SPECtrum Processing Routines

Users Manual

Version 3

(program SPECPR)

by

Roger N. Clark

U.S. Geological Survey, MS 964
Box 25046 Federal Center
Denver, CO 80225-0046
(303) 236-1332

rclark@speclab.cr.usgs.gov

USGS Denver Spectroscopy Lab
(303) 236-1411

USGS Open File Report no. 93-595
December 1993

TABLE OF CONTENTS

Acknowledgements

This manual documents the specpr program. However, the specpr program itself is the product of many people, some professional programmers, others scientists and students. The effort to write this manual pales in comparison to the programming time spent by myself and many others. Without these people, specpr would have died long ago.

Rodney Kam and Jeff Hoover were programmers at the University of Hawaii who shaped the software into the Unix environment and converted the early code from Fortran 66. Rodney and Jeff co-authored the first version of the specpr users manual in 1982. I often think of Jeff, a true Unix guru, and wish he were still alive. Lucy McFadden wrote some of the early special functions.

After I moved to the U.S. Geological Survey, specpr underwent many changes and extensions. Programmers at the University of Hawaii continued to make improvements and bug fixes in the new version. Kathy Kierein wrote the alias section, added to the radiative transfer routines and other code. Barry Middlebrook wrote the 3D read routines, a vital contribution for working with the new AVIRIS and GER imaging spectrometer data. Noel Gorelic wrote many sections, including the X-windows interface, variable parsing, the graphics window scaling, the continuum analysis and spectral features special functions, as well as bug fixes and enhancements to other parts of the program. Matthew Klejwa wrote the interactive band analysis routines. Wendy Calvin extended the radiative transfer routines and wrote the wavelength registration special function. Bob Brown and Pam Owensby have made numerous bug fixes and improvements.

Users contributed a lot to the development of specpr. They contributed ideas, found bugs and ways to work around them, ways to improve the program and occasionally fixed the bugs themselves. Some of the more vocal users (probably because I have worked closely with them) are Gregg Swayze, Marcia Nelson, Trude King, and Pam Owensby.

A little known fact is the origin of the name specpr. The name was suggested by Karl Hinck at the University of Hawaii in 1977 and the name has stuck ever since. Thanks Karl for a lasting name for a lasting program (smile).

There are probably several people I have forgotten. If I have forgotten you, I'm sorry. It simply reflects the large effort by so many people that have gone into this program.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The SPECtrum Processing Routines (SPECPR) is a large scale interactive program for general one dimensional array processing, and optimized for reflectance spectroscopy data and analysis. The program processes one-dimensional arrays up to 4852 data points and the operations include addition, subtraction, multiplication, division, trigonometric functions, logarithmic and exponential functions, and many more specialized routines.

The specpr user interface is a menu driven, character command system with all user input entered as ascii characters to the program. User commands are thoroughly checked for the context in which they will be used. This provides for essentially all user input mistakes to be caught and appropriate error messages to be issued. The user interface also allows for command aliasing, variable substitutions, command history and batch command processing to occur at any point in the program. The user can also control his or her own variables, increment and decrement them, and monitor certain internal variables. This flexibility allows for sophisticated programming, e.g. from specific computations, to management of a database.

All arrays are treated as one dimensional lists of numbers, each with an appropriate header for identification and history. Any array can be plotted versus another array, and each axis labeled appropriately. For example, you may plot a reflectance spectrum versus wavelength or wavelength versus reflectance. The array type is simply a label, so data like temperature versus time can be treated just as easily as a reflectance spectrum. Some routines are specific to certain purposes, however. For example, the Planck black body generator is specific to intensity versus wavelength, whereas a smoothing routine could operate equally on reflectance versus wavelength or temperature versus time.

Each array element can have an error bar associated with it and errors are propagated through all appropriate routines. Data points can be marked deleted and deleted points are tracked appropriately. These features allow the program to be effectively used on real world data.

This philosophy has resulted in a flexible system for which a user can manipulate data arrays efficiently. The program was designed for analysis of laboratory, field, telescopic, and spacecraft spectroscopic data, and although general in nature, there are biases built in. For example, horizontal axis labels default to wavelength in micrometers, but the defaults can be changed.

Because of the orientation in specpr to process planetary and terrestrial data, and the need to access such data, specpr has been extended to have access to non-specpr default file types. In particular, specpr can "skewer" an 3-dimensional (3D) data file along any of the three axes. Further, block skewers can be done with the standard deviation of the mean computed for each channel in the block. Currently, specpr can access any of the standard file types common in the terrestrial and planetary remote sensing communities. Details on 3D file I/O are given in Chapter 6.

Specpr has multiple record types available within a single specpr data file. Currently defined record types are data and text. In a data record, a standard one dimensional array is held, along with its header information. The header information includes a title, history, dates and time of data acquisition and when the data were last processed, the user who processed the data, information typical of a spectrum like temperature and viewing geometry, as well as pointers to wavelengths, resolution, and text. If a spectrum is more than 256 channels in length, then the data gets put in succeeding records in the file, and the following records are continuation records.

The second record type is the text, where a title and a block of text (up to 19 kbytes) can be stored. This record type is typically used for a description of samples, experiments, instruments, or data processing for a particular data set. It could also store the actual commands used to create a spectrum. The commands for a complicated plot for a publication might be stored in a text record. That way, you can easily regenerate the plot, or even write the commands to a non-specpr file, modify them with any editor and re-execute them.

The file types, text and command processing, combined with the math and special functions, provide for very powerful and general analysis tool. These facilities also provide the framework for database management.

1.2 History of Specpr

The program began in 1975 at the M.I.T. Wallace Observatory, on a Harris 2024 computer, as a short routine to subtract or divide 2 spectra from the Remote Sensing Laboratory Circular Variable filter spectrometer ("The Wedge"). Due to lack of processing software, the program grew to fill the Wedge data reduction needs. Over the years, the program has grown to serve more and more applications. The reason specpr has survived is because of the philosophy behind each application routine that was added: it should be of a general nature, have checking of user input so that data are not inadvertently destroyed, give a history of all operations, and track deleted points because most data are not perfect.

After the initial development of specpr at the MIT Remote Sensing Lab (RSL), the lab moved to the Institute for Astronomy of the University of Hawaii in June 1977. At that time, there was no software for reduction of spectral data. It was decided that the Wedge data reduction program should be put on the new computer (TI 980 B) as a general spectrum and one dimensional array processing system. In the next 3 years, the program was written to handle the many types of astronomical photometry being obtained by the group at the Institute.

In 1980, Specpr was moved from the overworked TI 980 B to the newly acquired LSI 11/23 system running the UNIX operating system.

The historical development of specpr up to this point solidified some of the commands used in the program. The initial specpr routines

(pre 1980) were designed with data reduction strategy for a particular method of astronomical observing. That strategy is still there, but must be modified for other methods, for example reduction of laboratory spectroscopy data. The TI 980 B and LSI 11/23 versions of specpr were based on 16 bit words whereas the MIT version was based on a 24 bit word machine, and that resulted in certain array sizes that are still in place. During the change to the TI 980 B, many of the letter command codes and array sizes were changed to fit more general cases. Also during the change to the LSI version all of the routines were rewritten in Ratfor with a few in C and the program was broken up into about 25 separate programs due to the lack of a decent overlay linker for the Fortran compiler. This Ratfor conversion apparently took place because the programmer who did it was a C programmer, and liked Ratfor better than FORTRAN 66 (this happened before FORTRAN 77 became widely available).

In 1984, the program was moved to the U. S. Geological Survey in Denver. The new spectrometers at the USGS necessitated many changes, mostly because the USGS machines produced more data points per spectrum. Research directions also played a role in the analysis routines developed. The study of high resolution spectra, absorption band analysis, radiative transfer mixing models, and the desire to manage spectral databases necessitated major changes to the functionality of specpr.

The latest analysis area for specpr development is access to imaging spectrometer data cubes. This access allows a user to query large data sets for spectra and then analyze those spectra in detail.

1.3 Specpr Design Philosophy

Specpr is designed to meet the needs of the remote sensing spectroscopist for detailed analysis of spectral data. In that sense, it is not designed to analyze whole image cubes (hundreds of thousands of spectra), but smaller numbers (ones to several thousands) where the user must become intimately knowledgeable about the data quality and results.

One of the main goals of the program is to be uncrashable. Since users sit at the terminal for many hours processing data, they are bound to make mistakes, so another goal is for the program to be somewhat "intelligent" and to try and catch mistakes. The program is completely free format allowing quick and easy input of commands. Spaces are required between two numbers only when there is no character between them. Otherwise, spaces can be completely left out or inserted wherever the user wishes.

Most commands are a single character. With the many programs and options in specpr, an individual letter is used for more than one command. For example in one section of the program, the command "l" may mean list a file, but in another, say a plot routine, it may mean line type. Thus a command is context dependent. The menus in each routine give the commands and options available to the users.

In many batch processing systems, the user types commands in a file, and then directs the files into a program for processing. If there was a

mistake, the job must be run again, and this wastes time. In the interactive version, as the commands are typed in, they are executed immediately and the results displayed. In this way, all intermediate steps are seen, and decisions can be made to change the processing in order to obtain the best results. Specpr commands can be read from a file starting and terminating at any point in the program, thus giving batch capability. Commands can also be saved in a text file as they are typed in. Then, if a mistake is made, the text file can be edited and executed again.

Specpr maintains a complete history (within reason) of each operation. This has proved invaluable for figuring out what was done to some data whether it is yours or someone else's. Any time there is a question of the results of a particular analysis, anyone can trace and verify whether or not everything was done correctly. However, for specpr to properly record histories, certain steps must be followed, for it is possible to subvert any history mechanism. One possible way to subvert the history is to put intermediate products in a temporary file, then delete the file and only keep the final product. Specpr records the history of the last operation of each spectrum, so to trace a complete history, all intermediate steps must be kept. This is not really a limitation, because spectral data are small in comparison to the size of modern computer disk space.

An interactive "smart" program of this nature requires an almost shocking amount of code. Specpr is now about 50,000 lines of Ratfor and C in length, contains over 400 subroutines, and takes about 1.5 megabytes of memory to run.

For reference, specpr was first described by Clark (1980, Publications of the Astronomical Society of the Pacific, 221-224).

CHAPTER 2

PROGRAM STRUCTURE

2.1 Block Diagram of Program Structure

Specpr is a menu driven program. The main menu allows access to the processing parts of specpr. From here the user types a letter or sequence of letters to get to other parts of the program, such as math or display. The main menu looks like:

```
 v = spdemos : f    1    w = *unasnd*: f    1    d = *unasnd*: f    1
 u = *unasnd*: f    1    y = *unasnd*: f    1    s = starpack: f    1
 lp: spoolfile       obs lat=    .000 deg   channels= 256 wav fl=C  256 h
 file protection: v    53,w     0,d     0,u     0,y     0,s     0 ltype= 0
-------------------------------------------------------------------------
 MAIN MENU: ********** Program Operations Control *********************

   INFO:     "in" to turn OFF information

   LIST:      l followed by v,w,d,u,or y  to list the contents
                          of the corresponding file
   DISPLAY:   t  to DISPLAY on screen, OVERLAP on screen
   MATH:      m  to do MATH operations
   TRANSFER:  t  to TRANSFER (COPY) files

   PLOT:      p  to PLOT SPECTRA on PLOTTER/printer
   SETUP:     b  to change SETUP PARAMETERS
   FILES:     r  to REASSIGN files and devices
   STARPACK:  s  to create a STARPACK for extinction corrections
   PRINT RST: f  to print summary of the current restart file

   EXIT:      EX  to exit program
```

Figure 2.1-1 shows the basic user control structure and the commands used by the user to access each section.

2.2 File Structure from the User's View

The file structure of SPECPR is the most complicated portion of the program. As far as the user is concerned, there are 6 "Devices" each labeled by a single letter (called "file ID letters"). Originally, these IDs referred to a style of processing (e.g. put raw data in the "raw-file" and processed data in the "savefile". Specpr has evolved beyond this simple concept, but the file ID letters are still in place, so the ID letters are now historical. They are:

v= saVefile (Vfile)
w= raWfile (Wfile)
d= workfile (formerly Diskfile at MIT) (Dfile)
s= Starpack file
u= Ufile
y= Yfile

Normally the user assigns, via the file assignment routine (section 5), all of the "devices" to disc files. Sometimes, processed data is stored on magnetic tape and thus needs to be transferred to the disc. Originally, devices u and y were intended to be used only for transfer and display and were not used in any math operation. However, due to increased needs, they have been made full working files (in version 1 specpr). Each of the 5 main data files (v, w, d, u, and y) may contain up to 9,999,999 records and may be assigned in the program to the named disc files or to magnetic tape drives. The starpack file contains a maximum of 50 extinction correction "starpacks" on disc.

The philosophy behind single letter file IDs is to minimize user typing. After all, why type a 10 or more character file name every time you want to access a spectrum. In practice, users quickly learn this abbreviation method to associate a single letter with a data file.

(WARNING: some operating system limitations may preclude implementation of direct tape I/O--check with your local administrator.)

(WARNING: with a limit of 9,999,999 records per specpr file, large specpr files will not fit on most tapes.)

Error bars (1 sigma standard deviations) are stored in the next record after the data when generated. This saves room since many files do not contain error values. In this manual as well as in the program, the word "error" is used to mean the error bars: the errors to the data, and not program errors, user command errors, or input/output (I/O) errors. Input/output errors and user command errors will be referred to more specifically.

In Specpr, many spectra are contained in one data file. The alternative method is to put each spectrum in a single file, and other data processing systems do this. The disadvantage to one spectrum per file is that each file must be unique in name, so you must type in long or complicated names to get access to a particular spectrum. Also, rapid access to many spectra quickly (like a library search) would be slow with extra overhead to open and close each file. With many spectra in one file, the user can list the file and see long titles to recognize the spectrum he or she wants and then type just a few characters to access it. Access to any single spectrum or a group of spectra is very quick--no open or close is necessary, just a simple read or write.

Each I/O operation is analyzed for I/O errors by the program. If an I/O error occurs, the error encountered is listed on the CRT so the user can try to figure out what happened.

2.2.1 Deleted point values

All data values for "data" or "wavelength" (y or x data) points are valid except

-1.23x10+34

which is taken to be a deleted point.

2.2.2 General data file I/O

The user accesses a data array (spectrum) by typing in the File ID letter and the record number in the file. In general, a lower case letter means to read (or write) a particular data set. An upper case letter means read (write is not allowed in this context) the particular data set, but treat the data as horizontal axis values (the x values of x-y paired data). In the case of spectra, a capital letter would signify the wavelengths.

```
Examples:    v23          # read data set in file v, at record 23.
             w126 W15     # read data set w126 and read data set
                            w15 as the "wavelengths" (or
                            horizontal axis array).
```

Also, transfers between all combinations of v, w, d, u, y, and s are allowed if the user-selected file protection permits such transfers (see File Protection, section 2.3).

2.2.3 Transfer Restrictions

If the device names are equal, one of the special restrictions below will be applied. For instance, if the name of "Vfile" is the same as the name of the "Ufile" (e.g. a data tape is on MT0 as "Ufile" and is to be transferred to the "Vfile" on disc), the restriction applies to tape to disc, disc to tape, disc to disc, and tape to tape transfers. These restrictions are enforced in an effort to preserve the histories which refer to a specific spectrum by data tape name and record number (sometimes called file number).

Transfers to tape including disc to tape and tape to tape transfers must be to corresponding record numbers but do not have to begin with record 1 (if the data file names are equal).

Transfers of tape to disc, but not disc to disc, must begin with record number 1 (if the data file names are equal).

Transfers of disc to disc must be to corresponding record numbers but do not have to begin with record 1 (if the data file names are equal).

2.3 File Protection

All files may be totally or partially protected, or completely unprotected. The fourth line of the CRT header gives the protection status for all six devices (devices v, w, d, u, v, and s).

- If the protection number is positive or zero, the device is a read/write device where you can read up to and including the protection, but you can only write to the protection +1 record. For

example, a value of 637 means you can read the first 637 records, and you can only write to record 638.

- A protection number of -1 means totally unprotected so you may read or write anywhere in the file randomly.

- A protection number of less than -1 means the device is a read-only device where you can read up to the absolute value protection number. For example, a value of -264 means 264 records are read only and you can't write to the file.

You can set the protections on many files on one command line. Example:

v0 d-1 u432 y-600 w-600 s-50

where v0 means to protect 0 files on device v (write to file 1 only, 0 files can be read); u432 means to protect up to record 432 (write to record 433); y-600 means that y is a read-only device with 600 records; similarly for w; and s is a read-only file with 50 records. If the protection number is zero or positive, it is incremented each time the device is written to.

2.3.1 Use of SPECPR File Protection

The SPECPR file protection is designed to protect the user from destroying existing data and to allow use of the program with minimal thinking of where data is going (so the user can be thinking about the science). Protection should be used at all times unless there is some necessary reason for not using it. Remember that a mistake is what will destroy data in an unprotected file--and everyone makes mistakes.

The following is an example of the use of protection when using specpr tape I/O. Some groups have a different philosophy regarding specpr tapes. For example, at the U. S. Geological Survey, Denver spectroscopy laboratory, specpr tape I/O is not used. Instead, specpr files are kept on line on disk and written to tape only with normal system backups. In any event, the following example illustrates the proper use of protections to pre- vent data loss, as well as to maintain histories.

Start SPECPR. Say you have 2 tapes to be transferred to disc and then do some processing. Load the tapes (we will call them A01 and B01): A01 on MT0, and B01 on MT1. Plan to assign u to MT0, y to MT1, v and w to disk files. Set the protection on v and w to 0 (this is the default). Type in the names of the devices u and v = A01; y and w = B01. Assign the devices as noted above. Now here is where the protection comes in. First, let us transfer the tape on MT0 (A01 = u). The protection is -9999. Say on the label on the tape, there are 673 records--do not believe this. We could have set the protection to -673, but what if the last time you added stuff to the tape you forgot to update the number or someone else added stuff and did not update it? If that happened and you then added stuff after record 673 and wrote it back to the tape, someone might get very mad. So--let the computer find the end of the data. Go to file transfer and type "u1 + 9999tv1". All the records up to the end of file

mark will be transferred. Say there were really 729 records on the tape. Then, after the transfer, the protection on v would be 729 and -9999 on u. Now transfer MT1 to the disc: "y1 + 1999tw1". Say there are 463 records on MT1. The protection after the transfer would be 463 on w and -9999 on y. Now go back to change protection routine and make the protection on u-729 and on y-463. All your data are fully protected.

Now say you did some processing and added 47 records to v and 21 to w. You must then transfer the stuff back to the tapes. Take the tape off MT0 and put in a write ring (if you had a write ring in before, you are asking for trouble). Go to the change protection routine and change the protection u from -729 to 729. If the tape was not at the load point when you put in the write ring, you must reset the record pointer in the program. This can be done in one of several ways: (1) change the tape name (change the name of u from A01 to A01 (yes, the same name); when the name is changed, the tape is rewound); (2) display record u1; or (3) transfer u1 to somewhere else (this is faster than a display). If d is assigned to disc and it is used as a work file (protection = -1), transfer "u1td1" or "u1td1000". If d is assigned to /dev/null, it will read u1 before you get the "illegal transfer, device assigned to /dev/null" message.

Now that the tape record pointer is reset, you transfer your stuff from v to u. Type "v730 + 999tu730". Note 730 is one more than the current protection. The transfer will continue up to the protection limit on v (776), and after transferring v776 to u776 the message "FILE REQUEST GREATER THAN FILE PROTECTION" will come on--press return.

It is now a good idea to list the tape from just before the point at which data was added to the end of file to make sure the tape does not have any obvious errors.

Now you must transfer the stuff on w to the tape. Change the protection on u to 463; then change the name of u to B01 (this puts in the correct name and resets the record pointer). Now go to file transfer and type "w464 + 99tu464". The program will stop at the protection limit on w of 484, transferring the added 21 records. Now list the type as before to check the newly added records.

Following this method will maximize the safety of your data. Failure to do so will cause no sympathy from the person whose data you accidentally destroy (and you must recreate it for him or her).

The specpr protection is closely tied to the history philosophy of specpr. The specpr history uses the "tape name" and record number as a unique identifier for a spectrum. Proper use of the history mechanism necessitates proper protection. Don't set the protection to -1 and write over data. Later data could be referencing specific records earlier in a file. For example if record 36 was a sample minus dark spectrum, and record 167 was record 36 divided by a standard, and then you wrote over 36 with another data set, then the history would be broken. Similarly, it is vital to keep file names indicative of the final archive source. Don't do work in a file called "junk" or "temp" and then rename it XYZ001 because all histories will be referring to file "junk" or "temp" and you will have no idea what that file really was a year from now.

2.4 Command Interpretation

The specpr terminal input routine looks for various special characters in the input. These special characters are: <, >, ?, %, =, !, ;, {, [, and $. All of these characters except ; {, [, and $ have special meaning only when it is the first non-blank character on the line. Additional special character sequences are also interpreted: "\#", "\\#", "==", and "px(". If a "{" or "[" is found on the command then the corresponding bracket, "}" or "]", will also have a special meaning. The effect of these characters is as follows:

< This character when followed by a file name causes specpr to read input from that file. You can also specify a starting and ending line number. For example

<inputcommands 20 40

will read commands from a file called inputcommands starting at line 20 and stopping after line 40 of the file. If the ending line number is omitted, the file will be read until the end of the file is reached. If the starting line number is also omitted the file is read starting at the first line.

When commands are being read from an input file, that command file may read commands from another input file. That input file can then read from yet another file, up to 9 deep. In a sense, a command file could set several variables and read another command file similar to a subroutine. In this case, the "subroutine" has access to all variables, thus it is similar to subroutines used in the Basic programming language. When one command file finishes, commands are read from the next line of the previous command file. For example, given two command files:

```
file "a1"           file "a2"
----------          ------------
line 1              line 1
line 2              line 2
<a2                 line 3
line 4
<a2
line 6
```

then the command sequence would be as follows.

```
<a1     # terminal input from user
a1 line 1
a1 line 2
a2 line 1
a2 line 2
a2 line 3
a1 line 4
a2 line 1
a2 line 2
```

```
a2 line 3
a1 line 6
# back to terminal input
```

> This character when followed by a file name causes specpr to copy all user input into the named file. When not followed by a file name it terminates the copying of the input. Example:

```
> outputfile  # copies all commands to file "outputfile"
>             # turns off copying of commands to "outputfile"
```

! This character when followed by any string will pass that string to the UNIX shell (/bin/sh, see the UNIX manual) for execution. For example typing

```
!ls
```

will give you a listing of the files in the current directory.

; A semicolon anywhere in an input line is treated as if the user had typed a carriage return instead of a semicolon. For example typing

```
c;0 1
```

is equivalent to typing

```
c
0 1
```

The special characters ?, %, =, and $ involve the "command file" which automatically keeps a record of the last 20 commands the user has typed, and in addition can contain up to 80 permanent commands saved by the user. The commands in the command file are numbered 1 to 100 with commands 1-20 being the last twenty commands typed by the user and commands 21-100 being the permanent commands. The effect of these 4 special characters is as follows:

? This character lists the contents of the command file on the user's terminal. The commands are printed in five groups of twenty commands. If the ? is followed by a digit from 1 to 5 the corresponding group of commands will be printed. If no digit is specified, the first group of twenty commands is listed. For example,

```
?2
```

will list commands 21-40.

= This command allows the user to type in the permanent commands. The = should be followed by the number of the command to be entered. The system will prompt the user with a question mark at which time the user should type in up to 80 characters for the new command. For example,

=25

allows the user to type in command 25.

% This command allows the user to copy a command from one entry in the command file to another. Commands may be copied only from commands 1-20 to commands 21-100. For example

% 10 25

will copy command 10 to command 25.

$ This command when followed by a number from 1 to 100 extracts the corresponding entry from the command file. For example, if command 25 contains

0.574

then typing

0 $25

is equivalent to typing

0 0.574

Care is needed when the command number is followed by another number. If the command ends with a number then the user should follow the command number with a comma if the number following the command is part of the desired final number. For example, if command 25 contains

0.574

then typing

0 $25,7

is equivalent to typing

0 0.5747

while typing

0 $25 7

is equivalent to typing

0 0.574 7

{ is used to parse internal variables. See section 2.6.

[is used to parse aliases. See section 2.5.

\# The character sequence "\#" means everything after and including these characters are comments and not to be interpreted by the command processor. If \# occurs as the first two characters on the command line, then the line is ignored.

\\# The character sequence \\# is an "escaped" comment meaning interpret the characters as \# and pass them to the command processor, in effect uncommenting the comment.

== The character string == means an alias function follows. See section 2.5 below.

Px(The character sequence "px(" means a pixel coordinate of the form px(i+j,k+l,m+n) where i, j, k, l, m, and n are integers describing an extraction from a 3D file. See section 6.2 for more details.

2.5 Alias Definitions and Substitutions

While running specpr, an alias list of common commands can be created using this option. This is used to reduce the amount of typing needed by substituting short alias words for long commands. Alias words can be created for any type of command used in specpr. These alias words can either be read into specpr from a file or set up every time specpr is run.

2.5.1 Creating Alias Words

Alias words can be created for any command used in specpr. These alias words are defined by typing ==[alias word] translation. The alias word must be inside square brackets and followed by its translation. An alias word can consist of any letters or numbers inside square brackets. If you use four or more capital letters, you do not need to use square brackets when you type the alias. At any point in the program specpr, an alias word can be defined or translated.

EXAMPLES	==[hi] hello	This sets the alias word 'hi' to translate into 'hello'
	==[DIRL] !ls	This sets the alias word 'DIRL' to give a listing of the current directory

2.5.2 Translating Alias Words

For an alias word to be translated, it must either be inside square brackets or be one word of four or more capital letters. The alias word can be entered at any point in the program and the translation will take place. The line will be reprinted on the screen showing the translation and then executing the command.

EXAMPLES	[hi]	The alias word is translated into hello
	hello	

DIRL	This alias word does not need brackets because it consists of four capital letters.
!ls	It translates into this command which then will execute a listing of the current directory

2.5.3 Listing and Saving Alias Words and Translations

The current list of alias words and translations is printed onto the screen using the command

==list

as the only character sequence on the command line. If a filename is typed after the list command, the list of alias words and translations will be saved in the specified file.

EXAMPLES	==list	This lists all the alias words onto the screen
	==[hi] hello ==[DIRL] !ls	
	==list alias.list	This saves all the alias words and translations in the file alias.list

2.5.4 Reading Alias Words into specpr from a File

To read a file of saved alias words and translations into specpr, type < and the filename. This will allow all the alias words in the file to be used in specpr.

EXAMPLE	<alias.list	This reads the alias words and translations from the file alias.list into specpr
	==[hi] hello ==[DIRL] !ls	

2.5.5 Deleting or Writing Over an Alias Word

An alias word can be deleted by entering ==![alias word] anywhere in specpr. This will not delete it from a file of stored alias words unless the alias list is resaved into the file. An old alias can be over-written by redefining the same alias word. The list must also be resaved for this alias translation to be changed in the file.

EXAMPLE	==![DIRL]	This deletes the alias word DIRL and its translation from the list

2.6 Variable Parsing

Variable parsing is done on strings enclosed in curly braces "{}". Variables may be set, incremented, decremented, or other simple math may be performed and the result inserted in place of the curly brackets on the command line. Variable parsing is useful in specpr command files.

The variable parsing routine has a limited understanding of math functions and variable parsing. The variables currently incorporated are:

```
ia-iz -- 26 user integer variables
ra-rz -- 26 user real variables (floating point)
iprtv -- protection of file v,
iprtw -- protection of file w,
iprtd -- protection of file d,
iprtu -- protection of file u,
iprty -- protection of file y,
```

The routines only understand the math functions of equivalence and addition (although addition of negative numbers is allowed). Here are a few examples of valid variable substitution strings.

```
{iprtv}    # replaced with the protection of device "v"
{ib=54}    # not replaced (sets ib to 54)
{ib+-5}    # replace with 49 (ib=54-5) (addition of negative
             numbers)
{ib=ib+iprtv}    # not replaced, (sets ib)
{ra+2.3+-6.5}    # replaced with -4.2 (user variables are
                   initialized to zero)
{2056+1123+-43+iprtd+ia+ib}  # (Only limit is 80 characters per
                               line)
```

NOTE: a minus in front of a variable, such as "{-ib}" will not work.

2.7 Order of Substitution and Command Interpretation

In specpr, the order of the interpretation of the special characters is $ substitution, {, ;, ?, ==, =, %, >, !, <, alias translation, and \#. Any of these special characters can be used in an alias translation because the program loops back to the beginning when a translation is performed. The special characters would then be interpreted the second time through the loop. This program would continue looping until all of the translations are finished.

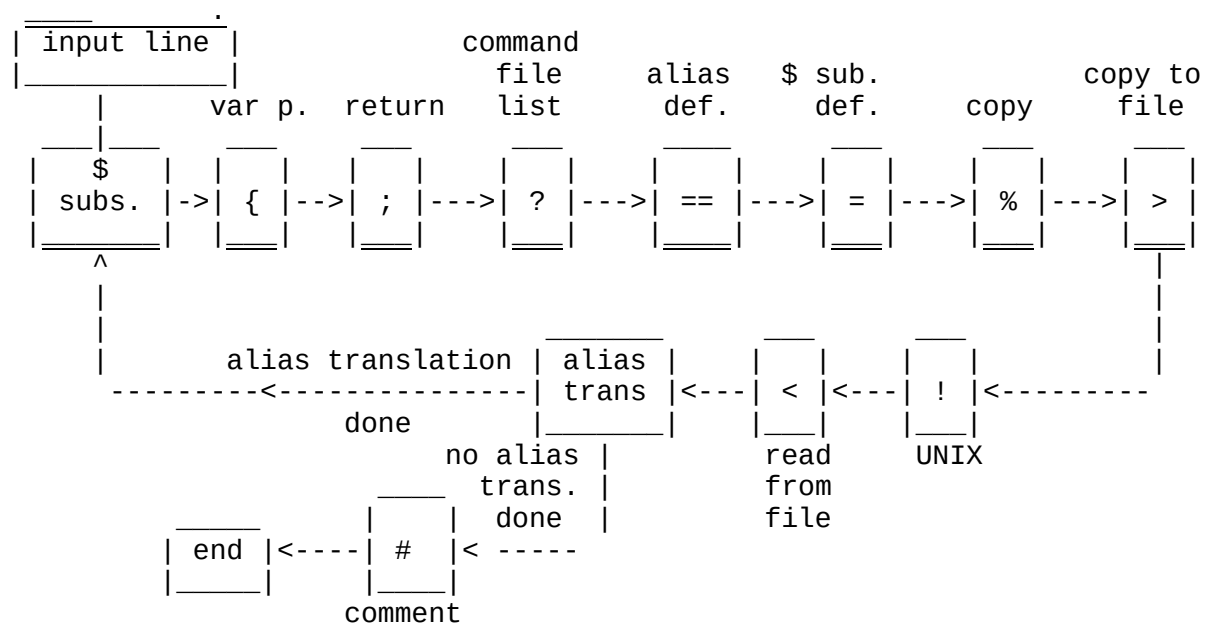
The $ substitutions can contain alias words because the translation takes place after the $ substitution is done. Both the $ substitutions and the alias translations can contain ; which are later processed. However, the \# comment statement can not contain a ";" because it is processed before the comment statement is found and everything after the ; would be stripped off for the next command. The flow chart below shows the order of these command interpretations.

```
EXAMPLES        ==[lv] lv;;;0 99        Assigning this alias would
                                        give the list command,
                                        return three lines, and
```

list from 0 to 99

==[lst] $21 This would assign the alias lst to the command in file number 21 If 21 has [DIRL] in it, the alias would translate to !ls

2.7.1 Flow Chart for Command Interpretation Order



2.8 Specpr File Types and Content

The specpr I/O routines support multiple file types. There are "standard specpr files" and other files. Currently, the other file types include 3-dimensional image type files, which me be Band Interleaved by Line (BIL), Band Interleaved by Pixel (BIP), or Band SeQuential (BSQ) files, with any header size, with any data type. Currently routines exist that read 16-bit integer (I*2) 32-bit integer (I*4) and 32-bit real (Real*4) data, but others can be added. It turns out that this general I/O package can read just about any format, because a 3D file can also access a 2D or 1D file since the higher dimensions have a dimension of 1.

The standard specpr record, however, is optimized for x-y paired data.

2.8.1 The Standard Specpr File Content

Specpr files are random access files, 1536 bytes per record. The data types are described in the file format document in the specpr programmers notes. Currently, there are four basic types of data formats (although many more can be easily defined for specific disciplines):

1) beginning spectrum (512 bytes of header and 256 channels),
2) continuation records for spectra with more than 256 data channels,
3) character records where the data space is ascii text, and
4) text continuation text records.

A data set can consist of as little as a single record (a single one-dimensional array) or have up to 12 continuation records (there is planned expansion to substantially increase the number of continuation records).

A spectrum typically consists of a wavelength set, data (e.g. reflectance) and the error bar. In specpr a data set is only one array, not three (e.g. waves, reflectance and error). The wavelengths are kept in an independent record. The error bars are kept in the record after the data values. Some data do not have error bars, so space is saved. A sample specpr file might look like this to the user (this is extracted from an actual data file):

Table 2-1

record	title	channels	time	date
0	reserved for future: intended for file description. This record has no user access.			
1	********** SPLIB001 **********	36 Characters of TEXT		
.				
.	other data records			
.				
15	USGS Denver Beckman STD wavelengths 1x	512	02:57:26.00	10/15/1985
17	USGS Denver BKMN 1x resolution	512	02:57:26.00	10/15/1985
.				
.	other data records			
.				
73	Description of Alunite Hunt 295.?B	3132 Characters of TEXT		
76	Alunite 295.3B .2-3um 1x ABS REF	512	12:03:14.00	08/20/1985
78	errors to previous record 76	512	12:03:14.00	08/20/1985
80	Alunite 295.3B .9-2.6um 2x ABS REF	680	09:49:27.00	08/21/1985
83	errors to previous record 80	680	09:49:27.00	08/21/1985
.				
.				
.				

In the case of record 76, the number of data channels is more than 256, so one continuation record is used (record 77; hidden from user view). Similarly, records 16, 18, 74, 75, 79, 81 and 82 are continuation records.

The following table shows how the data are spread over continuation records:

Table 2-2

Continuation record number	number of channels in record	Data channel range
-	256	1 - 256
1	383	257 - 639
2	383	640 - 1022
3	383	1023 - 1405
4	383	1406 - 1788
5	383	1789 - 2171
6	383	2172 - 2554
7	383	2555 - 2937
8	383	2938 - 3320
9	383	3321 - 3703
10	383	3704 - 4086
11	383	4087 - 4469
12	383	4470 - 4852

The following table shows how text data are spread over continuation records. Note that there is also a pointer to additional text, so the real text size is limited only by disk space.

Table 2-3

Continuation record number	number of characters in record	Character range
-	1476	1 - 1476
1	1532	1477 - 3008
2	1532	3009 - 4540
3	1532	4541 - 6072
4	1532	6073 - 7604
5	1532	7605 - 9136
6	1532	9137 - 10668
7	1532	10669 - 12200
8	1532	12201 - 13732
9	1532	13733 - 15264
10	1532	15265 - 16796
11	1532	16797 - 18328
12	1532	18329 - 19860

2.9 "Wavelengths" and Other Concepts of x-y Paired Data

"Wavelengths" and other axes are simply a data set and a label in specpr. The default label for the "x-axis" is "Wavelengths (microns)" and

any data set in the specpr data files can be used as the x-axis values. The x-axis label can be changed to something else by changing the fourth line of the manual history of the x-axis data record to: "\W string" where "string" is the character string for the label.

```
Example:    v23V19      # the data set is v23, the x-axis values are
                          in record v19 (specified as a capital
                          letter file id).
```

Then if the fourth line of the manual history is:

\W Intensity (seconds)

the crt plot would label the horizontal axis "Intensity (seconds)".

There is a special case for specifying a wavelength set: instead of a file ID, use a capital c, "C", and the number of channels and the x-axis values will be set equal to the channel number.

```
Example:    C236        sets the x-axis values to 1 in channel 1,
                        2 in channel 2, etc., to 236 in channel 236.
```

A data set can be called up with wavelengths specified by the data set if the wavelength pointer is set in the header info to that particular data set. For example, if v23 has its wavelength pointer set to 19, then the default x-axis data set is in record 19 of the same data file. The default x-axis values are found by specifying the character "&" instead of the upper case file ID and record number. Then

v23V19

and

v23&

are equivalent.

2.10 Help!

Specpr has a help facility where you may type the word help on the command line at any point and go to the help routines. You may also type help and a key word to get help on a particular subject. The help command with no keyword will show you a list of key words to choose from. See Chapter 3 for more details.

CHAPTER 3

STARTING SPECPR

3.1 Starting Specpr on Unix

To run specpr you must first log in to the system. It is advisable to do specpr work for a particular project in a separate directory. Specpr uses protective locks on files to prevent simultaneous specpr sessions from overwriting data. Thus, if you are starting a new project, create a directory in which to start, then cd to that directory.

You can start specpr in one of five ways, using the following commands.

specpr — starts specpr, first with current messages, then a start "from scratch" where a complete setup is required. After the word "specpr" type a carriage return.

specpr rfile — starts specpr in restart mode, where the restart file "rfile" contains all the information saved from the previous session. Before specpr is started, current messages are displayed.

specpr rfile –gN — starts specpr in restart mode, where the restart file "rfile" contains all the information saved from the previous session. The -gN option, where N is the graphics terminal type integer number given in section 5.7, sets the graphics type as specpr is started. The "N" may also be one of the strings "xterm" or "hpterm" (without the quotes. Thus the command "specpr rfile -gxterm" starts specpr with restart file "rfile" and starts an xterm X-windows graphics session.

specpr rfile - — starts specpr in restart mode, where the restart file "rfile" contains all the information saved from the previous session. The minus sign suppresses messages unless new messages have been added since the last time "rfile" has been written. After a basic setup, this is the command you will want to use to restart specpr (or the command below).

specpr rfile –gN - — starts specpr in restart mode, where the restart file "rfile" contains all the information saved from the previous session. The -gN option, where N is the graphics terminal type integer number given in section 5.7, sets the graphics type as specpr is started. The "N" may also be one of the strings "xterm" or "hpterm" (without the quotes. The minus sign suppresses messages unless new messages have been added since the last timer "rfile" has been written. After a basic setup, this is the command you will want to use to restart specpr (especially when starting specpr from X-windows). Thus the command "specpr rfile -gxterm -" starts specpr with restart file "rfile" and starts an xterm X-windows graphics session.

The restart file contains all information for restarting specpr and putting it in the same state as when you last quit the program. Specpr periodically updates the restart file during a processing session in case of a system or program crash.

The specpr startup messages are notes and information about changes, additions and bugs in the current version of specpr.

3.2 HELP!

The user may get help at any keyboard input point in the program. To get help, you simply type

help

or

help <keyword>

where "<keyword>" is a key word to search on. The index of key words and a description is displayed by just typing help.

Once in help, help is read with the Unix command defined by the environment variable PAGER (see section 3.3), or if PAGER is not defined, the Unix "more" command is used. When reading help, if there is more than one page you must follow the directions for that Unix command (e.g. for the more command, press the space bar to get to the next page or a "q" command to quit). When you have completed reading the index page or a keyword entry, the following message is displayed:

```
---------------------------------------- PRESS RETURN TO CONTINUE
with program, Type a keyword for help on that topic, or Type help
for the index page ----------------------------------------
```

If you want to read about another keyword, simply type that key word and press return. If you want to redisplay the index pages, type "help" (without the quotes) as the key word.

Example: to get help on file protection at any point in specpr, type

help protection

or, type

help

and once in the help facility, type the

protection

key word.

3.3 Configuration of Environment Variables

Specpr currently recognizes two environment variables. An environment variable is a variable that is set from the Unix command line. For example, in the c-shell, you type

setenv PAGER myprog

to set the environment variable PAGER. See your local system documentation on setting environment variables for other Unix shells.

Commonly, users define these variables in the login startup files, so they get activated each time the user logs in.

The environment variables currently used and defined by specpr are:

PAGER The program used for listing to the terminal screen. Currently it is used by the help facility. Default program if PAGER is not defined is the Unix "more" command.

EDITOR The editor to use. Currently, this is the editor to use when editing specpr text records in the header information change routines. Default editor is "vi".

CHAPTER 4

IMPORTANT RULES

4.1 IMPORTANT RULES

The following rules are important because they have been found to protect your data and make specpr operate properly. Study them well and follow them closely.

4.2 Protective Locks

Protective Locks are enforced to prevent more than one user at a time from running specpr in a directory. When specpr is started, a file called LOCK.specpr is created inthe directory. If another specpr is started, it will see that file and give you a message and quit. You should check to see if a user is really running specpr, and even ask him or her why the lock is still there if specpr is no longer being run (maybe specpr was terminated abnormally). If no one is running specpr, then you may delete the lock file and proceed as normal. When specpr terminates normally, the lock will be deleted. If specpr crashes, the lock will still be deleted because the lock activities are done in the startup shell for specpr.

4.3 Make Specpr Files Read Only When Outside Your Directory

There have been instances of people writing over data in specpr. On Unix systems, two users are allowed to write to the same file, and of course such incidents happened on occasion. Basically, two specpr users, working in different directories, assigned the same file (somewhere in the Unix file system) and each began writing to it. Of course, disastrous results followed as they wrote over each other's data! To prevent this, follow the simple rule: If a specpr file is located outside your directory, give it READ ONLY protection (protection value negative). Do not write to a file unless it is in your directory. If you need to write to a file not in your directory, move to that directory and run specpr from there. If that directory is owned by another user, check with that user before adding to a file. If they are logged on, check that they are not running specpr from the same directory. If two people run specpr from the same directory, restart files, command history, and data files will be overwritten (ordinarily this is prevented by the lock file discussed in the previous section, but could happen if someone deletes the lock)! If you add data to someone else's files, send them mail so they will not be surprised (and accidentally write over data).

4.4 Bug Reports

Whenever you encounter a bug, you should report all details so others can see them (and at least avoid them) and the specpr manager can fix them. Send bug reports directly from specpr with the command:

```
!specpr.bugs
  enter your report
<control d>  meaning type a "control d"
```

If you are not in specpr, enter the same command without the "!".

4.5 Keep Track of the Number of Channels and the Wavelength Set!

Specpr supports a "number of channels" value that ranges between 1 and 4852. You should be sure that the correct number of channels value is set BEFORE completing ANY MATH OPERATION. This is VERY Important. Further, if the operation requires "wavelengths", such as a Planck black body function, be sure the correct wavelength set is assigned. Novice users seem to have the most problems with this concept.

For example, if 32 channels are assigned, and you divide two arrays with 100 channels, only the first 32 channels are actually acted upon. If you then reset the number of channels to 100 after the operation, like in the CRT plot routines, 100 channels will be written to the data file after you finish with the plot. But channels 33 to 100 are actually garbage!

4.5.1 Rules of Operation for Channels

The math routines use the number of channels controlled by the wavelength set to determine what to operate on. For example if you have two 512 point spectra, and the wavelengths are set to 81 channels, then any math(e.g. divide) will only work on 81 channels. There are some exceptions (f14: edit, it will work on all 4852 channels). If you have any doubt, set the number of channels BEFORE entering the routine, or if it gives you the option, when you first enter it. All routines that operate on a certain number of channels inform you of how many channels they will use.

When a math operation is complete, the program goes to the CRT plot routines. The NUMBER OF CHANNELS RECORDED IN THE HEADER of the data is set after you exit the plot routines using the number of channels given by the wavelength set in use. For example, consider you had just divided two spectra of 512 channels. When you exit the CRT plot routines and the data are written, the number of channels is set by the wavelength set currently in use. If you changed the wavelength set in the CRT plot routines to 22 channels, when you exit the plot the program sets the channels to 22, copies 22 channels to the output I/O buffer and writes the results to disk and you don't get all your data. So, be careful. Most routines that have different output channels from input channels will automatically set the output to the correct values (e.g., f17: convolution).

4.5.2 If You Must Change The Number Of Channels

MATH: CHANGE the WAVELENGTH SET number of channels. DO NOT go to the header information change routines and change the value there: it gets overwritten by the value from the wavelength set when the data are written to disk (after you exit the CRT plot routines).

DISPLAY: Change protection on the data file to -1. Then transfer the data to itself. For example, if the data you wanted to change was in v23 and had a length of 512 channels and you really wanted 510, then you would set the protection on v to -1, and from transfer routines, use the command "v23citv23" (see section10.3.1), then change the number of channels value and exit the information change routines using "e" (exit and do pending write of the data back to disk). BE CAREFUL: if you change the channels to a bigger number, there is really no data in those new channels. Also, if you increase the number of channels too much, you will go over a record boundary and the next record could be written over other data! If you use a much smaller number, the output data could take up fewer records and then there would be a record that had its continuation bit turned on, but has no beginning data to it. A list of the data file will cause a read error at that record. In that case, you need to write a dummy data set in that record. To do that, copy (transfer) a data set whose number of channels is less than or equal to 256 to that record.BE SURE and RESET protection to the end of the file so you do not overwrite other data!

4.5.3 Setting Channels

Sometimes you must set the number of channels to a value that you do not have a wavelength set for. Use the capital c "C" and a number. For example, you have just started specpr in a new directory and have no data in any file. You can not display something because you have nothing to display. Go to MATH and operate on something that doesn't need input data. For example, do f16 (line segment generator). When you enter f16, set the number of channels to whatever you want and type in some values (e.g. set channel 1 to 0 and channel 2 to 0 and exit to the CRT plot). You can then write that data as a dummy data set. Many specpr users usually start every data file with a dummy data set where the title has the name of the data file (archive tape name) and maybe a description of the project.

CHAPTER 5

BASIC PROGRAM SETUP

5.1 Introduction

This section describes the steps necessary to basically configure specpr for working with data. This section describes the basic setup. To start with no setup, you must enter no arguments on the command line (see Chapter 3).

5.2 Beginning

As the program is started, messages relating to recent changes to specpr are printed. These messages can be skipped with command line flags; see Chapter 3 for details on how to start specpr from the operating system. However, to start specpr from scratch and do a basic setup, you can't skip messages.

When specpr starts with messages, the program is actually a Unix text viewing program (usually it is configured to be the program "more", but it could vary depending on your local specpr administrator). Assuming you are in the Unix "more" program, if you wish to skip these messages type "q" to quit, or type the space bar to see the next page of information.

After specpr messages have been printed, specpr asks you to press return before continuing. As soon as you do, the SPECPR version date will be written on the screen. Two options are available at this point:

Type

u <filename>

to create a new restart file or type

r <filename>

to use an existing restart file.

A restart routine is used so that SPECPR may be exited and later restarted with the same files assigned and protection and other parameters the same as the time of the exit. The restart parameters are stored in a disc file and are updated periodically, as new sections of specpr are entered during normal operation. See Chapter 3 for additional information on restart and restart files.

5.3 Protection

The next section for basic setup is the file protections. Protections are set before files are assigned, so that the protections will be in place when the files get assigned in a following step. Some users think this is backwards, but it is just another safety precaution. When assigning protections to each file ID, anticipate which files will be assigned in the file assignments section so that they will be correct. Plan your basic setup.

Specpr protection is designed to protect your data. Proper use is very important, because if your data are not protected, it takes only one little typing mistake to destroy something important! In specpr, all files may be totally or partially protected, or completely unprotected. The fourth line of the CRT header gives the protection status for all six devices (devices v, w, d, u, v, and s).

- If the protection number is positive or zero, the device is a read/write device where you can read up to and including the protection, but you can only write to the protection +1 record. For example, a value of 637 means you can read the first 637 records, and you can only write to record 638.

- A protection number of -1 means totally unprotected so you may read or write anywhere in the file randomly.

- A protection number of less than -1 means the device is a read-only device where you can read up to the absolute value protection number. For example, a value of -264 means 264 records are read only and you can't write to the file.

You can set the protections on many files on one command line. Example:

v0 d-1 u432 y-600 w-600 s-50

where v0 means to protect 0 files on device v (write to file 1 only, 0 files can be read); u432 means to protect up to record 432 (write to record 433); y-600 means that y is a read-only device with 600 records; similarly for w; and s is a read-only file with 50 records. If the protection number is zero or positive, it is incremented each time the device is written to. See section 2.3 for more details on file protection.

5.4 Observatory Location

The observatory location is used by specpr to compute air mass in astronomical data reductions, or for example, terrestrial field observations if the position of the sun were important. If you are not doing astronomical extinction corrections, this is not applicable and the not applicable, "na" flag should be set. The request for observatory location looks like:

```
 v = *unasnd*: f    1    w = *unasnd*: f    1    d = *unasnd*: f    1
 u = *unasnd*: f    1    y = *unasnd*: f    1    s = starpack: f    1
 lp: spoolfile       obs lat=    .000 deg   channels= 256 wav fl=C  256 h
 file protection: v     0,w     0,d     0,u     0,y     0,s     0 ltype= 0
-------------------------------------------------------------------------
  type in observatory or location: type  mko  for mauna kea observatory
                                   type  na   for not applicable
                                   type  wal  for wallace observatory
                                   type  mwo  for mount wilson observatory
                                   type  kpn  for kitt peak national obs.
```

type t to type in the coordinates of the observing site.

Three-letter codes are shown for often used observatories. If you wish another observatory location, type

t

and then the program will request the latitude in degrees, minutes, seconds, free format. Longitude is not needed because the sidereal time is contained with each spectrum. The observatory location is used only in computing the air mass in the object-sky subtraction routine.

5.5 Device and File Assignments

The device and file assignments menu with a new start looks like:

```
 v = *unasnd*: f    1    w = *unasnd*: f    1    d = *unasnd*: f    1
 u = *unasnd*: f    1    y = *unasnd*: f    1    s = starpack: f    1
 lp: spoolfile       obs lat=    .000 deg   channels= 256 wav fl=C  256 h
 file protection: v     0,w     0,d     0,u     0,y     0,s     0 ltype= 0
-------------------------------------------------------------------------
               *** FILE ASSIGNMENTS ***
               ---------------------------------
 to reassign files type letter and name:
    (74 characters max per file name)
 v = /dev/null
 w = /dev/null
 d = /dev/null
 s = /dev/null
 u = /dev/null
 y = /dev/null
 l = spoolfile
     e  or  x = EXIT this routine
```

To assign a file you type in the file ID letter followed by the file name. But there can be qualifiers, and Chapter 6 is devoted to the details of file assignments.

The listing device can be a file or the line printer spooler. To have the listing device assigned to the line printer spooler, specify a name of "spoolfile" for ID l as shown in the above menu. Specpr will write all lists to a file in the current directory, and when the list is complete, it will send the file to the spooler and truncate file length to zero length in preparation for the next listing.

5.6 Data File Names

The next step in basic setup is to assign the names to each file ID. The names entered here are VERY IMPORTANT for the history mechanism of specpr. The names entered here should reflect the archive name where the data will ultimately be stored. Traditionally, these names have been the

names of the mag tapes from which the data comes or to which it will be written. The names are a maximum of 8 characters. You must be certain these names are correct if you want the histories to be correct. See section 2.3.1 for more details of why this is so important.

5.7 Graphics Options

There are several graphics options available in this program. They were implemented to allow specpr to be run from different terminals, for example HP2623A, HP2648A, non-graphics terminals. Please note that the graphics option does not show up at the beginning of a new restart, but has to be set from the Initialize Parameter Routine via Program Operations. To change the terminal configuration, go to program operations control and type "g#" where # is a number from the table below. If you type g and a return with no number, it is equivalent to typing g4. Options not in the table default to HP graphics. These are the options available:

```
g2   HP2393A compatible graphics, Set White on Black mode
g3   HP2393A compatible graphics, Compliment, White on Black mode
g4   HP2393A compatible graphics, Jam mode, white on black (DEFAULT)
g11  HP2393A compatible graphics, Set Black on White mode
g13  HP2393A compatible graphics, Compliment, Black on White mode

g20  Tektronix Plot 10 compatible graphics terminals
g21  DEC VT240 with (sort of) Tektronix Plot 10 graphics
g22  GTERM (Tektronix window in a Sun gterm window)

g50  X-Windows using an hpterm window (see X-NOTE below)
g51  X-Windows using an xterm window (see X-NOTE below)

g99  Scrolling mode (non-graphics terminal)
g100 Televideo 914 alphanumeric terminal with no graphics
```

X-Windows NOTES:

- When running X-windows, the environment variable DISPLAY must be set to your X server.

- Every time you enter specpr, you must re-initialize the graphics window by typing g50 or g51 or by specifying the graphics mode on the command line as specified in section 3.1.

- User input always comes from the text window except for mouse button clicks which are entered in the graphics window.

- If the graphics window is hidden and then exposed, it will not be redrawn.

5.8 Automatically Checking File Protection

The basic setup menu (type b from the main menu) includes a command to evaluate current protections compared to what protections might be based on file sizes. This is done by the "f" command. The f command computes the number of records in the file from the file size and assumes the protection should be all records in the file. If the protection does not agree with the computed value, a warning message is printed and you are given the option to change the protection to the computed value. Further, you are given the option to make the file read only. You should use this command any time you are not sure of the proper file protection.

CHAPTER 6

DEVICE AND FILE ASSIGNMENTS

6.1 Device and File Assignments

The device and file assignment is accessed from Program Operations Control by typing "r", or by typing "b" to get to Basic Setup and then "r". The "r" stands for device and file reassignment. If you are starting specpr from scratch, specpr will automatically come to the device and file assignment menu.

The status of all files, online or not, is displayed on the terminal at the top 3 lines. File protections are displayed in line 4. The status menu, which appears at the top of most menus throughout specpr looks like the following when starting specpr from scratch:

```
 v = *unasnd*: f    1    w = *unasnd*: f    1    d = *unasnd*: f    1
 u = *unasnd*: f    1    y = *unasnd*: f    1    s = starpack: f    1
 lp: spoolfile       obs lat=    .000 deg   channels= 120 wav fl=C  256 h
 file protection: v     0,w     0,d     0,u     0,y     0,s     0 ltype= 0
-------------------------------------------------------------------------
```

Throughout the program, the device and file status in short form menu (above) is displayed in the top 4 lines on the CRT. This tells whether or not the devices are assigned, where assigned, the current file pointer position, the protections on all files, wavelength file in use, number of channels, and the file names that are used in the history.

WARNING: after assigning a new file name, you MUST set the PROTECTION and the FILE NAME VARIABLE to be used in the history.
See section 5.3 for a discussion of file protection, and 5.6 for a discussion on file names. Also in the basic setup menu (type b from the main menu), the "f" command evaluates specpr file sizes and based on those sizes, computes the protection, and if the protection does not agree with the computed value, a warning message is printed and you are given the option to change the protection to the computes value. Further you are given the option to make the file read only. You should use this command any time you are not sure of the proper file protection.

Devices assigned to /dev/null means these devices are not accessed. If access is requested when assigned to /dev/null, the program cannot respond and will tell you so.

Typing

e

or

x

will cause the program to exit to the next routine. Multiple commands are allowed in the device and file assignment section.

Example: Assign v to file lab501, d to file lab502, and l to the spooler.

v lab501 d lab502 l spooler

(NOTE: spaces are allowable between the file ID letters and the file names. Spaces are necessary before the file ID letters except the first one on the line.)

Assigning the list device to spoolfile (on Unix BSD machines, the keyword "spooler" will also work) will automatically dump the listings to the line printer as they are created. If you assign the list device to a file the listings will be appended to that file so that you can print them later.

When assigning a file, specpr tells you that it is either assigning a new file (creating it) or assigning an existing (old) one. If you are intending to assign an existing file, and specpr says it is creating a new file, then you probably misspelled it.

When assigning a file, a qualifier may be specified to assign nonstandard specpr files. The file type may be specified with the sequence:

<file ID letter> file name t=<keyword>

Currently, the only keyword defined is "3d" for a three-dimensional file type, such as data from an imaging spectrometer, see section 6.2.

6.2 3D File Types

Three-dimensional files enable the user to read and extract spectra from a three-dimensional data file by designating the pixel coordinates and the extraction direction from several types of file formats. The file setup is somewhat lengthy but required for proper reading of the 3D file. The following topics cover each aspect of the method in more detail and outline the procedure for actual use of the program.

6.2.1 3D File Parameters

File parameters are qualities of the data file which affect the way the program chooses which records are read. Currently, it takes 11 parameters to set up the file. They are listed here with descriptions and information on allowable values.

File name - Necessary to correctly open data file for i/o.

Organization - This variable is either band interleaved by line (BIL), band interleaved by pixel (BIP) or band sequential (BSQ).

File header length - Length (in records) of the header preceding the actual data cube.

Record length - Record length to read. Since Unix sees

data as a byte stream, designating a different record length than the one in which the data was written will cause the program to choose the wrong records to read. Maximum allowable value is 1536.

Record header length - Some records have headers so this was included to allow for that possibility. Must be less than record length, of course.

Data type - This is menu-selected and indicates what type of numbers the data file consists of. Allowable values are integer halfwords (2 byte), integer fullword (4 byte) or real fullwords (4 bytes).

The number of lines, number of samples, and number of bands which describe the size of the data set. Note: only a maximum of 4852 channels can be extracted along any one dimension, but the file could be larger.

Dimensions of the data cube. The conceptual method for addressing the cube is Line (or down, or scan direction, or y direction), sample (or across, meaning cross-track, x direction), and bands (or deep, z direction), where x,y,z are normal cartesian coordinates. Line 1, sample 1 is the upper left corner of the image. Line increases down, and sample increases to the right.

Data number (DN) offset and scale - These values allow a scaled range, usually used to represent a real number by an integer. The formula used to reconstitute the original value is:

Original value = (value stored + offset)*(scale).

6.2.2 3D Algorithm Description

This algorithm is divided into two major components; one that sets all of the parameters necessary to read the file correctly and another section which does all of the work required to read the data and extract spectra into arrays that specpr can handle.

The first component, essentially an information prompter, is initiated in the specpr REASSIGN file area. Follow the normal file assignment procedure but add the string "t=3d" after the device designator (u,v,w,y) and the file name. An example is, "u file3d.dat t=3d" which assigns the 3D data file file3d.dat to device u. What follows is a series of prompts for file parameters.

You need to know that the file is not actually opened until after the record length prompt because it is possible (and sometimes desirable) to use the last file setup that was entered. This is accomplished by answering the first prompt for the record length and then skipping to the parameter list. Just remember that unless you have allowed the program to go through the record length prompt the file has not been opened. The

parameter list will remind you of this by printing next to "Status =" the words "File not opened".

The secondary part of the program does the actual record reading according to the pixel coordinates input by the user in the TRANSFER, DISPLAY and OVERLAY area of specpr. The extraction direction is set by the position of the "*" in the pixel designation.
An asterisk in the z coordinate place indicates that the extracted spectra will have all z elements. A complete command example to read and extract a spectra from a 3D file assigned to device w is "wpx(n,m,*)". This translates as, for the file assigned to device w extract a spectrum aligned parallel to the z-axis. The coordinates of this pixel are x=m and y=n. Visualizing this type of extraction is not very hard, however, the mechanics are more difficult. This is why so many file parameters are required in order to read the records correctly. Listed below is a table of equations which determine the first record to read (a), the record increment (b), and the last record to read (c). Something of note when looking at the equations is that there are some simplifications imposed in order to facilitate understanding of how records are selected for reading. Also, these equations use integer arithmetic where fractions are truncated to whole numbers. Programmers will want to read the internal documentation in the source code in order to understand exactly how records are selected.

Dictionary of Variables:

rl	- records per line
pr	- pixels per record
fh	- file header length
rh	- record header length
rln	- record length
dx	- x dimension (# of Samples)
dy	- y dimension (# of Lines)
dz	- z dimension (# of Bands)
x	- x coordinate (sample) for pixel
y	- y coordinate (line) for pixel
z	- z coordinate (band) for pixel

General Equation for All Routines

```
pixels/record = rln/2   (i*2 data)
              = rln/4   (i,r*4 data)
```

Band Interleaved by Pixel (BIP)

```
records/line = dz/pr

extraction direction || x axis
        a = (y - 1)*dx*rl + z/pr + fh
        b = a + dx*rl - 1
        c = rl

extraction direction || y axis
```

```
                a = (x - 1)*rl + z/pr + fh
                b = a + (dy - 1)*rl*dx
                c = reclin * dx

        extraction direction || z axis
                a = (y - 1)*rl*dx + (x - 1)*rl + 1 + fh
                b = a + rl - 1
                c = 1
```

Band Interleaved by Line (BIL)

```
        records/line = dx/pr

        extraction direction || x axis
                a = (y - 1)*dz*rl + (z - 1)*rl + 1 + fh
                b = a + rl - 1
                c = 1

        extraction direction || y axis
                a = (z - 1)*rl + x/pr + fh
                b = dz*dy*rl
                c = rl*dz

        extraction direction || z axis
                a = x/pr + (y - 1)*dz*rl + fh
                b = a + (dz - 1)*rl
                c = rl
```

Band Sequential (BSQ)

```
        records/line = dx/pr

        extraction direction || x axis
                a = (z - 1)*rl*dy + (y - 1)*rl + 1 + fh
                b = a + rl - 1
                c = 1

        extraction direction || y axis
                a = (z - 1)*rl*dy + x/pr + fh
                b = rl*dy
                c = rl

        extraction direction || z axis
                a = (y - 1)*rl + x/pr + fh
                b = dz*dy*rl
                c = rl*dy
```

The algorithm is completed by reading the whole record into a character buffer as a byte stream, extracting the correct 4 or 2 byte section of it into full word or half word (respectively) number arrays (Max. length = 1536 bytes), applying the DN offset and scale correction to

reconstitute the original number, converting the result to a fullword real number and then loading the number into a specpr array.

6.2.3 3D I/O Setup Examples

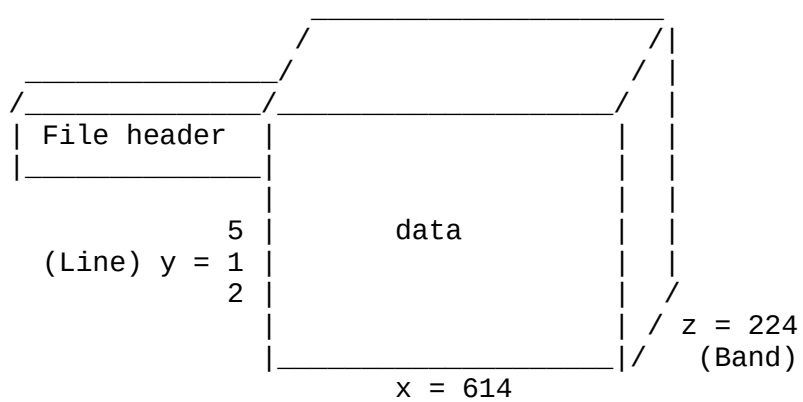
So far two different types of data have had more than preliminary testing. The AVIRIS image data file and dark current data file have both been used as sources for spectra extraction. The image data are in band interleaved by line (BIL) format and the dark current data are in band interleaved by pixel (BIP) format. The dimensions for both are as follows:

	Line y	Sample x	Band z
image	512	614	224
dark	512	1	224

Given this information (the rest is extraneous in our example) a setup and extraction procedure is listed below starting at the specpr main menu.

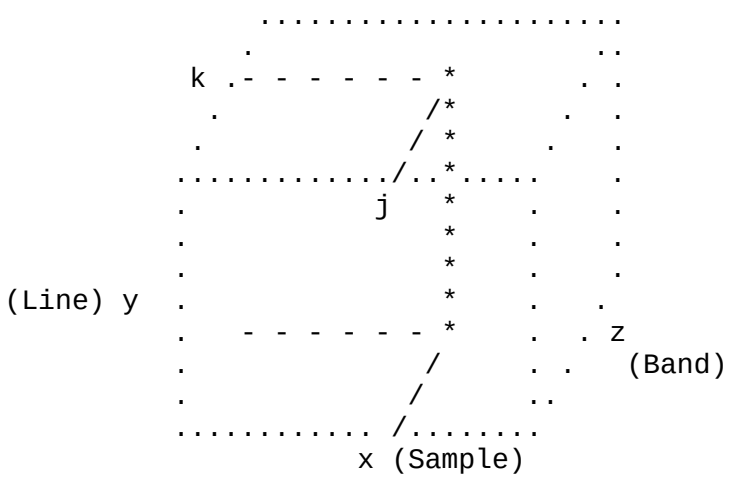
```
specpr>r                          # go to REASSIGN area of specpr
specpr>w image.dat t=3d           # assign image.dat to w
        .
        .                         # setup parameters for image.dat
        .
specpr>v dark.dat t=3d            # assign dark.dat to v
        .
        .                         # setup parameters for dark.dat
        .
specpr>e                          #  go back to main menu
specpr>t                          # TRANSFER,OVERLAY,DISPLAY
specpr>wpx(4,5,*)                 # extract pixel (4,5)
specpr>vpx(3,1,*)                 # extract pixel (3,1)
          ^
          |
          NOTE: NO space here - px(n,m,*) is req'd syntax.
```

In addition, an AVIRIS data set can be visualized as a cube of data similar to the figure below.

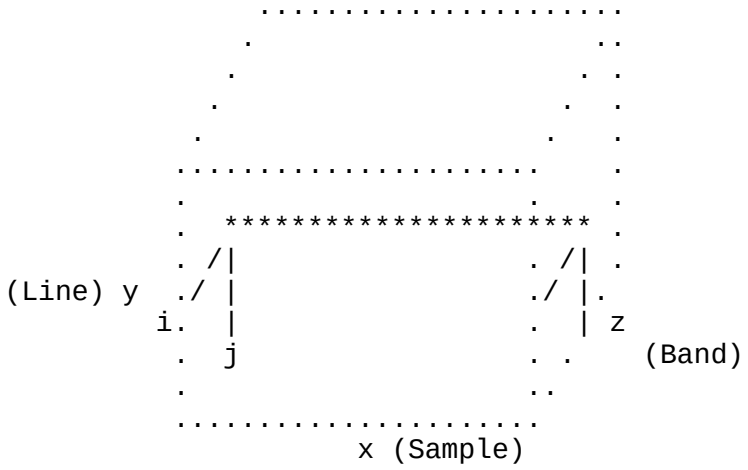


(Sample)

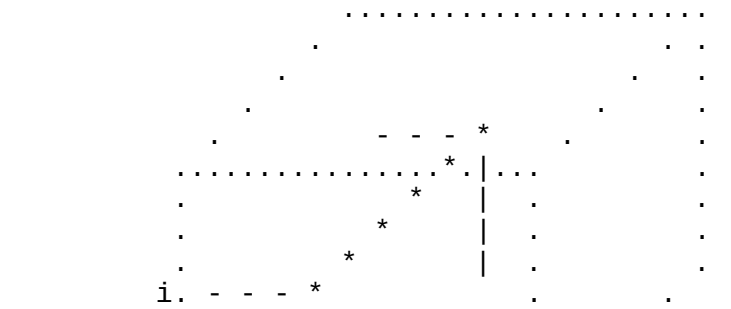
If we use the above file setups, an extraction indicated by wpx(*,j,k) will extract all values parallel to the y-axis at the xz coordinates j,k. It will appear in the above data cube as:

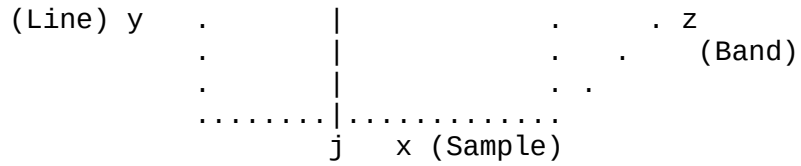


where the asterisks show which numbers are extracted. Similarly, an extraction along the x-axis such as wpx(i,*,k) looks like this,



and an extraction along the z-axis initiated by the command wpx(i,j,*) would resemble the following:





6.2.4 3D I/O Suggestions

Since 3d files have many parameters it might be easier to use a command file to do the file setup for you. Once you are in the REASSIGN area of specpr type a greater than symbol (>) and a file name. The information entered for file parameters will be saved in that file for use later. When you need to set that particular 3d file up again type (in REASSIGN area of specpr) a less than symbol (<) and the file name you used to store the information. If you can't remember what the name is type "!ls" (don't include the double quotes) for a listing. If you need to change one or two parameters in the command file type "!vi filename". Edit the parameters that need changing just as you would at any other time and then exit as usual. The program will return you to the REASSIGN area and you can type in <filename to have the command file enter the parameters for the file.

It is very important (and will save you time in the long run) to make sure that the parameters are correct when the program provides the checklist for your inspection.

If the dimensions were entered in the wrong order a message will usually appear telling you that one dimension must be a multiple of some integer number. Usually an i/o error message saying that an attempt was made to access an illegal record number (either < 1 or > max number of records) also appears. In this case, note which dimension has to be a multiple of the integer number and correct that dimension first. If you still aren't sure what order to enter the other dimensions, try one combination. Since there are only 2 possibilities only one combination is left to try if you're wrong.

If the file parameters are the same as the file previously setup the array holding the values can be used again. When the first prompt comes up enter the record length and then an s on the next prompt. You will skip to the parameter check list where you should check status to make sure the file opened successfully before continuing.

6.2.5 3D I/O DEMONSTRATION

What follows is an actual 3D file setup and spectral extraction procedure being executed. The screens you will see as you progress through this example are included. Explanations not part of the screen stuff are indicated by an asterisk in the first column. Entries by the user are to the right of the > prompt.

```
>specpr r1 -
```

```
* Translation:  Execute specpr, use a restart file named r1, do a
* non-verbose entry into specpr (-).  NOTE:  Type this in at system level
* prompt.


 Restarting
restart file= r1
------------------------------------------------------------------------
 v = spd0071 : f    4    w = splib001: f   19    d = spd0029 : f  169
 u = calger1 : f    1    y = spd0084 : f   25    s = starpack: f    1
 lp: spoolfile       obs lat=    .000 deg   channels=  61 wav fl=V    3 a
 file protect: v  -1358,w   -635,d   -168,u -55144,y    115,s   0 ltype= 3
------------------------------------------------------------------------
 MAIN MENU: ********** Program Operations Control ********************

   INFO:     "in" to turn OFF information

   LIST:      l followed by v,w,d,u,or y  to list the contents
                         of the corresponding file
   DISPLAY:   t  to DISPLAY on screen, OVERLAP on screen
   MATH:      m  to do MATH operations
   TRANSFER:  t  to TRANSFER (COPY) files

   PLOT:      p  to PLOT SPECTRA on PLOTTER/printer
   SETUP:     b  to change SETUP PARAMETERS
   FILES:     r  to REASSIGN files and devices
   STARPACK:  s  to create a STARPACK for extinction corrections
   PRINT RST: f  to print summary of the current restart file

   EXIT:      EX  to exit program

>r

* Translation:  Go to REASSIGN area of specpr

________________________________________________________________________
 v = spd0071 : f    4    w = splib001: f   19    d = spd0029 : f  169
 u = calger1 : f    1    y = spd0084 : f   25    s = starpack: f    1
 lp: spoolfile       obs lat=    .000 deg   channels=  61 wav fl=V    3 a
 file protect: v  -1358,w   -635,d   -168,u -55144,y    115,s   0 ltype= 3
------------------------------------------------------------------------
               *** FILE ASSIGNMENTS ***
               ---------------------------------
 to reassign files type letter and name:
    (74 characters max per file name)
 v = /d1/gswayze/data/ger/spd0071
 w = /d1/speclib/splib001
 d = /d1/fkruse/data/spd0029
 s = /dev/null
 u = /i1/tmp/GER/canon.city/calger2.dat
 y = spd0084
 l = spoolfile
     e  or  x = EXIT this routine
```

```
> u /i1/tmp/GER/canon.city/calger2.dat t=3d

* Translation:  Assign /i1/tmp/GER/canon.city/calger2.dat to device
* u.  Indicated 3D file by the "t=3d".
```

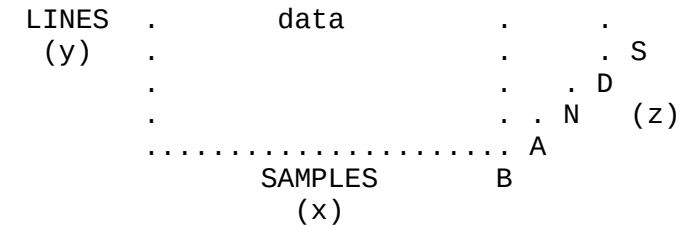
```
 ------------------------------------------------
 This program is designed to read multidimensional
 data files although at present it is limited to 3-
 dimensional data sets.  It is recommended that you
 put the input file parameters in a command file to
 save time if you find it necessary to change the
 specpr u,v,w,y or d setups often.  Currently,
 it takes 12 parameters to set up a file.
                 *************
 NOTE:  AT ANY TIME entering an e or x terminates
 the setup routines without opening the 3D file.
 ------------------------------------------------

 What is the length of each record (in bytes)?
 (e or x to exit)
>1024

 Opening file...

 File opened successfully...


 Select file organization:
                    1   BIL (band interleaved by line)
                    2   BIP (band interleaved by pixel)
                    3   BSQ (band sequential)
                    s   skip to parameter check list
                    e,x exit program
>1
 How many records long is the file header?
 (e or x to exit, b to back up, s to skip to list)
>0
 What is the length of the record header in bytes?
 (e or x to exit, b to back up, s to skip to list)
>0
 What is the DN offset?
 (e or x to exit, b to back up, s to skip to list)
>0
 What is the DN scale?
 (e or x to exit, b to back up, s to skip to list)
>.00005
                      ......................
                    .                    . .
   ..............                      .   .
 ....................................      .
 . File header  .                    .     .
 ...............                     .     .
               .                     .     .
```




```
 Please enter the dimension lengths of the input array
 which can be either 2 or 3 -dimensional (y,x,z):
 (e or x to exit, b to back up, s to skip to list)
>904  512 61
 Choose the file data type:
 --------------------------

 1   Integer*2 (half-word)
 2   Integer*4 (full-word) - specpr normal
 3   Real*4 (full-word)

 b   backup
 s   skip to parameter check list
 e,x to exit
>1
 What is the deleted point value?
>0
 Your input is as follows:
 -------------------------------------
           /i1/tmp/GER/canon.city/calger2.dat

 Parm.#
                  Status =  File opened
  1  Record Header length =  0
  2                Format =  BIL
  3    File Header length =  0
  4         Record length =  1024
  5             DN offset =  0
  6              DN scale =  4.99999E-05
  7           x-dimension =  512
              y-dimension =  904
              z-dimension =  61
  8             Data type =  Integer*2
  9       Point drop flag =  0
 -------------------------------------


 Enter the parameter # to be changed
 c     to CONTINUE
 e,x   to EXIT file setup
>c
 Continuing...
 Please type in the title (16 characters or less).
 -----------------|
>Canon City S CALIBRATED
 Please enter the time (hh mm ss) data was acquired
```

```
 in Universal or Civil time (r to return to main).
>19 00 00
 Enter the date (mm dd yyyy) of data acquisition.
>09 16 1988
 File opened.   Press return to continue.
> <CR>

* Translation:  Carriage return.
```

```
 v = spd0071 : f    4    w = splib001: f   19    d = spd0029 : f  169
 u = calger1 : f    1    y = spd0084 : f   25    s = starpack: f    1
 lp: spoolfile       obs lat=    .000 deg   channels=  61 wav fl=V    3 a
 file protect: v  -1358,w   -635,d   -168,u -55144,y    115,s   0 ltype= 3
```

```
               *** FILE ASSIGNMENTS ***
               ---------------------------------
 to reassign files type letter and name:
    (74 characters max per file name)
 v = /d1/gswayze/data/ger/spd0071
 w = /d1/speclib/splib001
 d = /d1/fkruse/data/spd0029
 s = /dev/null
 u = /i1/tmp/GER/canon.city/calger2.dat
 y = spd0084
 l = spoolfile
     e  or  x = EXIT this routine

>e

* Translation:  Exit to main menu.
```

```
 v = spd0071 : f    4    w = splib001: f   19    d = spd0029 : f  169
 u = calger1 : f    1    y = spd0084 : f   25    s = starpack: f    1
 lp: spoolfile       obs lat=    .000 deg   channels=  61 wav fl=V    3 a
 file protect: v  -1358,w   -635,d   -168,u -55144,y    115,s   0 ltype= 3
```

```
 MAIN MENU: ********** Program Operations Control ********************

   INFO:     "in" to turn OFF information

   LIST:      l followed by v,w,d,u,or y  to list the contents
                        of the corresponding file
   DISPLAY:   t  to DISPLAY on screen, OVERLAP on screen
   MATH:      m  to do MATH operations
   TRANSFER:  t  to TRANSFER (COPY) files

   PLOT:      p  to PLOT SPECTRA on PLOTTER/printer
   SETUP:     b  to change SETUP PARAMETERS
   FILES:     r  to REASSIGN files and devices
   STARPACK:  s  to create a STARPACK for extinction corrections
   PRINT RST: f  to print summary of the current restart file

   EXIT:      EX  to exit program
```

```
>t
 Transferring to display & math routine
```

```
 v = spd0071 : f    4    w = splib001: f   19    d = spd0029 : f  169
 u = calger1 : f    1    y = spd0084 : f   25    s = starpack: f    1
 lp: spoolfile       obs lat=    .000 deg   channels=  61 wav fl=V    3 a
 file protect: v  -1358,w   -635,d   -168,u -55144,y    115,s   0 ltype= 3
```

```
 ********** data DISPLAY, TRANSFER, and OVERLAY **********

 Type  i  to turn ON information

>upx(1,1,*)

Demonstration complete.
```

CHAPTER 7

MAIN MENU: PROGRAM OPERATIONS CONTROL

7.1 Introduction

The Program Operations Control is the main menu part of the program which allows access to math operations, file list, overlay, transfer, display, plot, file assignments, initialization and other applications programs such as the extinction routines. This is also the only point where the program should be terminated. The main menu looks like:

```
v = spdemos : f    1    w = *unasnd*: f    1    d = *unasnd*: f    1
u = *unasnd*: f    1    y = *unasnd*: f    1    s = starpack: f    1
lp: spoolfile       obs lat=    .000 deg   channels= 256 wav fl=C 256 h
file protection: v    53,w     0,d     0,u     0,y    0,s    0 ltype= 0
-------------------------------------------------------------------------
MAIN MENU: ********** Program Operations Control ********************

  INFO:     "in" to turn OFF information

  LIST:      l followed by v,w,d,u,or y  to list the contents
                        of the corresponding file
  DISPLAY:   t  to DISPLAY on screen, OVERLAP on screen
  MATH:      m  to do MATH operations
  TRANSFER:  t  to TRANSFER (COPY) files

  PLOT:      p  to PLOT SPECTRA on PLOTTER/printer
  SETUP:     b  to change SETUP PARAMETERS
  FILES:     r  to REASSIGN files and devices
  STARPACK:  s  to create a STARPACK for extinction corrections
  PRINT RST: f  to print summary of the current restart file

  EXIT:      EX  to exit program
```

7.2 Menu Information

Information on the screen can be suppressed by typing

in

or restored by typing

i

7.3 Listing the Contents of a Data File

To list the contents of the data files (v, w, d, u, or y), type l followed by the corresponding letter. For example to list the contents of file v, type:

lv

See Chapter 11 for instructions once you are in the list routine.

7.4 Terminating Program

To stop specr, type

EX

All files and devices will be closed properly, and specpr will print a summary of the files assigned and their protections. When you restart specpr, the files will be opened and protections set to the same state as when you quit at this point.

7.5 File Display, Transfer, and Overlap

To transfer (copy) files, or display or overlap data on the CRT, type

t

See Chapter 10 for information once you are in the display, transfer and overlap routines.

7.6 Changing Initialization Parameters

To change the basic setup parameters or reassign files and devices, type

b

The basic setup parameter menu is then displayed:

```
   v = spdemos : f   25    w =         : f    1    d = *unasnd*: f    1
   u = *unasnd*: f    1    y = *unasnd*: f    1    s = starpack: f    1
   lp: spoolfile       obs lat=    .000 deg   channels= 471 wav fl=V  23 a
   file protection: v    53,w     0,d     0,u     0,y    0,s    0 ltype= 3
  ------------------------------------------------------------------------
                      *** SETUP parameters ***


        type  o    to change the OBSERVATORY or observatory site
        type  r    to REASSIGN FILES and devices
        type  f    to EVALUATE PROTECTION vs file sizes
```

```
            (no response indicates all is consistent)
      type  g    and number to set GRAPHICS type (see manual)
      type  b    to toggle BELL
      type  v    to change the NAME of device v
      type  d    to change the NAME of device d
      type  u    to change the NAME of device u
      type  y    to change the NAME of device y
      type  w    to change the NAME of device w
      type  cp   to change the  FILE PROTECTION

      press return to go back to the MAIN routines.
```

See Chapter 5 for details on each command.

From the main menu, it is possible to enter a multiple command and go right to the basic setup section desired. From the main menu, type b<basic setup command>. For example to change protection from the main menu, type:

bcp

7.7 Device and File Assignments

To go to Device and File Assignments, type

r

See Chapter 6 for details on the assignment routines.

7.8 Extinction Routines

To do extinction calculations and create a starpack or list the starpack file by titles, type

s

See Chapter 12 for details on the extinction routines.

7.9 Plot Routines (to Hardcopy Plotters)

To plot spectra on the various supported Plotters, type

p

See Chapter 13 for details on the hardcopy plotting routines.

To go to the mathematics routines, type

m

See Chapter 8 for details on the math routines.

7.11 Restart Summary

Typing

f

will list the contents of the current restart file on the listing device.

CHAPTER 8

MATH OPERATIONS

8.1 Introduction

All mathematical operations are carried out in Math Operations except for extinction analysis. The math routines also contain the "special functions." Special functions do some operation on a data set, or generate a data set. While most special functions are math related, they need not be (for example sort a data set into increasing wavelengths). The general sequence the user sees in the Math Operations section is the following:

- At the math main menu, the user enters a math command
- The math function displays what it is going to do and the user works his/her way through the function-specific menus.
- When the math function completes normally, the user is asked where the data should be written (which v, w, d, u, or y data file and what record number). The data are not written at this time; that is the second to last step below. The ultimate location is requested at this point so that the spectrum is properly identified in case print-outs are made.
- The user is asked to type in a title to the data set.
- The data are plotted in the graphics terminal (with the crt plot routines, see Chapter 9). In the crt plot routines, the user can change or add to the header information.
- The data are written to the data file when the user soft exits from the crt plot routines (exit with e). See warnings and important rules below.
- The user is returned to the math operations menu.

WARNING: the number of data channels operated on by any math routine is determined by the number of channels in the data set in use (displayed in the top few lines of each routine). It is UP TO THE USER to be sure the number of channels is set properly. Improper number of channels is a very common reason for improper results.

WARNING: the number of channels in the data set written by the math routines is controlled by the number of channels in use AT THE TIME the user EXITS the CRT PLOT ROUTINES. If you change the number of channels in the header routines, it will have no effect--that value is overridden by the math writing routine.

IMPORTANT RULE: the wavelength pointer will be set automatically by the math routine at the time the data are written to the data file IF the wavelength set in use IS IN THE SAME DATA FILE.

> For example, if you divide two spectra in data file "w", and write the results in data file "v", use a

wavelength set that is already in "v". Thus if:

```
w 10 = wavelengths with 56 channels
w 11 = data set "a"
w 12 = data set "b"
```

and you want to divide w11 by w12, first put a copy of w10 into the "v" file (see Chapter 10 on how to copy/transfer), then assign the "v" wavelength set before you do the division. The wavelength pointer will then be set automatically when the results from the math operation are written to the v-file.

8.1.1 Multiple commands from the math command menu.

One entire line (80 characters) of commands may be typed in for execution. Each math command must be separated by a comma.

Example:

w11/w12,v23*c5.62e,y236/w12

After each math operation is completed, the next command is executed without returning to the math menu.

8.1.2 Menu Information

The information in the Math menu can beturned off by typing

in

or turned back on by typing

i

and nothing else on the command line.

8.2 Subtraction, Multiplication, and Division

Generally these operations contain a file ID (v, w, u, d, or y), record number, operation sign

- \- for subtraction,
- * for multiplication, and
- / for division,

then the second file ID (v, w, d, c, u, y, or s; s is used in division only), the second record number, and the options. Options include:

e to include 1 sigma error bars,
n to do a subtraction without airmass calculation,
b for turn on band normalization (section 8.9),
bn for turn off band normalization,
t for change sidereal time in subtraction with airmass calculation,
r for change RA and declination in subtraction calculation, and
p for production processing.

The production processing option stops only to request the title and writes the result to device v without doing the CRT plot. This option (p) makes processing equivalent to the time spent in performing batch processing. When you enter the output title (section 8.8), you may turn off (or on) production processing by typing "pn" (or "po" for on) after the title. See section 8.8.

The c in place of the second file ID is for constant. The value of the constant is then placed where the second record number is located.

Example:

w10-11, w12-13, v10*c1.3, v23/46eb, v12-d10np bn

Here file w record 11 is subtracted from 10 (the default when the second file ID is left out is the first file ID). Next w13 is subtracted from w12, then v10 is multiplied by the constant 1.3, next v23 is divided by v46 with 1 sigma standard deviation (errors) included and also a band normalization performed.

On the math command line, spaces may be included at any point except within a number, or they may be completely left out. The program processes the commands sequentially. When performing each operation, the program enters the appropriate routine, writes the titles of the requested files on the CRT, and pauses for a continue command (unless the Production option has been set). If you type

x

here, the program exits directly to math operations. If you type

e

the program exits that operation and begins processing of the next operation if one exists. The production option is set to no production at the end of each operation whereas the band normalization option stays on until turned off by a

bn

command.

When the operation is completed, the program goes to section 8.1.

Math Operations can be used to change the wavelength file set in use. Example:

v23/v45V2, v86*v27W46

where the capital file ID specifies a wavelength set within the corresponding data file. The number of channels in use will then be read from the wavelength set.

Special Note on the Division Routine

Division by a small number (-10-36 to +10-36) is defined as a deleted point

(-1.23x10+34 = A DELETED POINT)

x by this program.

Special Notes on the Subtraction Routine

Normal subtraction calculates airmass based on the coordinates found in the first 13 characters of the title of the first input file; if these coordinates are not entered in the correct format (HHMMSS+DDMMSS no spaces, declination sign must be explicitly specified as plus or minus, and any one of the first six digits must be non-zero), the routine issues the command

TYPE IN RIGHT ASCENSION AND DECLINATION

and awaits a correct response. The routine repeats this message until a correct response is encountered--e and x are NOT correct responses. The correct response is up to six numbers separated by spaces, any one of the first three of which must be non-zero, the fourth of which should be signed only if negative, and all the rest of which should be unsigned.

TO GET OUT OF THIS LOOP, enter any single non-zero number, and type e or x at the next possible opportunity.

(This format may be confusing because the interpretation routine from the first 13 characters requires an explicitly specified plus or minus sign for the declination, and the request for RA and DEC cannot interpret plus signs.)

8.3 Addition Routine

The addition routine can add from 2 to 128 spectra with deletion of zeros or requested points. To access the addition routine, type + and any options:

```
+     no options
+e    addition with propagation of existing error bars
+s    sum
+a    average
+se   sum with errors
+ae   average with errors
```

with no other math operations after that on the input line (because they will not be processed due to a programming restriction). Upon entering the addition routine, you will be asked whether you want to average or sum (unless you specified the a or s option as given above). Type

```
a    to Average,
s    to Sum and,
```

Next type:

```
z   to DELETE ZEROS in the input data,
v   to DELETE data values outside a user set limit, and/or
s   to DELETE data values outside a certain standard
    deviation limit.
```

Next you will be given instructions on entering the files to be added or summed. Type in the file ID [v, w, d, u, y or c (for constant)] and the record number or constant.

If you wish to delete points from this record, type d after the record number or constant. If you elect to delete channels, you may start the channel number list after the "d". You may use more than one line to input these numbers, but each channel to be deleted from the sum or average must be listed. When you are through entering numbers to be deleted, type a "c" to continue.

Example:

```
v23                    \# normal input, no deletions,
v23 d 1 2 26t32 c      \# delete channels 1, 2, 26 through 32.
```

When you are through entering record numbers, type

b

to begin analysis.

Note only 1 constant can be entered per addition run.

If you type

e

or

x

the program will exit directly back to math operations; otherwise, after the analysis is complete, the program goes to section 8.7.

8.4 Error Analysis

Error analyses are included in addition, multiplication and division, and subtraction and many of the special functions (see individual functions - section 8.6) In all cases of error analysis, the data numbers should be within the range of

1.0x10-15 to 1.0x10+15

to avoid overflow. If overflow occurs, the error is set to zero. If the data is zero in the multiplication or division routines, when using the propagation of errors, the error is set to zero for that channel. The errors represent 1 sigma standard deviation of the mean and are normally first computed in the addition routine as a standard deviation of the mean computation. The equations used in propagating errors are as follows:

Given $a \pm \sigma_a$, $b \pm \sigma_b$, $c \pm \sigma_c$ and the result $x \pm \sigma_x$ the equations used are:

division:

$$X = a/b$$

$$\sigma_x = ((\sigma_a/a)^2 + (\sigma_b/b)^2)^{1/2}X \qquad \text{(eqn 8.4.1)}$$

multiplication:

$$X = a\ b$$

$$\sigma_x = ((\sigma_a/a)^2 + (\sigma_b/b)^2)^{1/2}X \qquad \text{(eqn 8.4.2)}$$

subtraction:

$$X = a - b$$

$$\sigma_x = \{\sigma^2_a + \sigma^2_b\}^{1/2}$$

addition (when errors already exist):

(sum): $X = a + b + \ldots + c$, (eqn 8.4.3)

$$\sigma_x = \{\sigma^2_a + \sigma^2_b + \ldots + \sigma^2_c\}^{1/2}$$

(average of n spectra): $X = (a + b + \ldots c) / n$

$$\sigma_x = \{\sigma^2_a + \sigma^2_b + \ldots + \sigma^2_c\}^{1/2} / n$$

standard deviation of the mean is derived from (generate errors for the first time):

$$\sigma_x = \left(\left(\sum_{i=1}^{n} (x_i - \bar{x})^2\right) / (n-1) \right)^{1/2}$$

where n = number of spectra, x_i = each spectrum, and

$$\bar{x} = 1/n \sum_{i=1}^{n} x_i$$

8.5 Algebraic and Trigonometric Functions

A complete set of algebraic, trigonometric, and power functions exists similar to those on a scientific calculator. No error analysis is included with any of these functions. The functions are accessed by file ID, file number, colon (:), and the name of the function. Functions are:

Exponential:	exp
Natural logarithm:	ln
Common logarithm:	log
Ten to spectrum power:	10**x
Inverse:	1/x
Spectrum to the constant power:	x**c
Constant to the spectrum power:	c**x
Sine (in radians):	sin
Cosine (in radians):	cos
Tangent (in radians):	tan
Sine (in degrees):	sind
Cosine (in degrees):	cosd
Tangent (in degrees):	tand
Inverse sine (radians):	invsin
Inverse cosine (radians):	invcos
Inverse tangent (radians):	invtan
Inverse sine (degrees):	invsind
Inverse cosine (degrees):	invcosd
Inverse tangent (degrees):	invtand
Hyperbolic sine (radians):	sinh
Hyperbolic cosine (radians):	cosh
Hyperbolic tangent (radians):	tanh
Absolute value:	abs
Integer value:	int
Fractional part:	frac

Examples:

```
v397:ln       \# natural log of v397
w76:cos       \# cosine of w76.
```

In both the spectrum to the constant power and constant to the spectrum power, the value of the constant is after the command (x**c or c**x):

```
v397:x**c3.65       \# v397 to the 3.65 power,
v397:c**x3.65       \# constant 3.65 to the v397 power
```

No options are valid with any of these operations including errors (except 1/x can include errors), band normalization, and production processing. This is due to programming limits (there has to be some). Since these routines are not used too often, this should make little difference. What are they used for? Normal spectral processing does not need them, but say you wanted to make a plot of a phase function for a Lambert sphere:

$$\Phi(\alpha) = 1/\pi(\sin\alpha + (\pi - \alpha)\cos\alpha) \qquad \text{(eqn 8.5.1)}$$

You could generate data for increasing from 0 to 180 using the wavelength routines and then this function using the math functions. The result can then be plotted on the Gould printer plotter using the plot routines.

8.6 Special Functions

Special function subroutines are additional routines for operating on any kind of data. They do any operation: for example a data list, a data editor, or a specific computation. Some special functions operate on a data set (for example, a smoothing function) while others may create a new data set (for example, read one from an ascii text file or digitize one on a digitizing tablet, or simply compute one, like a Plank black body).

Special functions are called from math operations by one of two methods, depending on whether or not the function operates on an existing data set. If a function requires a data set, you enter the file id, record number, an "f" followed by the function number. For example, to smooth (function 7) data set v23, you would enter the command:

```
v23f7
```

Special functions also have alias names. For example, the smoothing function f7 has the alias "smooth". Thus

```
v23f7
```

and

```
v23smooth
```

are equivalent.

Some special functions do not need an input data set. In that case, just specify the function or its alias. For example, to compute a Planck black body, type:

f6

or

planck

and the function will be started.

Note there is no colon in the command line for special functions as compared to the algebraic and trig routines.

Special function 1 is a function which lists a catalog of all available special functions. Just type "f1" from the math operations command line. This list contains a code at the end of the short description which tells if the function needs a data file input before the function on the command line. If this is the case, you will see an "(f)" at the end of the description. If the function can also propagate the error bars to the data set, it will have an "(f,e)" at the end of the description.

To propagate error bars for those routines that support error propagation, you must specify an "e" option on the command line. For example, to propagate error bars in the cubic spline interpolation routine, function f12, alias cspline, and operate on data set v23, type the command:

v23f12e

or

v23cspline e

Specpr is designed to add special functions easily. Subroutine calls already exist in the program for many as yet unwritten routines. These routines are designed for easy linking to the present program. New routines can be user written with less effort than a completely separate program since, in most cases, SPECPR handles all the data management.

8.7 Return from a Math Operation or Function

When a routine such as addition, subtraction, special function, trig. function, etc., has finished, the program asks the user where he/she wishes the data to be written. The data can be written to any of the files: v, w, d, u, or y if allowed by the protection. If the protection on the device is positive or zero, the data can only be written to the protection value plus 1. Thus, it is not necessary to type in this value since it is the only value possible. Simply type the file letter ID and return; the record number will be set automatically.

The data set is not written at this point. It is written after the CRT plot routines. The user may also exit the current operation from this point. By typing

e

the current operation is terminated and the program begins execution of the next command if there is one or returns to Math Operations if there are no other command requests. If the user types

x

all processing stops, and the program returns directly to Math Operations.

If the user requests the data to be stored, then the program continues to the titles routine (section 8.8).

8.8 The Titles Routine

The titles routine displays 2 titles on the CRT which may be selected for the new data, or one of 25 user stored titles can be recalled, or the user can type in a new title and store it. The 2 titles displayed are as follows.

The "option p title"	This is the "present title" of the first data set read in for 2 file operations or the title of the last data set in the addition routine or many other routines. Some special functions may generate a suggested present title.
The "option l title"	This is the title requested for the last operation.

In the subtraction routine, the program decodes an "object-sky" title as the option p title. This is based on inputs from the "Wedge" CVF spectrometer data system and may not be valid with other types of subtraction.

All titles are 40 characters long, so to select the option p or l titles, type

p

or

l

with no other characters in columns to 40. To recall one of the stored titles, type

t#

where the "#" is the title number (1 to 25). There can be no other characters in columns 1 to 40. If other characters are detected here, the program will think this is a new title and use it. The title used, whether recalled or new, is stored in title file t1. To list the contents of the title file, type

tl

with no other characters on the command line. The titles will be listed and the program will again ask for a title.

Once a title has been selected (recalled or new), further commands can be placed in columns 40 or greater. To store a title, type

t#

where the "#" is the title number (3 to 25 are valid). Remember, the "t#" must be positioned in columns 41 or greater with the requested title (recalled or new) in columns 1 through 40.

The Band normalization or Production processing options can also be turned on or off by control characters in columns 41 or greater. Type

b

to turn on or

bn

to turn off Band normalization. Type

p

to turn on or

pn

to turn off the Production processing option (the n is for none).

The vertical scale of the pending crt plot can also be changed after column 41 of the title command line. Simply type "c and the lower and upper range. For example, to change the scale from 0.92 to 1.63, type

c 0.92 1.63

after column 41.

An example of a complete title entry would be

Scapolite HS351 / Halon 4x resol .2-3um t5 b c .5 1.5

which means to make the title of the current data set "Scapolite HS351 / Halon 4x resol .2-3um", store it in title location 5, turn on band

normalization and change scale of the pending CRT plot to lower bound 0.5 and upper bound 1.5.

The current operation can be terminated as in section 8.7 by typing

e

or

x

with no other characters on the line. Typing "e" will terminate the present operation, and the program will begin the next operation if there is one. Otherwise, it will return to Math Operations. Typing "x" will terminate all operations and will go directly to Math Operations.

If a title is requested, the program will then go to the Band Normalization Routine (section 8.9) if the Band Normalization option is on or to the CRT plot routines if the Band Normalization and Production options are off. If the Production and Band Normalization options are on, the program will go to the Band Normalization; otherwise, if only the Production option is on it will write the data and begin processing the next command.

8.9 Band Normalization

The Band Normalization is a least squares analysis over a user selected band with the spectrum scaled so the middle of the band is scaled to unity. The band can be one channel or however many channels are in the spectrum. The Band Normalization routine is called after each operation if the Band Normalization option is on.

When you are in the band normalization routine, you have several options to control the region that will be normalized.

Channels can be deleted inside the region of normalization by typing the

d

command. The program then asks for the points. One line can be filled with points to be deleted (numbers, no characters).

The band limits can be moved by typing the

m

command. Deleted channels can be reinserted by typing the

r

command.

The band area with the fitted line along with the correlation coefficient, the previous Band Normalization factor (the normalization factor of the data before this normalization), the current normalization factor (as determined by this normalization), and the future normalization factor (if this normalization is carried out) are plotted on the CRT. The future normalization factor equals the previous times the current factors.

WARNING: if the previous normalization factor was zero when the Band Normalization routine is entered, it is reset to 1.0.

To perform the normalization of the data, type

b

otherwise, type

e

to exit the routine without changing the normalization factor. The "e" soft exit continues to the next step which will normally be the CRT plot routine unless production processing is turned on. If production processing is turned on, the specpr will write the data when exiting band normalization with an "e" soft exit.

A hard exit, command "x" from the Band Normalization routine, will terminate all pending commands and return to the Math Operations command line.

Default Band Normalization limits when program is begun are 30 to 38.

F1: List of Special Functions **Alias: list**

Special function 1 provides a list of all available special functions. The list includes the function number, the function alias, and a code to indicate if you need to include an input data set before calling the function, and if the function can handle propagation of error bars.

The list is as follows.

```
Special Function 1: a list of the special functions
---------------------------------------------------

f 1 - list        : Description of all special function routines
f 2 - shif[t]     : SHIFTS data left or right a number of channels (f,e)
f 3 - sequ        : SETS UP SEQUENTIAL PROCESSING for the user
f 4 - exse[qu]    : executes SEQUENTIAL PROCESSING set up by f3. No user
                        access.
f 5 - cont[rem]   : CONTINUUM REMOVAL (use also f12)
f 6 - plan[ck]    : computes PLANCK BLACK BODY function
f 7 - smoo[th]    : SMOOTHING function  (f,e)
f 8 - tran[spose] : channel-file TRANSPOSE
f 9 - band[rem]   : BAND REMOVAL (reflection method)
f10 - sort        : SORTs channels into increasing wavelength order (f,e)
f11 - luna[rtherm]: Lunar THERMAL REMOVAL (f,e)
f12 - cspl[in]    : CUBIC SPLINE interpolation to new wavelength set,(f,e)
f13 - merg[e]     : MERGes 2 files (f,e)
f14 - edit        : EDITs file and errors (f,e)
f15 - gaus[sfmt]  : FORMATS Gaussian parameter file (f)
f16 - line[gen]   : user LINE SEGMENT GENERATOR
f17 - conv[olv]   : high to low resolution SPECTRAL CONVOLUTION (f).
f18 - bloc[kav]   : BLOCK AVERAGES and Statistics (f,e).
f19 - poly[fit]   : POLYNOMIAL FIT routine (f,e)
f20 - read        : READ/WRITE data from/to a TEXT file
f21 - calc[poly]  : Calculate N term POLYNOMIAL (f)
f22 - digi[tize]  : DIGITIZATION tablet input routine
f23 - pars[e]     : MATH PARSER
f24 - star[moontherm] : star/moon-spot reflectance THERMAL REMOVAL (f,e)
f25 - twoc[omlsq] : Two-Component LEAST SQUARE ANALYSIS (f)
f26 - not available yet
f27 - not available yet
f28 - not available yet
f29 - not available yet
f30 - not available yet
f31 - not available yet
f32 - not available yet
f33 - not available yet
f34 - not available yet
f35 - not available yet
f36 - not available yet
f37 - not available yet
f38 - not available yet
f39 - noise       : Random Number Noise Generator
f40 - leastsq     : compute LEAST SQUARES between two spectra (f)
                        under development
```

```
f41 - bin          : BINNING routine (f) under development
f42 - bandmap      : Map band depths and fit from a reference
                          spectrum (f) (under development)
f43 - fft          : FFT and Inverse FFT
f44 - suh          : Segmented Upper Hull automatic CONTINUUM ANALYSIS(f,e)
f45 - aba          : Automatic BAND ANALYSIS (f,e)
f46 - abaout       : Band Analysis output to a binary FILE (f)
f47 - abaspecrec   : Band Analysis SPECTRUM RECREATION (f)
f48 - hpdigit      : DIGITIZATION from an HP Terminal/Tablet
f49 - lininterp    : Linear interpolation (f,e)
f50 - wavreg       : wavelength registration
```

When f1 is started, it lists one page at a time and asks the user to type

e

or

x

to exit, or return to continue to the next page.

In the list, after each function number is the function alias. For example, in the above list, f50 is also "wavreg".

Each special function description has a code at the end of the short description which tells if the function needs a data file input before the function on the command line. If this is the case, you will see a "(f)" at the end of the description. If the function can also propagate the error bars to the data set, it will have an "(f,e)" at the end of the description.

F2: Shift Channels Left or Right **Alias: shift**

Special function f2 shifts data left or right an integer or fractional number of channels. A data set is required and errors can be included. Fractional shifts are done by linear interpolation.

When the routine has been entered, type:

the number of channels to shift, or e to soft exit, or x to hard exit.

A positive number (integer or real) indicates a shift right and a negative number left.

If X is the number of channels to shift, the value of X is split into the integer part (n) and the fractional part (XN). First, the integer part is shifted

$$D_{i+n} = D_i \qquad \text{(eqn 8.f2.1)}$$

where Di is each data channel in the spectrum. Next, the fractional part is interpolated

$$XN' > 0.0; \quad D_i = D_{i+1} - D_i * XN' + D_i \qquad \text{(eqn 8.f2.2)}$$

$$XN' < 0.0; \quad D_i = D_i - D_{i-1} * XN' + D_i \qquad \text{(eqn 8.f2.3)}$$

where XN' = -1.0 * XN. Data channels not in the shifted region are set to zero. The errors to the data are shifted in a similar manner.

Before June 24, 1979, there was a slight error in the program (XN' = XN) making the fractional shift appear to go in a different direction than the integer part shift. The history has been modified to signal this new change:

Old history (X = + 2.5 operation on WDG536 file 238):

"F2: wdg536 FILE 238 shift right 2.5 channels"

New History:

"F2: wdg536 r 238 shift right (+) 2.5 channels".

The subtle change is the sign (+ or -) in parentheses signifying the direction of the shift (very early version may have had the words in upper case).

F3: Sequential Processor **Alias: sequ**

This routine is very useful for repetitive calculations on many records for multiplication, division, subtraction and special functions only. This routine effectively sets up the command sequences as a loop, thus you do not have to type in the command over and over when you are incrementing a record number.

For example, if you wanted to multiply file v records 10 to 23 by the constant 1.3452, you would type f3 from Math Operations. The program will ask for the operation. In this example, you would type * for multiplication. (Other operations would be /, -, f2, f6 or any other function number except 3 and 4). Next the program will ask for the first file ID, beginning record number, ending record number and the record number increment. For this example, you would type

v10 23 1

All positive record increments are valid. If the record increment was negative, it is set to zero. Next you will be asked to type in the second file ID, beginning and ending record numbers and increment. For this case, you would type

c1.3452.

You will then be asked to type in options. If you wanted errors included, the record increment should have been 2 (or more if the data sets take up more than 256 channels) since the data and errors take up at least 2 records. Note that you can get in an infinite loop if the record increment is set to zero. However, you can exit back to Math Operations by typing

x

at any point. If you type

e

at any point in the processing, that operation will be skipped and the program will go on to the next operation. The operations are performed just as if you had typed in the commands one at a time or all on one line separated by commas in Math Operations. This function should save a lot of time and typing.

This function calls function f4.

F4: Sequential Processor (No User Access) Alias: none

This function sets up the commands from function f3 for execution and thus has no user access.

The f4 function makes a command to do an operation followed by a comma and f4. Thus f4 calls itself. Internally, it stores the sequential command and counters in an internal array and increments them each time the program returns to the f4 function. For example, f4 may build a command sequence like

v23/w26e,f4

so that the first operation (v23/w26e) is done and then f4 is called again. The f4 routine then increments whatever the user had requested and generates a new command and returns to the command interpreter.

When all incrementing is complete, f4 returns to the user command input routine.

F5: Continuum Removal **Alias: cont[rem]**

The f5 continuum removal is an old algorithm that has been superseded in most cases by newer routines such as f12 (cubic spline) and f44 (segmented upper hull). However, it does have an advantage in that it removes a continuum of a specific shape given by the user in the form of a spectrum.

To enter the routine, type f5.

The directions are fairly simple. The routine fits a given continuum to a spectrum. The routine will prompt for the continuum and spectrum, as well as the continuum points used for the fit. Basically, you select two channels and a straight line is generated through these two channels in both the given continuum and the spectrum. The continuum is modified by linearly changing its slope so that the same two channels on the continuum match the values in the spectrum. Once the fit is complete, the spectrum is divided by the modified continuum.

First, you will be instructed to enter the two channels for which you want to match the continuum. Usually that will be on each side of an absorption band.

Enter two values to which continuum is to be matched followed by h, a, or n representing units of channel, wavelength, and energy (inverse wavelength, and the wavelength file id and record number. For example:

12 33 h V36

or

1.2 1.65 a V36

would be equivalent if channel 12 = wavelength 1.2 and channel 33 = wavelength 1.65. The wavelength set is then V36.

The two points can be entered in whatever units are convenient for you, in free format (spaces between numbers, no commas). The wavelength file is read, and the two values entered are correlated with the appropriate channel in the wavelength set. You can exit the routine here if you wish with an e or x command.

Next, enter the spectrum file id and record number, e to include errors followed by the continuum file id and record number. Type e or x to exit the routine.

The message on the CRT will say "WORKING" when you hit the return key after entering the information. The title of the files used will be printed on the CRT as they are accessed by the program. The error file with the spectrum is accessed if it was requested.

The continuum spectrum is scaled to the spectral data at the two given points by calculating a scaled slope and an intercept (or vertical offset) and converting each data point of the continuum to the scaled continuum through the relationship $Y = mx + b$ where Y is the scaled continuum, m is the slope, x is the original continuum value at a given

channel, and b is the intercept. The spectrum is then divided by the scaled continuum. The errors are scaled by a factor of (scaled spectrum original spectrum).

Finally, the title and history are made and control is transferred back to the main program to write the file.

Original author: Lucy Mcfadden

F6: Black Body Computation **Alias: plan[ck]**

This function computes the Planck black body function at a given temperature (in degrees Kelvin) and any wavelength set. Currently, the output is in watts per square meter per micron per steradian and the input wavelength set is assumed to be in microns.

Upon entering the routine, you are asked to type in the temperature and the wavelength id and record number. When using extreme temperatures or extreme wavelengths, you should check the results for underflow or overflow in the CRT plot routines (Chapter 9). No file or errors are needed since this routine generates a new file instead of an operation on an old one. The equation used is

(eqn 8.f6.1)

```
cpile { E sub i (T) ={ 8 pi h c sup 2 } over {  lambda  sup  5  }
left [ e sup {{ h c}  over { lambda k T }} - 1 right ] sup -1
        above
        ={  1.4966  times 10 sup -5 } over lambda sup -5 left [ e
sup {13586 over { lambda T } } -1 right ] sup -1 }
```

where E is the energy for each channel at wavelength,

λ (in microns),
T is the temperature (degrees Kelvin),
c is the speed of light,
h is the Planck constant (= 6.6252x10-34 Joule-seconds),

and k is the Boltzmann constant (= 1.3806x10-23 Joule/K).

F7: Smoothing Function **Alias: smoo[th]**

This routine smooths a spectrum by performing a weighted smoothing of adjacent channels. First, the channels are sorted into increasing wavelengths; note that the smoothing is done in channel space only. The user then chooses how many channels to include on each side of each data point. Each point is weighted by the inverse power of 2 with the power increasing with increasing distance from the data point to be smoothed. If the parameter giving the number of channels to include on each side of a given channel is n, then for n = 1 data point D at channel I is computed by

$$D_i = [\ ^{1/2}D_{i-1} + D_i + {}^{1/2}D_{i+1}]/2. \qquad \text{(eqn 8.f7.1)}$$

For n = 2

$$Di = [\ ^{1/4}D_{i-2} + {}^{1/2}D_{i-1} + D_i + {}^{1/2}D_{i+1} + {}^{1/4}D_{i+2}]/2.5. \qquad \text{(eqn 8.f7.2)}$$

In general

$$D_i = \frac{\left\{ \frac{D_{i-n}}{2^n} + \ldots + \frac{D_{i-1}}{2} + D_i + \frac{D_{i+1}}{2} + \ldots + \frac{D_{i+n}}{2^n} \right\}}{\left\{ 1 + \frac{2}{2_1} + \ldots + \frac{2}{2_n} \right\}} \qquad \text{(eqn 8.f7.3)}$$

and

$$\sigma_i = \frac{\left\{ \frac{\sigma^2_{i-n}}{2^n} + \ldots + \frac{\sigma^2_{i-1}}{2} + \sigma^2_i + \frac{\sigma^2_{i+1}}{2} + \ldots + \frac{\sigma^2_{i+n}}{2^n} \right\}^{1/2}}{\left\{ 1 + \frac{2}{2_1} + \ldots + \frac{2}{2_n} \right\}} \qquad \text{(eqn 8.f7.4)}$$

The parameter n can vary from 1 to the number of channels divided by 2. When the smoothing is finished, the data are sorted into the original order. The wavelength record default is the current wavelength set in use (and is printed on the CRT). If another wavelength set is desired, type the wavelength file id and record number on the same line as the parameter n (which must be first on the line). Also, if data only in a certain range is to be considered, on the same line, type an "l" command (for limit) and the range limits:

l n1 n2

where n1 and n2 are the minimum and maximum data numbers to be included. Default limits on l are -1.0x1034 to +1.0x1034.

For example, the following command (all on one line):

2 V15 l -50 30000

sets the n = 2, wavelength set V15, and set limits to be -50 to 30,000.

If certain channels should be deleted, type

d

to delete points. You will then be asked to type in the points to be deleted. When you are finished typing in points to be deleted, type

c

to continue.

A data set must be specified before the f7 command, and errors can be included. The parameter n seems to give excellent results when it is 1 or 2. When n gets larger, there seems little difference in the smoothed result compared to smoothed data with n=2.

F8: Channel-Record Transpose **Alias: tran[spose]**

This routine transposes the array of channel (column) versus spectrum (rows) to spectrum versus channel. For example, if you had 10 spectra of 5 channels each where each spectrum was taken at 1 hour intervals, this routine transposes the matrix so that there are 5 output "spectra" of 10 channels each where each "spectrum" is intensity at 10 different times.

Example: given 4 spectra of 3 channels each:

Spectrum Number	Channel number 1	2	3
1	1.23	1.34	1.45
2	2.23	2.34	2.45
3	3.23	3.34	3.45
4	4.23	4.34	4.45

would be transposed to:

Spectrum Number	Channel number 1	2	3	4
1	1.23	2.23	3.23	4.23
2	1.34	2.34	3.34	4.34
3	1.45	2.45	3.45	4.45

WARNING: When you first enter the transpose routine, it prints the wavelength set in use. The wavelength set controls the number of channels (and thus the potential number of output spectra), so it must be set to the right value.

The first input request is an option to delete channels. If you delete input channels, the output spectrum will have that channel deleted. You delete channels by typing "d" and the channels to be deleted (see plot routines for a more complete description of the delete points routine), and a "c" to continue to the next step. For example, say you had 20 spectra of 100 channels each, taken as a function of time and you wanted to only have a data set that was the intensity of channel 33. You would delete all but channel 33:

d 1t32 34t100 c

(you may include spaces anywhere except within a number).

Next, you will be requested to enter each (non-deleted) channel. Input spectra are entered one per line (file letter ID and record number). When all are typed in, type

b

to begin transpose.

The routine will next ask for the beginning output location (type in file letter ID and record number) and then a common title for all the output spectra. The transpose will then begin and the output spectra will be output sequentially beginning at the beginning point in the file that you selected.

The number of output spectra is equal to the number of channels in use minus the deleted channels.

F9: Band Removal (Reflection Method) **Alias: band[rem]**

This absorption band removal routine takes one half of an absorption band and reflects it about the band minimum then, divides a given reflection spectrum by the calculated band.

The routine will request:

"Type in the file id and file # to be processed, e to include errors, followed by l or r for left- or right-side of band to be reflected. type e or x to exit."

The "l" or "r" refers to left- or right-side of the band minimum when the spectrum is stored in increasing wavelength from left to right. It refers to the side of the band that you want to be reflected to the other side.

Next you will be instructed to:

"enter outer limits of band (2 values), estimated half-height point on side to be reflected followed by h, a, n (channel, wavelength, inverse wavelength), and wavelength file id and number. type e or x to exit."

The two outer limits are on either side of the band minimum. The first limit tells where to stop the reflection process, the second tells where to stop looking for the band minimum. The program types "WORKING" and is doing the following:

- correlating the given band limits to channels in the wavelength file, default is that the limits are given in terms of channels.
- searching for the band minimum by looking for a datum that is smaller than the three values on either side of it, between the limits of the 1/2 height and the second outer band limit.
- puts values of one side of band into the position symmetrically opposite to it relative to the band minimum.
- divides spectrum by reflected band.
- scales the errors by a factor of (the residual the original spectrum).
- calculates title and history.

WARNING: This is a primitive means of removing a band. It was used in the 1970's but does not take into account the nonlinear nature of reflection spectra. The results for reflectance spectroscopy are not quantitative.

Original author: Lucy Mcfadden

F10: Sorting Routine **Alias: sort**

The function arranges data into increasing wavelength order. Entry is by the usual method for special functions: from the math routines, enter the file id and record number of the data set or be sorted. Errors may be included by an "e" option on the math command line.

After entering the sort routine, the title of the data set to be sorted is printed along with the wavelength set in use. The user may change the wavelength set at this point by entering the (upper case) file id and record number of the wavelength set.

WARNING: Only the data file is sorted, but by specifying the wavelength file when calling f10, the wavelengths may be sorted also.

For example, say the wavelengths are in v3 and the data in v6. First sort the wavelength set. From math operations, type the

v3f10

or

v3sort

command. Once inside the sort routine, set the wavelength set, in this example to the:

V3

wavelength set.

Once the wavelengths are sorted, sort the data set v6:

v6sort

and once inside you should not have to set the wavelength set again because V3 should already be in use. Simply press return to sort the data.

F11: Lunar Thermal Removal **Alias: luna[rtherm]**

This routine removes the thermal component from the reflectance object spectrum using the equation:

$$R_o = \{ R_o' + \frac{R_o'(1-R_s)P_s}{R_sF} - \frac{P_o}{F} \} \{1 - \frac{P_o}{F} \}^{-1} \quad \text{(eqn 8.f11.1)}$$

Refer to R. Clark, 1979, Planetary Reflectance Measurements in the Region of Planetary Thermal Emission, Icarus 94-103 (equation 14) for more information.

Variable definitions:

R_o = Reflectance of object with thermal component removed.

$R_{o'}$ = Reflectance of object with thermal component.

R_s = Reflectance of standard with no thermal component.

P_o = Temperature for Planck function of object (in Degrees Kelvin).

P_s = Temperature for Planck function of standard (in Degrees Kelvin).

F = (solar flux/)/(distance from sun in A.U.)2

The program requires that the file ID and record number of R_o' and the e to include errors, if any, be specified when called from Math operations. The program will ask for:

R_s: The file ID and record number.

The albedo at the normalization wavelength for R_o and R_s ($0.0 < A <= 1.0$). If the data are already in albedo (reflectance), then this constant should = 1.0 (the spectra are multiplied by this constant).

The wavelength data set (File ID and record number).

P_o and P_s in degrees Kelvin ($10 < T < 105$).

Solar flux/ data set (file ID and record number).

Distance from sun in A.U. (>0.0)

After entering albedo at normalization, the user has an option of either continuing, exiting, or changing what has been input so far (for the current screen information). User also has a similar option after entering distance from the sun. In calculating the thermal removal, R_o and R_s spectra are multiplied by their respective albedos at the

normalization point. Errors are propagated by multiplying individual error values by the Ro albedo at the normalization point.

F12: Cubic Spline Interpolation **Alias: cspl[in]**
F12: Derivative

This routine fits a cubic spline to an input data set, and then computes a new data set based on a new wavelength set specified by the user. This allows you to smoothly interpolate one data set to another (not this is not a convolution). Errors, if specified, are also interpolated. The input data set and its associated wavelengths are used to derive a cubic spline describing the curve. The curve may then be recomputed at other wavelengths, or because the curve is described by a set of polynomials, the derivative of the curve may be computed.

The input wavelength set can be the same as the output wavelength set, then with selected parts of the spectrum deleted, a continuum can be computed. For example, in this mode, you would simply delete all the channels comprising a particular absorption band. The computed spectrum is then the input spectrum without the absorption band. See

Clark, R.N., Water Frost and Ice: The Near-Infrared Spectral Reflectance 0.65-2.5 μm, J. Geophys. Res., 3087-3096, 1981.

or

Clark, R.N. and T.L. Roush, Reflectance Spectroscopy: Quantitative Analysis Techniques for Remote Sensing Applications, J. Geophys. Res., 6329-6340, 1984.

for examples of continua and a discussion of the theoretical aspects of continua.

The program requires that the input data set and its error bars, if any, be specified when called from Math Operations. For example;

v23f12e

or

v23csplin e

Upon entering the routine, the title to the data to be splined is displayed and you are asked select normal spline or derivative mode:

<return> for cubic spline mode,
d for derivative mode,
e or x to exit.

Next, you will be asked to input two wavelength data sets, the first for the input data set, the second for the output interpolation. The wavelength sets should be entered with capital letter file IDs. Example:

V24 U46

or

V15 V15

where the first interpolates to a new wavelength set, and the second example interpolates to the same wavelength set, presumeably with the intent of deleting some channels to use the result as a continuum.

Finally, you are asked if you want to delete points. If you do, just type in the channels to be deleted, separated by spaces, or ranges separated by a "t", and then terminated by a "c". For example:

1 2 59t63 120 c

deletes channels 1, 2, 59, 60, 61, 62, 63, and 120. You can use more than one line, and the deletion routine is not finished until it gets the "c".

WARNING: Cubic splining can be tricky. For example, glitches in data could lead to erroneous results, so you should delete those channels. The cubic spline routine used requires each wavelength set to be sorted in increasing order, and the interpolated wavelength to be within the bounds of the other wavelength set. F12 handles this, minimizing errors that could occur. However, it seems common to have problems. If you get one of the following "F12 errors," note the number and other pertinent information (e.g. data sets in use) and submit a bug report if you can't figure out a cause. NOTE, HOWEVER that a usual cause is bad input, most likely due to two wavelengths that have the same value (causing an infinite slope), and an extremely high data value (causing a near infinite slope), or some channel or wavelength not what you think it is (like not in the proper overlap range). Check your data carefully before reporting bugs.

If you have problems, you should carefully check all your data by both plotting it and examining the values with the data editor, f14. For example, if you find deleted channels in the input wavelength data set, you should include those channels in the list of channels to be deleted in the spline routine (f12).

Error codes from the cubic spline routine:

33: Found value(s) in interpolated wavelength set less than input wavelength set.

34: Found value(s) in interpolated wavelength set greater than input wavelength set.

129: The dimension of spline coefficient matrix is less than one less the number of channels.

130: The number of channels is less than 2 (user should check wavelength data sets in use).

131: Wavelength files are not ordered. (Since the routine sorts the data, this should not happen. It may indicate fundamental problem with the input data).

The spline function is very useful for interpolating data to new wavelength values and also for generating continua. When interpolating to

a new wavelength set, the resolution of the spectra should be equal (use the smoothing function of F17 to degrade a high resolution spectrum to that of the low resolution spectrum). Spurious data points should be deleted. For generating a continuum to a spectrum, specify the same wavelength file for the input and output spectra, then delete data points which are not on the "continuum." This can be tricky depending on the data and what you are trying to do so talk to people who have used the routine first (e.g. Roger Clark or Bob Singer).

How the Cubic Spline Interpolating Function Works

The cubic spline interpolating function may be visualized as follows. Bend a flexible strip (like a plastic ruler) so that it passes through each of the data points in the spectrum to be interpolated

$$[f(x_1), f(x_2), \ldots f(x_n)].$$

The physics of the bent strip shows that the equation of the strip can be represented as a series of cubic polynomials with appropriate boundary conditions. A different cubic polynomial is calculated for each interval in the spectrum

$$[\text{i.e. } S_1(x) \text{ on the interval } (x_1, x_2),$$

$$S_2(x) \text{ on } (x_2, x_3) \ldots S_{\{n-1\}}(x) \text{ on } (x_{\{n-1\}}, x_n)$$

where S_n represents a cubic polynomial

$$A_n x^3 + B_n x^2 + C_n x + D_n = S_n(x)].$$

The boundary conditions are that the spline must go through each function value

$$[S_1(x_1) = f(x_1), S_1(x_2) = f(x_2) = S_2(x_2) = \text{etc.}],$$

and the first and second derivatives of the cubic polynomials are continuous

$$[S_{\{m-1\}}'(x_m) = S_m'(x_m), S_{\{m-1\}}''(x_m) = S_m''(x_m)$$

for m on the interval 2 to (n-1) (inclusive)]. Finally, the curvature is forced to zero at the endpoints

$$[S_1''(x_1) = 0, S_{\{n-1\}}''(x_n) = 0].$$

The spline used here is NOT a smoothing function. Therefore, data with large noise spikes present should be pre-smoothed before fitting with the interpolating spline (or the data points with anomalous values should be deleted). For a more complete discussion see Carnahan and J. O. Wilkes, "Digital Computing and Numerical Methods," John Wiley & Sons, New York, New York. p 307. (1973).

F13: Merge Two Spectra to One **Alias: merg[e]**

This function will combine two input data sets (of presumably different wavelengths) into a single output data set. Perhaps this routine should be named "splice" because you can only merge two at a time, although you can actually have multiple splice points, enough that it is indistinguishable from a merge concept. It will not automatically merge the two input data sets according to their respective wavelengths; rather, it will combine the two sets according to a channel sequence specified by the user. (Errors will be included if specified.)

The following files must exist prior to calling the function:

A) a first input data set (+ errors),

B) a second input data set (+ errors), and

C) a wavelength set with enough channels to accommodate the desired output data set (the value of the wavelengths is inconsequential; only the number of channels matters--4852 maximum at present)

The first input data set must be specified when f13 is called. The wavelength set must also be specified at call time unless the last operation or display in the program used a wavelength set with the same number of channels.

Example of use:

Given two input data sets of different wavelengths:

A) w127 (25 channels) + errors
B) v321 (120 channels) + errors

1) Decide how the two data sets should be combined. For this case, combine the two data sets so that w127 channels 1-24 are followed by v321 channels 4-57 and channels 63-115; all other channels will be omitted.

2) Create a wavelength set with enough channels to accommodate the output data set. For this case, set the wavelength set to C131.

You would then call f13 from math (the ending "e" includes error bars):

w127f13eC131

Then in f13, type in the second data set when prompted (data set "b"):

v321

and finally, you will be prompted to enter the channels to merge. On the merge command line, "a" refers to the first data set (w127 in this case) and "b" refers to the second data set. So we have for this example:

a1 24 b 4 57 b 63 115

IMPORTANT NOTE: The merge routine works on one data set at a time. You must merge the wavelength sets in the same manner. Be very careful to make sure the data channels line up with the proper wavelength channels. It is usually a good idea to overlay the merged data sets with the originals.

F14: Edits Spectral Data and Error Value **Alias: edit**

This function allows editing of a data set and, if e is specified at call time, its associated error set as well.

To call f14, specify the file and record number, f14, and, if errors are desired, e.

Example:

v317f14e or **v317f14**

F14 accepts the following formatted commands to perform particular tasks:

channel data/e error	change
d[c] channel [channel1 t channel2] c	delete
i channel	insert after
l channel1 channel2	list (crt)
pd	print data (lp)
c	continue (exit with write)
e	soft exit
x	hard exit

Note: both e and x terminate the editing session without writing the results. Use "c" to exit and write results.

To CHANGE a data value (and its error, or just its error), specify the channel number of the value to be changed and the new data value (and the new error value); should you wish to change only the error, specify the channel number, e, and the new error value. (If you wish to change only the data value but you specified e at call time, you must specify three parameters; the channel number, the new data value, and the old error value.) Incorrect or uninterpretable format will cause an error message to be printed on the terminal screen. (Parameters may be entered either as integers or as real numbers but not in exponential format.)

Examples:

Call setup	Command	Task enacted
v317f14	11 13.567	change channel 11 data value to 13.567
v317f14e	11 13.4 .2	change ch. 11 data to 13.4, error to .2
v317f14e	11 e .2	change ch. 11 error value to .2
v317f14e	11 13.567	ILLEGAL ENTRY (errors were specified at call time, but no error value was entered)

To DELETE data values, specify d (or "dc" to delete and compress); the routine will then accept only channel number or channel ranges until the parameter c (do not confuse the parameter c with the option c) is encountered, at which time the delete task is performed. Without option c, the data values of all specified channels will be set to -1.23e+34 (the

standard value for SPECPR deleted points) [and the associated errors will be set to 0.0]. However, if the dc command is specified, the specified points will first be deleted, and then the channel space will be compressed to totally eliminate the specified channels. (Note: Compression of the channel space does not take effect until all specified points have first been deleted.) Channel numbers for the points to be deleted may be specified either individually or as ranges; the character t specifies a range of channels, starting with (and including) the number preceding t, and ending with (and including) the number following t, the command d, [the option c,] individual channels, channel ranges, and the parameter c may all be strung together; spaces are required as delimiters only between two numbers.

Examples:

Command	Task enacted
d6 7 12t13 18t21c	Data values for channels 6, 7, 12, 13, 18, 19, 20, 21 are set to -1.23e34:[errors for same channels are set to 0.0]
dc1t3 7c	channels 1, 2, 3, 7 are deleted totally; old ch. 4 becomes new ch. 1; old ch. 5 becomes new 2; old 8 becomes new 4.
dc	routine requests "enter more deletions or type c to continue"

To INSERT data [and error] values after a specified channel, specify i and the channel number after which you wish to insert values. (If no channel is specified, the routine defaults to channel 0, in which case the insertions occur at the start of a spectrum.) The routine will then accept only data [and error] values, one data [and error] value per line, until either a blank line or two carriage returns in a row are encountered; either integers or real numbers (but not exponential format) are acceptable for data and errors.

Example:

Call setup	Command sequence	Task enacted
v317f14e	i	routine prepares to accept data & errors, starting at channel 1
	12 1.3	new channel 1: data = 12; error = 1.3; original ch. 1 becomes new channel 2
	15.6 .9	new channel 2: data = 15.6; error = .9; original ch.1 becomes new channel 3

i2	routine prepares to accept data & errors starting after current channel 2
16 2	new channel 3: data = 16; error = 2
17.9	ILLEGAL ENTRY--no error was specified; routine requests last line to be retyped
17.9 1.6	new channel 4: data = 17.9; error = 1.6; original ch.1 is now ch.5
<return>	insertion routine stops

To LIST data and error values on the terminal screen, specify l and either a channel number or else two channel limits. Data [and error] values will be listed in exponential format along with channel numbers. (Spaces are required as delimiters between limits.) All data values within the channel limits are listed first; error values follow.

Example:

l 1 60	channel number & data value for channels 1 through 60 are listed on the terminal [followed by channel number & error value for channels 1 through 60]
l 5	5 and data value for ch. 5 [and 5 and error value for ch. 5] are listed on the terminal

To PRINT the data [and error] values on the line printer, specify pd. The output format is the same as when pd is specified in the CRT plot routine.

To CONTINUE normally and write the edited file, specify the command c; the routine will exit to the write-to-file routine. Note: You cannot return to f14 and issue additional commands once CONTINUE is enacted.)

A SOFT or HARD EXIT will occur if at any time the commands e or x, respectively, are encountered; control will return to MATH OPERATIONS. (Exception: In CHANGE, e is not interpreted as an EXIT command if it follows a channel specification.)

IMPORTANT NOTE: This routine does NOT record a history of the edit commands performed; should you desire a record of your editing commands, you must create a manual history yourself. However, edited changes can be easily seen by comparing with the original data set which is specified in the history.

Minor notes:

All channel numbers should be positive integers; if a real number is entered by mistake, the routine will truncate it to integer form. All channel range limits must list the lower bound first and the upper bound last.

If at any time during this routine you find yourself lost or confused, try pressing <return> once or twice until the EDIT COMMAND FORMAT message appears. Alternatively, you may type e or x to exit f14.

You may insert spaces as delimiters to improve the clarity of commands; use of such spaces will not affect the operation of f14.

F15: Formats Gaussian Parameter File **Alias: gaus[sfmt]**

NOTE: At this version of specpr, GFIT is not implemented beyond specpr version 1. However, f15 is still included to format Gaussian parameter files from GFIT output done for those sites still running version 1 specpr and GFIT. A future implementation of specpr may include GFIT.

This routine takes GFIT data from the SPECPR files and formats the data in a neat, legible manner. If no file was input, an error message is written, and the routine soft exits. The routine also checks to make sure that the input file is of the the correct type. If not, the routine soft exits.

First, the routine writes out the Gaussian terms (if there are any), followed by the continuum terms (if there are any); these are followed by the integrated intensities which are equal to the height times the width of the Gaussian, the percent error of the Gaussians, the mean value and sigma.

The routine then checks the manual history to see if the fit was in inverse wavelength space or natural logarithm space. If the fit was in inverse wavelength space, then the data are converted to wavelength space for the center positions and widths. If the fit was done in log space, then the items printed are

(1) the relative band depth which is equal to 1 - EXP (Height),
(2) Band depth - Error = 1 - EXP (Height - Error) and
(3) Band depth + Error = 1 - EXP (Height + Error).

In all divisions, the denominator is checked for zero. If it is zero, it is reset to 1.0x10-36, and an error message is printed on user's terminal.

F16: Line Segment Data Generator **Alias: line[gen]**

This routine generates arrays of data by linearly interpolating user (x,y) pairs to a wavelength set. Given data values at specific wavelengths or channels typed in by the user, this routine computes line segments between the points the user has input.

Upon entering the routine, the user has the option of changing the current wavelength set in use by typing the file ID letter (upper case) and the record number, or assigning the channel numbers as "wavelength values" with the maximum number of channels given by the wavelength record. To do this, type h and the wavelength record number to assign channel numbers with the number of channels given by the wavelength record number. Examples:

V23	Assign wavelength data set in v23.
h482	Assign the array value = the wavelength data value. (channel 1 = 1.0, channel 2 = 2.0, ... channel 482 = 482.0).
C482	same as h482, but the wavelength routines assign the wavelength set before it is handed to the f16 routine.
<return>	Use existing wavelength set in use.

After this initialization, the routine repeatedly asks for the first two data values--the (x,y) coordinates for user point #1, then #2, etc., until you are through. Enter the data values separated by a space. At each command point, a number of control options can be entered:

e or x will cause the routine to exit.

rn return to step n, where n is the step number. This allows the return to any previous input step so it may be changed (and then you may return to the last step before you went to step n). Note that you cannot change an x value (wavelength) to a value greater than that in the next step--to do that, you must delete the following steps.

dn delete steps. Using this command, all data values after and including step n will be deleted; command control returns to the first step deleted.

ln is a list command which will list up to 25 entries previously entered with the step number. Here (n) is optional. If it is used, the routine lists from step number n to step n+24.

b for begin analysis. The routine calculates all data values in the array between each set of ordered pairs.

No input spectrum is required because you are creating data, not operating on data. If an input file was called, an error message is printed, and the routine will hard exit.

All operations are checked for validity. If a given operation produces an error, a message will be printed stating what caused the error and whether or not the routine can continue. If it can, the routine asks again for the data. If not, the routine will exit, hard or soft, depending on the severity of the condition. Note that all x data values (e.g. wavelengths) must be in increasing order, or error messages will result.

F17: High To Low Resolution Convolution **Alias: conv[olv]**

Routine f17 convolves a high resolution data set to a low resolution data set. It has 2 modes: 1) use Gaussian profiles whose widths and centers are specified by the user, or 2) use user-specified filter profiles. The routine outputs two data sets: (1) the resultant wavelength values (center values) and (2) the convolved spectrum. The output of the center values is optional. A typical use would be to use mode 1 (Gaussian profiles) for convolving one spectrometer to another, and mode 2 for convolving a spectrum to a specific filter response, such as a LANDSAT TM system.

To run the function, go to the math routines and enter the file ID and number for the input spectrum and f17. Example:

v23f17

or

v23conv

The routine starts with a query for the use of the default wavelength assignment, which is displayed at the top of the screen. To change the default assignment, enter the wavelength data set file ID (upper case) and the record number. If you don't want to change the wavelength set, just type return.

Next you will be asked whether you want to normalize the data. This is a convolution normalization. Normally you want to do this. If you normalize,the resulting spectrum is divided by a convolved spectrum whose input data values are all one. If you don't normalize, the convolution output will vary as a function of the area under the curve of each response function. Unnormalized convolutions can be used for tracking voltages from a sensor. If you simply want to compare the spectral response, calibrated to, say, radiance or reflectance, and your input data are calibrated in those units, you want to normalize the convolution.

Next you are asked if you want to use the Gaussian mode or the filter bandpass mode. Type:

g	Gaussian mode.
<return>	bandpass filter mode.

Gaussian Profile Convolution

You are first asked for a data set describing the center values of the Gaussians for the output data set (these are the wavelengths you are convolving to). Enter the file ID and record number of the data set.

Next you are prompted to enter the Gaussian bandwidths. These are the Gaussian Full Width at Half Maxima (FWHM) for each Gaussian center value entered in the previous step. Enter the file ID and the record number of this data set. The Gaussian bandwidth data set and the Gaussian center values must have the same number of channels.

Next you are asked to input the bandwidths of the input data set. The spectral resolution of the input data set must be known to properly convolve to the output data set. Enter the file ID and record number of the input spectrum resolution. This data set must have the same number of channels as the input wavelength data set in use.

The convolution routine computes the convolution and then asks if you want to output the center wavelengths. See below.

Bandpass Filter Convolution

If you selected the bandpass filter convolution mode, the routine prompts you for the first filter profile and asks how many filters will be convolved. Each filter profile produces one output channel. The filter profile must be sampled at the wavelengths of the input data set (if it is not, you could reinterpolate them using a routine like the cubic spline, f12). For example, say you were convolving a laboratory spectrum of kaolinite to LANDSAT TM filters (7 bands). Say the filter response function sampled at the wavelengths of the laboratory spectrum was at w46 (the other 6 filters must be sequentially located in the same data file). You would then type:

w46 7

The input filter response functions are read in and the output spectrum is calculated. Next the user is asked if the output of center wavelengths is desired.

Center Wavelengths

After the convolution is complete (either Gaussian or bandpass mode) you are queried on whether or not to output the center value wavelengths. If you wish to write the center values, type in the file id and record number of the file where you wish to write the center values, or type return to ignore writing the center values. Next enter the title for the center values when prompted. Because this is an output data set (like any other), use a lower case file id (not upper case).

Note that f17 cannot include errors in its calculations. However, you can do error propagation by simply convolving the error data set in the same manner as the regular data set.

The equation used to calculate the spline is

$$R_j = \int_{x_0}^{x_1} S\, B_j\, dx, \qquad \text{(eqn 8.f17.1)}$$

where R_j is a data point in the resultant spectrum, x_o is the first value of the input spectrum, x_l is the last value of the input spectrum, S is the input spectrum, and B_j is the spectral bandpasses.

The center values are calculated by the following:

$$C_j = \int_{x_0}^{x_1} x\ S\ B_j\ dx, \qquad \text{(eqn 8.f17.2)}$$

where C_j is the center value for that particular spectral bandpass, x_l, S, B_j and x_o are the same as in equation 8.f17.2, and x is the wavelength.

The integrals are approximated by using numerical integration by summing the rectangular segments. Each rectangle is computed by using the channel separation of the input spectrum:

$$R_j = \sum_{i=1}^{n} S_i\ B_{j,i}\ x_i, \qquad \text{(eqn 8.f17.3)}$$

where n is the number of channels and x_i is the spectral bandpass of channel i

$$\delta x_i = \begin{cases} x_2 - x_1, & i = 1 \\ x_{i+1} - x_{i-1}, & 1 < i < n \\ x_{n-1} - x_n, & i = n \end{cases} \qquad \text{(eqn 8.f17.4)}$$

Similarly, the center values (wavelength) of each convolved channel are computed from the equation

$$C_j = \frac{\sum_{i=1}^{n} x_i\ S_i\ B_{j,i}\ \delta x_i}{R_j} \qquad \text{(eqn 8.f17.5)}$$

During the calculations, divide operations are checked for a zero denominator. If this happens, an error message is printed for the user, and the denominator is reset to 1.0x10-36 instead of crashing the program.

If no file has been input to the program at the start, an error message will be printed, and the routine exits.

Note that if the number of bandpass files times five is greater than the number of channels, a warning message will be printed explaining that, with data arranged in such a manner, the resulting accuracy of the data is questionable. This is due to using the summations (equations 3 and 5) and the equation 4 to approximate the integrals. The message is only a warning—you must determine if the bandpasses are complex enough that accuracy may actually be in error due to a small sample.

IMPORTANT CONSIDERATION: if the filter bandpass or the Gaussian profile is close to the input spectral resolution, you may want to interpolate the input spectra to finer channel spacing. Because the numerical integration rectangles are controlled by the channel spacing, interpolating to finer spacing will increase the accuracy of the numerical

integration. You must interpolate both the input spectra and the resolution "spectrum" to the finer wavelength set. Remember, for example, if you had 0.001 m spacing and 0.001 m resolution, and you interpolated to 0.0001 m spacing, the resolution is still only 0.001 m, thus the spectrum is oversampled. This oversampling is only useful in terms of making the numerical integration more accurate. If your convolved spectra show a "staircase" effect, it is an indication of the limitation of the numerical integration and you should resample the input data to finer spacing.

F18: Block Averages and Statistics **Alias: bloc[kav]**

Special function f18 averages data blocks into single data channels and computes the statistics (standard deviation of the mean) for each block. The size of the block is defined by the user. The blocks can be a number of channels, or a range from the wavelength set. The blocks can be overlapping, as in Nyquist sampling, or contiguous (half Nyquist sampling).

To start f18, you must supply a data set from math operations. For example, from math, type:

v23f18

or

v23block

As in most specpr routines, when you first enter f18, the title of the current data set being operated on is displayed along with the wavelength set in use and the number of channels. You may change the wavelength set in use at this point by typing in the new wavelength set file ID(upper case letter)and the record number. If you do not wish to change the wavelength set, press return.

Next you will be asked if you want the blocks defined as channels or wavelengths. Type:

c to define block in units of channels, or
w to define blocks in units of the wavelength set.

Now you are asked to enter the block size. If in channel mode, enter the number of channels to be averaged in each block. If in wavelength mode, enter the wavelength range in the same units as your wavelength set. For example, if you have a wavelength set going from 0.3 to 2.5 m and you want blocks of 0.1 m, enter 0.1 (be sure you are in wavelength mode). The block size must be less than or equal to the number of channels in channel mode, or less than or equal to the upper end of the wavelength range.

The block always starts at 0 and in includes the upper end of the block. For example, in channel mode a block size of 5 will include channels 1, 2, 3, 4, and 5. In wavelength mode, a block size of 0.1 will include all channels whose wavelengths fall from 0.00 to 0.10.

Next you are asked for half or full Nyquist sampling. Enter

h for half Nyquist sampling, or
f for full Nyquist sampling.

Say the block size is 6 and you select full Nyquist sampling. The input to each block would be as follows:

Input channels	Output channel
(1+2+3+4+5+6)/6	1

(4+5+6+7+8+9)/6	2
(7+8+9+10+11+12)/6	3
(10+11+12+13+14+15)	4
(13+14+15+16+17+18)/6	5

However, for half Nyquist sampling, the input would be:

Input channels	Output channel
(1+2+3+4+5+6)/6	1
(7+8+9+10+11+12)/6	2
(13+14+15+16+17+18)/6	3

The average of a block j is computed from the equation

$$x_j = \frac{\sum_{i=n_1}^{n_2} x_i}{(n_2 - n_1 + 1)}$$

where $(n_1 - n_2 + 1) = N$ if n_2 is less than or equal to the number of channels.

If errors have been included, the standard deviation is calculated by the following

$$\bar{\sigma}_j = \frac{\{\sum_{i=n_1}^{n_2} \sigma_i^2\}^{1/2}}{(n_2 - n_1 + 1)}$$

where σ_j is the new standard deviation. If the errors have not been read in, the following equation is used:

$$\bar{\sigma}_j = \frac{\{S_{i=n_1}^{n_2} \frac{(x_j - \bar{x}_j)^2}{(n_2 - n_1 + 1)}\}}{(n_2 - n_1 + 1)^{1/2}},$$

which computes the standard deviation of the mean.

F19: Polynomial Fit (10 Term) **Alias: poly[fit]**

Special Function 19 does a least squares fit to a data set with a polynomial of the form:

$$y = a_1 + a_2 x + a_3 x^3 + a_4 x^2 + \ldots + a_n x^{n-1}$$

The program requests the number of terms to use for the polynomial fit and will currently handle up to ten terms. To run this routine type the file id and record number of the spectrum you wish to fit followed by f19, or use the alias:

v23f19

or

v13poly

The program will allow you to change the wavelength set if the current set is not the proper one. The program then calculates the least squares curve and prints the polynomial coefficients and the reduced chi square for the fit.

You have the option of saving the fitted curve, computed with the wavelength set in use, or saving the coefficients. The coefficients can be used as input to f21 (calculate a polynomial).

F20: Text File Input and Output **Alias: read**

Special Function 20 has two parts to it. The first part reads data from a text file (for example, one that was generated by any program or by using the system editor). The second part writes data from a specpr file into a text file.

With both parts there are options to include errors or wavelengths or both. Data, errors, and wavelengths may be written either in standard Fortran format 'f15.6' or in exponential '1pe15.6' format, depending on the user's option. The program works with one set of numbers per line i.e. wavelength, data and errors. Without the wavelength option, the order is data, errors, and without the error option the order is wavelength, data.

When writing the data to a specpr record (i.e v123), the program will put the error values in the next consecutive specpr file (i.e v124).

Using Function 20

To run function 20, type

f20

from math operations. The routine will prompt you for what part (reading or writing) you want to do. Type

r for reading data,
re for reading data and errors,
w for writing
we for writing data and errors.

If you are writing to a text file, you are given an option for format. Type

f for f format (Fortran format F15.6), or
e for exponential format (Fortran 1pe15.6).

Next if you are writing, you are asked if wavelengths are to be included. Type

y for yes to include wavelengths, and
<return> or any other character for no.

Then you are instructed to enter the filename of the text to which you will read or write data. Currently, the file name must be 40 characters or less.

Be sure that when you are reading, the file is in your directory, or you must give the full file name (i.e /u/smith/dataf).

When reading, f20 will prompt you for the starting column number where the data begins (e.g. col. 1 - 80). Default is column one(1). Columns are character columns, not columns of numbers. The routine does

not count tabs properly, and if a tab or control character is encountered, it is treated as an error and the read is terminated.

After reading the data, you can exit normally to plot and write the data, or elect to read the data again (type r to redo).

If you are writing data to a text file you will be asked the file id and record number of the specpr file that the data are from (v, w, d, u, or y). When the operation is finished you have the option to redo the operation or exit. Type r to redo the operation at this point or e or x to exit. You can thus write many text files without leaving f20.

Setting Up the Text File

There are two ways of setting up a UNIX text file that can be read by f20.

First, and the easiest, is directed output. Output the data to the terminal and direct it to a file. In your program write to the terminal:

```
      write(6,10)data
10    format(1x,1pe15.6)
```

or with errors

```
      write(6,10)data,error
10    format(1x,1pe15.6,1pe15.6)
```

and run the program using this method

```
progname > filename
```

where progname is the program executable file and filename is the file to be read by f20.

Second, you can write your data directly into a file. You must open the file and write to it. See your local Fortran manual on how to write ascii files.

When reading, enter f20 from math operations and when prompted for your input/output filename, type in the filename of that you assigned to the text file in the above examples. The program will print out the channel number and the data value of that channel. The last channel number printed is the number of channels in the file (or the number of data values read by f20).

An f20 Example (square the data and add 1.0)

1) Operate on data using your own programs:

2) Put these values in v34 (with, for example the editor f14):

channel 1: 0.99

```
                2: 0.90
                3: 0.70
                4: 0.40
```

3) Write v34 to file 'testa' using f20 and e format

4) Run program 'testprog <testa >testb'

5) Read testb into specpr using f20

6) List the 4 channels

Fortran program "testprog":

```
                        program testprog
                        real x,y
                        do 100 i = 1, 4852
                            read (5,1) x
                    1       format (1pe15.6)
                    c do a computation:
                            y = x*x +1.0
                            write (6,1) y
                    100     continue
                        end
```

F21: Calculate Nth Term Polynomials **Alias: calc[poly]**

This routine calculates nth term polynomials where the maximum number of terms or coefficients is 256. This routine operates on the data set entered from math operations. Then it prompts you for a data set of coefficients (file ID and record number) for the polynomial:

$$Y(n) = C(1) + C(2)X(n) + C(3)X(n)^2 + C(4)X(n)^3 + ...,$$

where Y is the output array, X is the input array, C is the coefficient array.

To execute the f21 function, in math operations type the file id and record number of the input file and f21. For example:

v23f21

or

v23calc

After the routine title is printed type in the file id and record number of the file containing the coefficient values, then a space, then the number of terms of the polynomial. For example, if the coefficients were in v24 and there were 6 coefficients, you would type:

v24 6

The program will then do the calculations and return to the math routine prompting you for the location to put the output file and the title etc.

WARNING: be careful of overflows! When the number of terms, n, becomes large (like near or greater than 10), x^n can get very large if $x > 1.0$.

F22: Graph Digitization using a Tablet **Alias: digi[tize]**

This routine is a graphics digitization program. **Note: this routine is device-specific and must be tailored specifically for your site; see your system administrator for details)**.

This program asks the user to define a graph by digitizing the graph limits. Then it accepts points from the tablet that are within the limits of the graph and converts the x and y values into data values as defined by the graph. If a digitized point is outside the bounds of the graph, the program will say so and ask if you want to redo the point, begin over, write the data or exit.

When you are finished digitizing, the program will prompt you for the file id (v, w, d, u, or y). The file must have positive protection set. The program will first write the x data values and then the y values in the following record. The protection is updated as appropriate.

To run f22, type

f22

from math operations. The program will return with the title page of f22.

Press return to continue.

Digitize the upper left bound of the graph, lower left bound (origin), and the lower right bound of the graph. The program will return with the values of the points in tablet coordinates, the amount that the points are off perpendicular, and the angle between the two axes. If these values are acceptable type "yes".

The program will rotate the axis should it be not exactly perpendicular. Then you must enter the values of the digitized points starting with the upper left, origin, lower right points. Then you must enter the limits of the horizontal axis and the limits of the vertical axis of your graph.

When the above is finished the program will return with "ENTER POINTS FROM TABLET". Now you can digitize points from your graph.

If you somehow enter a point that is outside the limits of your graph, the program will return with "POINT OUT OF BOUNDS". You have various options at this point. Type

r to reenter the point.
This will not affect the data count.

b to begin entering the points again.
This allows you to start over without having to reenter the limits etc.

w when finished entering points. This will put you in the writing mode to

write your data to a file.

x will exit you from the program.

If you are finished entering points from the tablet, and wish to write the data, digitize a point outside the graph. The program will return with "POINT OUT OF BOUNDS". Type a

w

to write the data.

To write the data, the program will prompt you to enter the file id of the file and record number where you want the data to be written. Then you are prompted to type in the title of the data set that will contain the x (horizontal) values, then after that the title of the data set that contains the y (vertical) values. The data are then written.

After all that you will be asked if you want to digitize another graph. Answer yes or no, if no then the program exits.

F23: Mathematical Parser **Alias: pars[e]**

This routine allows the user to type in arbitrary mathematical expressions involving data sets and constants. The expressions allowed are any combination of addition, subtraction, multiplication, division, and exponentiation (exponentiation is represented by ^).

WARNING: f23 is site specific because of the interdependency between Fortran and C so it may not be linked at all sites. Check with your system administrator.

To use f23 type from math operations:

f23

or

f23e

if errors are to be included in the computation. The program will list the wavelength data set and the number of channels it will be using. To change the wavelength record at this point type the file ID (upper case) and the record number of the new wavelength set. If the default wavelength set is acceptable just type a carriage return. If you wish to abort the routine type

e

or

x

After choosing the appropriate wavelength set the routine will print out the current file protections and allow you to type in the expression to be evaluated. For example, if you wanted to add data set v21 to a Gaussian with a height of -1.5, half width of 2.3 microns (at 1/e point), centered at 1.32 microns you would type:

```
v21+-1.5*2.71828e0^(-((v3-1.32)/2.3)^2)
```

assuming you are using wavelength set v3 (note that the wavelengths are treated as data here, so the wavelength set uses a lower case file ID).

F24: Star Moon Thermal Removal **Alias: star[moontherm]**

This routine removes the thermal component from the reflectance of a lunar spectrum using a standard star as a reference. This routine was written by the University of Hawaii, Planetary Geosciences Division as a modification to special function 11 (Lunar Thermal Removal, page 8-f11.1). See that manual page for related operation. The specific manual page for this function will be written by people at the UH.

F25: Two Component Areal Mix Least Squares Alias: twoc[omlsq]

This special function computes the best areal mixture of two "known" spectra as compared to an observed spectrum. The user provides his or her own guess as to what the end-member spectra are and the routine computes the least squares fit of the two.

WARNING: This routine was a developmental version at the University of Hawaii and has never been certified as to its method. It currently works for spectra without error bars and with channels less than or equal to 256. If you do use this routine, you must verify for yourself the accuracy of the method. A full manual entry is waiting on proper certification of this routine.

F39: Random Number Noise Generator **Alias: noise**

This routine uses the system random number generator to generate noise with a normal distribution and a standard deviation of 1.0. The user can then multiply the resulting numbers by a constant to generate noise with a different standard deviation, and the noise may be added to a spectrum.

The routine generates a random number for each channel in use. The user then gives a scale factor to scale the noise to the desired interval. For example if you wanted noise of 3% (standard deviation of 0.03), you would scale the noise by 0.03.

F40: Least Squares Between Two Spectra **Alias: leastsq**

This function computes the least squares between two spectra. You enter two data sets and the routine computes the sum of the squared differences between the two spectra.

It is currently under development and should not be used. See your site specpr manager if you need this routine.

F41: Binning Routine **Alias: bin**

This routine bins data into various bins that are selected by the user. It is not complete and can't yet be run by users. See your site specpr manager for current updates.

F42: Fit Band Profile from a Reference Spectrum Alias: bandmap

This routine fits an absorption band from a reference spectrum to an observed spectrum by minimizing the least squares. The reference absorption band depth is changed by a simple equation (thus the reference spectrum chosen should have an absorption band profile similar in saturation level to the observed spectrum in order to provide the best fit).

This routine is described in detail in the paper

Clark, R.N., A.J. Gallagher, and G.A. Swayze: Material Absorption Band Depth Mapping of Imaging Spectrometer Data Using a Complete Band Shape Least-Squares Fit with Library Reference Spectra, Fifth Airborne Imaging Spectrometer Workshop Proceedings, JPL Publication 90-54, 176-186, 1990.

To use f42, the user types the observed spectrum and f42 from the math command line. For example, if v76 is the observed spectrum, you would type:

v76f42

Next you will be prompted to enter the reference library spectrum file ID and record number.

Then you are asked to enter the left and right continuum values. The continuum is located on each side of the absorption band and consists of one or more data channels. You enter the beginning and ending channels for each side. For example, if the channels describing the left side of the continuum included channels 331, 332, and 333 and if the right side continuum included channels 385, 386, 387, and 388, you would enter the following.

331 333 385 388

If the continuum included only one channel, you still must enter 4 numbers, so the beginning and ending channel would be the same. For example if the continuum was only channel 331 on the left and 339 and 340 on the right, you would type:

331 331 339 340

Next you are asked what data to output to the file. You can save the fitted reference spectrum (press return, this is the default), or the continuum-removed observed spectrum by typing "o" (you don't type the quotes).

The Band depth Fitting Algorithm

The absorption band depth, D, is defined relative to its continuum:

$$D = 1 - R_b/R_c \qquad \text{(eqn 8.f42.1)}$$

Where R_c is the reflectance of the continuum at the band center, and R_b is the reflectance at the band center. The definition originates from Clark and Roush, (JGR, 1984). The process of absorption band analysis is to first remove a continuum (see Clark and Roush, 1984 for details about continuum analysis). Once the continuum is removed, a band can be characterized by determining how well the feature matches a reference library spectrum.

A simple model to change a continuum-removed absorption feature's contrast is to add a constant to the data at all wavelengths. In this case, the feature will not be represented properly if the band saturation changed significantly, due to say a major change in grain size between the reference and observed spectra. The algorithm presented here uses the simple case of an additive component because it is computationally fast compared to a full radiative transfer model. The method can easily be adapted to the full model however, or even to a method that examines the band depth and chooses an appropriately saturated absorption feature from a library of materials at several grain sizes.

The algorithm first removes a straight line continuum from the library reference spectrum using channels on each side of the absorption feature that is to be mapped. The continuum is removed from the observed spectrum using the same method and spectral channels. The user may select several channels on each side of the band so that noise in the continuum is averaged. The continuum is removed by division:

$$L_c(\lambda) = \frac{L(\lambda)}{C_l(\lambda)} \quad \text{and} \quad O_c(\lambda) = \frac{O(\lambda)}{C_o(\lambda)}, \qquad \text{(eqn 8.f42.2 a and b)}$$

where $L(\lambda)$ is the library spectrum as a function of wavelength, λ, O is the observed spectrum, C_l is the continuum for the library spectrum, C_o is the continuum for the observed spectrum, L_c is the continuum removed library spectrum, and O_c is the continuum removed observed spectrum.

The contrast in the reference library spectrum absorption feature is modified by a simple additive constant, k:

$$L_c' = \frac{L_c + k}{1.0 + k}, \qquad \text{(eqn 8.f42.3)}$$

where L_c' is the modified continuum-removed spectrum that best matches the observed spectrum. This equation can be rewritten in the form:

$$L_c' = a + bL_c, \qquad \text{(eqn 8.f42.4)}$$

where

$$a = k/(1.0 + k), \text{ and}$$

$$b = 1.0/(1.0 + k). \qquad \text{(eqn 8.f42.5)}$$

In Equation 8.f42.4 we want to find the a and b that gives a best fit to the observed spectrum Oc. The solution is done using standard least squares:

$$a = (\sum O_c - b\sum L_c)/n,$$

$$b = \frac{\sum O_c L_c - \sum O_c \sum L_c/n}{\sum L_c^2 - (\sum L_c)^2/n},$$

and

$$k = (1-b)/b, \qquad \text{(eqn 8.f42.6)}$$

where n is the number of spectral channels in the fit.

The algorithm produces a band depth, indicating its spectral abundance in the observed image, a goodness of fit (correlation coefficient) which gives a measure of confidence in the accuracy of the resulting fit, and the continuum value at the band center in the observed spectrum. The correlation coefficient is computed by

$$r = \mathrm{sqrt}(b_1 b'_1) \qquad \text{(eqn 8.f42.7)}$$

where b_1 is from Bevington (Data Reduction and Error Analysis for the Physical Sciences, Mcgraw Hill, New York, 1969), his equation 7.2 (which he calls b) on page 120, and b'_1 is from Bevington (1969), his equation 7.3 (which he calls b') on page 121.

F43: Fast Fourier Transform (FFT) and Inverse FFT Alias: fft

This routine performs a Fast Fourier Transform (FFT) on a spectrum. The spectrum can be any number of channels; the algorithm pads the spectrum with zeros to a power or 2 number of channels in order to do the FFT. For example, if you have 931 channels, the algorithm pads the data to 1024 channels. The resulting FFT is NOT folded: zero frequency occurs at channel 1 and the maximum channel, and the maximum frequency occurs in the middle channel.

The f43 routine does not save both the real and imaginary components at this time. Thus the user must run f42 twice: once to save the real component and the second time to save the imaginary component.

The inverse FFT requires both real and imaginary components. If you have only the real, you could use a data set of zeros for the imaginary components (but this is not strictly valid).

F44: Segmented Upper Hull Continuum **Alias: suh**

This program derives a continuum by draping line segments over the top of the spectrum. The user can remove the spectrum and then drape line segments into the remaining absorption features. The iteration level can continue until the continuum matches the entire spectrum. The method is described in detail in:

Clark, R.N., T.V.V. King, and N.S. Gorelick, Automatic Continuum Analysis of Reflectance Spectra: Proceedings of the Third Airborne Imaging Spectrometer Data Analysis Workshop, JPL Publication 87-30, 138-142, 1987.

F45: Automatic Band Analysis **Alias: aba**

This routine uses the segmented upper hull (see f44) to isolate absorption features in a spectrum and then for each feature the following are derived:

Band Center
Full Width at Half Maximum (FWHM)
Band Depth
Error in band depth (assuming that error bars were included with the input spectrum)
Full Width at Quarter Maximum (FWQM)
Band Asymmetry
Continuum reflectance level
Segmented Upper Hull iteration number

The output is a data set using 9 channels to describe each feature (the 8 items above and one additional that is currently unused). Thus if the input spectrum contained 26 features, then the output data set would have 234 channels.

This routine was used to generate the features in the USGS Digital Spectral Library.

For a description of the continuum algorithm, see Clark, R.N., T.V.V. King, and N.S. Gorelick, Automatic Continuum Analysis of Reflectance Spectra: Proceedings of the Third Airborne Imaging Spectrometer Data Analysis Workshop, JPL Publication 87-30, 138-142, 1987.

Once the continuum is removed, the band center is found by finding the lowest channel in a feature and fitting a parabola to the lowest channel and the channel in each side of the lowest one. The FWHM, and FWQM are found by linear interpolation. The error bar is found based on the error bars on the data. The asymmetry is found by ratioing the distance from the band center to each FWHM point. See also section 9.8, Interactive Band Analysis (it uses the same algorithms).

F46: Band Analysis Output **Alias: abaout**

This routine reads specpr data records produced from the Automated Band Analysis routine, f45, and outputs the data to a binary file in order to generate a spectral features data base. The independent program "spfeatures" reads the data base created by this routine.

In addition to the features, this routine adds the title and spectral identification (specpr file name and record number) to the features binary data base.

F47: Spectrum Recreation from F46 output **Alias: abaspecrec**

This routine reads specpr f46 spectral features output and attempts to use the feature parameters to compute the original spectrum. The features are triangular in shape at the bottom because straight line segments are used from the band center to the Full width at Quarter Maximum and the Full width at Half Maximum and to the continuum level. The routine is useful to see how well the f45 automated band analysis routine works.

In the initial evaluation of f45, this routine proved useful. The key in evaluating the output is the question "is the spectrum identifiable?" For example, do the features for the spectrum of kaolinite get reassembled into a spectrum that is recognizable as kaolinite? The authors tested many spectra and concluded "yes."

F48: HP Graphics Tablet Digitization **Alias: hpdigit**

This routine digitizes spectra using an HP tablet that is an optional part to HP 2623A graphics terminals. The setup and operation are identical to the digitization routine, f22. See f22 for details.

F49: Linear Interpolation **Alias: lininterp**

This routine works like the cubic spline, f12, but the interpolation is done linearly instead of with a spline. Use the f12 manual for operation of f49.

F50: Wavelength Registration **Alias: wavreg**

This function allows for calibration of the wavelengths of a spectrum against a reference spectrum. It was originally designed to correct the wavelengths of the Mariner 6 and 7 infrared spectrometer data sets, using a spectrum of a carbon dioxide atmosphere. Four files are required to run the registration.

1) The original data which requires calibration in wavelength.

2) An approximate wavelength file for (1)

3) Data for a reference spectrum, which contains absorptions at known wavelengths.

4) The wavelength file for the reference data (3).

The output of the function is a new wavelength set for the original data. (1) and (2) should have the same number of channels as should (3) and (4) however the original data and the reference do not need to have the same number of channels.

To begin F50,from math operations, input the file ID and file number for the original data and f50. Then enter the wavelength file for this data. If this wavelength file is the default wavelength file no input is required. You will be prompted to enter the wavelength file and data file for the reference spectrum. The function will then overlay a plot of the original and reference spectra The reference spectrum will be in the dotted line. If the reference spectrum needs to be scaled vertically, that can be done at this point.

It is useful to determine the number of "plot windows" or regions required to adequately cover all the absorption features which will be used for calibration. Both the reference and original absorption feature must appear in the plot window. To determine the number of plot windows use the standard crt plot scale changing commands (see section 9.3). Enter the number of plot windows and change scale to the plot window at the smallest wavelength.

Within a plot window determine the number of absorption features that will be used for wavelength calibration. The program currently allows for 20 features total to be used. Then, using the graphics cursor, position the cursor on the absorption in the reference spectrum. The program uses only the horizontal position so the vertical position of the cursor is irrelevant. Then position the cursor on the corresponding absorption feature in the original spectrum. Repeat this process for each feature in the plot window and in subsequent plot windows. Be sure to always move from shorter to longer wavelengths.

The accuracy of the wavelengths is limited by the graphics resolution, so be sure your plot windows magnify each set of features adequately for the wavelength precision you desire.

The program uses the wavelength value obtained from the reference spectrum and the channel number from the original spectrum. In this way, the channel of a particular absorption feature in the original data is associated with a new (correct) wavelength. A linear interpolation in wavelength between each absorption feature is then constructed and used to determine the wavelengths for all channels of the original data. If the first and last absorption feature are not the first and last channel then the interpolation derived for the first and second (or last and second-to-last) absorption features is extrapolated to the end channels.

It should be noted that an "absorption feature" can be a peak or a spike or any feature of the original spectrum that can be associated with a definitive wavelength. Also these features need not necessarily be present in the reference spectrum, if the wavelength value is accurately known from other means. (In this case, just pick the wavelength value off the x-axis, rather than from the reference spectrum).

Finally, the program will re-display the reference spectrum and the original spectrum, now using the newly derived wavelengths for the original. If the calibration was reasonably successful this new wavelength file can be saved. If not, you can begin again with more, or different calibration points. In practice, this seems to work best if you exit the function and then start completely over again.

This routine and manual page was written by Wendy Calvin.

CHAPTER 9

CRT PLOT ROUTINES

9.1 Introduction

The CRT plot routines are a set of routines for data display, overlay, header information display and change, data print-out, and specific graphics-oriented routines to read data values off of the plot and to analyze specific absorption features. The CRT plot routines are called from the completion of a Math Operation (Chapter 8) or from the File Display, Transfer and Overlay routines (Chapter 10).

Any x-y paired data may be plotted. The horizontal axis may be labeled to properly describe the data; the default horizontal-axis label is "Wavelength (microns)", but the default can be overridden. Data may also be plotted as a function of channel number, or the x-axis values inverted in which case the default units of microns is assumed and the program converts the units to wavenumber in inverse centimeters.

The CRT plot routines also have a facility for removing spikes or glitches in data by assuming the glitch was caused by a bad bit in the data. The user may also elect to change any channel to a deleted point.

9.2 Plotting Mode

The plot mode is selected by the command letters

h for channel,
a for wavelength, or
n for energy (inverse wavelength).

When you have a plot on the screen, change the plot type by entering h, a, or n and return. The menu for these commands is always located at the bottom of the plot screen. Note the characters chosen are the second letter of channel, wavelength, and energy since the first letters are used for other commands. The energy mode assumes the wavelengths are in microns, and the inverse wavelengths are computed in wavenumber.

9.3 Changing Scale

The scale of the CRT plot is controlled by four values: the minimum and maximum for the horizontal ("wavelength") axis and similarly for the vertical axis. The plot scale may be changed by a command where the user types the desired limits, or by graphically selecting a window within the current window.

9.3.1 Changing Scale by Typing in the Range

The scale is changed by typing the

c

command. You are then given the following menu.

```
   Current Scale: VERTICAL= 0.0000E+00   2.0000E+00
                  HORIZ.  = Automatic

   To scale the plot, type in the mode (n or w) and horizontal axis limits
first.  When the vertical scale is entered, the routine will exit to the
plot.

   HORIZONTAL:
   type  n  and left and right hand limits for INVERSE WAVELENGTH, or:
   type  w  and left and right hand WAVELENGTH limits
            (if you type  w  only,  the program will AUTOSCALE the limits
            from the current wavelength set)

   VERTICAL:
   Type lower bound and upper bound values for the VERTICAL AXIS, or:
   type  A  to AUTO SCALE (the VERTICAL AXIS), or:
```

In order to change the horizontal axis range, you MUST change its range before the vertical axis. As soon as the vertical axis range has been changed, the routine automatically exits and the plot is redone. For example, to change the scale from 0.0 to 1.1 from the CRT plot, type

```
c
0 1.2
```

command sequence. If you wish to change the horizontal scale to 0.2 to 3.0 and the vertical scale to 0.5 to 1.3, type the

```
c
w.2 3
0.5 3
```

command sequence. The format is completely free: there is no need to add a decimal point on integers. If you type a mistake, the routine will flag it and ask you to retype the values.

Specpr allows commands to be separated by semicolons, so it is often desirable to see the plot while changing the scale. This can be done all from the CRT plot screen: enter the "c" command and the ranges separated by semicolons. Thus

```
c;w.2 3;0.5 3
```

is equivalent to the

```
c
w.2 3
```

0.5 3

command sequence.

In order to autoscale the horizontal range, type a "w" command with no minimum or maximum. To autoscale the vertical range within the current horizontal range window, type an "A" command. Thus to autoscale both the horizontal and vertical ranges from the CRT plot screen, type the

c
w
A

or

c;w;A

command sequence.

9.3.2 Changing Scale by Graphics Sub-Window

A window within the current window may be selected to change the plot scale using the graphics cursor. **NOTE: this option currently only works on X-windows or Hewlett Packard compatible graphics terminals.**

To change the sub-window, type the

S

command from the CRT plot command screen. The CRT menu will be erased and a new menu written asking you to position the graphics cursor at the position on the plot where you want the upper left corner of the new sub-window. Place the graphics cursor at the correct position and press the return key. Next you will be instructed to select the lower right corner. Place the graphics cursor at the desired lower right corner and then press the return key. The CRT plot will then be rescaled to the new window.

Note that you can not use this command to scale to windows larger than the current window. It is, however, a very useful command for enlarging small features on the plot.

9.4 Line Type

Each data set on a CRT plot may have a specific line type. The line types available may be dependent on the terminal type, but the line type also has a few different ways of plotting the data, for example by including a small cross at the actual point on the plot. Typing the

l

command (the lower case letter L) followed by a number from 0 to 9 will change the line type used in the CRT plot. The following shows the characteristics of line types on HP series graphics terminals.

Line	Description
0	error bars included, crosses, not connected (+ + +)
1	error bars included, crosses, connected (+--+--+)
2	error bars excluded, points connected (--------)
3	only small error bars excluded, points connected (-----)
4	error bars excluded, line: -- - -- - --
5	error bars excluded, line: ------ ------ ------
6	error bars excluded, line: ---- ---- ----
7	error bars excluded, line: .....
8	error bars excluded, line: ---- - ---- - ---- -
9	error bars excluded, line: --- --- ---

9.5 Horizontal Axis Labels

The default axis label for the specpr CRT plot is "Wavelength (microns)", but this label can be changed. The change is done in the wavelength set by changing line 4 of the manual history. For example, say you had a data set of intensity versus time, and the time data set was in seconds. Change line 4 of the manual history to read "\\W Time (seconds)" (no quotes in the manual history), and the CRT plot horizontal axis label will read "Time (seconds)". The key sequence for axis labels is the \\W (the backslash is important).

9.6 Changing the Wavelength Set

The wavelength set contains the "x-axis values" which make the plot of x-y paired data possible. The number of channels plotted is also controlled by the wavelength set in use. For Math operations, the number of channels that are written to the data file is controlled by the wavelength set in use AT THE TIME THE USER EXITS THE CRT PLOT ROUTINES. A different wavelength set can be selected by entering the uppercase file-ID-letter and the record number. For example:

V234

selects file "v", data set record 234 as the current wavelength set. You may also assign channel numbers to the "wavelength set" by typing a "C" and the number of channels. For example:

C2314

sets channel 1 to be "wavelength" 1.0, channel 2 to be 2.0, and so on to channel 2314 which is set to 2314.0.

9.7 Graphics Cursor Position

WARNING: the graphics cursor read routine currently only works with X-windows or HP-compatible graphics terminals.

Often you want to know a particular data value that is displayed on a plot. The graphics cursor read routine allows you to place the cursor on a particular data point, press return, and the closest channel number, wavelength, and data value are displayed, along with the equivalent wavelength and data value for the cursor position on the graph. This position capability also allows you to graphically interpolate a value if you are trying to estimate a data value.

To begin the graphics cursor read routine, type a

G

command (be certain it is upper case g, because lower case g is the glitch removal routine). Next, use the graphics cursor arrow keys on the graphics terminal to position the cursor to the desired location on the CRT plot. Then press return to get the position displayed. To exit the cursor position routine, type an

e

or

x

command.

9.8 Interactive Band Analysis

The interactive band analysis routine is a graphically oriented absorption band analysis system. The user selects continuum points graphically, the system then removes the continuum, and the user can select the band center or let the routine find it. The Full Width at Half Maximum (FWHM) works similarly. The routine can write the results in ascii to a user selected file.

WARNING: this routine currently ONLY works with X-windows or Hewlett-Packard compatible graphics terminals.

The routine is entered from the CRT plot routines by typing an

A

command. The routine then prompts the user for one of three actions: the changing of the vertical and horizontal scales, the opening of a new file, or the analysis of a band.

The changing of the plot scales is by a

c

command. The scale change command calls the plot scale routine, so it proceeds exactly as the normal scale changing is done in plotting a spectrum (see section 9.3).

If the user wishes to open a new output file for writing the ascii results of the analysis, type an

o

command. You will then be prompted for the file name and a title for the data which is to be written.

The "automated, interactive band analysis" is started by issuing an

a

command. You are first prompted for the first point which lies on the continuum which surrounds the band of interest. After this prompt is given, you are expected to use the graphics cursor position keys to move the cursor to the first continuum point and then to enter a return. A box is then drawn around the nearest channel in the data set (i.e. the continuum points can be selected by locating their proper horizontal position alone and ignoring the current vertical position of the graphics cursor). After the first point has been selected you are then asked whether this point is at the right position, if it is not, you may redo it by typing an

r

or you may continue by entering a return. Once you have selected the first continuum point, the process is then repeated for the second continuum point. If both points are correct you enter a return and the data are scaled by dividing out the continuum and then the continuum-removed spectrum is plotted.

At this point in the analysis you must determine whether you would like to move the locations of the continuum points (command "m"), change the plot scales (command "c") or have the band center selected (return).

If you decide to do another continuum the new one is done using the same process by which the original continuum was selected. The only exception is that the data for the y axis which is used to select the new continuum is no longer the original data but is instead the continuum-removed spectrum currently displayed on the screen.

If you wish to rescale the data, it is done in the same manner as in the normal spectrum plotting routine.

If you decide that all is correct and the band center should now be selected, the program begins its analysis with a return command. The program first searches the continuum-removed data to find the minimum channel between the two selected continuum points. Once the minimum point has been found, a parabola is fit to the minimum point and the two points

nearest to this minimum. The center wavelength (x-coordinate) of this parabola is then calculated as well as the reflectance (y-coordinate) on the parabola at this point. A solid vertical line is then plotted from this calculated set of coordinates to the continuum.

The reflectance level of the Full Width Half Max (FWHM) is then calculated and the FWHM is then itself drawn as a dashed line between the sides of the band with its endpoints being solid boxes. A vertical dashed line at the mid-point of the FWHM is then drawn from the reflectance level of the absorption feature to the continuum.

If there are enough channels (10 channels are currently being used as the limit) between the two end-points of the FWHM the routine will draw a cross at the Full Width Quarter Max (FWQM). This routine repeats this process determining whether it is possible to find the subsequent full widths (e.g. Full Width Eighth Max) by examining the number of channels as determined by the previous full width. The last full width that can be determined is the Full Width Sixty-fourth Max. The last FWQM (if present) or subsequent width which has been located is drawn as a dotted line with its endpoints being two small solid boxes. The center of this width is then computed and a vertical dotted line is draw from its intersection with the data to the continuum.

You are now asked whether you would prefer to use the fit selected center (return) or to manually select the band center by an

s

command. If the fit selected center is chosen the FWHM is calculated, as well as the asymmetry parameter and the band depth with the band center and FWHM being written to the terminal.

If you choose to manually select the band center you must manually move the graphics cursor (as before) to the correct horizontal position. A Full Width Half Max is then drawn on the graphics screen and is calculated and then written to the terminal. After the band center has been selected you have the option to redo the center with an

r

command, or to venture forward (return). Next, you have the option to select the FWHM manually with an

f

command, or to have the width selected automatically. If manual selection is chosen, a horizontal line is drawn at the proper reflectance level and you then select two points on that line in a manner similar to choosing the original continuum (Note: with noisy data, manual selection may need to be used).

After the FWHM has been determined by either of the methods above, you are

then asked whether the error associated with the horizontal data should be determined automatically (return) or picked by the user with an

m

command. The automatic error selection finds the first two points on either side of the channel nearest the band center and takes their absolute difference divided by four as the horizontal error value. If you wish to determine the horizontal error manually, you enter two points in a fashion similar to the selection of the continuum or the manual determination of the FWHM.

Once the error has been obtained by either method, you are asked whether the data should be written to an ascii text file (default). If you choose not to write the data to a file with an

n

command, the routine then asks if you would like to exit (command e) or begin the entire process again (return). If, however, you wish to write the data, the program prompts you for a file and title for the data if a file is not currently open. You then receive a prompt for a 15 character comment (longer comments are silently truncated) which describes the band (you may wish to standardize the content of this field when analyzing many bands). After a comment has been entered the program asks you if you want to exit with an

e

command or to begin the entire process once more.

Known crashes:

1) If one attempts to run the program to analyze a band with a large number of deleted channels, the routine seems to get confused and crashes specpr.

9.9 Deleting Individual Data Channels

Individual data channels may be marked as deleted by typing the

r

command followed by a list of channel numbers followed by a

c

command. Internal to specpr, the deletion is done by setting the value of the data in the specified channels to -1.23x1034. Thus, once a data point has been deleted, its value has been changed and it can't be undeleted. The list of channel numbers consists of numbers separated by spaces and by pairs of numbers separated by a "t" (meaning to or through). For example:

r 1 2 55t63 77 117 t 120 c

means to delete channels 1, 2, 55, 56, 57, 58, 59, 60, 61, 62, 63, 77, 117, 118, 119, and 120. If you forget the "c", the routine will ask for another line of deleted points until a c, for complete, continue, is entered.

9.10 Glitch Removal

Typing

g

from the CRT plot will call the glitch removal routine. The routine tries to identify glitches by looking for data points which are greater than 6 percent of the total data range and, by the use of a simple pattern recognition routine, checks 4 conditions. The data points which are thought to be glitches are identified by a small diamond shaped symbol. The user can then select which data points are "actual" glitches and can then correct them.

The glitches are assumed to be wrong by some power of 2 from the true data. This follows from a binary counter where one of the bits has been set wrong. This routine was written specifically for the University of Hawaii "Wedge" CVF spectrometer which has these type of counters, but most modern digital instruments are similar and this routine may work for them.

WARNING: once the data have been multiplied or divided, the glitch removal is an estimate of the actual data and thus strictly speaking, is "fudging" the data.

The user indicates which channels are to be "deglitched" by selecting all data points marked on the graph, only those indicated by a list, or all those selected by the program except for those in a list. By typing an

a

command, the user tells the program that ALL the points identified plus the channels typed in after the "a" are glitches to be corrected. By typing an

o

command, the user tells the program ONLY those channels typed in after the letter "o" are to be corrected as glitches. By typing the

b

command, all the channels identified by the program are glitches BUT those channels which are typed in after the "b". At least one space must occur between channel numbers.

The glitch routine searches for 15 glitches at one time so it may take more than one pass to remove a lot of glitches. Sometimes more than one bit is wrong, and it will take more than one pass. Glitches which occur next to each other are not recoverable by this routine since the routine tries to correct the point to the surrounding data using the nearest power of 2. If after 2 passes on the same point the data value is not restored, it is probably lost.

A note on removing glitches: you are fudging the data. If you are not very careful and use the utmost restraint, you may create some absorption or emission features you had not counted on!

9.11 Information Display and Information Change

Every standard specpr data set contains extensive header information, including the title, history, dates and times of data acquisition and data processing, as well as many others. Appendix A lists the specpr format, and all the header information values may be displayed and changed in the Information Change routines.

To change header information, type the

i

command from the CRT plot screen. The header information is contained on many pages, the first of which displays the title. Pressing return goes to the next page, and typing

r

from any page returns to the first page. To change information on any page, type the indicated letter. You will then be instructed to input the appropriate data.

In the case of the manual history, which is displayed in 4 lines and can be changed one line at a time type

m

and the line number or simply the line number 1, 2, 3, or 4, or to change all four lines, type "m" and no line number.

In the case of the Band Normalization factor, scan time, or total integrating time, the number can be integer, floating point, or scientific notation. In the case of scientific notation, the number is typed in as an integer or real number, then the letter e, then the exponent (to the power of 10, an integer). Thus

1.4e12

is equivalent to

1400000000000.

There are 3 ways to exit the information change routine. When at the last page, pressing return with no input will exit to the CRT plot. Typing

g

from any page will also exit to the CRT (graph) plot. Typing

e

will return to the calling routine (Math Operations or File Display and Transfer). The "e" exit command will not terminate other processing; it only skips the CRT plot. Thus, if a file write is pending (as in the Math Operation, Chapter 8, or file transfer with display, Chapter 10), it will be completed in the type e exit. If you wish to exit and terminate pending file writes, type

x

for a hard exit.

Note that, from the data display routine (Chapter 10), no information is changed on the stored data unless there is a transfer involved (see Chapter 10).

9.12 Printer Listings and Printer Plots

The entire header information and data (in scientific notation) can be listed on the printer by typing the

pd

(print data) command. The data can be plotted as a printer plot with the current vertical scale by typing

p

and the number of copies (10 or less). The data can be plotted in a printer plot only as a function of channel number. The vertical-axis resolution is 1 part in 100 (100 print positions for the plot). For each data point printed, the wavelength, channel number, and data number are given. Four pages are required for 256 data points and 2 pages for 120 points. When the number of channels is less than 120, the plot is scaled to fit from 1 to 2 pages.

9.13 Multiple Commands in the CRT Plot Routines

One entire line (80 characters) can be input to the CRT plot routine at one time for execution. For example, to change the wavelength data set to file v, record 2, change scale, print a printer plot, and set the line type to 3, you would type the

V2cpl3

or

V2 c p l3

Spaces can be inserted wherever desired but are not necessary. This multiple command capability greatly speeds up processing since it may take several seconds to plot the data on the CRT and would take a long time to replot the CRT after each command (on a standard graphics terminal; X-windows is typically only a fraction of a second).

9.14 Exiting the CRT Plot Routines

There are 3 commands for exiting the CRT plot. The normal method of exiting the CRT plot routines is by typing an

e

command. In this case, the user soft exits from the CRT plot in the normal fashion and the program executes the next command.

The "hard exit", terminate all pending commands, is the

x

command. With the "x" exit command, specpr returns to the calling routine (which is either Math Operations or File Display, Transfer, and Overlay).

After a soft exit (e) from the CRT plot under Math Operations, the data set is written to the requested location (see section 8.7). If errors are included, the program writes on the CRT where they will be written and gives the user the option of not writing the error data set (by typing an

x

command). Otherwise, press return or type a

c

command to continue.

After a soft exit from the CRT plot under File Display and Transfer, the data set is written only if there is a transfer (see Chapter 10); otherwise, the next command is executed.

The third way to exit the CRT plot routines applies to the Math Operations routine only. If the Band Normalization option was not turned on, the user may type a

b

command. This will turn on the Band Normalization option, exit the CRT plot, and go to the Band Normalization routine (section 8.9). When the Band Normalization routine is left, the program will return to the CRT plot. If the "b" command is entered while in the file display, transfer, and overlay routines, the CRT plot will soft exit (as a type "e" exit above).

CHAPTER 10

DATA DISPLAY, TRANSFER, AND OVERLAY

10.1 Introduction

The data display, transfer, and overlay routines are used to display data on the graphics terminal (or X-window device), transfer data records from one file to another (or to another place in the same file), and overlay data on X-windows or graphics terminals. To transfer means to copy; specpr does not (nor is it intended to be able to) remove the original. Historically, the transfer routines were written to "transfer" a copy of the data on a magnetic tape to a disk file. Because of this need, there are restrictions when transferring data records from disk to tape or tape to disk so that the records will properly coincide and the proper histories will be maintained.

10.2 Data Display

A spectrum can be displayed on the CRT using the CRT plot routines (section 9) by typing in the file ID (v, w, d, u, or y), the record number, and any options. For example,

v23

displays the data in file v, record 23 with the current wavelength set and graphics line types. Options can be included to change the wavelength set in use, change the number of channels, include error bars and do other things.

Options:

```
e   to include error bars,
a   to auto scale within the current plotting window,
b   to auto scale and make a printer plot.
l#  to change the graphics line type (see Chapter 9),
i   to display header information first, not the graphics,
&   use the wavelength pointer with the data to get
        the wavelength set.
+n  sequentially plot the next "n" records, where n is
        an integer.

V, W, D, U, or Y, followed by the record number sets the
    wavelength set in use to the new value.
C#  sets the wavelengths equal to channel number with
        the number of channels = "#".
h   sets the plot type to channel mode.
```

For example, if v23 contains a reflectance spectrum of the mineral Alunite, and v20 contains the wavelengths to the spectrum, you could type

v23V20

to plot reflectance on the vertical axis versus wavelength. Alternatively, if you typed:

v20V23

the plot would have wavelength on the vertical axis and reflectance on the horizontal axis.

If many records in sequence are to be plotted, then directly after the record number type "+" and the number of records to be plotted. Thus, to plot spectra from device v, records 10 to 32, type

v10+22

where the +22 means the next 22 records after the requested record should also be processed. Note that the + specification will yield strange results with option e (include errors) since the record increment is always 1. Thus, after plotting data with errors (e.g. record v10 = data and record v11 = errors), the record increment will be one, and the errors will be plotted as data with the next record plotted as the errors to the errors! To plot many spectra in sequence which all have error records, they must be typed in sequentially such as v10e v12e v14e v16e v18e v20e v22e. Note no spaces are required, but they can be inserted if desired. No commas are necessary (this is different from Math Operations).

If your spectra take up more than one record, you must count by records, not spectra. Because the increment is by records, multiple records to one spectrum increments the counter by the number of records not by the number of spectra. For example, if you have 512 channel spectra, and you want to overlay 3 spectra (each spectrum takes 2 records in the data file) you would use the current spectrum record number plus 5 records. If the first spectrum was at v20, then the next two are at v22 and v24, so type "v20+5" (which is equivalent to "v20 v22 v24").

10.3 File Transfer

Files can be transferred by specifying the file ID, record number, the transfer specification

t

and the file ID and record number where the transfer is to go. Note that "transfer" means to copy. Many records may be transferred sequentially by using the plus specification. For example,

v10td23

transfers (makes a copy of) file v record 10 to file d record 23. Also,

v10+99td23

transfers file v record 10 plus the next 99 records to file d record 23 plus the next 99 records. Thus, file v record 109 will be the last record transferred and will go to file d record 99 + 23 = 122.

10.3.1 File Transfer with Plot or Information Change

Records that are going to be transferred can be plotted (and thus can have points deleted, deglitched, or information changed, see section 9) before the transfer is actually carried out by the option

c

which stands for "crt plot routines." Thus, to transfer v10 plus the next 5 records to d23 and display them before transferring, type:

v10 + 5 ct d23

(the spaces are not necessary, but are included for clarity). The first file ID and record number represent a request to read the data. The "t" option means to set up a transfer (copy). The "c" option means to also enter the crt plot routines for anything you may wish to do. The transfer is always the last step processed regardless of the order of the options. However, the destination of the transfer must come after all other options.

If information only needs to be changed, the CRT plot can be skipped and the information change routine called directly by a

cit

specification as in

v10 + 5 cit d23.

The c is required since the information change routine is part of the CRT plot routines. The "i" then signals the CRT plot routines that the information change routine only is requested.

Note, when transferring data to a protected (positive or zero protection value) file, the record number to transfer to must be 1 plus the protection value. For example, if you transfer to v1 but 100 records are protected, the program will say there is a protection violation. However, you do not need to specify the destination record number if the file is protected with write at the end of the file (protection value is positive). For example, to transfer from w36 to file v whose protection is 100,

w36tv

and

w36tv101

are equivalent.

10.3.2 Starpack Transfers

Starpack transfers involve 3 regular data records. Thus, transferring starpack 1 (s1) to v10 (s1tv10) will result in the starpack being put into v10, v11, and v12. When the starpack is transferred back,

the program automatically collects the 3 records for starpack s1. Thus, to transfer 6 starpacks (1 - 6) to file v starting at record 29, type

s1 + 5t v29.

The starpack will be put in records 29 through 46. To transfer these starpacks back, type

v29 + 5 t s1.

When the program sees that starpacks are being transferred, it knows the +5 means 5 starpacks at 3 regular records per starpack.

10.4 Overlaying Data Sets

Overlaying spectra on the CRT is an excellent way to compare data for reproducibility. The overlay is an option used in file display (see section 10.2). To overlay spectra, type the file ID, record number, and the option letter o. The o must be included in every spectrum to be overlaid. Errors can be included, and the overlay can be plotted as a function of channel, wavelength, or inverse wavelength (the program assumes the wavelength values are in microns and converts the values to wavenumber). Also, you can change the graphics line type with each spectrum so that all data sets are distinguishable. For example, to overlay v23, v30 and w56 on the same plot, each with a different line type, you could type:

v23ol3 v30ol5 w56ol7

where the line types 3, 5, and 7 are used (see Chapter 9 for descriptions of the line types). To overlay data sets with different wavelength sets, include the wavelength set as an option. For example if y25 has wavelength set y20 and d104 has wavelength set d3, you could type:

y25Y20ol3 d104D3ol7

The plot type (wavelength, inverse wavelength, or channel) can be set as an option on the first file to be overlaid. However, when in wavelength mode, you should not be in the auto-scale mode. The minimum and maximum wavelengths can be different for different data sets; thus the CRT plot routine will scale the data differently if the wavelengths are auto-scaled. To avoid this problem, simply set the minimum and maximum wavelengths to accept the range of all of your data sets.

Many spectra in sequence can be plotted and overlaid by using the plus specification as in section 10.2. The program tells you when the last file in a sequence is overlaid on the CRT. The plus specification will give strange results when including errors as noted in section 10.2.

Another example of overlaying many spectra is:

v10V8l6o v11el3o d23D20 + 5o y51oY30.

Note that the letter o is required on the last file to be overlaid. Note that the "o" is not always in the same place: it doesn't matter what order the options are in. The "e" means include error bars. If an "o" option is not included with each spectrum, the screen will be erased, and that particular spectrum will be displayed as in section 10.2.

Note that the wavelength set stays the same until it is changed again.

10.5 Multiple Commands in Data Display Transfer and Overlay

One entire line (80 characters) can be input for execution. These commands can be mixed freely between data displays, file transfers, or data overlays. Spaces may be included anywhere on the line, but none are necessary. An example of multiple commands is:

v1+3ctd1 w34+10W4 v23+14oV6 v38eo v102+243tv236 v346+10citd480.

First there is a display and transfer of 4 spectra, then 11 spectra are displayed, then 16 spectra overlaid (the last with error bars), next 244 records are transferred, then 11 are transferred with information change first.

In any of these routines, typing an

e

will exit to the next command, (including the next one in sequence) and typing

x

will exit back to Data Display, Transfer and Overlay (except in the information change, see section 9) without execution of the remaining commands.

10.6 Extraction of Data from 3D Files for Display and Transfer

The 3-dimensional (3D) data extraction routines are best used from the data display and transfer routines. Here, for example a spectrum can be extracted from an imaging spectrometer data cube, displayed on the graphics device and the copy transferred to a regular specpr format data file for further processing.

IMPORTANT NOTE: Because of the history mechanism, it is strongly recommended that you extract data from 3D files in this routine rather than directly in the math routine. When a 3D data segment is extracted, a title and history that describes the exact data set is created. If the extraction was done in math as an input to a routine, the math history generated loses the 3D file origin and will only give the file name and

the first record number (of the extraction sequence). The proper way to track histories then is to extract the 3D data, transfer the data to a standard format specpr file and do math operations, plotting and other functions from the regular specpr file.

Examples. If file v is assigned to an AVIRIS imaging spectrometer data cube containing 512 lines, 614 samples (across track) and 224 bands and you wish to display the spectrum at pixel (line,sample) = (200,132) you would type

vpx(200,132,*)

If you wanted to extract a block with the upper left pixel at 200,132 and 3 lines by 5 samples, you would type

vpx(200+2,132+4,*)

and 15 spectra would be extracted and the error bars (the standard deviation of the mean) would be computed.

To transfer a copy of the extracted data to a regular specpr file (which in the example below we will use file w), you would simply do a transfer using "t":

vpx(200+2,132+4,*) t w56.

The result of the 15 pixel average would be copied to file w record 56 and the error bars put in record 57.

As described in the file assignments chapter (Chapter 6, section 6.2), the extraction direction can be in any one of the 3 orthogonal directions. For example, to extract a profile across track (along the "sample" direction) at line 200, wavelength channel 12, you would type

vpx(200,*,12)

and the data would be plotted on the graphics device.

To extract a profile along the flight line at sample 132, wavelength channel 12, you would type

vpx(*,132,12)

and the data would be plotted on the graphics device.

CHAPTER 11

DATA FILE LIST

11.1 Introduction

The data file list routines read the contents of a specpr data file and prints header information and data on the terminal or to a print spooler. The user can select different modes to print different fields, search for strings and print only those found. The search capability along with the ability to read selected records within a file (given as a list of record numbers) can be used for simple data base operations.

The list routines can be entered from Program Operations Control, Data Display Transfer and Overlay, or from Math. To enter the list routines, you type

l<ID>

where "<ID>" means one of the specpr file letter ID's: v, w, d, u, or y.

At all user command points in the list routines, an "e" or "x" command will terminate the routine and return you to program operations.

NOTE: there is a new X-windows based, "point and click" specpr data file viewer. It is currently called "sp" and is available as part of the pic-works (pw) software package.

11.2 Listing Mode

When you first enter the list routines, you are given the option to print in "laboratory" or "telescopic" mode:

```
 type  t  for TELESCOPIC or  l  for LABORATORY (DEFAULT)
or beginning and ending files to be listed
```

The telescopic mode prints the observation air mass whereas the laboratory mode prints channels in that column. The mode is different for both the terminal list and the printer list. The printer listing always assumes a 132-character wide printer and is able to show more items than the terminal listing. The default printer listing also shows the sidereal time in telescopic mode.

11.3 CRT Listing Options

After you select the telescopic/laboratory mode, you are asked which items to show on the terminal.

```
 *** CRT print options ***
n  print # only on CRT when printing to lineprinter
a  print wavelength record numbers
h  print history
t  print text
```

```
  r  read and print a list of comma separated record numbers
  w  write to a file, a list of comma separated record numbers
     You will be prompted for the file name after you select r or w.
```

The "n" option speeds the operation when all you desire is a printer listing. The "a" option lists the wavelength pointers in place of the channel numbers in laboratory mode. The "h" option adds the history to the listing (this takes more than one line on the terminal). The "t" option prints the text if the data record type indicates a text record.

As noted in the introduction, the list routines can read only selected records. Similarly, it can write the records it finds. For example, if you do a complicated search, you can record the record numbers found by the search and then read only those numbers in subsequent listings. The file containing the record numbers must have each record number separated by a comma. If the number is the last on a line, it must still have a comma. The "r" option directs the list routines to read one of these lists, while the "w" option directs it to create a list. You can both read and write a list at the same time.

11.4 Printer Listing Options

Next you enter the options to decide what to print on the printer.

```
LINEPRINTER OPTIONS (return for CRT only)
   p  print titles + selected fields given by print mode
   a  print titles with no pausing at CRT page full

   with p or a, also select:
   h  print history
   m  manual history if not blank
   t  print text on text data files
   b  print all header info except manual history
   c  print all except data
   d  print all and data
```

If you press return, no printer output is created. To create a printer listing, you must type either

p

or

a

If you type "p", the terminal stops each time the terminal screen is full, but if you type "a" (for "automatic, no stopping"), the list continues until the user limits are reached.

With a "p" or "a" command, you can select options as described above. For example, to make a listing including the history, type

11.5 Search Capability

The next command prompt is for search capability:

```
Enter SEARCH STRING: letter "string1" letter "string2" etc.
       where letter is for t  title
                           h  history
                           d  date
                           m  manual history
       maximum number of strings is 4
       comparisons is and (default), use | for or, & for and
```

Pressing return does no search selection; all records are printed. To limit the listing to a specific search parameter, enter the search field (t, h, d, or m) and the search string. The search string is a full Unix regular expression. See the Unix string searching documentation with your computer for more details. A few examples will be given below.

You may enter multiple search fields, as long as they are less than 80 characters or less. They may include multiple entries of the same type, however, such as two different title strings. The date field is converted from its numerical value to a string for this search capability. The date field searched is the date of data acquisition.

Examples:

```
t"Kaolinite"           \# search for all Kaolinite
t"[Kk]aolinite"        \# search for all Kaolinite or kaolinite
t"Kaolin" t"NN56"      \# search for all entries that have Kaolin
                           and NN56 in the title.
t"Kaolin" d"198[56]"   \# search for all Kaolin spectra obtained in
                           1985 or 1986
t"Kaolin" |t"Alunite"  \# search for all entries of Kaolin or
                           Alunite
```

In the above examples, a "&" was not necessary because "and" id the default logical operator.

11.6 Listing Limits

The final prompt before the listing/search occurs is to enter the limits of the search:

```
Type in the beginning and ending files to be listed
```

If the lower limit is zero, it is reset to 1. If the upper limit is greater than the protection limit, it is reset to the protection value.

11.7 Continuing/Ending the List

When the listing is complete, or the terminal page is full, you can continue, or exit to one of 3 places as described by the prompt:

```
Type  c  to continue, e or x  to exit,  Type in new record numbers to list
type  r  to return to options, type t to go to file display and transfer
or type m to go to math routines
```

If you type c to continue and you are at the upper limit given above, but that upper limit is less than the protection value, the listing/search will continue up to the protection limit. To continue with a new range, do not type "c," just the new range of record numbers.

11.8 Sample Listing

A sample listing using the default laboratory mode looks like the following.

```
 file           title                              chans    time        date
    73  Description of Alunite   Hunt 295.?B      3132 Characters of TEXT
    76  Alunite 295.3B .2-3um 1x ABS REF           512  12:03:14.00  08/20/1985
    78  errors to previous file   77               512  12:03:14.00  08/20/1985
    80  Alunite 295.3B .9-2.6um 2x ABS REF         680  09:49:27.00  08/21/1985
    83  errors to previous file 80                 680  09:49:27.00  08/21/1985
    86  Alunite 295.3B 1.3-1.7um 2x ABS REF        264  11:43:26.00  08/22/1985
    88  errors to previous file 86                 264  11:43:26.00  08/22/1985
    90  USGS Den BKMN STD waves 4x 2.2738-4.00um   256  04:23:02.00  11/13/1985
    91  USGS Den BKMN 4x resolution                256  04:23:02.00  11/13/1985
    92  Alunite 295.3B 2.36-2.555um 4x ABS REF      75  14:00:38.00  08/21/1985
    93  errors to previous file 92                  75  14:00:38.00  08/21/1985
    94  Alunite 295.3B 1-1.55um 4x ABS REF         442  16:49:01.00  08/21/1985
    96  errors to previous file 94                 442  16:49:01.00  08/21/1985
    98  Description of Beryl   (WNS)             1737 Characters of TEXT
 ------------------------------------------------------------------------------
 Type  c  to continue, e or x  to exit,  Type in new record numbers to list
 type  r  to return to options, type t to go to file display and transfer
 or type m to go to math routines
```

Note that the continuation record numbers are hidden from the user.

CHAPTER 12

EXTINCTION ROUTINES

12.1 Introduction

The extinction routines are used to correct astronomical data for the effects of the terrestrial atmosphere. The extinction analysis is a least squares fit of a straight line to the base 10 logarithm of the data versus airmass for each channel. The intercept and slope of the line, along with a goodness of fit (correlation coefficient), are stored for later extinction corrections with objects observed and is together called called a "starpack."

Extinction analysis methods and examples can be found in

McCord, T.B., and R.N. Clark, Atmospheric Extinction 0.65-2.50 m Above Mauna Kea, Pub. Astron. Soc. Pac., 571-576, 1979.

The extinction routines are accessed by typing

s

from Program Operations Control. The screen will then list available commands on the CRT.

12.2 Starpack List and Display

Type

d

to list the starpack by title. You will then be instructed to input a record number (1 to 50) or type the number 0 to list all starpacks in the starpack file. Typing a number (1 to 50) will read that starpack into memory so that it can be displayed (see section 12.4). If zero was typed, the last starpack listed is in memory (assuming there is no I/O error).

IMPORTANT NOTE: If you have just created a new starpack, do not list the starpack file as this writes over the starpack in memory.

Note that, when reading a starpack from disk, you cannot do airmass versus log intensity plots because this information is not stored in the starpack. It is only stored in memory after an extinction calculation. In order to do these plots, or deletion and restoration of runs and channels (section 12.5), you must recreate the starpack from scratch using the data indicated in the history.

12.3 Extinction Calculation

Type

c

to perform extinction calculations. The computer then asks the user for up to 20 spectra for extinction calculations. The data are input by

typing in the file ID (v, w, u, y, or d) and the record numbers. The program is designed so that the user can type on one line as in the example:

w12 13 14 15 16v1 2 3 d4 5 6c.

Typing the file ID and then a string of numbers after the ID causes the program to store the string of numbers along with that file ID. When all files have been typed in, the user types

c

to indicate the finish. More than one line (at 80 characters per line) may be used. Note: At the beginning of a new line, a file ID must be typed; otherwise, erroneous record numbers will be stored.

After typing in the data sets, the user is asked to type in a 24-character title for the starpack. After storing the title, the computer will read in the records and list the titles on the CRT, then calculate the extinction values (it takes a few seconds), and display the slopes in a CRT plot. At this point in extinction calculations, all the data are stored in memory and the extinction coefficients have been calculated, but the starpack has not been written to the starpack file. The user should look over the calculations using the CRT plot and then write the starpack on disc (see below).

Possible deviations from normal operation:

If the number of spectral scans (revs) in each data set are not the same, the sets whose scans differ from the first set read in are flagged.

If a file ID has not been included, a read error is encountered, the list of options is displayed, and the user must begin again.

12.4 Extinction CRT Plot

Because the extinction coefficients are in 3 arrays (slopes, intercepts, and goodness of fit), they are displayed one at a time on the CRT. When the plot routine is entered [automatically from extinction calculation or by typing

p

(for plot) from extinction commands], the slopes are displayed. The scale can be changed by typing

c

and the intercepts can be plotted by typing

i

and the goodness of fit can be plotted by typing a

g

command. The slopes can then be plotted again by typing an

s

command. A lineprinter plot can be made of either slopes, intercepts, or goodness of fit by typing

p

(note that the printer spooler must be assigned). To exit the plot routines, type an

e

or

x

command. Both these exit commands are the same since there can be no pending commands.

12.5 Airmass Versus Log Intensity Plots and Deletion and Restoration of Runs and Channels

To investigate the fitted line, and the scatter in the data, the airmass versus log intensity must be plotted. To improve the fit, entire runs or single channels must be deleted (and restored). Typing

t

will call this routine. The program will then ask you to type in a channel number (with which an airmass versus log intensity plot is to be made) or type

d

to delete runs and channels or

r

to restore runs and channels.

Up to 10 channels in each run can be deleted, or the entire run can be deleted. Because up to 20 data sets can be used to calculate extinction, this routine refers to the sets as run numbers 1 to 20 in the order that they were typed in by the user. To delete a run, type the run number and 0. The system then asks for n channel numbers to be read in. The whole procedure can be repeated until the user has finished deletions. A least squares fit is done in this routine, and the routines list of options is displayed upon completion. The least squares fit takes a few seconds.

Selecting restoration causes the display of an array of numbers that is 12 wide and 20 high. The left-hand column is the run number, and the column next to this gives the number of channels deleted or, if the number is -1, it indicates that the run has been deleted. To restore a run, type in the run number and zero; to restore a channel, type in the run number and the channel number. This process can be repeated, if necessary, and the array can be plotted out again showing any changes. When restoration is complete, a least squares fit is applied, and the routines list of options is displayed. The least squares fit takes a couple of minutes. To exit this routine, type e.

Note that no history is kept of runs and channels that are deleted or restored. There is a 592-character manual history (see section 12.7) for any notes or listing runs and channels which have been deleted, but the user must do this manually.

12.6 Writing a Starpack

After extinction calculations are finished, the starpack is stored in memory but is not written to disc (to the starpack file). This is not done automatically because the user must look over the data and decide whether to delete runs or channels or whether or not the starpack is any good at all. Thus, the user must remember that an extinction analysis has been completed and that the starpack must be written to the starpack file. (If you have been able to keep the slightest attention while reading this manual, you will be able to remember when you have finished an extinction analysis). To write a starpack to the starpack file, type an

s

command. You will then be asked for the record number (1 to 50).

12.7 Starpack Manual History

There is a 592-character (8 lines at 74 characters per line) manual history for special notes or for listing runs and channels which have been deleted. Type

m

from the extinction routines section. You will then be asked to type in the line number or zero for all lines or

e

to exit the routine.

CHAPTER 13

PLOT ROUTINES FOR WORK AND PUBLICATION

13.1 Introduction

The specpr hardcopy plot routines are designed to provide publication quality plots for general x-y data sets. Specpr builds two intermediate files: a vector file and a text file. The files are plotter device independent. Specpr then starts a Unix daemon process that reads the files and plots the data on a specific plotter. The daemon process is device dependent. While some users have written programs to use several different plotters and even laser printers, the most commonly used plotter is an HP7550-compatible plotter. This chapter is specific to the HP-compatible plotters, but will work in general for all other plotters so far implemented.

The plot routines are called from Program Operations by typing a

p

command.

When you first enter the routine, the first command will list the available plotters. This may be site dependent.

13.2 Plotting Mode Type

The first step in setting up a plot is to select the plot mode. Type

w

to plot in value versus wavelength,

n

to plot in value inverse wavelength. In this case, you specify a wavelength set in microns and the plot routines convert to wavenumber where wavenumber increases to the right. To plot in wavenumber, as above, but with wavenumber increasing to the left (reverse of the above mode), type an

r

command. If you want to use your own horizontal axis label, type

y

(the plot will be linear). You will then be asked to type in the horizontal axis label (24 characters). For example, you might type:

TIME (SECONDS)

or whatever your data correspond to. Thus, you may do any kind of one-dimensional array processing you wish.

13.2.1 User-selected wavelength limits

If you type only a letter for the plot mode, the horizontal axis is scaled to the range covered by all the wavelength sets used in the generation of the plot. However, you may select the wavelength limits for the plot, regardless of the wavelength sets used. When you type in the "w" for wavelength mode (or n, r, or y), also type (on the same line) the minimum and maximum wavelengths for the plot. Type in wavelengths the inverse wavelength modes (n and r) also. Example:

```
w 0.325       2.60
n 0.325       2.60
r 0.325       2.60
y 0.325       2.60
```

The actual limits on the plot will be 2% greater than the range (maximum-minimum) on each side (see section 13.5 for details). The left bound is then minimum minus 2% of the maximum-minimum range, and the right bound is the maximum plus 2% of the maximum-minimum range.

You can suppress the addition of the 2% by adding the option

```
n
```

after the limits. For example,

```
w 0.325  2.60
```

would produce actual limits of 0.2795 to 2.6455. However,

```
w 0.325  2.60n
```

would keep the actual limits at 0.325 and 2.60.

To exit the plot routine, type an

```
e
```

or

```
x
```

command anywhere in the routine.

13.3 Vertical Axis Label

Several vertical scale labels are displayed for you to select. They are numbered 1 to n, where n is currently 8. To select one of these, type in the number. If you wish to type in your own label, type the number (currently number 9) that says "type in your own label," and you will be instructed to type in a label of not more than 60 characters. If you wish to plot the vertical axis in log (base 10) of the input data, type

```
10
```

and you must type in your own vertical axis label. The program will then proceed to the Delete From All Spectra section. Type

e

or

x

to return to the beginning (section 13.1).

13.4 Delete From All Spectra

If, from all the spectra you are going to plot, you wish to delete one or more channels common to all, type a

d

command. You will then be instructed to type in the channels that will be deleted from all spectra to be plotted. At least one space should be left between each pair of channels, and more than one line can be used for input. All channels may be deleted if desired. When you are through, type

c

to continue to the scale section. Type

e

or

x

to return to the beginning (section 13.1). When typing in a sequence of channels, you may give the beginning channel, a

t

and the end channel. For instance, to delete channels 1, 2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 23, and 24, you could type

d1t8 15t17 23 24c.

13.5 Scale of Plot

The vertical scale of the plot is selected by the user. Type in the lower and upper bounds of the plot. The plot size is 13.5 centimeters (vertical axis) by 21.7 centimeters (horizontal axis) for the data plot area on the HP plotter. However, the plot size does vary somewhat with other plotters.

If your data are to be plotted in log space on the vertical axis, the lower and upper bounds should be input as ordinary numbers. The

program will convert the scale limits and data to logs. The scale is then labeled as the log of the number. Type

e

or

x

to return to the beginning of the plot routines (section 13.1); otherwise, the program will go to the Plot Scale Factors section (section 13.6).

13.6 Plot Scale Factors

Next you are asked to enter the plot horizontal and vertical scaling factors. These factors must be less than 1.0 and scale the size od the data area of the plot. Thus if the original size of the plot was 17 centimeters high, a vertical scale factor of 0.5 would make it 8.5 centimeters. These scale factors do not affect the text or the text size.

13.7 Data Set Input and Options

The data sets to be plotted are input as the file ID (v, w, u, y, or d), the record number, and the options. For all data sets entered, the wavelength set must be specified. If it is not, unpredictable plots can occur. A maximum of 50 spectra may be plotted at one time. These may be all on one graph or separated into many plots by an option request. The options are listed below.

Option B means to plot symbols as Black (assuming the black pen is used on HP compatible plotters), the same color as the axes labels. The default is to use the same pen color as the data.

Option c means to Connect the data points. If the wavelengths are not in sequence, or if a channel has been deleted, the points involved are not connected.

Option C means to Always Connect the data points. This option permits the wavelengths to be NOT in sequence, but will connect them anyway.

Option d means to delete points from this data set only (as opposed to delete from all data sets as in section 13.4). The program will ask for channels to be typed in when it is ready. More than one line may be used to input the channels to be deleted. When you are finished entering channels, type

c

to continue.

Option e is include Errors.

Option E is include Errors, but only when the error bars are large enough to be recognized on the plot.

Option g means the spectrum after this one is to be plotted on a new graph. Thus, 50 spectra may be plotted on a single graph (overlaid), or they may be plotted singly or any combination. The g (new Gould plot) selects when a new plot is to begin. The "g" works the same for non-Gould plotters also.

Option l followed by a number 1 through 8 selects the line color for the plot. The line color for HP-compatible plotters is the pen number. The default is size 1.

The line color for some implementations is device daemon dependent. For the Gould electrostatic plotter this number represents the line thickness. The axis is fixed at 3. The number 1, 3, 5, or 7 is the width of the line in Gould units. Size 1 is about 1/200 of an inch (0.127 mm). The line size changes error bars and the line connecting points.

Option L is the Line type. For HP-compatible plotters it varies from solid to various dot and line patterns. See Figure 13.7.1 for specific examples.

Option m More copies: In the options, specify "m" and the number of copies desired (1 to 9) default=1. Note that you need this option only on the last spectrum to be put on the plot.

Option n means that, if a point lies outside the lower or upper bound, it should be deleted. If this specification is not given, the point will be plotted at the limit of the graph.

Option o means to offset the data, followed by the amount to offset. You can use any positive or negative number. This number is added to the data before it is plotted. For example to offset data set v23 with its wavelength set v22, you could type:

v23V22o2.2

to offset by 2.2 data units.

Option p followed by a number is the point size. This number can be 0 to 5. If the number is zero, no point will be plotted. Point size 1 is about 1.27mm on a side, size 2 is twice this, size 3 is three times, etc. Note, if point size zero is specified and

points are not connected, no data will be plotted, and you will have a blank plot! A line drawing can be made by specifying point size zero and connecting points. The default point size varies with the total number of channels per spectrum. It is size 3 for 30 channels or less, size 2 for 31 to 100 channels, and size 1 for more than 100 channels. The symbol type changes with each spectrum plotted. See Figure 13.7.2.

Option r Because mistakes are bound to happen when typing in all these data sets, you may return to the last step for re-entering your data. This is done by typing an "r" in place of a file ID and record number. Example: You type in the spectra

1) "v23eV2p3"
2) "v29eV2p4"
3) "v30p1c"
4) "v33V5".

You now notice the step #3 should have been "v31V3p1c". You are at step 5: you type "r"; the computer says

"RETURN TO 4; CONTINUE".

You then type "r" again, and the computer responds

"RETURN TO 3; CONTINUE".

You may then retype step 3 (and also all steps after 3). You may not return to a step before the first data set entered (if you need to, you must type "e" or "x" to exit).

Option s Symbol type: To change symbol type, specify "s" and the symbol number (1 to 8). See Figure 13.7.2. The symbol size is given by the "p" specification (point symbol size). The symbol type is incremented for each spectrum input and is reset to 1 for each new plot ("g" specification).

Option t means to include text on the graph. Text may be written as 15 sets of 70 characters per set for each spectrum entered. The text position is given in centimeters [horizontal direction (x), then vertical (y)] from the lower left-hand corner of the axes of the plot. After the x and y position, input the text angle (90 is horizontal and 0 is vertical) and then the text to be added. When all text is entered, type a

c

to continue to the next data set.

Symbols may be plotted using the text mode. Instead of x, y, degrees, text, you enter

s <symbol type> <size> x y

where <symbol type> is the symbol described under the "s" option. The symbol size is also described under the "s" option.

Terminate entry of the text mode and continue entering data sets by typing a

c

command.

Option V, W, D, U or Y followed by a record number sets the wavelength set to be used (the horizontal axis values).

Notes on Text entry: You may also return to the previous text entry using the "r" specification as with the data set input. You may not return to a previous entry before text entry 1. If you have an error in a previous file entry, type "c" to continue to next step after text entry; then return to the last data set input as described above.

The specpr plot routines allow subscripting, superscripting, backspacing and greek and math characters in the text. Special characters are used to enter the different modes and can be used wherever text is written on the plot (horizontal or vertical axis labels or other text). The characters are:

Character	Effect
\ (Backslash)	Backspace to previous character
\{ (Left Brace)	Go into subscript mode
} (Right Brace)	Go into superscript mode
@ (At sign)	End last scripting mode
! (Exclamation Point)	Toggle greek/normal mode (see table 13.7.1)
| (Vertical Bar)	Toggle math/normal mode (see table 13.7.2)

NOTE: the \ is necessary before "{" because these brackets are used by the command interpreter for variable parsing (section 2.6). The "\" escapes the "{" so that it is not interpreted by the command line variable parser.

Table 13.7.1

Greek character set

A	α	A	a	N	ν	N	n
B	β	B	b	Ξ	ξ	C	c
Γ	γ	G	g	O	o	O	o
Δ	δ	D	d	Π	π	P	p
E	ε	E	e	P	ρ	R	r
Z	ζ	Z	z	Σ	σ	S	s
H	η	Y	y	T	τ	T	t
Θ	θ	H	h	Y	υ	U	u
I	ι	I	i	Φ	φ	F	f
K	κ	K	k	X	χ	X	x
Λ	λ	L	l	Ω	ω	W	w
M	μ	M	m				

Example: 1.04μm would be entered as:

1.04!m!m

Table 13.7.2

Mathematical Symbols

A	infinity (∞)	_	_
B	improper superset (⊇)	a	⇑
C	proportional (∝)	b	↓
D	union (∪)	c	section (§)
E	root extender (‾)	d	@
F	bell ()	e	double dagger (‡)
G	plus or minus (±)	f	|
H	less than or equal to (≤0)	g	improper subset (⊆)
I	greater than or equal to (≥)	h	\\
J	square root (√)	I	circle (O)
K	terminal sigma (ς)	j	partial derivative (∂)
L	integral sign (∫)	k	empty set (∅)
M	subset (⊂)	l	{
N	superset (⊃)	m	}
O	intersection (∩)	/	/
P	not (¬)	"	"
Q	clover leaf ()	#	#
R	angstrom (°)	′	′
S	gradient (∇)	+	+
T	times (x)	-	-
U	divide (÷)	*	*
V	identically equal (≡)	<	<
W	approximately equal (≈)	=	=
X	not equal (≠)	>	>
Y	←	^	^
Z	→	‘	‘

It was assumed that these characters would not need to be printed so they have been reserved for the above special use. Backslash can be used to cause characters to be overprinted. An "@" (at sign) terminates the last use of a brace. All characters in between are printed in script mode.

Examples:

A\{1@ Will print as A_1

SAM___ Will print as SAM

X}2@-Y Will print as X^2 - Y

Typing

e

or

x

in place of a file ID will cause the program to return to the beginning of the routine (section 13.1).

Typing

b

will begin building the vector and text files on the disc. As soon as these files are complete, specpr starts the plot daemon job and returns for more input. Thus while the plotter is plotting, you may do other tasks, including creating more plots.

CHAPTER 14

RADIATIVE TRANSFER ROUTINES

14.1 Introduction

The radiative transfer routines allow you to compute a reflectance spectrum given the optical constants of a material, compute reflectance spectra of mixtures of materials given their optical constants, invert a reflectance spectrum to absorption coefficient given the index of refraction spectrum, and other computations. All computations can be done at specific grain sizes and viewing geometries (including geometric albedo).

Because of the memory requirements, the radiative transfer routines have been split from the specpr program, and are now called "radtran." See the radtran user's manual for details.

NOTE: the radtran manual is currently in preparation.

CHAPTER 15

STANDALONE UTILITIES

15.1 Introduction

This chapter describes specpr support utilities. They are independent programs to do tasks related to specpr. Each utility has a Unix-style manual page. You may want to copy these manual pages and put them into your local list of user-available commands.

Spprint is a routine that prints the contents of a specpr file, similar to the "list" function in specpr (section 11 of the specpr manual), but the advantage in using it is you do not have to start a specpr process, assign a file, and then go to the list routines. It is useful for checking the contents of a specpr file you happen across. This routine is used to provide rapidly searchable lists of online specpr files at the U.S. Geological Survey, Denver Spectroscopy Laboratory. Spprint is described in section 15.2.

Spfeatures is a program that reads a data file created by specpr special function f46. Thus, spfeatures queries a spectral features database and allows a user to find spectra with certain user-defined features. Spfeatures is described in section 15.3.

The **data translation** routines are used to convert specpr data files to ascii and those ascii files back to specpr format data files.

15.2 Spprint

NAME
spprint - print a summary of a specpr format file

SYNOPSIS
spprint file

DESCRIPTION
This program uses the specpr listing routines to provide a listing to the standard output of the header and data, similar to the specpr list routines, Chapter 11 of the specpr manual.

FILES
"file" is a user-specified specpr file

AUTHOR
Matthew Klejwa

15.3 **spfeatures**

NAME

spfeatures - search spectral features data base

SYNOPSIS

```
spfeatures [-c range] [-w range] [-d range] [-e range]
           [-a range] [-F file] [-o file] [-h] [-v] [-b]
```

DESCRIPTION

This program searches the spectral features database selecting entries based on the band center, width, depth, asymmetry, or error on the depth of the feature. All numbers refer to microns in the database.

Options:

```
-c c1 c2 : Band Center --  Causes program to select
             all features with a band center within
             the bounds of c1 and c2.   c2 must be
             greater than or equal to c1.

-w w1 w2 : Band Width --  Causes program to select
             all features with a band width within
             the bounds of w1 and w2.  w2 must be
             greater than or equal to w1.

-d d1 d2 : Band Depth --  Causes program to select
             all features with a band depth within
             the bounds of d1 and d2.   d2 must be
             greater than or equal to d1.

-e e1 e2 : Error on band depth --  Causes program
            to select all features with an error
            on the band depth that is within the
            bounds of e1 and e2.   e2 must be
            greater than or equal to e1.

-a a1 a2 : Band Asymmetry --  Causes program to
            select all features with a band asymmetry
            within the bounds of a1 and a2.  a2
            must be greater than or equal to a1.

-F file  : Database file --  Causes program to use
            "file" as a database, rather than the
            default file.

-o file  : Output file -- Causes output to be written
            to "file", rather than standard output.
            "File" is outputted in binary form, and
            is not compatible with the "-b" option.
                    (-h and -v are ignored)

-h       : Header -- Causes the appropriate header
            to be included in the output.
```

(default is no header)

-v : Verbose -- Prints the entire title and the comments to each band. (default is a seventeen character name, and no comments)

-b : Bandanal format -- Causes program to use the output from Specpr Interactive Band Analysis routine, see section 9.8 of the specpr manual. This option is not compatible with the -o option, and causes a fatal error. (-v is ignored, and -h header is changed)

When more than one option is given, the search is by "and" as compared to an "or" search. If an "or" search is desired, run the program separately, and append the output to a file.

EXAMPLES

Find all features with a center between 1.38 and 1.41 microns:

```
spfeatures -c 1.38 1.41
```

Find all features between 1.38 and 1.41 microns with a full width at half maximum less than 0.01 micron:

```
spfeatures -c 1.38 1.41 -w 0 0.01
```

Find all features between 1.38 and 1.41 microns with full width at half maximum less than 0.01 micron, and belonging to Alunite:

```
spfeatures -c 1.38 1.41 -w 0 0.01 | grep Alunite
```

Find all features between 1.38 and 1.41 microns with full width at half maximum greater than 0.01 micron, and all features between 2.1 and 2.4 microns that belong to Alunite, and put the output in a file called temp1 with a header:

```
spfeatures -h -c 1.38 1.41 -w 0.01 9999 | grep Alunite > temp1
spfeatures -c 2.1 2.4 | grep Alunite >> temp1
```

FILES

/d1/samples/features/FWHM_46	binary features database
/d1/samples/features/FWHM	bandanl features database
*	user's features database

NOTE: the above files are be site dependent.

NOTES

1) There will be no output if there are no options (null default).

AUTHORS

Noel Gorelick, Roger Clark, Matthew Klejwa

15.4 **Data Translation**

15.4.1 sptoascii

NAME

sptoascii - convert a specpr format file to ascii data

SYNOPSIS

sptoascii file

DESCRIPTION

This program converts a specpr file to ascii so that it may be easily transferred to a different machine and retranslated to the local machine format, thus avoiding the writing of special floating point number conversion routines.

FILES

"file" user-specified specpr file

AUTHOR

Robert Burtzlaff

15.4.2 **asciitosp**

NAME
asciitosp - convert an ascii data file to specpr format

SYNOPSIS
asciitosp file < asciifile

DESCRIPTION
This program converts an ascii file to specpr format, version 2 (version 3 specpr has the same format as version 2). The ascii format must be the same as that generated by sptoascii or oldsptoascii.

FILES
"file" user-specified specpr file

AUTHOR
Robert Burtzlaff

15.4.3 **oldsptoascii**

NAME

oldsptoascii - convert a specpr version 1 format file on a DEC VAX to ascii data

SYNOPSIS

oldsptoascii file

DESCRIPTION

This program converts a specpr version 1 file to ascii so that it may be easily transferred to a different machine (or the same machine) and retranslated to the local machine format, thus avoiding the writing of special floating point number conversion routines.

This program MUST be compiled and run on a VAX Unix (or VAX Eunice) system.

The output is written to standard out, which may be redirected to another file.

FILES

"file" user-specified specpr file

AUTHOR

Robert Burtzlaff

15.4.4 **cgastosp**

NAME

cgastosp - convert an ascii data file from the NASA Infrared Telescope Facility, Cooled Grating Array Spectrometer (CGAS) to specpr format

SYNOPSIS

cgastosp file < asciifile

DESCRIPTION

This program converts an ascii file made at the University of Hawaii, from the NASA Infrared Telescope Facility, Cooled Grating Array Spectrometer (CGAS) to specpr format, version 2 (version 3 specpr has the same format as version 2).

FILES

"file" user-specified specpr file

AUTHOR

Roger N. Clark

APPENDIX A

SPECPR STANDARD FORMAT OF DATA FILES

SPECtrum Processing Routines
Data Format
3/4/88

Variable	Description	Length (bytes)
icflag	32 one bit flags: (low bit = 0, high bit = 31) bit 00 continuation data flag =0 first record of a spectrum consists of: header then 256 data channels =1 continuation data record consisting of: bit flags followed by 1532 bytes of real data (bit 1=0) (383 channels) or 1532 bytes of text (bit 1=1). A maximum of 12 continuation records are allowed for a total of 4852 channels (limited by arrays of 4864) or 19860 characters of text (bit 1=1). bit 01 text/data flag: =0 the data in the array "data" is data =1 the data in the array "data" is ascii text as is most of the header info. bit 02 flag to indicate whether or not the data for the error bar (1 sigma standard deviation of the mean) is in the next record set. =0: no errors, =1: errors in next record set. bit 03 RA, Dec / Long., Lat flag: =0 the array "ira" and "idec" corresponds to the right ascension and declination of an astronomical object. =1 the array "ira" and "idec" correspond to the longitude and latitude of a spot on a planetary surface. bit 04 variable iscta universal time/civil time flag =0 cta is civil time =1 cta is universal time bit 05 variable isctb universal time/civil time flag =0 ctb is civil time =1 ctb is universal time bit 06 unused bit 07 unused bit 08 unused bit 09 unused bit 10 unused bit 11 unused bit 12 unused bit 13 unused bit 14 unused bit 15 unused bit 16 unused bit 17 unused	4

	bit 18 unused	
	bit 19 unused	
	bit 20 unused	
	bit 21 unused	
	bit 22 unused	
	bit 23 unused	
	bit 24 unused	
	bit 25 unused	
	bit 26 unused	
	bit 27 unused	
	bit 28 unused	
	bit 29 unused	
	bit 30 unused	
	bit 31 unused	
	bit 32 unused	
**********	case 1: bit 00 = 0, bit 01 = 0 **********	
icflag	Bit flags: 32 bits, see above.	4
ititl	40 character title which describes the data (ascii, character*40).	40
usernm	The name of the user that created the data record (ascii, character*8).	8
iscta	Civil or Universal time when data was last processed (integer*4 in scaled seconds). Scaled by 24000. See flag #04.	4
isctb	Civil or Universal time at the start of the spectral run (integer*4 in scaled seconds). Scaled by 24000. See flag #05.	4
jdatea	Date when data was last processed Stored as integer*4 Julian Day number *10	4
jdateb	Date when the spectral run began Stored as integer*4 Julian Day number *10	4
istb	Sidereal time when the spectral run started. (integer*4 in scaled seconds). Scaled by 24000. See flag #05.	4
isra	Right ascension coordinates of an astronomical object, or longitude on a planetary surface (integer*4 numbers in seconds *1000) (RA in RA seconds, Longitude in arc-seconds) See flag #06.	4
isdec	Declination coordinates of an astronomical object, or latitude on a planetary surface (integer*4 number in arc-seconds *1000). See flag #06.	4

itchan	Total number of channels in the spectrum (integer*4 value from 1 to 4852)	4
irmas	The equivalent atmospheric thickness through which the observation was obtained (=1.0 overhead scaled: airmass*1000; integer*4).	4
revs	The number of independent spectral scans which were added to make the spectrum (integer*4 number).	4
iband(2)	The channel numbers which define the band normalization (scaling to unity). (integers*4)	8
irwav	The record number within the file where the wavelengths are found (integer*4).	4
irespt	The record pointer to where the resolution can be found (or horizontal error bar) (integer*4).	4
irecno	The record number within the file where the data is located (integer*4 number).	4
itpntr	Text data record pointer. This pointer points to a data record where additional text describing the data may be found. (32 bit integer)	4
ihist	The program automatic 60 character history. (ascii, character*60)	60
mhist	Manual history (4 lines of 74 characters each. Program automatic for large history requirements (ascii, character*296).	296
nruns	The number of independent spectral runs which were summed or averaged to make this spectrum (integer*4).	4
siangl	The angle of incidence of illuminating radiation (Integer*4 number, in arc-seconds*6000). (90 degrees=1944000000; -90 deg <= angle <= 90 deg) integrating sphere = 2000000000 Geometric albedo = 2000000001	4
seangl	The angle of emission of illuminating radiation (Integer*4 number, in arc-seconds*6000). (90 degrees=1944000000; -90 deg <= angle <= 90 deg) integrating sphere = 2000000000 Geometric albedo = 2000000001	4
sphase	The phase angle between iangl and eangl (Integer*4 number, in arc-seconds*1500). (180 degrees=972000000; -180 deg <= phase <= 180 deg)	4

	integrating sphere = 2000000000	
iwtrns	Weighted number of runs (the number of runs of the spectrum with the minimum runs which was used in processing this spectrum, integer*4).	4
itimch	The time observed in the sample beam for each half chop in milliseconds (for chopping spectrometers only). (integer*4)	4
xnrm	The band normalization factor. For data scaled to 1.0, multiply by this number to recover photometric level (32 bit real number).	4
scatim	The time it takes to make one scan of the entire spectrum in seconds (32 bit real number).	4
timint	Total integration time (usually=scatime * nruns) (32 bit real number).	4
tempd	Temperature in degrees Kelvin (32 bit real number).	4
data(256)	The spectral data (256 channels of 32 bit real data numbers).	1024
	case 1: total (bytes):	1536
**********	case 2: bit 00 = 1, bit 01 = 0 **********	
icflag	Flags: see case 1	4
cdata(383)	The continuation of the data values (383 channels of 32 bit real numbers).	1532
	case 2: total (bytes):	1536
**********	case 3: bit 00 = 0, bit 01 = 1 **********	
icflag	Flags: see case 1	4
ititle	40 character title which describes the data (ascii, character*40).	40
usernm	The name of the user who created the data record (ascii, character*8).	8
itxtpt	Text data record pointer. This pointer points to a data record where additional text may be may be found. (32 bit integer)	4

```
itxtch          The number of text characters (maximum= 19860).       4

itext           1476 characters of text.  Text has embedded        1476
                newlines so the number of lines available is
                limited only by the number of characters available.
-------------------------------------------------------------------------
        case 3: total (bytes):                                     1536
=========================================================================

*************** case 4: bit 00 = 1, bit 01 = 1 ******************

icflag          Flags: see case 1                                     4

tdata           1532 characters of text.                           1532
-------------------------------------------------------------------------
        case 4: total (bytes):                                     1536
=========================================================================
```

APPENDIX B

HOW TO OBTAIN SPECPR

HOW TO OBTAIN SPECPR

The specpr software, other software and spectral libraries are published as a series of USGS Open File Reports. The hardcopy spectral library, and users manuals are available from:

USGS/Dept. of the Interior
Books and Open-File Reports Section
U.S. Geological Survey
Box 25425, Federal Center
Denver, CO 80225

USGS Books, Open File Reports and Maps:
Phone number: (303) 236-7476

The relevant Open-File documents are:

93-595 SPECtrum Processing Routines Users Manual Version 3 (Program SPECPR), 202 pages, (this document). (Clark, 1993)

93-592 Spectral Library paper version, 1326 pages. (Clark et al, 1993). This document will also be published as a USGS Bulletin. See below for digital data.

93-594 Spsearch Users Manual, 23 pages. (Gorelick and Clark, 1993)

93-593 Spview manual, approximately 15 pages. (Includes the digital spectral library and spectral library reader software on 3.5-inch floppy disks for IBM-PC compatible computers.) (Livo et al., 1993)

The specpr software, related programs, the digital data for the USGS spectral library and digital copies of the above manuals are available via anonymous ftp on the internet:

```
ftp speclab.cr.usgs.gov

login as anonymous
password is your userid@machine

get README (Note: examine this file for what is available.)

cd specpr
binary     (set binary mode for copy)
mget *
```

In order to get the USGS Digital Spectral Library:
(note this is cd ../pub/spectral.library from the specpr directory)

```
cd pub/spectral.library

get README
```

Follow instructions in the README file for obtaining the data. The pub/spectral.library directory will contain all the different versions listed in Table 5, as well as additional ones as they become available (again see the README file for details).

Similarly, obtain the spsearch software in the pub/specpr and pub/spsearch directories. The specpr distribution also includes an independent Fortran program, spprint, that reads a specpr format file and prints titles. For independent subroutines in C that read a specpr file, see the README file.

After you have retrieved specpr, or the library, please send mail to rclark@speclab.cr.usgs.gov with your name, address, phone number and email address. We will put you on a mailing list for future announcements and updates.

Alternatively, contact the author at his address, or send electronic (internet) mail to:

```
rclark@usgs.gov
```

if you have questions.