

PTWH	Determines for which array the fit is calculated. Is not changed.

ADVERBS

NGAUSS	The number of gaussians to be fit. NGAUSS must be between 1 and 6.
BGAUSS EGAUSS	The explicit sample numbers where the fit is to begin and end.
CENTER(N)	The sample number where gaussian N has its center. Iterated by the routine.
HEIGHT(N)	The height in degrees Kelvin of gaussian N. Iterated by the routine. Must be set before entering REGAUSS if GAUSS has not been previously executed as REGAUSS will not set a first-guess itself.
HWIDTH(N)	The width in number of points of gaussian N at half its peak width. Iterated by the routine.
NITER	The number of iterations which will be made to try to find a fit. Default value is 8.
FIXC	If FIXC is less than zero, REGAUSS will not iterate CENTER.
FIXHW	If FIXHW is less than zero, REGAUSS will not iterate HWIDTH.

-continued-

RELATED VERBS

GAUSS	Refines the user's first guesses of CENTER(N) and HWIDTH(N) for N Gaussians. Determines HEIGHT for each gaussian.
GMODEL	Evaluates the parameters of the gaussian(s), constructs a model of the gaussian (or sum of the gaussians), and replaces the data in the currently referenced array with the model. Also prints the parameters of the gaussians and the RMS error of the fit.
RESIDUAL	Evaluates the parameters of the gaussian(s) for each point and subtracts the total from the correctly referenced array.
GPARTS	Evaluates the parameters of, constructs and displays each gaussian separately.

REMARKS

REGAUSS is identical to GAUSS except that the routine assumes that the values in HEIGHT(N) are meaningful. REGAUSS does not change the contents of any of the three arrays.

ERRORS

If the fit is unsuccessful after NITER iterations, the message FIT FAILED is printed.

EXAMPLE

After making the fit in the example given in GAUSS you feel that convergence was not reached and you would like to iterate further. Specify

```
REGAUSS
```

To see the improved parameters, use TELL GPARMS or specify

```
PRINT CENTER(1) HEIGHT(1) HWIDTH(1)
```

RESHOW

This routine plots the contents of the currently referenced array using the Y-scaling used by the last SHOW. RESHOW does not draw or label the axis.

PTWH	Determines which array is plotted. Is not changed.

ADVERBS

BDROP(CH)	The number of points at each end of each receiver
EDROP(CH)	channel which will not be plotted.

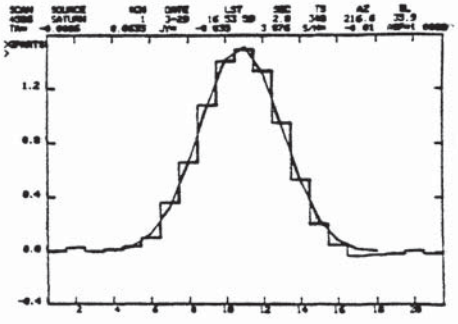
RELATED VERBS

RHIST	Sets the plotting mode for RESHOW to HISTOGRAM.
RLINE	Sets the plotting mode for RESHOW to LINE.
RPOINTS	Sets the plotting mode for RESHOW to POINTS.

REMARKS

RESHOW is often used to plot a polynomial baseline model, or gaussian model, on top of the data for which the model was generated. RESHOW can also be used to plot the reduced data on top of the original.

A particularly effective display format is to SHOW data as a HISTOGRAM and RESHOW a fit as LINE, as in the example on the right. These are the default display modes in Tucson.



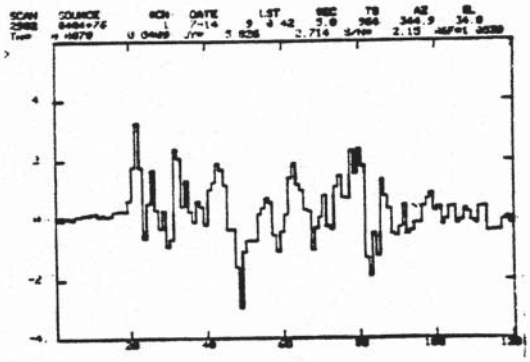
-continued-

EXAMPLE

To plot the 12-meter scan 2902, specify:

```
GET 2902
PAGE SHOW
```

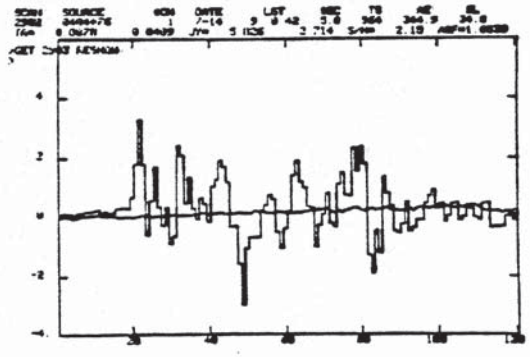
to produce the display to the right. Now you wish to look at the second receiver of the same scan, plotted on top of the current display for purposes of comparison.



In this case, the scan number of the second receiver is 2903, so specify:

```
GET 2903
RESHOW
```

to produce the final result illustrated to the right.



RESIDUAL

This routine takes the parameters of specified gaussians (usually refined by GAUSS), and subtracts the sum of the gaussians from the currently referenced array.

PTWH — Determines from which array the total model is subtracted.
Is not changed.

ADVERBS

NGAUSS — The number of gaussians.

BGAUSS
EGAUSS — The first and last points of the region over which the gaussian(s) are subtracted.

CENTER(N) — The center points of the gaussians.

HEIGHT(N) — The heights of the gaussians.

HWIDTH(N) — The half widths of the gaussians.

RELATED VERBS

GAUSS — Refines the user's first guesses of CENTER(N) and HWIDTH(N) for N gaussians. Determines HEIGHT for each gaussian.

GMODEL — Constructs the sum of the N gaussians using the parameters CENTER, HWIDTH, and HEIGHT, and replaces the currently referenced array with the sum.

PEAK — Finds the CENTER, HWIDTH and HEIGHT of a single gaussian. Also sets BGAUSS and EGAUSS.

REMARKS

GAUSS may or may not have been called before RESIDUAL.

-continued-

EXAMPLE

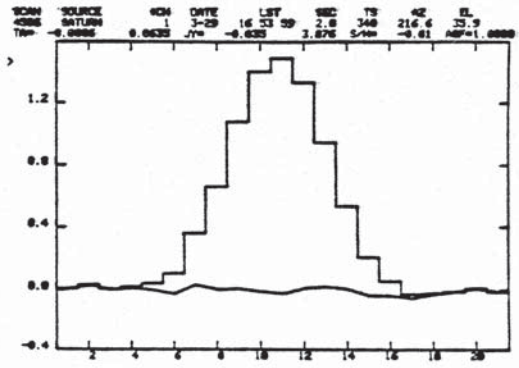
You had

GET 4586 BASELINE PAGE GAUSS SHOW

Now you want to remove the gaussians and see what the residual looks like. Specify

TRH RESIDUAL RESHOW

You will have



RESTART

RESTART re-initializes the program memory. It zeros the three arrays (WORK, TEMP, and HOLD), empties the procedure definition space, and sets the values of all the adverbs to their initial values. RESTART puts the program into the same condition it has immediately after loading the program.

PTWH	Is not used, but is set to the WORK array.

ADVERBS

All adverbs	Are set to their default values.

RELATED VERBS

STORE page#	Copies the procedure definition space and adverb values into a disk storage area called "page" page#.
RESTORE page#	Copies disk page page# into the program memory.

REMARKS

STORE, RESTORE, and RESTART are useful for control of the program memory space. RESTART is used to get a fresh page on which to write new procedures and to return the adverbs to their initial values.

EXAMPLE

You were trying to define a procedure and got the message

```
BLEW CORE
```

which tells you that you are out of procedure space. To empty the procedure space, enter

```
RESTART
```

but realize that once you have done this, the three arrays will be empty, your adverbs will return to their initial values, and all of your procedure definitions will be gone.

RESTORE

RESTORE page# restores to memory the procedure definition space stored on disk by the command STORE page#. Both procedure definitions and adverb values are restored.

PTWH — Is not used, but is reset to the value it had before the STORE page# command.

ADVERBS

All Adverbs — Are reset to the values they had before the STORE page# command.

OBJECTS

Page# — The disk page which is copied into memory. Page# must be an integer between 1 and 6.

RELATED VERBS

STORE page# — Creates a copy of the current state of memory in the specified disk page#.

RESTART — Restores the program memory to its original or default state.

REMARKS

STORE, RESTORE, and RESTART are used to control the program memory space. RESTORE restores the program to a prior condition saved for that purpose.

EXAMPLE

Remember that copy of memory you saved yesterday with all the procedures and adverb values just the way you wanted them? (See STORE example.) Well, you can get back to where you were before by specifying

```
RESTORE 5
```

RETURN

RETURN is required in procedures which are called by other procedures. The RETURN statement should immediately precede the FINISH statement when it is used.

PTWH — Is not used or changed.

ADVERBS — None.

RELATED VERBS

See PROCEDURE

REMARKS

A procedure which is called by another procedure must be defined before the procedure which calls it. That is, procedures must be defined in a "bottom up" fashion.

ERRORS

If the RETURN statement is omitted from a called procedure, the program exits that procedure to the EXECUTE communications mode rather than to the calling procedure.

If a procedure which is called by another procedure is not defined before the calling procedure, the message SYMBOL ? will be generated (and the program will return to EXECUTE mode) when the name of the not-yet-defined procedure is used while defining the calling procedure.

-continued-

EXAMPLE

To define a procedure which can be called by another:

```
 PROCEDURE BUSY(T)
:R = T * T
:PRINT T R
:RETURN
:FINISH
```

Now define the calling procedure:

```
 PROCEDURE EG
:T = 12
:BUSY(T)
:PRINT 'THATS ALL FOLKS'
:FINISH
```

Now you can execute BUSY through EG by

```
 EG
```

and you will get

```
 12.0000  144.0000
THATS ALL FOLKS
```

RHIST RLINE RPOINTS

These routines control the type of plot made by the RESHOW verb. RHIST sets the RESHOW display mode to histogram. RLINE sets the RESHOW display mode to a continuous line and RPOINTS sets the RESHOW display mode to points.

PTWH — Is not used or changed by RHIST, RLINE or RPOINTS.

ADVERBS — None.

RELATED VERBS

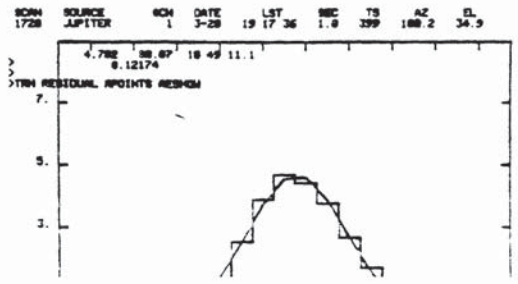
HISTOGRAM	Sets the SHOW display mode to a histogram.
LINE	Sets the SHOW display mode to a continuous line.
POINTS	Sets the SHOW display mode to discrete points marked by crosses.
RESHOW	Graphs the currently referenced array on the CRT, no labels or borders, using the scaling last used by SHOW.
SHOW	Graphs the currently referenced array on the CRT with labels and borders.

EXAMPLE

You have just fit a double gaussian to scan 520 and modeled the gaussian over the data. Now you would like to also see the residual by specifying

PMH RESIDUAL RPOINTS RESHOW

and you get



RMS

This routine computes the sample root mean squared of the scan in the currently referenced array, using the specified region or regions. The result is stored in the variable VRMS(CH).

PTWH — Determines for which scan the RMS is computed. Is not changed.

ADVERBS

BDROP(CH)
EDROP(CH) — The number of points at each end of each receiver channel which will be ignored by the routine.

NREGION(CH) — Specifies the region or regions of the scan to be used to compute the RMS. N is an integer between 1 and 8.

BBASE(CH)
EBASE(CH) — The number of points at each end of each receiver channel (not including the points dropped by BDROP and EDROP) which will be used to compute the RMS if NREGION(CH,1)=0.

RELATED VERBS

None.

REMARKS

The RMS computed is stored in the variable VRMS.

-continued-

EXAMPLES

To compute the RMS of the scan in the currently referenced array, specify

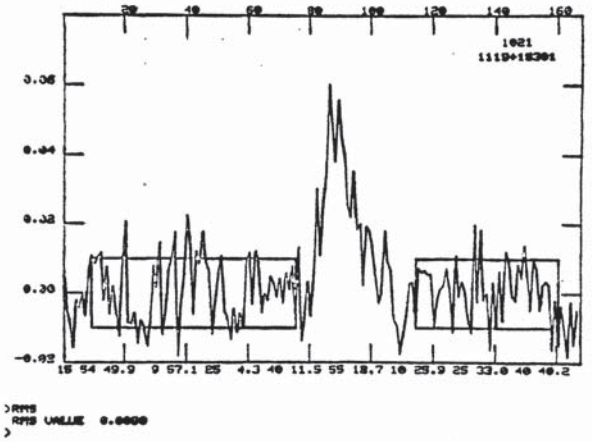
```
BMARK=1;
PAGE SHOW
```

You will see something like the display below. The boxes indicate the regions over which the RMS will be computed. To compute and print the RMS, specify

```
RMS
```

In this case, the results are

```
RMS VALUE 0.0090
```



SAVE

SAVE copies the scan in the currently referenced array into one of the save areas on the disk called "bins". Both the scan and its header information are copied into the save bin named by the adverb NSAVE.

PTWH — Determines which scan is saved. Is not changed.

ADVERBS

NSAVE — The number of the disk bins into which the scan in the currently referenced array is put. NSAVE must be an integer between 1 and 80 (Green Bank) or 154 (Tucson).

RELATED VERBS

RECALL — Copies the scan in bin NSAVE into the currently referenced array.

ERRORS

If NSAVE is greater than 80 (Green Bank) or 154 (Tucson), the message

```
NSAVE ?
```

is generated.

EXAMPLE

You are observing between noon and six o'clock. It is now 5:59 and the next observer is anxious to get on the computer. You have a partly reduced scan in the WORK array and an incomplete stack of scans in the HOLD array. To save both of these until you can get back on the computer, specify

```
NSAVE = 65; PMW SAVE
NSAVE = 66; PMH SAVE
```

SCALE

SCALE multiplies the data in the currently referenced array by the values of the adverb FACT(CH). FACT may be specified independently for each receiver channel, and may have any real value.

PTWH	Determines which scan is scaled. Is not changed.

ADVERBS

FACT(CH)	The value by which each receiver channel is to be multiplied.

RELATED VERBS

BIAS	Also uses the adverb FACT. BIAS adds FACT(CH) to the data in each receiver channel.

EXAMPLE

SCALE is often useful when the values of the y-coordinate are so large that the tic mark labels overflow the formats used by SHOW. If the numbers are too large, set FACT to a small number and SCALE the data:

```
FACT = 1/1000
SCALE PAGE SHOW
```

SCALE is also useful for negating a scan. The trick for adding two scans is to negate one:

```
FACT = -1; SCALE
```

and then to use the verb DIFF to subtract them. You may need to negate the result again.

SCLEAR

SCLEAR is a routine which clears the accumulator (HOLD) array before stacking data.

PTWH — Is not used or changed by SCLEAR.

ADVERBS — None.

RELATED VERBS

ACCUM	Adds the scan in the WORK array to the contents of the HOLD array.
AVE	Divides the HOLD array by the number of the accumulated scans.

EXAMPLE

See ACCUM.

SCRATCH

SCRATCH deletes the specified procedure from the program, but does not return the procedure space used by the procedure to the program.

PTWH — Is not used or changed by SCRATCH

ADVERBS — None.

OBJECT — The name of the procedure to be deleted.

RELATED VERBS

See PROCEDURE

SEDIT, SEDITS (Tucson only)

SEDIT deletes the specified pair from the on-off scan in the currently referenced array and recomputes the source temperature and RMS with the pair deleted. SEDITS deletes a series of pairs from the on-off scan in the currently referenced array and recomputes the source temperature and RMS.

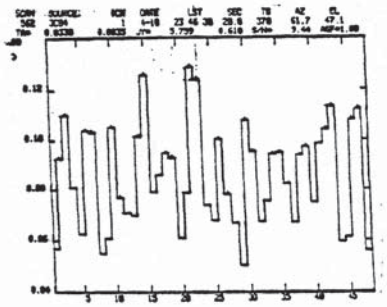
PTWH — Determines from which scan the pair or pairs are to be deleted.
It is not changed.

ADVERBS — None.

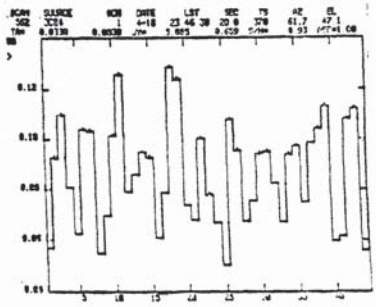
RELATED VERBS — None.

EXAMPLE

Scan 562 looks like the scan to the right and you want to delete the third pair.



You specify 3 SEDIT XX and get



SHOW

SHOW plots the contents of the currently referenced array on the CRT screen. The routine also draws and labels the axis with the default as follows:

y axis	antenna temperature	Kelvins
x axis	sample number and position (Green Bank only)	Points

SHOW will automatically compute the scaling needed to display the scan in the currently referenced array: however, the user may control the scaling by means of the adverbs BDROP, EDROP, YMIN and YINCR and the verbs FREEY, HOLDY, and RANGE.

PTWH — Determines which array is plotted. Is not changed.

ADVERBS

BDROP(CH)
EDROP(CH) — The number of points at each end of each receiver channel which will not be plotted.

BMARK — If = 1, the regions last used to compute a baseline or RMS will be indicated by boxes.

CMARK(N) — N is an integer between 1 and 6. If CMARK(N) is greater than 0, a vertical line will be drawn at point CMARK(N).

SLABEL — Determines how certain documentation information are printed on the display (Tucson only).

TMARK(N) — N is an integer between 1 and 6. If TMARK(N) is greater than -999 a horizontal line will be drawn at temperature TMARK(N) in degrees K.

YMIN(CH) — The minimum temperature that is to appear on the graph.

YINCR(CH) — The user may specify how many degrees are to be between each y-axis tic mark. If YMIN is greater than -999, SHOW will not compute the y-axis scaling, but will use the values specified by YMIN and YINCR.

ZLINE — If set to 1, a horizontal line will be drawn at zero K.

-continued-

RELATED VERBS

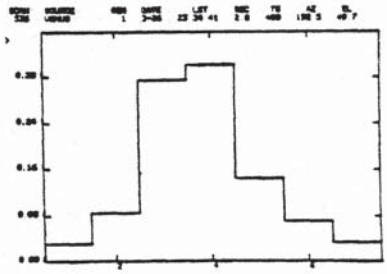
AUTO	Alias for FREEY (below).
FREEY	Sets YMIN to -90000. This causes SHOW to automatically scale the y-axis.
HISTOGRAM	Sets the SHOW display mode to a histogram.
HOLDY	Causes SHOW to retain the last determined y-axis scaling.
LINE	Sets the SHOW display mode to a continuous line.
POINTS	Sets the SHOW display mode to points marked by crosses.
RANGE	Defines the values of YMIN and YINCR according to the specified max and min values.

EXAMPLE

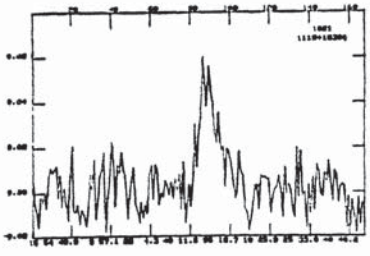
After retrieving Tucson scan 536 from disk, you may use:

PAGE SHOW

to get the display to the right.



A similar display of Green Bank scan 1021 might look like the display to the right.



SIN

A built-in sine function. Functions the same as the Fortran SIN.

PTWH — Neither used nor changed.

ADVERBS — None.

OBJECT

argument — An argument assumed to be in radians. The argument may be a constant, a variable, or an arithmetic expression, enclosed in parentheses.

RELATED VERBS

see COS, TAN, ATAN

EXAMPLE

```
PRINT SIN(0.7854)
0.7071
```

SLENGTH

This routine places the number of points in the scan in the currently referenced array into the adverb NPOINTS.

PTWH — Defines the scan whose length is returned. Is not changed.

ADVERBS

NPOINTS — Set to the number of points in the scan.

RELATED VERBS — None.

Example

You wish to know how many points are in the scan 4580. You proceed as follows:

```
GET 4580
SLENGTH PRINT NPOINTS
 21.00
```

SMOOTH

SMOOTH smooths the scan in the currently referenced array by convolution with the weight function SMWGT specified by the user.

PTWH — Determines which scan is to be smoothed.

ADVERBS

BDROP(CH)
EDROP(CH) — The number of points at each end of each receiver channel which will be ignored.

SMWGT (12) — The smoothing function array. The first element of the array specifies the number of elements in the function - not greater than 11. Other elements specify the weighting of each point. The total of all weights should equal unity.

RELATED VERBS

BOXCAR — Smooths a scan by averaging a specified number of points together.

HANNING — A smoothing routine which averages three points together with the center point getting twice as much weight as either side point.

EXAMPLE

To mimick commonly used smoothing functions,

```
SMWGT = 3, 0.25, 0.50, 0.25; SMOOTH
```

is the same as HANNING and

```
SMWGT = 5, 0.2, 0.2, 0.2, 0.2, 0.2; SMOOTH
```

is the same as BOXCAR with NBOX=5. An example of a five point non-distorting smoothing function is

```
SMWGT = 5, -0.073427, 0.293706, 0.559441, 0.293706,
            -0.073427
```

SNAP (Tucson only)

SNAP is an informational verb which prints the header information for the currently referenced array on the printer.

PTWH	Determines which header is printed. Is not changed.
ADVERBS	None.

RELATED VERBS

HEADER	Prints the same documentation on the CRT.
TABLE	Prints the array values for the currently referenced array.

ERRORS

If there is no scan in the currently referenced array when SNAP is specified, the message "LOAD SCAN" will be generated.

EXAMPLE

```
GET 2516 SNAP
```

SOLVETIP

SOLVETIP uses the antenna temperature values measured at a number of pre-defined elevations to compute the receiver temperature and atmospheric absorption coefficient at the zenith. The values of these are printed on the screen. The solution of SOLVETIP is not applied directly to the data and the user should do this via the verb SCALE (Green Bank only). The correction can automatically be applied at the 12-meter telescope by setting the value of ATTN in the control system.

PTWH — Determines which scan is used for the fit.

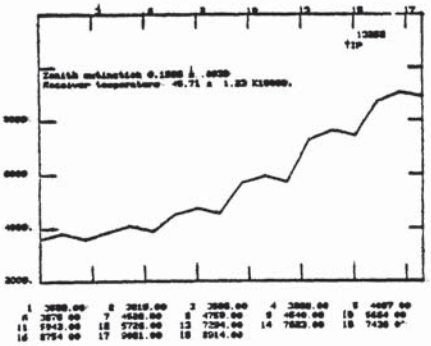
ADVERBS — None.

RELATED VERBS

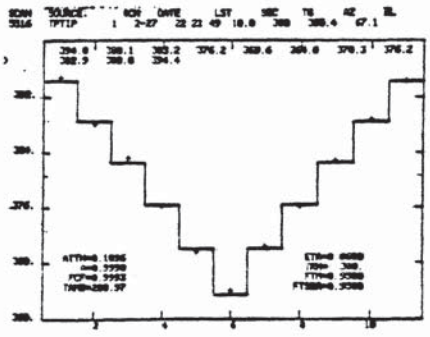
SCALE — Multiplies the points in a scan by the constant in the adverb FACT(CH).

EXAMPLE

The Green Bank version of SOLVETIP is matched to the TIP observing procedure in use at the 140-foot telescope. An example is shown at right.



At the 12-meter telescope, the observing and analysis procedures are TPTIP for a total power tip or SPTIP for switched power data. A TPTIP example is shown to the right.



SQRT

A built-in square root function. Functions the same as the Fortran SQRT.

PTWH — Neither used nor changed.

ADVERBS — None.

OBJECT

argument — The argument of SQRT may be a constant, a variable, or an arithmetic expression, enclosed in parentheses.

EXAMPLE

```
PRINT SQRT(144)
 12.000

NPOINTS = 144
PRINT SQRT(NPOINTS)
 12.000
```

STACK

A routine which lists all scan numbers currently in the STACK.

PTWH — Is not used or changed.

ADVERBS

ACOUNT — A pointer whose value is the number of scan numbers in the stack.

ASTACK — An array containing the stack of scan numbers.

RELATED VERBS

A # — Inserts one scan number into the STACK.

ADD (#,#) — Inserts a series of scan numbers into the STACK.

DELETE # — Deletes one scan number from the STACK.

EMPTY — Empties the STACK.

EXAMPLE

After defining the STACK by

```
500 A;  506 A
512  520  ADD
```

STACK will produce the following output:

```
500. 506. 512. 514. 516. 518. 520.
```

Note that this was two-channel Tucson data, since only even numbers between 512 and 520 were added to the stack.

STORE

STORE page# copies the procedure definition space into a disk storage space called a "page". It does not change the present state of memory; it merely makes a copy of memory for later retrieval. Both procedure definitions and adverb and variable values are stored.

PTWH — Is not used or changed.

ADVERBS — No adverbs are changed, but all adverb values are stored.

OBJECTS

Page# — The disk page into which the copy of memory is to be put. Page# must be an integer between 1 and 6.

RELATED VERBS

RESTORE page# — Copies disk page# back into memory.

RESTART — Wipes out the current state of memory by replacing it with the default or original state.

REMARKS

STORE, RESTORE and RESTART are used to control the program memory space. STORE saves the current state of memory for later restoration and use.

EXAMPLE

You are one of three observers presently using the telescope. You have gone to a great deal of trouble to set up your adverb values and procedures and now it is time to get off the telescope. You are not thrilled at the prospect of having to redefine your procedures and reset your adverbs tomorrow. Pick a number between 1 and 6, say 5, and specify

```
STORE 5
```

Now in disk page 5 there is a copy of the procedure definition space exactly like that in program memory.

SUM

A routine that is used to stack scans in the HOLD array with user supplied weighting. SUM after an AVE, SCLEAR or a program restart copies the header information and the data values in the WORK array to the HOLD array. Subsequent uses of SUM add the data values contained in the WORK array to the contents of the HOLD array. Subsequent uses of SUM do not change the header stored in the HOLD array, except for the integration time, which is incremented, and the system temperature, which is averaged with the specified weights. SUM increments the internal stack each time a scan is accumulated, and stores the numbers of the first 17 and the last scan accumulated.

PTWH	Must be pointing to WORK array. Is set to the HOLD array.

ADVERBS

WEIGHT(CH)	The weighting used in accumulating the current scan.

RELATED VERBS

ACCUM	Accumulates scans with no weighting.
AVE	Divides the accumulated scans by the number of scans accumulated. Sets the internal stack counter to zero.
TELL CSTACK	Prints the internal stack counter and the numbers of scans accumulated.
SCLEAR	Sets the accumulator flag and internal stack counter to zero.

EXAMPLE

To stack scans weighted inversely by an RMS, define the following procedure:

```
PROCEDURE ADDS
FOR I = 1 TO ACOUNT
   GET ASTACK (I); RMS WEIGHT (1) = 1./VRMS (1); SUM END
AVE PAGE SHOW
FINISH
```

Load the appropriate scans in the STACK and execute ADDS.

SUMMARY (Green Bank only)

SUMMARY produces a listing of information about the scans on the analysis disk under a specified user number. A SUMMARY contains more information than a TELL DISK output since the actual data, and not just the scan number table in memory, are referrred to. Besides the scan number, a SUMMARY lists the source name, type of scan, right ascension and declination (epoch 1950.0), scanning rate, scanning direction, number of samples in the scan, and the calibration factor. At the 12-meter telescope, similar information can be obtained with the procedure LDISK.

PTWH — Not used but set to the TEMP array.

ADVERBS — None.

OBJECT

user# — NRAO Computer Department assigned user number, associated with your data through your setup or source cards.

RELATED VERBS

CRT — Directs the print output of certain routines, including SUMMARY, to the CRT screen.

PRINTER — Directs the print output of certain routines, including SUMMARY, to the Versatek printer.

REMARKS

SUMMARY destroys the contents of the TEMP array.

If you have no idea what data are on disk, but need to know more than the scan numbers, first use WIPE to get a list of the user numbers represented on disk. Then do a SUMMARY of each user number.

EXAMPLE

NRAO CONDAR SYSTEM

SUMMARY

USER 28 BURKE JAN84

SCAN	SOURCE	SCAN TYPE	RA	1950	DEC	RATE "/MIN		NUM PTS	CAL FACTOR
1001.	1108+281@1	NO	11	6	56.2	17 34 45.	90.00	V	252	0.00031
1011.	1111+488@1	NO	11	11	34.2	48 25 20.	90.00	V	251	0.00031
1021.	1119+183@1	NO	11	18	7.1	15 44 13.	90.00	V	253	0.00031
1031.	1123+264@1	NO	11	22	5.3	24 42 16.	90.00	V	352	0.00031
1041.	1128+385@1	NO	11	26	27.3	35 53 57.	90.00	V	421	0.00031
1051.	1140+223@0	NO	11	39	4.3	19 46 16.	90.00	V	295	0.00031
1061.	1144+402@1	NO	11	43	48.3	39 14 8.	90.00	V	295	0.00031
1071.	1150+497@1	NO	11	49	3.3	47 18 45.	90.00	V	395	0.00031
1081.	1156+295@1	NO	11	55	26.2	27 14 3.	90.00	V	296	0.00031
1091.	1206+439@1	NO	12	4	57.4	41 10 57.	90.00	V	421	0.00031
1101.	1213+356@1	NO	12	11	48.2	37 28 6.	90.00	V	422	0.00031

TABLE

TABLE is an informational verb. In Tucson, TABLE prints the antenna temperature (Kelvins) for each point and labels the information with scan number. In Green Bank, each point is labelled with its sample number, but no scan number is printed.

PTWH — Determines for which scan the table is printed. Is not changed.

ADVERBS — None.

OBJECTS — None.

RELATED VERBS

HEADER — Prints the header information for the scan in the currently referenced array.

EXAMPLE

A Tucson user wants a list of the data values of scan 546. Specifying:

```
GET 546 PAGE TABLE
```

produces the following:

```
SCAN 546
0.060   0.063   0.058   0.060   0.133   0.385   0.950   1.985
3.507   5.107   6.131   6.246   5.568   4.409   3.144   2.116
1.478   1.143   0.930   0.735   0.530
```

A Green Bank TABLE looks like the following:

```
 1    -0.00    2    -0.00    3    -0.01    4    -0.01    5    -0.00
 6    -0.00    7    -0.00    8    -0.00    9     0.00   10     0.00
11     0.00   12     0.00   13     0.01   14    -0.00   15     0.00
16    -0.00   17     0.00   18    -0.00   19    -0.01   20     0.00
21     0.02   22    -0.00   23    -0.00   24    -0.00   25    -0.01
26    -0.00   27    -0.00   28    -0.01   29    -0.00   30     0.00
31     0.00   32     0.01   33    -0.01   34    -0.00   35     0.00
```

TAN

A built-in tangent function. Functions the same as the Fortran TAN.

PTWH — Neither used nor changed.

ADVERBS — None.

OBJECT

argument — The argument of TAN is expected to be in radians. The argument may be a constant, a variable, or an arithmetic expression, enclosed in parentheses.

RELATED VERBS

see COS, SIN, and ATAN

EXAMPLE

```
PRINT TAN(0.7854)
 1.000
```

TCUR

This routine activates the horizontal crosshair and returns the temperature value for the crosshair position on the current display when any key except RETURN is struck.

PTWH — Is not used or changed by TCUR.

ADVERBS — None.

RELATED VERBS

CCUR — Returns the sample number of the current vertical crosshair position.

CROSSHAIR — Activates the horizontal and vertical crosshairs and prints the point number and temperature values at the crosshair position.

EXAMPLE

You want to expand the y-axis of the SHOW display between two temperatures designated by the horizontal crosshairs. A procedure for doing so is defined below:

```
PROCEDURE YSET
X1 = TCUR; X2 = TCUR;  X1  X2  RANGE
RETURN
FINISH
```

TELL

TELL is an informational verb requiring an object. The five permitted objects of TELL are DATA (Tucson only), DISK, CSTACK, GPARMS (Green Bank only) and KSCANS. TELL DISK lists the scans in the disk index; TELL CSTACK prints the stack counter and the numbers of the scans that have been accumulated in the HOLD array; TELL GPARMS prints the current gaussian fit parameters; TELL KSCANS lists the scans in the KEEP file.

PTWH — Is not used or changed.

ADVERBS — None.

OBJECTS

DATA — Is used to request a listing of the continuum scans in the disk index (Tucson only).

DISK — Is used to request a listing of the scans in the disk index.

CSTACK — Is used to request a listing of the scans that have been accumulated. Only the numbers of the first seventeen and the last scan that was accumulated will be printed.

GPARMS — Is used to print the current gaussian fit parameters in the format used by GMODEL. Used as an alternative to PRINT CENTER, etc. where the output is the sample number.

KSCANS — Is used to request a listing of the scans in the KEEP file.

RELATED VERBS

CRT — Used to direct the output of TELL DISK to the terminal screen.

PRINTER — Used to direct the output of TELL DISK to the hard copy device.

-continued-

The output of TELL DISK looks like:

```
>TELL DISK
5855  5856  5857  5858  5859  5860  5861  5862  5863  5864
5865  5866  5867  5868  5869  5870  5871  5872  5873  5874
5875  5876  5877  5878  5879  5880  5881  5882  5883  5884
5885  5886  5887  5888  5889  5890  5891  5892  5893  5894
5895  5896  5897  5898  5899  5900  5901  5902  5903  5904
5905  5906  5907  5908  5909  5910  5911  5912  5913  5914
5915  5916  5917  5918  5919  5920  5921  5922  5923  5924
5925  5926  5927  5928  5929  5930  5931  5932  5933  5934
5935  5936  5937  5938  5939  5940  5941  5942  5943  5958
5959  5960  5961  5962  5963  5964  5965  5966  5967  5968
5969  5970  5971  5972  5973  5974  5975  5976  5977  5978
5979  5980  5981  5982  5983  5984  5985  5986  5987  5988
5989  5990  5991  5992  5993  7798  7799  7800  7801  7802
7803  7804  7805
```

After KEEPing 11 scans to disk, the output of TELL KSCANS is:

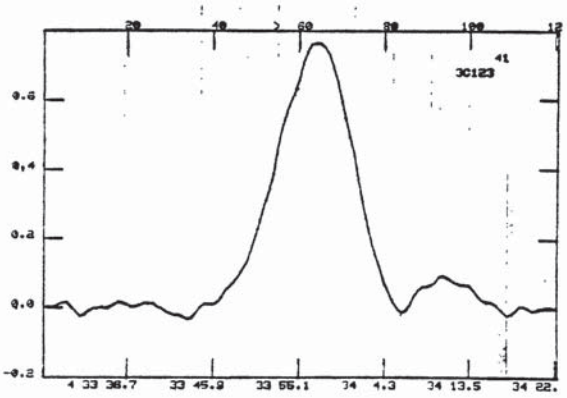
```
>TELL KSCANS
   21   31   41   51   61   71   81   91  101  111
  121
>
```

After accumulating 5 scans, the output of TELL CSTACK is:

```
>TELL CSTACK
    5
 2807  2808  2809  2810  2811
>
```

After using GAUSS to fit a gaussian to the scan shown below, the output of TELL GPARMS is:

```
>TELL GPARMS
      0.774          1.94  4 33 56.8
          0.02418
```



TEMPS (Tucson only)

TEMPS lists the source temperature for each individual pair of the on-off scan in the currently referenced array.

PTWH Determines which scan is used.

ADVERBS None.

RELATED VERBS None.

REMARKS

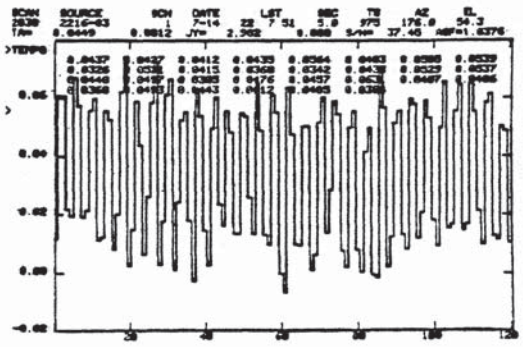
TEMPS is handy for determining bad pairs to be deleted from an on-off scan.

EXAMPLE

You want to know the individual pair temperatures of scan 2838. You specify:

```
2838 S
TEMPS
```

and get



TITLE (Tucson only)

TITLE writes one line of documentation about the current scan on the CRT screen including scan number, date, source name, RA, DEC, number of points, SEC and scan time.

PTWH — Determines which array is used by TITLE.

ADVERBS — None.

RELATED VERBS

HEADER — Prints header information about the currently referenced array on the CRT.

SNAP — Prints header information about the currently referenced array on the printer.

EXAMPLE

You want to compute the area under the curve of several profiles and document the results. You might define the following procedure:

```
PROCEDURE AREA (BS,ES)
FOR I = BS TO ES BY 2
    GET I BASELINE MOMENT TITLE PRINT SIZE(1)
    END
FINISH
```

TRH TRT TRW

These routines copy the contents of the currently referenced array to the indicated array:

TRH copies to the HOLD array

TRT copies to the TEMP array

TRW copies to the WORK array

PTWH — Determines which scan is transferred. Is changed to the indicated array:

TRH sets PTWH to the HOLD array

TRT sets PTWH to the TEMP array

TRW sets PTWH to the WORK array

ADVERBS — None.

OBJECTS — None.

RELATED VERBS — None.

TWO (Tucson only)

TWO sets the CONDAR processing mode to dual channel.

PTWH — Is not used or changed.

ADVERBS

SINCR — The scan increment is set to two.

RELATED VERBS

ADD (#,#) — Inserts a group of scans into the STACK with SINCR as the increment between scans.

ONE — Sets the processing mode single channel (SINCR = 1).

REMARKS

The five point fitting procedure, F, also uses SINCR to determine what scans are to be included in the STACK for the five point. The LOGS procedure uses SINCR as the increment between scans listed.

EXAMPLE

You wish to print a log of the first channel of dual channel scans, 612 - 680. You specify

```
TWO
612 680 LOGS
```

WHILE

WHILE begins a logical construction of the form

```
WHILE (test condition)
     (statements to be iterated)
END
```

WHILE can be used only in procedures. The WHILE logical construction is a conditional looping device similar to the DO WHILE looping facility in PL/I.

(test condition) can be any expression which has a single true or false result. As long as (test conditions) is true, the (statements to be iterated) will be repeated. The (statements to be iterated) may be any group of statements. END is required to complete the construction. WHILE loops may be nested and arranged on one or more lines; if arranged on one line, the statements to be iterated must be separated from the logical statements by semicolons.

PTWH — Is not used or changed.

ADVERBS — None.

RELATED VERBS

END — Completes a logical construction.

FOR — Begins a logical construction of the form:

```
FOR(VARIABLE) = (ALPHA) TO (BETA) BY (GAMMA)
     (statements to be repeated)
END
```

IF — Begins a logical construction of the form:

```
IF(test condition)
     THEN(do this)
     ELSE(do this)
END
```

See PROCEDURE

-continued-

EXAMPLE

(1) WHILE construction on several lines in a procedure

```
PROCEDURE EG(X,Y)
WHILE X>Y
      Y=X*Y/2+1
      PRINT X,Y
      END
FINISH
```

(2) WHILE construction on one line in a procedure

```
PROCEDURE EGALT
WHILE X>Y; Y=X*Y/2+1; PRINT X,Y; END
FINISH
```

(3) WHILE constructions nested in a procedure

```
PROCEDURE EG2(X,Y)
WHILE X>Y
      WHILE Y < SQRT(2)
      X=X+SQRT(2)
      END
END
FINISH
```

WINDOW (Green Bank only)

WINDOW is used to limit processing and display of a scan to those positions on the x-axis between the values specified in XWIND(1) and XWIND(2). The positions in XWIND are in radians.

PTWH — Determines which scan is windowed. It is not changed.

ADVERBS

XWIND(1)
XWIND(2) — The minimum and maximum x-coordinates for windowing. NOTE: the units for XWIND are radians.

RELATED VERBS

BESTVIEW — Sets the starting and stopping points of a scan to the more restrictive of (1) the current value of XWIND or (2) the endpoints of the scan in the currently referenced array.

REMARKS

WINDOW may be used in conjuction with the verb BESTVIEW to display a map which spans the maximum range in the x-coordinate shared by a group of scans. See the example under BESTVIEW.

If you have multi-feed data, you should use FOFFSET to correct for the feed offsets from the center of the receiver box before using WINDOW or BESTVIEW.

WIPE (Green Bank only)

This routine is provided for wiping out scans on the analysis disk that are filed under a specified user number. References to scans having the specified user number are deleted from the scan directory but no disk space is actually freed. After invoking WIPE, the data disk is searched to provide a list of user numbers represented on disk. You are then asked to specify, one at a time, the user numbers of the scans to be wiped.

PTWH Not used or changed.

ADVERBS None.

RELATED VERBS None.

REMARKS

WIPE is useful for speeding up scan access if the scan directory is quite long.

WIPE is especially useful in cases where another observer has scans with the same scan numbers as yours, preventing you from seeing your data.

EXAMPLE

A sample use of WIPE is illustrated:

```
>WIPE

  THE FOLLOWING USER NUMBERS APPEAR
  IN THE DISK INDEX.
915
533
198
999
 YOU MAY DELETE ONE USER NUMBER AT A TIME
 FROM THE INDEX. ENTER THE VICTIM'S FULL
 THREE-DIGIT USER NUMBER, WITH LEADING
 ZEROS IF NECESSARY. TO EXIT FROM THIS
 ROUTINE, ENTER THE NUMBER 0.
999

ANOTHER USER NUMBER? (ENTER 0 IF FINISHED)
0
>WIPE

  THE FOLLOWING USER NUMBERS APPEAR
  IN THE DISK INDEX.
915
533
198
 YOU MAY DELETE ONE USER NUMBER AT A TIME
 FROM THE INDEX. ENTER THE VICTIM'S FULL
 THREE-DIGIT USER NUMBER, WITH LEADING
 ZEROS IF NECESSARY. TO EXIT FROM THIS
 ROUTINE, ENTER THE NUMBER 0.
0
>
```

*

This operator, when typed immediately after a caret (>) or colon (:) indicates to the program that the current line is a comment. The program does not interpret the line.

PTWH — Is not used or changed.

ADVERBS — None

OBJECT

comment field — The comment field may contain any remarks or notes desired, but may not exceed the length of the line.

RELATED VERBS — None

ERRORS

If the asterisk appears in any column except the first, it is interpreted as a multiplication operator.

EXAMPLES

To include a comment on a display of reduced data, use the *:

```
* STACK OF SCANS  500-520,  BBASE = 12,  EBASE = 12,  NFIT = 2
```

?

An alias for PRINT. ? <field> prints the field requested on the CRT screen. <field> may be one or more variable or adverb names, literals, or arithmetic expressions. If <field> is an adverb or variable (a variable is an array or scalar defined in a procedure) name, the value(s) of the adverb or variable will be printed. Arrays of more than one dimension are printed with the first index varying most rapidly, as in Fortran. If <field> is a literal, it must be enclosed in quotation marks, and it will be printed without the quotation marks. If field is an arithmetic expression, the result of the expression will be printed.

PTWH — Is not used or changed.

ADVERBS — None.

OBJECTS

<field> — As described above.

RELATED VERBS

READ <field> — Reads the values requested by <field> from the CRT. In this case <field> may be one or more variable or adverb names, and may not include expressions or literals.

PRINT <field>
<field>? — Same as ? <field>.

PRINTER — Directs the PRINT output to the line printer.

CRT — Directs the PRINT output to the CRT screen. CRT is the default assignment, but must be invoked after PRINTER to return output to the CRT.

EXAMPLE

See PRINT

A D V E R B

D I C T I O N A R Y

ACOUNT

A counter for the ASTACK adverb. Determines how many scans are in the STACK. Is used by STACK verb, and in procedures.

ACOUNT is modified by

A ADD DELETE EMPTY

The initial value of ACOUNT is 0.

REMARKS

When ACOUNT = 0, the stack is empty.

EXAMPLE

Suppose you are observing in Tucson with a dual channel receiver, and you wish to average scans between 500 and 508 for a single receiver. After specifying:

```
TWO                 Sets the increment between scan numbers to two.
                    (Dual channel continuum)
EMPTY               Assures that the STACK is empty.
500  508  ADD       Inserts scans into the STACK.
```

the value of ACOUNT will be 5. The scans inserted into the stack are incremented by 2, so the output of STACK is

```
5
500  502  504  506  508
```

ASTACK

ASTACK

The STACK that is listed by STACK and used in procedures to operate on up to 100 scans.

ASTACK is modified by

A ADD DELETE EMPTY

The initial values of ASTACK are 0.

REMARKS

When ASTACK = 0, the stack is empty.

EXAMPLE

After specifying:

```
ONE                 Sets the increment between scan numbers.  In
                    Green Bank, the increment between scan numbers
                    is set to ten.  In Tucson, the increment is
                    one.
EMPTY               Assures that the stack is empty.
600  614  ADD       Inserts scans into the STACK.
```

STACK will produce the following output on the terminal:

```
600  601  602  603  604  605  606  607  608  609
610  611  612  613  614
```

BBASE
EBASE

BBASE EBASE

The number of points at the beginning and end of the data to be used for computation of a baseline. BBASE and EBASE are arrays, and may be set independently for each receiver. BBASE is the number of points at the beginning of the data; EBASE is the number of points at the end of the data. BBASE(CH) and EBASE(CH) will not be used if NREGION(CH,1) is different from zero.

BBASE and EBASE are used by

BASELINE BMODEL BSHAPE DCBASE RMS

The initial values of BBASE(CH) and EBASE(CH) are 25.

REMARKS

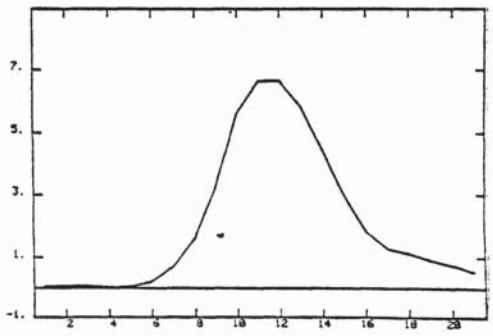
The values can be changed by the user as needed. No verbs change the values of BBASE and EBASE. BBASE and EBASE can be specified separately for each receiver.

EXAMPLE

Your data look like the display to the right, and you want to fit a baseline in the windows from channels 1 to 4 and from 17 to 21. Specify:

```
BBASE = 4; EBASE = 4
BASELINE
```

to remove a baseline fitted to the desired regions.



BDROP
EDROP

BDROP EDROP

The number of points at the beginning and end of the data which will be ignored by the routines listed below. BDROP and EDROP are arrays, and may be set independently for each receiver.

BDROP and EDROP are used by

SHOW	RESHOW	MAPSHOW	LABEL	GMODEL	CROSSFCN
BMODEL	BASELINE	BSHAPE	DCBASE	BMSWITCH	
BOXCAR	HANNING	SMOOTH	RESIDUAL		

The initial values of BDROP(CH) and EDROP(CH) are 0.

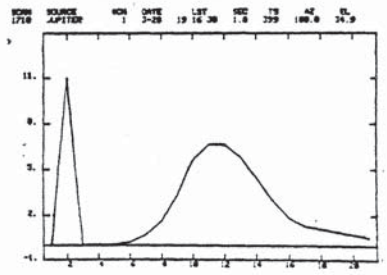
REMARKS

The values can be changed by the user as needed. No verbs change the values of BDROP and EDROP. BDROP and EDROP can be specified separately for the different receiver channels.

EXAMPLE

Your data look like the display to the right. To get rid of the spike at the left of the scan, drop some channels from the beginning of the scan:

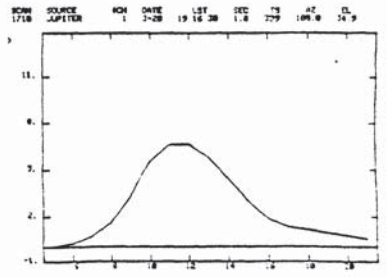
```
BDROP = 6
```



A new plot, produced with:

```
PAGE SHOW
```

will then show the scan without the spike at the beginning.



BEAMHW
BEAMSEP

BEAMHW BEAMSEP

BEAMHW is set to the half-width (in arcminutes) of the gaussian used by BMSWITCH in constructing a double gaussian model of a beam-switched response.

BEAMSEP is the separation of the two feeds (in arcmin) used in constructing the above model.

BEAMHW and BEAMSEP are used by

BMSWITCH [Green Bank only]

The initial values of BEAMHW and BEAMSEP are 0 and 0.

REMARKS

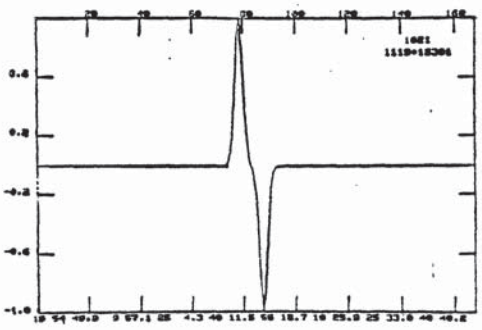
BMSWITCH will not reset BEAMHW and BEAMSEP. Note the values must be entered in arcmin.

EXAMPLE

You wish to construct a model of two gaussian beams, each of half-width 2.9', separated by 7.05'. Specify:

```
BEAMHW = 2.9: BEAMSEP = 7.05
BMSWITCH PAGE SHOW
```

to produce the result illustrated to the right.



BGAUSS
EGAUSS

BGAUSS EGAUSS

The explicit sample numbers between which a gaussian is fit by GAUSS.

BGAUSS and EGAUSS are used by

GAUSS GPARTS RESIDUAL

The initial values of BGAUSS and EGAUSS are 1 and 256.

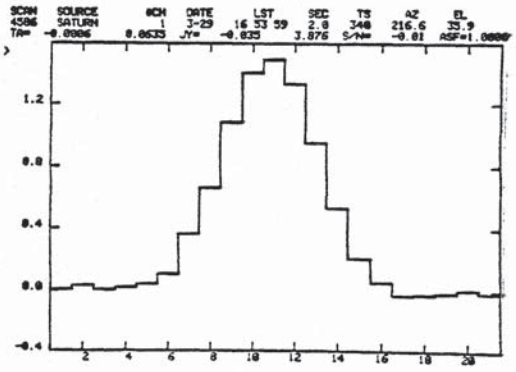
REMARKS

PEAK will set the values of BGAUSS and EGAUSS (see PEAK) or the user may specify their values before using GAUSS or GMODEL.

EXAMPLE

Your data look like the display on the right, and you want to fit a gaussian between channels 4 and 18. Specify:

```
NGAUSS = 1;
BGAUSS = 4; EGAUSS = 18
CENTER = 11; HWIDTH = 6
GAUSS
```



If the fit were successful you get a caret prompt. To print the parameters of the fit, use TELL GPARMS (Green Bank only), or specify:

```
PRINT CENTER(1) HEIGHT(1)
      HWIDTH(1)
```

to get:

```
10.82  1.521  5.141
```

in sample numbers and Kelvins.

BMARK

This adverb acts as a switch. If BMARK = 1, the verb SHOW will draw boxes indicating the baseline fitting regions last used or the regions in which an RMS was last computed. If BMARK = 0, the boxes will not be displayed.

BMARK is used by

SHOW

The initial value of BMARK is 0.

REMARKS

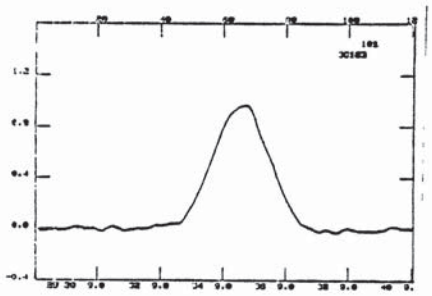
The user can change BMARK as desired. No verbs alter the value of BMARK. BMARK cannot be specified separately for each receiver.

EXAMPLE

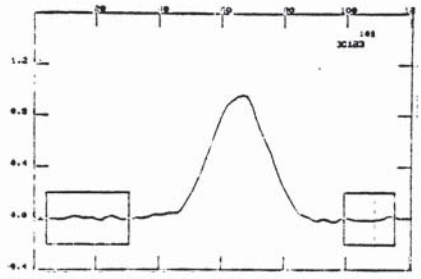
If BMARK = 0, then:

```
BASELINE PAGE SHOW
```

produces the display at right.



If BMARK = 1, however, the result of the same commands is shown at the right, where the boxes mark the regions used to calculate the baseline (or RMS).



BMOMENT
EMOMENT

BMOMENT EMOMENT

These adverbs specify the region of a scan which is used to compute a moment. BMOMENT is the first point of the region and EMOMENT is the last point of the region. BMOMENT and EMOMENT are arrays, and may be set independently for each receiver.

BMOMENT and EMOMENT are used by

MOMENT

The initial values of BMOMENT(CH) and EMOMENT(CH) are 0.

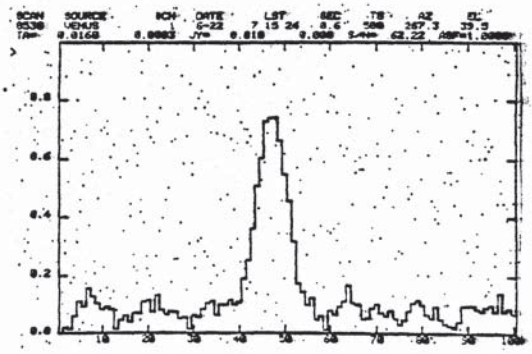
REMARKS

The user must specify values for BMOMENT and EMOMENT before using MOMENT. These adverbs are not changed by any verbs and may be specified separately for the different receiver channels.

EXAMPLE

Your data look like the display to the right, and you want to compute the area under the data points between samples 41 and 55. Specify:

```
NMOMENT = 0
BMOMENT = 41; EMOMENT = 55
MOMENT
```



BRANGE ERANGE (Green Bank only)

These are the minimum and maximum right ascensions (Epoch 1950.0, in hours) of scans to be loaded from a KEEP-tape.

BRANGE and ERANGE are used by

LOAD [Green Bank only]

The initial values of BRANGE and ERANGE are -1.

REMARKS

If BRANGE or ERANGE is set less than zero, then no check is made on the corresponding RA limit. BRANGE must be less than ERANGE. To LOAD scans in the RA range from 20 hours to 4 hours, first LOAD 20 to 24 hours and then LOAD zero to 4 hours.

EXAMPLE

You wish to read from your keep-tape all scans between a RA of 09h and 15h 30m. Specify:

```
BRANGE = 9.
ERANGE = 15.5
LOAD
```

BSCAN
ESCAN

BSCAN ESCAN (Green Bank only)

These are the mimimum and maximum scan numbers to be loaded from a KEEP-tape.

BSCAN and ESCAN are used by

LOAD [Green Bank only]

The initial values of BSCAN and ESCAN are -1.

REMARKS

If BSCAN or ESCAN are set less than zero, then no check is made on the corresponding scan numbers.

EXAMPLE

You wish to read from your KEEP-tape all scans between number 29300 and 29900. Specify:

```
BSCAN = 29300.
ESCAN = 29900.
LOAD
```

CENTER

An array of length six containing the sample numbers of the centers of up to six gaussians.

CENTER is used by

GAUSS GMODEL GPARTS RESIDUAL REGAUSS

The initial values of CENTER(N) are 0.

REMARKS

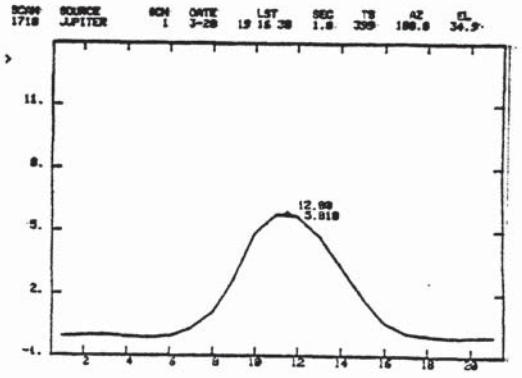
The user must specify NGAUSS values of CENTER before calling GAUSS. These values will be iterated by the routine GAUSS unless the adverb FIXC is less than zero (see FIXC). The values of CENTER (usually refined by GAUSS) are used by GMODEL, GPARTS, and RESIDUAL. PEAK will find the approximate center of a single gaussian.

EXAMPLE

Your data look like the display on the right. Using CROSSHAIR, you have determined that the center of the desired gaussian is near sample 12. Specify:

```
NGAUSS = 1
CENTER = 12
```

Before using GAUSS, however, you must also specify HWIDTH, and may have to change BGAUSS and EGAUSS.



CLIPMIN
CLIPMAX

CLIPMAX CLIPMIN

These adverbs are the maximum and minimum temperatures (in Kelvin) that will be found in the data after it is CLIPped. CLIPMIN and CLIPMAX are arrays, and may be set independently for each receiver.

CLIPMAX and CLIPMIN are used by

CLIP

The initial values of CLIPMAX(CH) and CLIPMIN(CH) are 99999.9 and -99999.9 respectively.

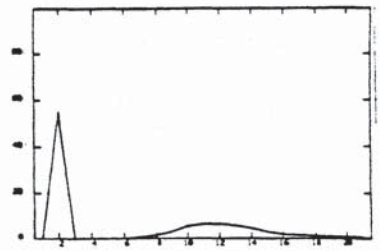
REMARKS

The user may specify any values for CLIPMAX and CLIPMIN. No verbs change the values of CLIPMAX and CLIPMIN. CLIPMAX and CLIPMIN may be specified separately for the different receiver channels.

EXAMPLE

Your data look like the display to the right. The interference spike at the left has overwhelmed the y-axis scaling chosen by SHOW. To eliminate this problem, specify:

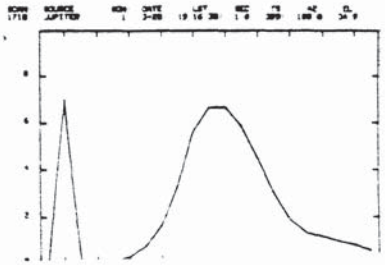
```
CLIPMIN = 0; CLIPMAX = 7
CLIP
```



After CLIPping, specify:

```
PAGE SHOW
```

to produce the display at right.



CMARK

An array of length six containing the sample numbers at which to draw a vertical line when a scan is displayed. Up to six different points may be specified.

CMARK is used by

SHOW

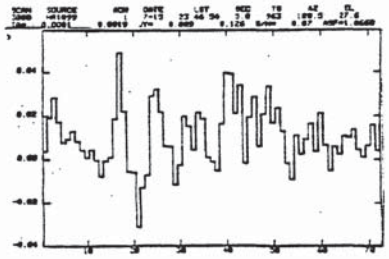
The initial values of CMARK(N) are 0.

REMARKS

If CMARK(N) is less than or equal to zero, no vertical line will be drawn. The user may change the values of CMARK. No verbs change CMARK.

EXAMPLE

Your data look like the display to the right. You want to draw a vertical line down the center of your scan, which is sample 40 in this case, and through sample 17. Specify

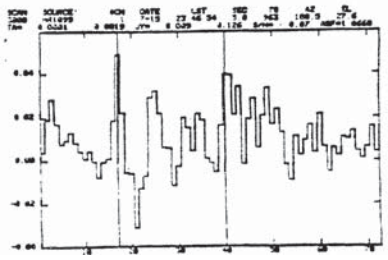


```
CMARK(1) = 40
CMARK(2) = 17
```

or

```
CMARK = 40, 17
```

(the comma is necessary). Then a subsequent SHOW will produce the display to the right.



DCPCT

DCPCT

The percentage of points to be made negative when you fit a baseline with PCBASE.

DCPCT is used by

PCBASE

The initial value of DCPCT is 25.

EXAMPLE

You want to subtract a baseline value from your data such that 30% of the data points will be negative. Before using PCBASE, specify:

```
DCPCT = 30.
```

FACT

An array of numbers supplied by the user which may be used:

(1) by SCALE to multiply the data points for each receiver by a constant, or

(2) by BIAS to add a constant to the data points for each receiver.

FACT is used by

SCALE BIAS

The initial values of FACT(CH) are 1.0.

REMARKS

FACT may be set by the user. It is not changed by any verbs. It may be specified separately for the different receiver channels.

EXAMPLES

(1) You want to multiply a scan by .001 so you specify:

```
FACT = .001
SCALE
```

(2) You want to subtract ½°K from a scan so you specify:

```
FACT = -0.5
BIAS
```

FIXC
FIXHW

FIXC FIXHW

These adverbs are used as switches. If FIXC or FIXHW is less than 0, then the gaussian routines will not iterate the user's guesses for the centers or the halfwidths of the gaussians, respectively.

FIXC and FIXHW are used by

GAUSS REGAUSS

The initial values of FIXC and FIXHW are 1.

REMARKS

For these default values, GAUSS will iterate the values of CENTER and HWIDTH specified by the user. For values of FIXC or FIXHW less than zero, GAUSS will not iterate the CENTER or HWIDTH values. FIXC and FIXHW may be changed by the user and are not set by any verb.

EXAMPLE

If you want to fit a gaussian with center at sample 79, specify:

```
CENTER = 79
FIXC   = -1
```

(FIXC = -1 will fix all the center values if you are fitting more than one gaussian.)

GCRV
GGRID
GMIN

GCRV GGRID GMIN

GCRV(N) A table of the gain calibration values spaced at equal values of declination. The values could be given in units of flux density per Kelvin, relative gain, etc. Must contain 12 entries.

GGRID The declination spacing between adjacent entries in the GCRV table in decimal degrees.

GMIN The minimum declination of the scale factors in the gain table in decimal degrees.

GCRV, GGRID AND GMIN are used by

GAINCRV

The initial values of GCRV(N), GGRID and GMIN are 0, 1, and 1, respectively.

REMARKS

GAINCRV will not reset these adverbs.

EXAMPLE

You have measured your gain table in five degree steps of declination starting from 20° and going up to 75°. Enter adverb values as follows (values are only illustrative):

```
GCRV = 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1
GGRID = 5.
GMIN = 20.
```

HEIGHT

HEIGHT

An array of length six containing the heights (in Kelvins) of up to six gaussians. First guesses for the values of HEIGHT are determined by GAUSS. Therefore, HEIGHT need not be specified before GAUSS is called. The values of HEIGHT are used by GMODEL, GPARTS, REGAUSS, and RESIDUAL.

HEIGHT is used by

GMODEL GPARTS RESIDUAL REGAUSS

The initial values of HEIGHT(N) are 0.

REMARKS

The values of HEIGHT are set each time GAUSS is called. If GAUSS is not called before GMODEL, GPARTS, REGAUSS or RESIDUAL, NGAUSS values of HEIGHT should be set by the user. PEAK will set HEIGHT to the highest data value in the scan, excluding points dropped from the ends.

EXAMPLE

To fit one gaussian, you may specify:

```
CENTER = 79
HWIDTH = 12
NGAUSS =  1
```

The value of HEIGHT will be calculated by GAUSS. After GAUSS, the calculated value of HEIGHT may be printed by

```
PRINT HEIGHT(1)
```

or with TELL GPARMS (Green Bank only).

HWIDTH

An array of length six containing the widths at half the peak height (in number of points) of up to six gaussians.

HWIDTH is used by

GAUSS GMODEL GPARTS RESIDUAL REGAUSS

The initial values of HWIDTH(N) are 0.

REMARKS

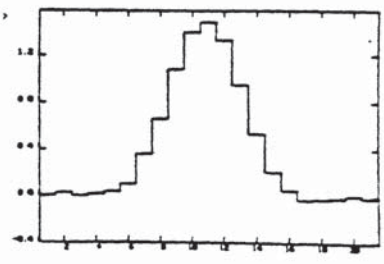
The user must specify the values of HWIDTH before calling GAUSS. The values of HWIDTH will be iterated by GAUSS unless the value of the adverb FIXHW is less than zero. (See FIXC FIXHW.) The values of HWIDTH (usually refined by GAUSS) are used by GMODEL, GPARTS, and RESIDUAL. PEAK will set the approximate HWIDTH of a single gaussian.

EXAMPLE

Your data look like the display to the right. From the display, you determine that the half-width is near six channels. Specify:

```
HWIDTH(1) = 6
```

Also specify the CENTER before using GAUSS to refine your guesses.



LEVS

The temperatures (in degrees Kelvin) at which a contour is to be plotted by MAPSHOW. Up to twenty contour levels (in monotonically increasing order) may be specified.

LEVS is used by

MAPSHOW LABEL

The initial values of LEVS (M) are -1 E+5

REMARKS

The user must specify the levels he wants drawn before he invokes MAPSHOW. No verb changes the values of LEVS. The first contour level beyond the highest level you wish to plot, should be less than -999.

EXAMPLE

You want to get a contour map with contours at every half degree from 1. to 3,

You specify

```
LEVS(1) = 1
LEVS(2) = 1.5
LEVS(3) = 2
LEVS(4) = 2.5
LEVS(5) = 3
LEVS(6) = -1000
```

or more conveniently specify

```
LEVS = 1, 1.5, 2, 2.5, 3, -1000
```

(the commas are necessary).

MRATIO

The ratio between x and y scaling of contour maps.

MRATIO is used by

MAPSHOW LABEL

The initial value of MRATIO is 1.

REMARKS

The size of a contour map along the x-axis is fixed. The size in the y-axis direction is determined with respect to the x-axis. If your map is too large in the y-axis direction, make MRATIO smaller. No verbs change MRATIO.

NBOX

NBOX

An odd number of points to be averaged together by the BOXCAR baseline removal routine, or sorted by the MDBASE baseline removal routine (Green Bank only).

NBOX is used by

BOXCAR MDBASE

The initial value of NBOX is 3.

REMARKS

The user may change NBOX to any other positive odd integer. Attempting to use BOXCAR or MDBASE when NBOX is even will result in an error message. No verbs set NBOX and it cannot be specified separately for each receiver.

EXAMPLE

You want to average seven points together for the purpose of smoothing your data. Before you BOXCAR, specify:

```
NBOX = 7.
```

NFIT

The order of the polynomial baseline to be computed.

NFIT is used by

BASELINE BMODEL BSHAPE

The initial value of NFIT is 1.

REMARKS

The user may specify any other value for NFIT between 1 and 12. NFIT is not changed by any verbs and cannot be specified separately for each receiver.

EXAMPLE

You want to remove a cubic baseline from your data; before you use a polynomial baseline routine, specify:

```
NFIT = 3.
```

NGAUSS

NGAUSS

The number of gaussians to be fit by GAUSS or evaluated by GMODEL, GPARTS, or RESIDUAL.

NGAUSS is used by

GAUSS GMODEL GPARTS RESIDUAL REGAUSS

The initial value of NGAUSS is 1.

REMARKS

The user may specify a value of NGAUSS between 1 and 6. No verbs change NGAUSS. If GAUSS is to be called next, NGAUSS values of CENTER and HWIDTH should be specified. If GMODEL or RESIDUAL is to be called next, NGAUSS values of CENTER, HWIDTH and HEIGHT, should be specified.

EXAMPLE

To fit or model three gaussians, specify:

```
NGAUSS = 3.
```

NITER

The number of iterations which a gaussian fitting routine will attempt before giving up.

NITER is used by

GAUSS REGAUSS

The initial value of NITER is 8.

REMARKS

The user may change NITER if desired. NITER is not changed by any verbs.

EXAMPLE

Your attempt to fit a gaussian failed although you were certain your guesses for HEIGHT, CENTER and HWIDTH were accurate. Specify:

```
NITER = 12
```

and try again. Use REGAUSS if the values of HEIGHT are reasonable; otherwise, use GAUSS to recompute initial guesses for the heights.

NMOMENT

NMOMENT

The order of the moment to be calculated by the verb MOMENT. If NMOMENT = 0, the area under the curve (between BMOMENT and EMOMENT) will be calculated. Higher moments are not currently implemented.

NMOMENT is used by

MOMENT

The initial value of NMOMENT is 0.

REMARKS

No other values of NMOMENT are allowed at this time. No verbs change NMOMENT and it may not be specified separately for each receiver.

EXAMPLE

See MOMENT

NPOINTS

An adverb set to the number of data points in the currently referenced array by the verb SLENGTH.

NPOINTS is used by

SLENGTH

The initial value of NPOINTS is 0.

REMARKS

If there is no facility in the CONDAR system to do the kind of data manipulation that you desire, and you resort to direct manipulation of the data in the TWH array, you will need to know how many data points are present to control the range of the loop that processes the data. Since the number of data points is stored in the TWH array as an integer, it cannot be read directly from there. The verb SLENGTH and the adverb NPOINTS are therefore provided for this purpose.

EXAMPLE

Suppose you are sitting in Green Bank making up examples for this manual. You decide to give the scan in the WORK array a sloping baseline so you can demonstrate the effect of baseline removal. Your baseline is to span about one degree from beginning to end, and is constructed in the TEMP array. Consulting the Appendix, you find that in Green Bank, the first data sample is in real word 88 of the TWH array. You define the following procedure:

```
PROC SLOPE
SLENGTH
N2 = 87+NPOINTS
FOR I = 88 TO N2
   TWH( I,1 ) = I - 88
   END
FACT = 1/NPOINTS; SCALE
FINISH
```

NREGION

NREGION

NREGION is a two-dimensional array containing pairs of beginning and ending sample numbers for up to four data windows for each receiver. The first array index is the receiver number for the scan in the currently referenced array. The second array index labels the list of beginning and ending window samples, and runs from 1 to 8. These data windows are used primarily for baseline fitting.

NREGION is used by

BASELINE BMODEL BSHAPE DCBASE
RMS

The initial values of NREGION (CH,I) are 0.

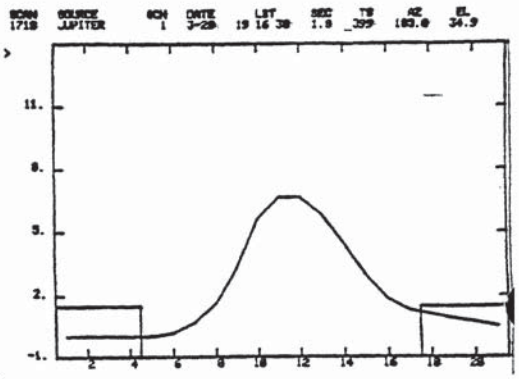
REMARKS

NREGION has two dimensions so that baseline fitting windows may be specified independently for each receiver. NREGION is not set by any verbs.

Beginning and ending channel numbers must be assigned consecutively along the second dimension of the array. If the second index to NREGION is odd, the value of that element is a beginning point of a region. If the second index is even, the value of that element is an ending point of a region, which must be larger than the beginning point. If the begining point of the first pair for a given receiver is zero, then the routines mentioned above will look at BBASE and EBASE, rather than NREGION, to define baseline fitting windows.

-continued-

EXAMPLE



Your data look like the display to the right. You want to fit a baseline through samples indicated by boxes, which are: 1 to 4 and 18 to 21. To specify these windows for the first receiver:

```
NREGION( 1,1 ) = 1
NREGION( 1,2 ) = 4
NREGION( 1,3 ) = 18
NREGION( 1,4 ) = 21
```

Setting the same windows for the second receiver could be done by:

```
NREGION( 2,1 ) = 18
NREGION( 2,2 ) = 21
NREGION( 2,3 ) = 1
NREGION (2,4 ) = 4
```

since the order in which pairs are listed does not matter. To go back to using BBASE and EBASE, specify:

```
NREGION( 1,1 ) = 0
NREGION( 2,1 ) = 0
```

specifically, or more compactly:

```
NREGION = 0
```

NSAVE

NSAVE

The disk bin into which a scan is to be SAVEd or from which it is to be RECALLed. NSAVE is the index to a disk file containing copies of the scan registers in memory.

NSAVE is used by

SAVE RECALL

The initial value of NSAVE is 0.

The user must select a save bin between 1 and 80 (Green Bank) or 154 (Tucson) before using SAVE or RECALL. NSAVE is not changed by any verb.

EXAMPLE

You have reduced scan which you wish to save so that you can show it to your colleague when he returns from coffee. Thirty four is your favorite number. Specify:

```
NSAVE = 34
SAVE
```

When your colleague arrives, you can recall the scan as follows,

```
NSAVE = 34            (in case you had reset its value)
RECALL
```

PTWH

The adverb used by the program as a pointer to the "currently referenced array:"

```
PTWH = 1  points to the TEMP array,
PTWH = 2  points to the WORK array, and
PTWH = 3  points to the HOLD array.
```

The pointer may be changed by an assignment statement (e.g., PTWH=2), or by using the verbs PMT, PMW, and PMH, which are acronyms for "pointer moves to (array name)."

PTWH is used by

PMT PMW PMH,

and all verbs which use or modify the contents of the currently referenced array. See the verb dictionary for the effect of a verb in question.

The initial value of PTWH is 2.

EXAMPLES

To find out which array is the currently referenced array:

```
PRINT PTWH
2.000
```

In this case the result indicates that the WORK array is the currently referenced array.

NSAVE

NSAVE

The disk bin into which a scan is to be SAVEd or from which it is to be RECALLed. NSAVE is the index to a disk file containing copies of the scan registers in memory.

NSAVE is used by

SAVE RECALL

The initial value of NSAVE is 0.

The user must select a save bin between 1 and 80 (Green Bank) or 154 (Tucson) before using SAVE or RECALL. NSAVE is not changed by any verb.

EXAMPLE

You have reduced scan which you wish to save so that you can show it to your colleague when he returns from coffee. Thirty four is your favorite number. Specify:

```
NSAVE = 34
SAVE
```

When your colleague arrives, you can recall the scan as follows,

```
NSAVE = 34            (in case you had reset its value)
RECALL
```

PTWH

The adverb used by the program as a pointer to the "currently referenced array:"

```
PTWH = 1  points to the TEMP array,
PTWH = 2  points to the WORK array, and
PTWH = 3  points to the HOLD array.
```

The pointer may be changed by an assignment statement (e.g., PTWH=2), or by using the verbs PMT, PMW, and PMH, which are acronyms for "pointer moves to (array name)."

PTWH is used by

PMT PMW PMH,

and all verbs which use or modify the contents of the currently referenced array. See the verb dictionary for the effect of a verb in question.

The initial value of PTWH is 2.

EXAMPLES

To find out which array is the currently referenced array:

```
PRINT PTWH
2.000
```

In this case the result indicates that the WORK array is the currently referenced array.

SINCR

SINCR

SINCR is the increment between scan numbers being put on the stack by the verb ADD. It is meant to be set to the scan number increment between scans from the same receiver. In Green Bank, this increment is always ten. In Tucson this increment depends on the number of receivers in use, and may be one or two; if there are two receivers, then every other scan belongs to a given receiver.

SINCR is used by

ADD ONE TWO

The initial value of SINCR is 10.0 in Green Bank
The initial value of SINCR is 1.0 or 2.0 in Tucson, depending on the receiver.

REMARKS

In Tucson, use ONE to set SINCR for one receiver operation, and use TWO to set SINCR for two receiver operation.

SIZE

SIZE is an array used by the verb MOMENT to return its results. The array index is the receiver in use.

SIZE is used by

MOMENT

The initial value of SIZE is 0.

REMARKS

After using the verb MOMENT, you must

```
PRINT SIZE
```

to see the result of the calculation.

EXAMPLE

See MOMENT.

SMWGT

SMWGT

An array of length twelve containing a template of weights for data smoothing. The verb SMOOTH smooths a scan by convolving the template with the data. The first element of SMWGT is the number of points in the weighting function (not greater than 11). The following elements define the template of weights. The sum of the weights should be one.

SMWGT is used by

SMOOTH

The initial values of SMWGT are 0.

EXAMPLE

To mimick HANNING smoothing, where the template spans three data points, specify:

```
SMWGT( 1 ) = 3
SMWHT( 2 ) = 0.25
SMWGT( 3 ) = 0.50
SMWGT( 4 ) = 0.25
```

or more compactly:

```
SMWGT = 3, 0.25, 0.50, 0.25
```

(where the commas are necessary).

For other examples of smoothing templates, see the verb SMOOTH. To see the effect of one kind of smoothing, see the verb BOXCAR.

SUMT

SUMT is an adverb used by the verb RMS to return the sum of the temperatures of the points used to compute the rms.

SUMT is used by

RMS

The initial value of SUMT is 0.

REMARKS

Finding the total area under a number of data windows may be done more conveniently using the adverb array NREGION and the verb RMS, with the result going to the variable SUMT, compared to using BMOMENT, EMOMENT, and the verb MOMENT.

SUMT2

SUMT2

SUMT2 is an adverb used by the verb RMS to return the sum of the squares of the temperatures of the points used to compute the rms.

SUMT2 is used by

RMS

The initial value of SUMT2 is 0.

TMARK

An array of length six containing the temperatures at which to draw a horizontal line when a scan is displayed. Up to six different temperatures may be specified.

TMARK is used by

SHOW

The initial values of TMARK(N) are -9999

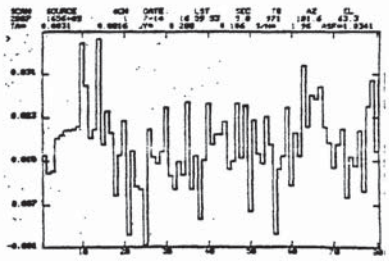
REMARKS

When TMARK is less than zero, no horizontal lines will be drawn. The user may change the values of TMARK. No verbs change TMARK.

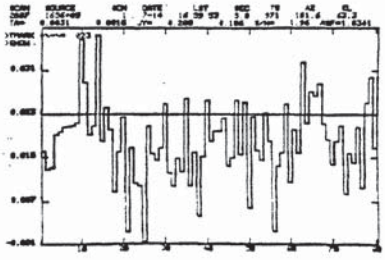
EXAMPLE

Your data look like those to the right. You want to verify the line temperature associated with a given tick mark on the display. Specify:

```
TMARK(1) = .023
SHOW
```



Your display will then look like the one to the right.



TWH

TWH

TWH is a two dimensional array which constitutes the three scan registers or arrays in memory. The second index runs from one to three, and is the scan pointer. The program uses the pointer PTWH to refer to the currently referenced array. The first index is the real word in the scan. The content of each word in the scan is given in this manual under the heading Disk Format of Continuum Scans. Only the real words in these formats can be accessed; integer and character fields cannot be changed and can be read only to the extent provided for through special routines, like HEADER and SLENGTH.

TWH is used by

All routines that refer to data in scans.

The initial values of TWH are 0.

Attempting to perform an operation requiring data in TWH before any data have been loaded results in the error message:

```
LOAD DATA!
```

REMARKS

Except for the restriction to real words in the disk formats, direct access to header and data values in the TWH array gives you the ability to perform almost any data manipulation.

EXAMPLE

Suppose you are in Green Bank, observing on the 140-foot telescope. You want to scale up your data for atmospheric attenuation, which depends on zenith distance z as

```
exp( tau / cos(z) )
```

where tau is the optical depth at zenith. You find that the zenith distance is in word 39, so the scale factor is:

```
FACT = EXP( TAU/ COS( TWH( 39,PTWH ) ) )
```

VRMS

VRMS is an array used by the verb RMS to return the rms values. The array index is the receiver in use.

VRMS is used by

RMS

The initial value of VRMS is 0.

REMARKS

The rms results are printed by RMS, but are available for calculations through VRMS.

EXAMPLE

Suppose you are observing with two receivers, and scans 151 and 152 are on receivers one and two. You decide to average the two scans, weighted by the inverse of the square of the rms noise in each scan. First set windows for RMS using NREGION or BBASE and EBASE. The windows may be set independently for the two receivers. Use the verb SUM and the array WEIGHT to take a weighted average:

```
GET 151 RMS; WEIGHT = 1/VRMS( 1 )**2; SUM
GET 152 RMS: WEIGHT = 1/VRMS( 2 )**2; SUM AVE
```

WEIGHT

WEIGHT

WEIGHT is an array of weighting values used to accumulate scans using non-standard weighting. The value of WEIGHT multiplies the default weighting of scans, but the CONDAR program default is to weight all scans equally.

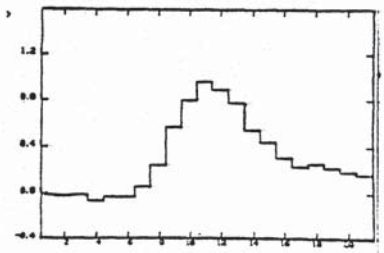
WEIGHT is used by

SUM

The initial values of WEIGHT(N) are 0.

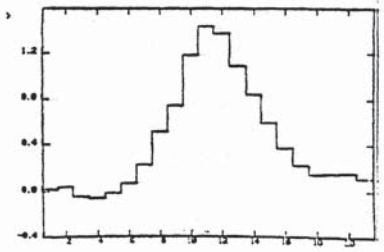
EXAMPLE

Scan 1500, observed with the first receiver, looks like the top display on the right. The peak source temperature is about 0.95 K.



Scan 1501, observed with the second receiver, looks like the middle display on the right. The peak source temperature is about 1.45 K. To add these two scans together, weighting scan 1501 by twice the weight of scan 1500, specify:

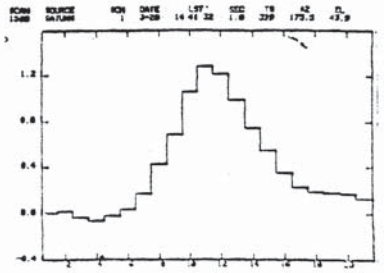
```
WEIGHT(1) = 1; WEIGHT(2) = 2
GET 1500 SUM; GET 1501 SUM
```



To see the average, specify:

```
AVE PAGE SHOW
```

to produce the bottom display to the right. The peak source temperature is about 1.3 K, confirming that the second scan was given more weight.



XWIND (Green Bank only)

XWIND is a two-element array containing the minimum and maximum x-axis coordinates used to window scans. They are to be set in radians. The purpose is to provide a way to window a group of scans to map only the x-coordinates spanned by all of the scans.

XWIND is used by

BESTVIEW WINDOW [Both are Green Bank only]

The initial values of XWIND(N) are 0.

The verb BESTVIEW is used to see if the scan in the currently referenced array requires more restrictive values of XWIND, and to so set XWIND, if necessary. Before invoking BESTVIEW, XWIND(1) and XWIND(2) should be set to values less than and greater than, respectively, the miminum and maximum value expected.

EXAMPLE

Since XWIND is set in units of radians, before entering BESTVIEW one could safely specify:

```
XWIND(1) = -9
XWIND(2) = 9
```

YINCR
YMIN

YINCR YMIN

Are arrays used to set the Y scaling of the data graphs. YINCR equals the number of Kelvins that will appear between each of six y-axis tick mark. YMIN equals the lowest number of Kelvins that will appear on the graph. The array index is the receiver number.

YINCR and YMIN are used by

SHOW RESHOW

The initial values of YINCR (CH) are 0.
The initial values of YMIN (CH) are -99999.9.

REMARKS

If YMIN is less than or equal to -9999, SHOW will automatically set the y-axis scaling. If you set YMIN to greater than -9999, SHOW will use your values of YMIN and YINCR. So if you set YMIN you must also set YINCR.

Four verbs set the value of YMIN. FREEY and AUTO set YMIN to greater than -9999, so that SHOW and RESHOW will set the y-axis scaling. HOLDY sets YMIN to a large negative number (-9E10) so that SHOW will not update its y-axis scaling. RANGE sets the values of YMIN and YINCR according to specified minimum and maximum temperatures to be plotted.

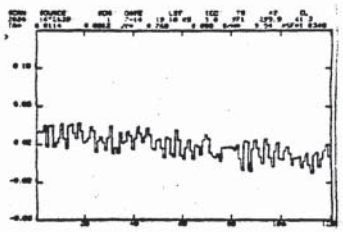
If set, you must set both YMIN and YINCR. YINCR and YMIN may be specified separately for each receiver.

-continued-

YINCR
YMIN

EXAMPLES

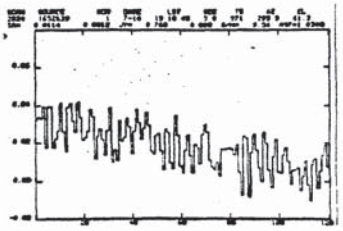
Suppose SHOW produced the top display to the right.



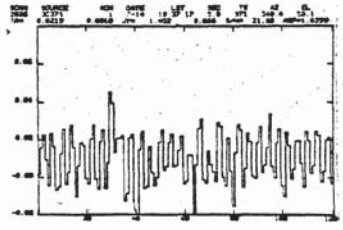
To expand the vertical scale, specify:

```
YMIN = -.02
YINCR = .02
PAGE SHOW
```

to produce the middle display on the right.



Now you want to compare two different scans but you are frustrated because SHOW plots them on different scales. To get the second graph to the same scale, specify HOLDY before you specify SHOW for the second scan to produce the bottom display on the right.



To get back to automatic scaling, specify FREEY or AUTO.

ZLINE

ZLINE

This adverb is used as a switch. If ZLINE = 1, a horizontal line is drawn at zero Kelvins. If ZLINE = 0, the line is not drawn.

ZLINE is used by

SHOW

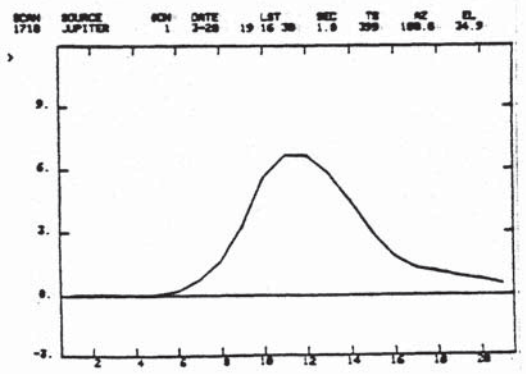
The initial value of ZLINE is 0.

REMARKS

The user may change ZLINE as desired. No verbs change ZLINE.

EXAMPLE

If ZLINE = 1, SHOW will draw the zero line as shown:



Green Bank Format of CONTINUUM Scans on Disk

CONDAR

10 April 1984

Integer ITWH	Real TWH	Double DTWH	Type	Contents	Units
1-2	1		R*4	Scan Number	
3-4	2		R*4	Subscan Indicator	
5-10			I*2	Source Name (12)	
11			I*2	Observer's Number	
12-21			I*2	Observer's Name (20)	
22-24		8	DR	Julian Date	
25			I*2	Solar Day of Year	
26			I*2	Month	
27			I*2	Day	
28			I*2	Year	
29-30	15		R*4	LST	Rad
31-32	16		R*4	EST	Rad
33			I*2	Telescope	
34			I*2	Observing Program	= 3
35			I*2	Scan type code	
36			I*2	MODE	
37			I*2	First Channel	
38			I*2	Number of Channels	
39			I*2	Points this scan	
40			I*2	START	
41			I*2	STOP	
42			I*2	Map START	
43			I*2	Map STOP	
44-47			I*2	RESERVED TO ALIGN WITH LINE	
48			I*2	VREF	
49			I*2	VDEF	
50			I*2	Position Code	
51			I*2	Scan Direction (H-V)	
52			I*2	Total subscans	
53			I*2	Feed Number	
54			I*2	Total subcans	
55			I*2	Status word 1	
56			I*2	Status word 2	
57-58	29		R*4	Sample rate	Sec
59-60	30		R*4	LAMBDA	
61-62	31		R*4	Beam Size Factor	Rad
63-64	32		R*4	Epoch of Obs	Years
65-66	33		R*4	H Coordinate Rate	
67-68	34		R*4	V Coordinate Rate	
69-70	35		R*4	RA Indicated	Rad
71-72	36		R*4	DEC Indicated	Rad
73-74	37		R*4	Focus	mm

75-76	38	R*4	Orientation	Rad
77-78	39	R*4	Zenith distance	Rad
79-84	40-42	R*4	Descriptive Origin (3)	
85-90	43-45	R*4	Unused	
91-106		I*2	Environment Values (16)	
107-108	54	R*4	Epoch RA	Rad
109-110	55	R*4	Epoch DEC	Rad
111-112	56	R*4	Apparent RA	Rad
113-114	57	R*4	Apparent DEC	Rad
115-116	58	R*4	Galactic l	Rad
117-118	59	R*4	Galactic b	Rad
119-120	60	R*4	Observed H Position	Rad
121-122	61	R*4	Observed V Position	Rad
123-124	62	R*4	Horiz. Description	Rad
125-126	63	R*4	Vert. Description	Rad
127-128	64	R*4	RHO	Rad
129-130	65	R*4	THETA	Rad
131-132	66	R*4	Noise Tube	K
133-134	67	R*4	Cal Value	V
135-136	68	R*4	Cal Factor	K/V
137		I*2	Unused	
138		I*2	# scans accumed	
139-174	70-87	R*4	Accum stack	
175-256	88-1280	R*4	Data Points	

All pointers marked with a star (*) are one less than the actual word.

All pointers marked with lB sign (#) are different from the telescope tape format due to reformatting by feed when the foreground program writes the data to the analysis disk.

Green Bank Format of CONTINUUM On-Off Scans on Disk

CONDAR

10 April 1984

Integer ITWH	Real TWH	Double DTWH	Type	Contents	Units
1-2	1		R*4	Scan Number	
3-4	2		R*4	Subscan Indicator	
5-10			I*2	Source Name (12)	
11			I*2	Observer's Number	
12-21			I*2	Observer's Name (20)	
22-24		8	R*8	Julian Date	
25			I*2	Solar Day of Year	
26			I*2	Month	
27			I*2	Day	
28			I*2	Year	
29-30	15		R*4	LST	Rad
31-32	16		R*4	EST	Rad
33			I*2	Telescope	
34			I*2	Observing Program	= 4
35			I*2	Scan type code	
36			I*2	MODE	
37			I*2	First Channel	
38			I*2	Number of Channels	
39			I*2	Points this scan	
40			I*2	START	
41			I*2	STOP	
42			I*2	Map START	
43			I*2	Map STOP	
44-47			I*2	RESERVED TO ALIGN WITH LINE	
48			I*2	VREF	
49			I*2	VDEF	
50			I*2	Position Code	
51			I*2	Scan Direction (H-V)	
52			I*2	Subscan Number	
53			I*2	Feed Number	
54			I*2	Total subscans	
55			I*2	Status word 1	
56			I*2	Status word 2	
57-58	29		R*4	Sample rate	Sec
59-60	30		R*4	LAMBDA	
61-62	31		R*4	Beam Size Factor	Rad
63-64	32		R*4	Epoch of Obs	Years
65-66	33		R*4	H Coordinate Rate	
67-68	34		R*4	V Coordinate Rate	

69-70	35	R*4	RA Indicated	Rad
71-72	36	R*4	DEC Indicated	Rad
73-74	37	R*4	Focus	mm
75-76	38	R*4	Orientation	Rad
77-78	39	R*4	Zenith distance	Rad
79-83	40-42	R*4	Descriptive Origin (3)	
85-90	43-45	R*4	Unused	
91-106		I*2	Environment Values (16)	
109-110	55	R*4	Epoch DEC	Rad
111-112	56	R*4	Apparent RA	Rad
113-114	57	R*4		Rad
115-116	58	R*4	Galactic l	Rad
117-118	59	R*4	Galactic b	Rad
119-120	60	R*4	Observed H Position	Rad
121-122	61	R*4	Observed V Position	Rad
123-124	62	R*4	Horiz. Descriptive	Rad
125-126	63	R*4	Vert. Descriptive	Rad
127-128	64	R*4	RHO	Rad
129-130	65	R*4	THETA	Rad
131-132	66	R*4	Noise Tube	K
133-134	67	R*4	Cal Value	V
135-136	68	R*4	Cal Factor	K/V
137		I*2	Unused	
138		I*2	# scans accumed	
139-174	70-87	R*4	Accum stack	
175-256	88-1280	R*4	Data Points	

All pointers marked with a star (*) are one less than the actual word.

All pointers marked with lb sign (#) are different from the telescope tape format due to reformatting by feed when the fore-ground program writes the data to the analysis disk.

CODE DESCRIPTIONS

Scan type code 0=no cal 1=pulsed cal 2=timed cal

ENVIRONMENTAL VALUES
1=LOCAL TEMPERATURE
2,3,4=UNUSED
5=DEWPOINT
6=OUTSIDE TEMPERATURE
7=AIR PRESSURE
8-16=UNUSED

Tucson Format of CONTINUUM Scans on Disk

CONDAR

10 April 1984

Integer ITWH	Real TWH	Double DTWH	Type	Contents	Units
	1		R*4	Scan Number	
3-8			I*2	Source Name (12)	
9-18			I*2	Obsr. Initials (4)	
				Opr. Initials (4)	
19			I*2	Channel Number	
20			I*2	# Blocks for this Scan	
		6	R*8	Julian Date	
25			I*2	Month	
26			I*2	Day	
27			I*2	Year	
28			I*2	Not Used	
	15		R*4	LST	Rad
	16		R*4	UT	Rad
33			I*2	Telescope	
34			I*2	Type of Observing	
35			I*2	Scan Type	
36			I*2	Number of Points	
37			I*2	START	
38			I*2	STOP	
39			I*2	Number of Samples	
40			I*2	Number of Rows	
41 - 46			I*2	Not Unsed	
	24		R*4	Scan Time	Sec
	25		R*4	Not Used	
	26		R*4	Sample Rate	Sec
	27		R*4	% Efficiency	%/100
	28		R*4	Azimuth Offset	Sec Arc
	29		R*4	Elevation Offset	Sec Arc
	30		R*4	Azimuth	Rad
	31		R*4	Elevation	Rad
	32		R*4	Focus Offset	MM
	33		R*4	RMS	
	34		R*4	Attenuation	%/100
	35		R*4	Tolerance	Sec Arc
	36		R*4	Baseline	Deg K
	37		R*4	Scanning Angle	Rad
	38		R*4	Cell size	Sec Arc
	39		R*4	HP	Sec Arc
	40		R*4	Not Used	

Integer ITWH	Real TWH	Double DTWH	Type	Contents	Units
	41-42		R*4	Five Point Offsets	Sec Arc
	43-44		R*4	Reference Offsets	Sec Arc
	45		R*4	#C	
	46		R*4	#CP	
	47		R*4	Commanded RA	Rad
	48		R*4	Commanded DEC	Rad
	49		R*4	TC	Deg K
	50		R*4	System Temperature	Deg K
	51		R*4	Source Temperature	Deg K
	52		R*4	Source Temperature RMS	
	53		R*4	Bandwidth	MHz
	54-125		R*4	Not Used	
	126		R*4	Wavelength	MM
	127		R*4	Ambient Temperature	Deg C
	128		R*4	Elevation Axle Temp.	Deg C
	129-384		R*4	Data Values	Deg K

INDEX