National
Optical
Astronomy
Observatories

Kitt Peak National Observatory

IRAF REDUCTION OF COUDE CCD DATA

edited by
Daryl Willmarth

Kitt Peak National Observatory
National Optical Astronomy Observatories††
February 15, 1990

Contributors

Jeannette Barnes	Rm.86	Ext.381
Phillip Massey	Rm.147	Ext.271
Caty Pilachowski	Rm.37	Ext.289
Robert Seaman	Rm.98	Ext.248
Daryl Willmarth	Rm.10	Ext.333

††Operated by the Association of Universities for Research in Astronomy, Inc. under cooperative agreement with the National Science Foundation.

Table of Contents

Table of Figures

1. Introduction

This manual is intended to describe the general procedures available to reduce "coude-like" spectroscopic CCD data with IRAF version 2.8. This can include most spectra of single objects with a short or long slit, or data taken with the coude fiber image scrambler. These procedures will include removal of the "instrumental signature" (calibrations) and extraction to one-dimensional spectra. Finally, measurements such as equivalent widths and radial velocities will be discussed. A flowchart summary of these procedures is shown in figure 1. A more complete discussion of the IRAF command language can be found in ***A User's Introduction to the IRAF Command Language***, available in the Computer support documentation room, or in Volume 1A of the 4 blue binders that compose the IRAF documentation.

Observers are encouraged to complete as much of the reductions on the mountain Sun computers as possible. Features such as easy transfer of data from CCD computer to Sun computer, large amount of user disk space, and being able to see the final data quality make this desirable. If this is not possible, one can continue the reductions with the identical software downtown or at the home institution. The flowchart shown in figure 1 may be used to determine subsequent operations. In you have used the mountain CCD computer reduction program (SPECPROC), start the reductions either at "Telluric Line Divisions" (§3.9) or "Wavelength Calibrations" (§3.10).

2. Calibration Frames

2.1. Philosophy of Calibrations

Most of the calibration data is intended to remove "additive" effects: the electronic bias or "pedestal" level (measured from the overscan region), the preflash level (measured from bias frames), and if necessary, the dark current (from dark frames). The flat-field data (projector flats) will remove the multiplicative gain and illumination variations across the chip. Figure 1 shows a typical data reduction flowchart which might be applicable to most kinds of coude CCD data.

When the data frames are obtained at the telescope, the output signal is "biased" by adding a pedestal level of several hundred ADUs (Analog-to-Digital Units). We need to determine this bias level for each frame individually, as it is not completely stabilized, and will vary slightly (≃5-30 ADUs), mainly with dewar position and temperature. Furthermore, the bias level is usually a slight function of position on the chip, varying primarily along columns. The bias level can be removed to first-order by using the data in the *overscan region*, the (typically) 32 columns at the right edge of the data frames (see figure 2). We will average the data over all the columns in the overscan region (except perhaps the first and last columns of the overscan), and fit a constant to the average of the columns (i.e., average in the "x" direction within the overscan and fit a constant to the "y" values). This value will be subtracted from each column of the data frame. At this point we will chop off the overscan region, and keep only the part of the image containing useful data. This latter step usually trims off not only the overscan region, but the first and last couple rows and columns of the data.

If the CCD was pre-flashed with light before each exposure, there will still be a non-zero amount of counts superimposed on each image ("fat zero"). This extra signal is also an additive amount, and needs to be subtracted from the data. In addition, there may be column-to-column variations in the structure of the bias level which would not have been removed by the above procedure. To remove both the pre-flash (if any) and the residual variation in the bias level (if any) we will make use of frames that were obtained with zero integration time. These are referred to in IRAF as "zero" frames but are called "bias" frames in KPNO and CTIO lingo. We need to average many of these, process the average as described above, and subtract this frame from all the other frames.

"Dark current" is also additive. If necessary, one can remove the dark current to first order by taking "dark" exposures, processing these frames as above, and then scaling to the exposure time of the object frames before subtraction. However, experience has shown that the dark current seldom scales linearly, so care must be taken. Furthermore, at least 3 dark frames are needed in order to remove radiation events ("cosmic rays"), and unless one has taken a large number of darks to average, the signal-to-noise of the objects may be decreased. The bottom line is unless you really need to remove the dark current, don't bother.

The next step in removing the instrumental signature is to flat-field the data. This will remove the pixel-to-pixel gain variations, and in the case of long-slit spectroscopy, the larger scale spatial variations. The flat-field frame may also introduce (or remove!) a tilt in the spectrum due to the color temperature of the lamp.

The type and number of required calibration frames will depend on the instrumental configuration and the nature of your scientific program. It is difficult or impossible to reproduce these data images after your run, so be sure to take sufficient numbers of calibration frames while at the telescope. The goal is to not let the quality of the calibration data degrade the signal-to-noise of your object data in any significant way. If your data is in the regime where the read-noise of the chip is the dominant source of noise on your program objects, then subtracting a single bias frame from your data would increase the noise by $\sqrt{2}$. However if you instead use the average of 25 bias frames, the noise will be increased by only 10%. At the other extreme where the signal is very high compared to read-noise, or if you have high sky background on low dispersion spectra so that read-noise is again immaterial, then the signal-to-noise will be little affected by whether you have only a few bias frames. In this regime however, the quality of your flat-fielding is all important if you want to get the most out of your data.

2.2. Calibration Frames: Bias, Dark, Flat, Comparison

The following paragraphs describe the types of calibration images you may need, and provides some guide as to how many of each kind are necessary:

bias: These are zero second (just the read-out time) integrations with the same preflash (if any) and on-chip summation factors that are used for program object spectra. Take 20 - 30 bias frames *per night*. These may be taken during dinner or twilight.

dark: These are long integrations taken with the shutter closed. If your longest exposure is over 15 minutes you may want to take an equal length dark frame, subtract a bias frame from it, and decide if the residual dark current will be significant. The dark current is sometimes high if the chip has been exposed to too much light or if power has been interrupted to a cold chip. It is useful to at least take three darks, median average them, and look for hot pixels or bad regions on the CCD. Applications where dark current *will* matter are long-slit spectroscopy and surface brightness studies - cases where the background is not removed locally. If you do find that dark current should be subtracted, take at least 3 and preferably 5 to 10 dark frames during your run, each with an integration time equal to your longest exposure.

flat: At a minimum, flat-field exposures are used to remove pixel-to-pixel variations across the chip. You will want to expose the flats to get sufficient counts to not degrade the signal-to-noise of the final images, but be careful not to exceed the linearity limit in any single exposure. These limits are about 8000 ADUs for TI CCDs, 10,000 ADUs for RCA CCDs, and up to saturation for TEK CCDs. Please note however, each chip has its own peculiarities and on-chip summation will affect these values. It is better for most purposes to obtain 20 flats at the level of your most intense object spectrum than to try to obtain all of the photons in 1 exposure. Use a decker wide enough to provide even illumination across the region where the spectrum falls. It may be useful to obtain at least two sets of flats during the night as chip motion does occur.

comparison: Obtain comparison lamp exposures as needed for your scientific program. The position of the CCD in the dewar does change slowly during the night as the liquid nitrogen evaporates and can be as large as 0.5 pixel. For radial velocity measurements this can mean errors as large as a few km-s^{-1}. A comparison exposure taken once an hour will provide good compensation for this motion. Otherwise, a single comparison per night per wavelength region may suffice.

telluric spectra: If you need to remove telluric lines from your spectra, be sure to obtain several exposures of a rapidly rotating B star.

If you are not sure what data will be needed, discuss the problem thoroughly with your instrument assistant. He or she should also advise you how to obtain these images, and how to do preliminary reductions on them on the mountain. You will have the CCD computer and the mountain Sun computer almost completely to yourself, so some time and effort may be saved by accomplishing as much as possible of the reductions on the mountain. Note that an automatic reduction program resides on the coude CCD computer which will calibrate and extract your spectra. The ***Coude Spectrograph CCD Camera Manual*** should be consulted for details of this program. If this procedure will satisfy your needs you may proceed to sections of this manual beyond spectral extraction for further analysis. Otherwise, the CCD computer and/or the Sun/IRAF computer may be used for reductions and analysis.

Raw data, including calibrations and objects, should be written to magnetic tape for future processing, and the reduced data should be written to a separate tape. Note that NOAO tapes are to be used only for temporary backup (6 months) and as a transport for taking data to the downtown computers; any tapes needed for transporting data elswhere must be supplied by the observer.

3. IRAF Procedures

3.1. Introduction

It will be very useful to have reviewed ***A User's Introduction to the IRAF Command Language*** by Shames and Tody before starting as well as any manuals pertaining to the particular system you will be using.

3.2. Getting Into IRAF

There are three main avenues of IRAF use at NOAO, all very similar but differing mainly in data I/O procedures: the mountain Sun, the downtown Suns and the downtown VAXs. Visitors will need to obtain an account to use the downtown computers. Please inquire at the computer support office (room 100) or see an IRAF support person.

3.2.1. Mountain Suns

A listing of logins should be posted at each Sun station and a manual of procedures: ***IRAF on the Sun Computer***, by Bret Goodrich, available at each site. This manual will describe logging in, data transfer between CCD computer and Sun, and the use of IRAF in the *SunView* windows environment. To login to a remote VT100 terminal connected to a Sun (such as at the coudé feed), type <**return**> and respond with the appropriate account name (e.g. **feed**). IRAF will be automatically loaded in a few seconds. Note that when transfering pictures with the **getpix** command (a UNIX routine), the pixel files (∗.pix) will now be placed in the appropriate pixel directory instead of the user directory as was true of previous versions of **getpix**.

3.2.2. Downtown Suns

The downtown Sun computers operate almost identically to the mountain Suns except I/O must be through the tape drives located on Orion or Tucana. These drives are referred to as **orion!mta** (or **orion!mtb**) or **tucana!mta** in all tape commands. The three diskless Sun nodes, Pictor, Pyxis, and Phoenix are suitable for one-dimensional spectral analysis, the pixel files being stored on Orion "tmp" disks. See ***A Quick Look at Sun/IRAF on the Tucson Sun Network*** for more information. Orion may also be accessed through a VT100 terminal as class 14.

3.2.3. Downtown VAXs

While 4 VAX computers are available for downtown IRAF reductions, only Draco (class 6) has sufficient speed and disk space to handle IRAF reductions in a reasonable manner. To login, type **RETURN** and when it responds with ENTER CLASS, type **6**, then type **RETURN** a couple of times. You should then be prompted for your login or username, and your password. All visitor accounts have passwords - check with Jeannette or Rob if you are not sure what your password is. Visitor accounts on Draco (class 6) run IRAF from a subdirectory, so change to the appropriate directory: **sd .iraf**. Now type **cl** to load IRAF.

3.3. Tasks and Task Parameters

All IRAF tasks are grouped into collections of similar tasks called packages. Three packages are loaded initially - the SYSTEM, LISTS and LANGUAGE packages. Type **?system** to look through the available tasks in that package. Type **help system** for a listing of one-liner definitions for the tasks in that package. To load another package, just type its name. For example the **rfits** task (read FITS tape) is in the DATAIO package, loaded by typing **dataio**. To find out which packages are currently loaded, type **package**. To list the tasks available in the above example, type **?dataio**. The details of individual tasks can be listed on the terminal by typing **help *taskname***.

Each IRAF task has several parameters associated with it that control the manner in which it executes a task. A quick listing of the task parameters can be obtained on the screen with the command **lpar *taskname***. A parameter shown in parentheses is a "hidden" parameter. Its default value is used automatically unless the value is explicitly altered, either before the task is executed, or on the command line for the task. Parameters not in parentheses must be specified each time a task is executed. Before executing a task, one should list the parameters, and set them to the desired values. These two operations can be performed together with the parameter editing task **epar**. For example

```
epar flatten
```

lists the task parameters and enables the user to move down or up (arrow keys) the parameter list typing in new values followed by a **return**. Exit the "epar" mode with a **CTRL/Z**. These new parameters are "learned" by IRAF and are now the new default values. Hidden parameter values may also be changed by typing the task name, parameter and new value as in the following example:

```
flatten.minflat=3000.
```

To reset all the parameters of a task to their original values, use the **unlearn** task as in

```
unlearn flatten
lpar flatten
```

In the pages that follow, the parameters for many IRAF tasks are listed and discussed. Since some tasks are used for many different functions, not all parameters are relevant to the task at hand.

3.4. Reading CCD Data Into IRAF

While CCD data can be transferred directly over the CCD/Sun link on the mountain, downtown the data must be read in from FITS or CAMERA format tapes. It is recommended that each night's data be kept in a separate subdirectory. For example to create a directory and then move to it:

```
mkdir nite1
cd nite1
```

Table 1 lists the available tape drives for each downtown computer, their IRAF designations, densities and location.

Table 1. Tape Drives			
computer	IRAF drive name	density	location
Draco	mta	1600/6250	room 101
	mtb	1600/6250	"
	mtc	800/1600	"
	mtd	800/1600	"
	mtg	800/1600	"
	mth	1600/6250	"
Pictor	orion!mta	1600/6250	room 101
Pyxis	orion!mtb	1600/6250	"
Phoenix	tucana!mta	1600/6250	B-50
Argo			
Orion	mta	1600/6250	room 101
	mtb	1600/6250	"
	tucana!mta	1600/6250	B-50

If your data tape is in FITS format, type **dataio** to load the **rfits** task. Now examine the **rfits** parameters using **epar rfits** and set them to agree with figure 3. Allocate the desired tape drive with for example, **allocate mta**, mount the tape (remove write-ring!), and read in the data:

```
rfits mta 1-999 nite1 data=r >n1list &
```

This will read in all of the pictures on the tape and name them nite1001, nite1002, nite1003,..... The **&** allows the terminal to be used for other tasks while the tape is being read, and is a generally useful feature. Reading in the data as real numbers ("data=r") will assure there is sufficient disk space for picture arithmetic operations which create real data as output. The *root* name, nite1, may be any useful combination. Another valuable feature is the **rfits** parameter ***oldirafname*** which will restore the picture name to what it was when last stored on the tape. This can be valuable for continued reductions on partially reduced data from the Sun, as record keeping is greatly simplified. Yet another useful root name is of the type nite1.0 which creates images of the type nite1.0001, nite1.0002,..... This 4 digit numerical extension is a useful format for some of the tasks, and is recommended if your spectra are already extracted with the mountain CCD computer. If your data is in CAMERA format, load **noao** and **mtlocal**, set the **rcamera** task parameters similar to **rfits**, and read in the data as above.

If your data takes up more than one tape, **rewind mta**, exchange tapes, and again execute **rfits**, but now inserting the current picture number for the ***offset*** parameter. When finished with the tape drive, type **deallocate mta** and **bye** to exit the DATAIO package.

3.5. Widely Used IRAF Commands

When we actually reduce our CCD data we will use the **ccdproc** task within the CCDRED package, but beforehand it may be useful to examine some of the data to become familiar with its appearance with some generally useful IRAF tasks. First load the necessary package by typing **noao**.

imheader: The image header may be listed as in

```
imhead star001
```

which results in

```
star001.imh[132,800][real]:hd123456 11apr89
```

The header gives the size of the image, its pixel type and its title. Note that IRAF task names, as well as package names can be abbreviated as long as the command is not ambiguous:

```
imhe star001 l+
```

will list all of the information stored in the image header, not just the title. The pointer to the pixel file is also listed with the "long+" parameter. To list the titles of all images with a particular root name, type

```
imhe spec*.imh
```

To print on paper the titles of images for later reference, just pipe the output of **imheader** into the **lprint** task. Type **show printer** to check the default printer device. Type **devices** to see what printers are available. The printer may be changed with **set printer = lw1** for example.

```
imhe spec*.imh | lprint &
```

imcopy: This command will make a new copy of an image, which is good for safety, especially when you are new to the system and are testing things out.

```
imcopy star001 temp
```

imdelete: This command will remove pictures from your directory. The data pixels of your images are not actually stored in your directory (just the header information is stored in the .imh file in your directory), and **imdelete** removes not only the header file from your directory, but also the pixel file stored on a scratch disk.

```
imdel temp
```

Of course, you could (if you wanted to) delete all pictures with a given root name with the command:

```
imdel star*.imh
```

or delete several pictures with one command (note the use of commas and no blanks to list a string of images):

```
imdel star001,temp,temp1
```

Text files can be deleted with the command **delete *filename***.

implot: This command will plot the middle line (or a specified line) across an image:

```
implot star001
```

It offers many interactive cursor commands for examining a plot; once a spectrum is plotted, type **?** or **:.help** for lists of cursor commands. For example, to plot a column of the image, set the vertical cursor on the column you want and type a **c**. To plot another line of the image, again set the vertical cursor at the desired line number on the horizontal axis of the plot, and type an **l**. One can also use **:ln** or **:cn** where n is the desired line or column number, to plot a new line or column. To exit, just type a **q**.

dir: List all the files in your current directory.

CTRL C: Abort a task. Use **flpr** (**fl**ush the **pr**ocess cache) afterward to clean up loose ends.

edit: Text files can be edited with the command **edit *filename***. On NOAO computers, the edit command invokes the ***vi*** editor. A summary of ***vi*** commands is available from the computer support office. If you wish to invoke some other editor, type

```
set editor=edt
```

for example. You may also change the line defining the default editor in the *login.cl* file and remove the "#" sign, so the default will be changed the next time you log on. It is preferable however, to make such changes in the *loginuser.cl* file since the *loginuser.cl* file will not be deleted with a **mkiraf** command, while the *login.cl* file will be deleted.

files: The system command **files** can be used to create a file list for use in processing a group of similar files:

```
files star*.imh >stars
and
files more*.imh >>stars
```

where ">" creates a new file, and ">>" appends to an existing file.

@file: The text file created with the **files** command or through other text editing is just a list of pictures which can be used as the object of a task:

```
imhead @stars
```

will list short headers for all the pictures listed in the file *stars*.

names: Creates or adds to an existing file specified files within a given file rootname:

```
names spec 1-2,5,10-20 >>stars
```

will add the files spec.0001, spec.0002, spec.0005, spec.0010....spec.0020 to the file *stars*.

%: The string substitution operator, %, can be used to substitute one string for another from a filename:

```
files star%A%B%*.imh
```

will find all occurences of files "starA*.imh" and replace the filenames with "starB*.imh" in the output list.

ehistory: Typing this command will enter you in the command history file editor. Previous commands can be accessed with the up and down arrow keys, and the left and right arrow keys used to position the prompt to add or delete to the left of the prompt.

A final note about the use of **&**. If you log off a Unix system after submitting a background job with the **&**, the job will keep running. On a VMS system, however, the same job will terminate if you log off the system. Thus, in VMS, you must submit your background job as a batch job if you plan to log off before its completion. This is done with **& batch**.

3.6. Combining Calibrations

As discussed earlier, spectroscopic data may be reduced completely on the CCD computer, partially reduced on the CCD computer and finished with IRAF, or reduced solely with IRAF. It is convenient and time-saving to at least average or median the bias frames and flat-field frames, and to debias and trim all pictures with the CCD computer. On the other hand, the IRAF tasks for these procedures are very fast and can handle larger groups of pictures for median filtering and other "cleaning" algorithims. You may wish to discuss these matters with your instrument assistant or a data reduction specialist to arrive at the most efficient plan. ***A User's Guide to CCD Reductions with IRAF*** by Phillip Massey has discussions of these steps.

For initial CCD reductions with IRAF, we will use the **ccdproc** task within the CCDRED package, and this task will depend upon the image headers to perform the proper operations. Therefore our first task must be setting up the translation table between the image header and the various things the tasks in CCDRED need to know (see ***User's Guide to the CCDRED Package*** by Francisco Valdes). These things include what a "zero" frame is called, what distinguishes a flat-field at one wavelength from another or at different times of the night, and so on. The command for setting up this translation file is **setinstrument** in the CCDRED package. Load **noao**, **imred**, and **ccdred**, and then type **setinstrument**. If you type a **?** you will get the list shown in figure 4. The proper instrument selection will probably be "coude", although "fibers" may be appropriate for the fiber image scrambler.

After entering the instrument choice you will be in the parameter editor for the package parameters for CCDRED, as shown in figure 5. Check that ***pixelty***=real, and set any other parameters as you wish. See the help page for CCDRED for a description of all parameters. When you are done, exit with **CTRL-Z**, which invokes the parameter editor for **ccdproc** which can be used to set the parameters as in figure 6. Exit again with **CTRL-Z**.

As the CCDRED package is "header driven", we must insure that the picture headers contain sufficient information to reduce the spectra in our intended manner. The pictures usually leave the mountain with the correct data type in their headers, but if spectra of varying wavelengths, or spectra grouped in time intervals are to be handled properly, we must edit the headers. This is easily done with **ccdhedit**: for example,

ccdhedit @stars filters A

will add the keyword FILTERS='A' to all the pictures in the list "stars" (see §3.5). Adding this keyword to all flats, comps, and objects of a particular data set will cause **ccdproc** to recognize this distinction in the reductions.

We are now ready to average the bias and flat-field pictures to create a combined picture of each group that has radiation events removed. There are various options for combining which are discussed in the help pages for **combine**. Type

epar zerocombine

and set the ***combine*** option as desired (see Figure 7). The "avsigclip" or "median" options do a good job of excluding radiation events. Now type

zerocombine nite1*.imh

to combine the bias frames which will result in output similar to figure 7.

Next, we will do the same thing with our projector flats (pflat) after typing

epar flatcombine
flatcombine nite1*.imh

and end up with the result similar to that shown in figure 8. **Flatcombine** produces images with names like *FlatA* or *FlatB* according to the FILTER parameter in the picture headers. Note that both of these examples keyed on the parameter value for ***ccdtype***.

You may at this time wish to examine the combined images with **implot** and then delete the individual bias and pflat frames if the results look satisfactory. An easy and safe way to do the picture deletion is by putting the names of the individual calibration frames into a @file with a command like

ccdlist nite1*.imh names+ ccdtype=zero > tempfile

followed by

ccdlist nite1*.imh names+ ccdtype=flat >> tempfile.

If the files have been named according to data type, one may use

files bias*.imh > tempfile

followed by

files flat*.imh >> tempfile.

The point is, there are many ways to do these things depending upon the initial file naming and how clever one wants to be. It is good practice to do a **type tempfile** or **ed tempfile** to make sure there is nothing in it you want to save! Now one can type

imdel @tempfile

followed by

del tempfile.

3.7. CCDPROC - Debiasing, Trimming, Flat-fielding

Up to this point we have set up the instrument translation file with **setinstrument**, identified data groups with **ccdhedit**, and combined bias and flat frames with **zerocombine** and **flatcombine**. Before we actually run **ccdproc**, we need to examine the flat-field frames and a typical object frame to identify the areas containing data and the parts of the picture to be trimmed off. Let us first examine a flat: **implot flatA** plots a middle line of the image. The s/n can be increased by averaging a few lines with the **a** cursor command (figure 9). The profile can also be expanded with the **e** cursor command (type **?** for a complete list) and positions of the cursors noted by typing **C** (twice). We need to determine where the edges of the flat-field illumination are and if the illumination is uniform across the columns where the object spectra lie. The latter is important to preserve the relative signal-to-noise of the flat-fielded pixels. It is for this reason a larger decker should be used for pflats than the seeing disk at the slit. For fiber observations of course the flat and object spectra will be the same width. See Appendix A for a discussion of this type of data. If the flat-field spectra are not uniform across the area under an object spectrum, you may wish to "flatten" the flat-field frame with **flat1d** to normalize each column (or line) of the spectrum to 1 (see below).

Now type the **o** (overplot) cursor command followed by

:i star001 †
:l 300 400

to overplot the cross-section of an object spectrum with the cross-section of the flat frame. This will show us the relationship of the stellar spectrum to the flat-field frame. Generally, the stellar spectrum will lie within the bounds of the flat-field and we can trim all frames to the bounds of the flat-field profile.

Also examine in one of these same pictures the overscan bias area for edge defects and note the useable columns. Then plot some columns and check for bad rows which are generally the first and last one or two showing up as "hot".

We are now ready to edit and run **ccdproc** to subtract the bias level and trim the images of unuseable pixels. Type **epar ccdproc** and set the parameters as in figure 6, but using the "biassec" and "trimsec" values determined from **implot** above. These parameters are designated as follows:

[102:130,2:799] = [first overscan col:last overscan col,first overscan line:last overscan line]
[40:60,2:799] = [first data col:last data col,first data line:last data line]

Now type

ccdproc nite1*.imh

or whatever you need to designate the data-set to be processed. This pass of **ccdproc** will first process the average bias and flat frames that you designated in the **ccdproc** parameter set, and then debias, trim, and flat-field the object spectra.

If the profile of the flat-field image is not fairly flat across the area occupied by the object spectrum, you may wish to normalize each column (or row) of the flat before flat-fielding your objects. Otherwise, the columns (or rows) of the object data will undergo changes in relative intensity, affecting the signal to noise of the extracted spectrum. The flat image should first be processed with **ccdproc** to debias and trim the image. Then type **epar flat1d** (after loading the

†The names of image files used here are chosen to insure clarity in types of files; in practice, "star001" might be named "nite1020" for example, depending on its order on the data tape.

GENERIC package) and set the parameters as in figure 10, with ***minflat*** equal to the intensity value of the flat corresponding to the column or row outside of which no object column or row will fall (figure 9).

flat1d FlatA NflatA

will now modify the flat-field image by scaling the average value of each column (or line) to 1 and create a new picture, NflatA. All columns or rows below the value of ***minflat*** will be set to 1. The header of NflatA should contain a keyword CCDMEAN (added during **flatcombine**) which will be the scale factor for subsequent flat-fielding. If you wish to have the intensities of your flat-fielded spectra resemble those of the raw data, this keyword should be set to 1.:

ccdhedit NflatA ccdmean 1.

Ccdproc may now be run to debias, trim and flat-field the object spectra as before:

ccdproc nite1*.imh flat=NflatA

If more than one flat will be used (flat-fields combined with the ***subsets*** option) the parameter ***flat*** may be set equal to "Nflat*" or equal to a file list of flats, "@flats". The comparison spectra should also be processed similarly by setting *ccdtype* = comp in the above reductions.

There may be cases where one wants to **aptrace** the spectra before flat-fielding, in which case turn off this option while running **ccdproc**:

ccdproc nite1*.imh flatcor=no

3.8. Extraction of One-Dimensional Spectra

You now have flat-fielded two-dimensional images which are ready for extraction to one-dimensional spectra. If not still in the IMRED package, type **imred** and **coude**. If you do not already have an @ file for the object spectra, it will be convenient to create one with a list of all the spectra to be extracted. The first step is to set the ***dispaxis*** parameter in the object headers so that the IRAF tasks know which way the spectrum runs (i.e. along the columns or along the lines). To set this parameter assuming the spectra are parallel to the columns, type

setdisp *.imh dispax=2 (dispax=1 for spectrum parallel to lines)

The next step is to set the parameters of several tasks in the APEXTRACT package: **apedit, apfind, aptrace** and **apsum**. Do an **epar** on each of the tasks in turn, setting the parameters as in figure 11. Be sure to set the ***width*** in **apedit** to roughly the FWHM of the spectrum, but not less. A description of each parameter may be found in the "help" pages for the tasks. Note that this discussion assumes the data are single aperture spectra. Fiber data or long slit data may require other parameter settings (See Appendix A).

We are now ready to extract the spectra:

apsum @starfiles

Apsum works by using each of the tasks listed above in the logical order to find, edit, trace and sum the data in each spectrum. First, **apfind** will attempt to find the center of the profile by summing the middle 10 lines together (***nsum***=10), find the single most prominent feature (***nfind***=1), and then find the center of this feature which is really our stellar profile (***radius***=2, ***width***=5). A plot will then be displayed of these summed rows and the feature will be marked along with a default aperture width. **Apsum** will ask you innumerable questions; at this time answer **yes** to all of them.

Now **apedit** allows you to adjust the extraction aperture by setting the vertical cursor on the lower (mark with **l** key) and upper (mark with **u** key) edge of the profile, or setting the horizontal cursor to the lower limit (mark with **y** key) of the profile. You can also change any of the other parameters at this time; type **?** to list all of the options. When you are satisfied with the way it found your object and marked the aperture, type **q** to go on.

At this point **aptrace** will step along the spectrum (***nstep***=10, ***nsum***=10) and attempt to find new centers along the way relative to the previous center. A plot will then be displayed of the x-y trace with an initial fit overplotted. The interactive curve fitting package is now available enabling you to adjust the type of fitting function and its order until you are satisfied. Type **?** for the cursor options. Notice the fitting parameters listed at the top of the plot (figure 12). When the fit is satisfactory, type **q** to go on.

Apsum will now sum the spectrum over the aperture you have accepted according to the summing parameters (profile, variance, clean, etc.). The final extracted spectrum will be plotted, and if saved will be named *inputname.0001*. **Apsum** will then continue through your input list in a similar manner displaying the profiles, the traces, and the extracted data. It is possible, after one gains confidence with how things are going, to skip some of the intermediate plots. But since you have chosen this way to extract your coude data, you probably want to see everything. If not, see the help pages for **apsum**.

We can now extract comparison spectra if desired. The comparison spectra are usually extracted using the **aptrace** parameters of a star taken nearby in time as the reference. Assuming we will extract two comparison spectra, one before and one after the stellar spectrum, type

```
apsum comp001,comp002 interact- recen- trace- ref=star001
```

In this example the trace for the object "star001" will be used to extract the comps taken before and after star001, thus we turn off many of the options. In like manner, extract any other comparison spectra.

3.9. Telluric Line Divisions

If the need to remove telluric lines from your spectra is anticipated, obtain high signal to noise spectra of a rapidly rotating B star observed at a similar airmass. Reduce these spectra (or spectrum) in a manner identical to the program spectra. You may then use the arithmetic options in **splot** to adjust the strengths of the telluric lines, and divide the data spectrum by the telluric spectrum. We recommend that you remove telluric lines before wavelength calibration, since the wavelength calibration process rebins the data to linearize the wavelength scale.

If shifts between the two spectra have occured due to flexure or chip motion, use **shiftlines** (in the IMAGES package) to adjust the B star spectrum to match the data spectrum. The amount of shift needed can be determined with **splot**, using its **e** key to measure accurate positions of telluric lines in both spectra. Don't use the cursor positions to determine the relative shift - the cursor isn't accurate enough. Then shift the B star spectrum by the appropriate amount; both integral and non-integral pixel shifts are allowed. For non-integral pixel shifts, use a high order polynomial fit (like POLY5) to preserve the noise characteristics of the original spectrum.

After the B star spectrum has been shifted to match the data spectrum, plot the B star spectrum with **splot**. In the *function* mode (the **f** key), take the log of the spectrum (natural or base 10 - it doesn't matter). Multiply the spectrum by a constant, the ratio of the airmass of the data spectrum to the airmass of the B star spectrum (or any constant that works!) and then take the anti-log of the B star spectrum. Finally, divide the B star spectrum by the data spectrum (the / key) and take the inverse (the **i** key) to obtain the corrected data spectrum. If it looks acceptable, **q** the function mode, and write the new spectrum to a file with **i**.

You may have to try several multiplicative constants to get the spectra to match exactly. The spectra match if the telluric lines are gone and are not replaced by spikes in the data spectrum; the spikes indicate that the telluric lines are too strong in the B star spectrum.

3.10. Wavelength Calibrations

If you have extracted data from the mountain, and don't need to do telluric line removal, this is the step at which you will probably want to start. Otherwise, see the previous section. In general, it is advisable to do the wavelength calibration of spectra before continuum rectification since rebinning the data for uniform dispersion can cause the continuum to tilt. This is especially true of echelle data which is quite nonlinear in wavelength. Comparison spectra are necessary for establishing a rest wavelength scale upon the stellar spectra, and to monitor and correct for any motion of the CCD during the night (typically a few tenths of a pixel). If radial velocities are to be measured you will want to do the wavelength calibration carefully; otherwise, a comparison at the beginning and end of the night may be sufficient. Once the wavelength solutions are established, the comparison spectra themselves may be deleted. It is the database entries for the wavelength solutions that are important. It may be useful to save one extracted comparison for examination of the instrumental profile.

The steps in wavelength calibration of spectra consist of extracting (**apsum**) comparison spectra, **identify**ing spectral lines and fitting wavelength solutions, assigning these comparison spectra as reference spectra to the stellar spectra (**refspectra**), and applying the dispersion solution to the stellar spectra (**dispcor**). The comparison spectra are usually extracted using the trace parameters of a star taken nearby in time as the reference (see end of §3.8).

Wavelength solutions are calculated through the **identify** task on one comparison spectrum, and propagated to other similar comparison spectra with the **reidentify** task. **Identify** is used to mark spectral lines, to determine wavelengths of other lines from a stored list of comparison line wavelengths, and to calculate the dispersion solution. We assume, in these examples, that your comparison spectra were obtained with a thorium-argon lamp, but a helium-neon-argon wavelength list is also available in the IRAF database (*onedstds$idhenear.dat*). You can also create your own linelist - create a file with the editor containing one wavelength entry per line. Then set the ***coordlist*** parameter in **identify** to the name of this file. This linelist can be created interactively in the **identify** task itself just by marking and typing in wavelengths for as many lines as desired. In this case, set ***coordlist*** = "".

The extracted spectra are written to tape at the telescope as 2-dimensional images, although there is really only one row of data. Further reductions will be less tedious if the data is converted to simple row images at this time. This can be done very easily. The following example converts all the images called "nite1*" to simple row images and writes them back into themelves.

```
imcopy nite1*.imh[*,1] nite1*.imh
```

Compare the parameters of **identify** with figure 13. The ***match*** parameter is in Ångstroms. Only those features in the comparison list will be used that are found by the preliminary fit to be within this ***match*** value. Use a low order, nearly linear function to start the identification, then change to a higher order when more lines are identified.

```
identify comp001
```

Mark two lines in the plotted spectrum using the cursor and the **m** key, identifying each with its proper wavelength. Now type **l** to search the linelist for additional identifications. Several new lines should be found and marked in the comparison spectrum. The colon command **:labels user** can be used to display the wavelength of each feature marked. Type an **f** to perform a dispersion solution and enter into the interactive curve fitting routine where you can iterate the solution. You can delete or reinsert points, change the function and the order, and then rapidly refit the data points to arrive at a good solution. You should be able to obtain a solution with residuals of less than 0.05 to 0.10 pixel; use the **j** key (note the residuals are in Ångstroms). When you are finally satisfied, type **q** to go back to the "identify mode", followed by another **q** to exit the task. The final dispersion solution will be written into the database, if you choose.

The task **reidentify** (figure 14) can be used to determine wavelength solutions for the remaining comparison spectra for the first night once you have done the first solution:

```
reidentify comp001 comp*.imh  v+
```

In this example, comp001 is the reference solution to be applied to the rest of the comparison spectra. The ***v+*** option gives us some useful information on the terminal regarding the shifts from the reference spectrum. All solutions are stored in the database directory.

The next step is to assign reference comparison spectra to each object based on the time of observation relative to the comparison spectra. The task **refspectra** will insert into the image header the reference spectrum name or pair of names under the keywords REFSPEC1 and REFSPEC2. The final wavelength scale for each object will then be the time-weighted average of the wavelength scales for the preceeding and following comparison spectra. The time given in the picture header, UT, is at the beginning of the integration however, necessitating the calculation of UTMIDDLE. This can be done by running the task **setairmass** (in IMRED package) on the objects and comps:

setair star*.imh,comp*.imh

The keyword UTMIDDLE will be added to each spectrum header. Check the parameters of the **refspectra** task against figure 15. Be sure that the ***sort*** parameter is set to "utmiddle". Now type for example,

refspec star*.imh ref=comp*.imh
or
refspec star*.imh ref=@compfiles

We are now ready to wavelength calibrate the object spectra with these wavelength solutions using the **dispcor** task. The parameters for **dispcor** are shown in figure 16. The input spectra are dispersion corrected to a linear wavelength coordinate system (possibly linear in the logarithm of the wavelength) by interpolation using the dispersion solutions of the reference spectra specified in the image header. The input spectra must have one or both of the reference spectra definitions REFSPEC1 and REFSPEC2 in the image header. When there are two reference spectra the spectrum names will be followed by a weighting factor (assumed to be 1 if missing). The wavelength of a pixel is then the weighted average of the wavelengths from the reference spectra dispersion solutions. A high order polynomial function, such as POLY5, is best for interpolating the spectrum to preserve the original noise characteristics of the data. For radial velocity cross-correlation measurements, the standard star and each object must have identical wavelength dispersion values. This is achieved by setting the ***global*** parameter in **dispcor** to "yes". Consider the example of dispersion correcting a series of spectra named "star*.imh" and renaming them "stardc*.imh":

dispcor star*.imh %star%stardc%*.imh

If the image names are in record number extension format, we might type

dispcor star stardc 1-10,15-20 recform+

where the record numbers are appended to the input and output spectra root name.

3.11. Continuum Fitting

The task for fitting the continuum to a one-dimensional spectrum is called **continuum**, which uses the interactive curve fitting task. **Continuum** fits the specified function to the spectrum in the input file. Functions such as a spline3, chebyshev or legendre are available. Data points which deviate from the fit by more than the ***high_reject*** or ***low_reject*** sigma limits are rejected, and a new fit is derived. The fit is iterated the number of times specified by the parameter ***niterate***. The parameters for **continuum** can be set initially as in figure 17. To execute the task, using one input spectrum, type for example

continuum nite1001 cnite1001

The routine is interactive such that the order of the function, the number of iterations, and the upper and lower sigma limits may be changed to improve the fit. Notice the use of the **s** key to mark specific regions to fit while excluding others, such as broad hydrogen lines. Generally, the value of ***high_reject*** should be larger than that of ***low_reject***. Different types of spectra will require different values for many of the parameters, so you will need to experiment to see what works best for your data. When you are satisfied with the fit, type **q** to exit the routine. The input data spectrum will be divided by the fit, and the results written to the output image. If you want to save the fit rather than the ratio, just change the hidden parameter ***type*** to "fit". To do a batch of spectra consecutively, perhaps all the spectra from one night of observing, use @files for the input and output.

4. Measurement of Spectra

The interactive routine **splot** may be used to measure equivalent widths, deblend lines, smooth, measure signal-to-noise, etc on extracted spectra. Radial velocities may also be measured by fitting a gaussian to individual line profiles using the **d**eblend key. The other keys **e**, **k**, and **v**, do not have sufficient accuracy for radial velocity measurements. See the help pages for **splot** for details. The corrections for reducing geocentric velocities to heliocentric velocities can be computed easily with the **rvcorrect** task (in the ASTUTIL package). Since **rvcorrect** reads the UT in the picture header, you may wish to add the UTMIDDLE keyword to the picture header (§3.10) if you have not done so previously. You can then convert UT to UTMIDDLE with the header editor,

```
hedit nite1*.imh ut '(utmiddle)'
```

Radial velocities may be measured at NOAO with a prototype radial velocity package using various cross-correlation techniques. Please discuss your needs with one of contributors of this document.

5. Taping Reduced Data

When your reductions are complete you will probably want to save the reduced or partially reduced spectra on magnetic tape. Two output options are available: FITS format or card-image format. After loading the DATAIO package, allocate the desired tape drive (§3.4). Writing a FITS tape is done with the task **wfits**. Each IRAF image is a file on the tape. Approxiamtely 320 individual spectra (1024 points) can fit on a 600 ft. FITS tape. The following example will output data for a FITS formatted tape:

```
files nite1*.imh >dataout
files more*.imh >>dataout
edit dataout
allocate mtb
wfits @dataout mtb[1600] new+ bit=32 >flist &
lprint flist
```

An easy and reliable way to tape files is to create a @file (e.g. dataout) which contains the list of images you want to write to tape. All the data files in the list in the file "dataout" will be written to tape as 32-bit integers with autoscaling. The drive *mtb* is used, and the tape is assumed to be a new tape. A list of the files as they are written to tape is created in the file "flist" This file may be printed out as a record of the tape contents.

Writing a card-image tape can be done with the task **widstape**. You will need to load the MTLOCAL package. Each execution of **widstape** writes a file on the tape according to the input specifications. Type **lpar widstape** to check the parameters. The card image tape format is identical to the old IDSOUT tape that you may be familiar with.

```
widstape mtb @dataout "" new+ &
```

The term "new+" is an abbreviation for ***new_tape***=yes. This will write an output tape that has 1 ASCII coded file with 40 card-images (3200) characters per block. The spectra are automatically written with 1024 data points; the extra points are set to zero. (Thus, 1-d spectra with more than 1024 points are not supported by this task.)

```
widstape mtb @dataout "" new+ bl=10640 &
```

will write the same data to tape except now the blocking size is 10640 characters. This blocking size will correspond to one spectrum per block.

A card-image tape can also be generated with tasks **wtextimage** and **wcardimage**, both in the DATAIO package.

NOTE: A word about tape densities may be in order at this time.

1. On Unix systems with variable tape drive densities, the tape density for a new tape can be specified at the time of execution of the IRAF tape writing task, e.g.

   ```
   wfits @dataout mta1600 new+ ...
   ```

2. On VMS systems with variable tape drive densities, the tape density for a new tape must be specified with a VMS system command before the tape is allocated within IRAF:

   ```
   cl>!init/den=6250 mua0: label
   cl>allocate mta
   cl>wfits @dataout mta new+ ...
   ```

   The VMS "init" command writes a label on the tape at the specified density; the tape drive name is the logical VMS name for the drive. The IRAF **allocate** then mounts the tape at the proper density. The IRAF **wfits** task overwrites the label with "new+". If the tape already has data on it, the proper density will be sensed with the IRAF **allocate** command, and the **!init/den** command should not be used.

6. Appendix A. Triple Fiber Feed Reductions

The triple fiber feed consists of 3 separate fibers: one for the stellar image, and two to monitor a comparison lamp during each integration. The output spectrum format is a stellar spectrum flanked by the two comparison spectra. The value of the fibers lies in their ability to scramble the input beam to remove most spatial assymetries such as guiding errors or atmospheric refraction effects. The purpose of the comparison fibers is to provide a precise reference frame for removing any motion due to the CCD or spectrograph. Their use is probably not justified unless the spectra are aligned parallel to the CCD rows (lines). Otherwise, the CCD charge transfer deficiency will still be a significant source of error. If the spectra are aligned parallel to columns, good results can still be achieved by taking a comparison lamp exposure through the stellar fiber every hour in the same manner as if a normal slit were being used. Reductions for this observing mode are the same as used elsewhere in this manual. Reductions using all three fibers are the same as slit spectra until the **apsum** step. Here, an experimental task, **dofibers** (NEWIMRED package) which is presently available only at NOAO, can be used to automatically extract and wavelength calibrate triple fiber feed data.

Two important images are needed for the aforementioned reductions: a flat-field image with the flat lamp through all three fibers, and a comparison image with the comparison lamp also in all fibers. These images may be obtained by switching the mirrors to illuminate the different fibers for the appropriate exposure times, or by combining (after debiasing and triming) frames taken first through one fiber and then immediately taking another frame through the other fiber. The combined flat frame could also be used for flat-fielding the object spectra if sufficient numbers of each are taken for good signal to noise. In this case use **flatcombine** (§3.6) to create an average flat for the stellar fiber and an average flat for the comparison fibers.

If the spectra are parallel to the CCD lines (rows) there will almost always be hot or dead columns running perpendicular through the data. It is best to fix these using **ccdproc** with ***fixpix***=yes after first setting up a bad pixel file. This file consists of a list of bad columns and lines which will be replaced by linear interpolation from neighboring lines and columns. See the help pages for **ccdproc** for details on setting up this file. Now **ccdproc** can be run to debias and trim the average stellar fiber flat and the average comparison fiber flat. Also process the corresponding comparison spectra. Combine the appropriate frames to create the aperture template and the comparison template:

```
combine sflat,cflat scflat combine=average
combine scomp,ccomp sccomp combine=average
```

In the above example, sflat and cflat refer to flats taken through the stellar and comparison fibers respectively, while scomp and ccomp are the corresponding spectra taken with the thorium lamp. The first of these images, scflat, will be used as a reference to locate the comparison and stellar fibers in all subsequent spectra. We can also use scflat to flat-field the object spectra, but first it should be processed with **flat1d** to normalize it to 1 across the 3 apertures (see §3.7):

```
flat1d scflat Nscflat minflat=15
```

Set your value of ***minflat*** after examining the image with **implot**. Now **ccdproc** can be used to debias, trim, fix bad columns, and flat-field the object spectra. **Ccdproc** the combined comparison image (sccomp) if not already done.

The **dofibers** task can now be used to automatically (with some user intervention if desired) recenter the stellar aperture and apply the same shift to the comparison apertures. Comparison spectra are **identify**ed from the combined comparison spectrum (sccomp) and the subsequent comparisons on each object picture are **reidentify**ed. A two-dimensional interpolation is done to transfer the wavelength solution to the stellar spectrum. Finally, a one-dimensional spectrum is output with a ".ms" appended to the input root name. We now have dispersion corrected, one-dimensional spectra that can be futher analyzed as discussed in sections 3.11 and 4.

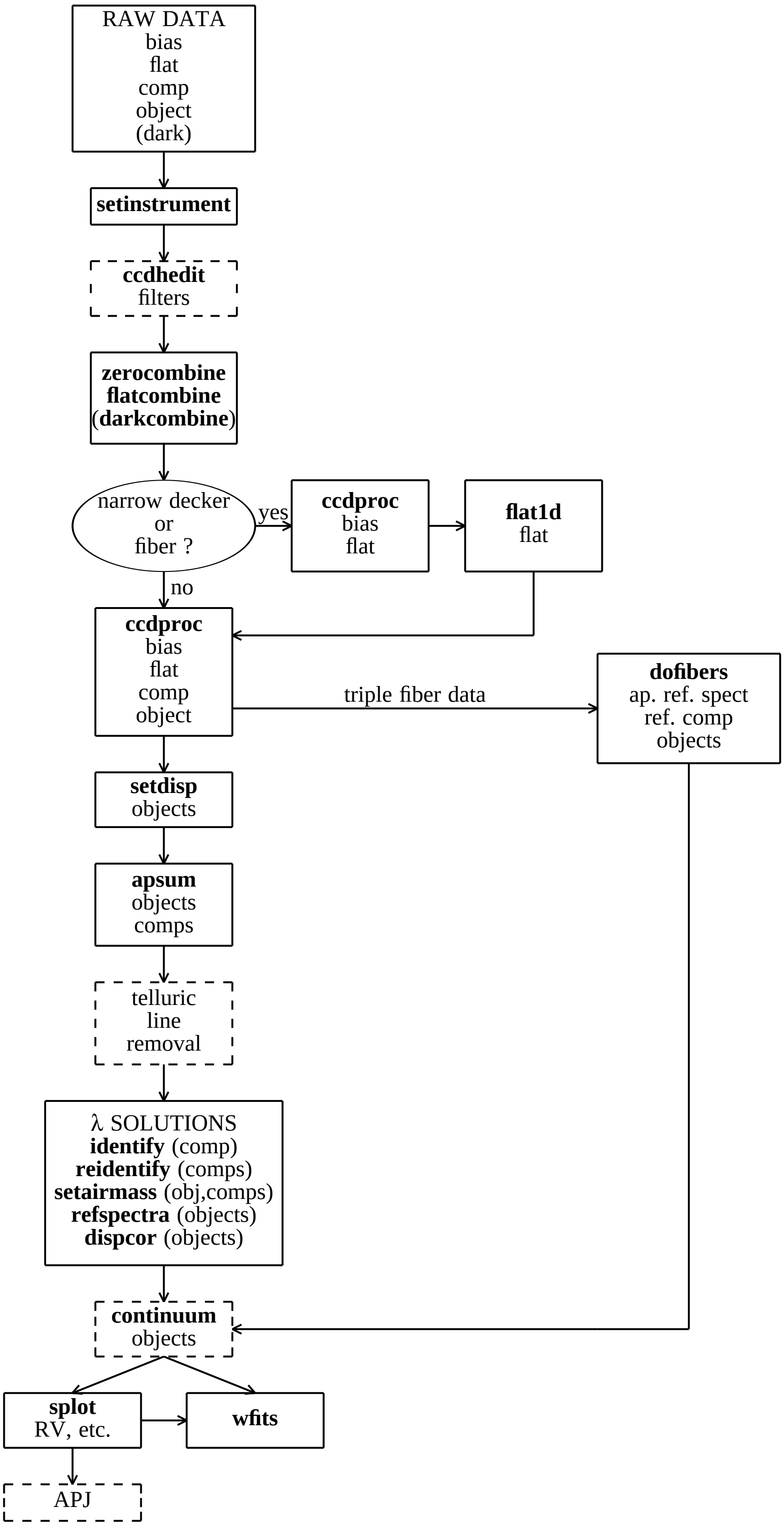

Figure 1. Flowchart of Coude CCD Reductions.
dashed box indicates optional procedure