APEXTRACT Package Revisions Summary: IRAF Version 2.10

Francisco Valdes

IRAF Group - Central Computer Services
National Optical Astronomy Observatories††
P.O. Box 26732, Tucson, Arizona 85726
September 1990

ABSTRACT

This paper summarizes the changes in Version 3 of the IRAF **apextract** package which is part of IRAF Version 2.10. The major new features and changes are:

- New techniques for cleaning and variance weighting extracted spectra
- A new task, **apall**, which integrates all the parameters used for one dimensional extraction of spectra
- A new extended output format for recording both weighted and unweighted extractions, subtracted background, and variance information.
- Special featurers for automatically numbering and identifying large numbers of apertures.
- New tasks and algorithms, **aprecenter** and **apresize**, for automatically recentering and resizing aperture definitions
- A new task, **apflatten**, for creating flat fields from fiber and slitlet spectra
- A new task, **apfit**, providing various types of fitting for two dimensional multiobject spectra.
- A new task, **apmask**, for creating mask images from aperture definitions.

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

APEXTRACT Package Revisions Summary: IRAF Version 2.10

Francisco Valdes

IRAF Group - Central Computer Services
National Optical Astronomy Observatories††
P.O. Box 26732, Tucson, Arizona 85726
September 1990

1. Introduction

A new version of the IRAF **apextract** package has been completed. It is Version 3 and is part of IRAF Version 2.10. The package will be made available as an external package prior to the release of V2.10. This paper describes the changes and new features of the package. It does not describe them in detail. Full details of the algorithms, functions, and parameters are found in the task descriptions. Reference is made to the previous version so familiarity with that version is useful though not necessary. There were three goals for the new package: new and improved cleaning and variance weighting (optimal extraction) algorithms, the addition of recommended or desirable new tasks and algorithms (particularly to support large numbers of spectra from fiber and aperture mask instruments), and special support for the new image reduction scripts. Features relating to the last point are not discussed here.

Table 1 summarizes the major new features and changes in the package.

Table 1: Summary of Major New Features and Changes

- New techniques for cleaning and variance weighting extracted spectra
- A new task, **apall**, which integrates all the parameters used for one dimensional extraction of spectra
- A new extended output format for recording both weighted and unweighted extractions, subtracted background, and variance information.
- Special featurers for automatically numbering and identifying large numbers of apertures.
- New tasks and algorithms, **aprecenter** and **apresize**, for automatically recentering and resizing aperture definitions
- A new task, **apflatten**, for creating flat fields from fiber and slitlet spectra
- A new task, **apfit**, providing various types of fitting for two dimensional multiobject spectra.
- A new task, **apmask**, for creating mask images from aperture definitions.

2. Cleaned and Variance Weighted Extractions: apsum and apall

There are two types of aperture extraction (estimating the background subtracted flux across a fixed width aperture at each image line or column) just as in the previous version. One is a simple sum of pixel values across an aperture. In the previous version this was called "profile" weighting while in this version it is simply called unweighted or "none". The second type weights each pixel in the sum by its estimated variance based on a spectrum model and detector noise parameters. As before this type of extraction is selected by specifying "variance" for the weighting parameter.

Variance weighting is often called "optimal" extraction since it produces the best unbiased signal-to-noise estimate of the flux in the two dimensional profile. It also has the advantage that wider apertures may be used without penalty of added noise. The theory and application of this type of weighting has been described in several papers. The ones which were closely examined and used as a model for the algorithms in this software are *An Optimal Extraction Algorithm for CCD Spectroscopy*, **PASP 98**, 609, 1986, by Keith Horne and *The Extraction of Highly Distorted Spectra*, **PASP 100**, 1032, 1989, by Tom Marsh.

---

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

The noise model for the image data used in the variance weighting, cleaning, and profile fitting consists of a constant gaussian noise and a photon count dependent poisson noise. The signal is related to the number of photons detected in a pixel by a gain parameter given as the number of photons per data number. The gaussian noise is given by a readout noise parameter which is a defined as a sigma in photons. The poisson noise is approximated as gaussian with sigma given by the number of photons. The method of specifying this noise model differs from the previous version in that the more common CCD detector parameters of readout noise and gain are used rather than the linear variance parameters "v0" and "v1".

Some additional effects which should be considered in principle, and which are possibly important in practice, are that the variance estimate should be based on the actual number of photons detected before correction for pixel sensitivity; i.e. before flat field correction. Furthermore the uncertainty in the flat field should also be included in the weighting. However, the profile must be determined free of sensitivity effects including rapid larger scale variations such as fringing. Thus, ideally one should input the unflat-fielded observation and the flat field data and carry out the extractions with the above points in mind. However, due to the complexity often involved in basic CCD reductions and special steps required for producing spectroscopic flat fields this level of sophistication is not provided by the current package.

The package does provide, however, for propagation of an approximate uncertainty in the background estimate when using background subtraction. If background subtraction is done, a background variance is computed using the poisson noise model based on the estimated background counts. Because the background estimate is based on a finite number of pixels, the poisson variance estimate is divided by the number (minus one) of pixels used in determining the background. The number of pixels used includes any box car smoothing. Thus, the larger the number of background pixels the smaller the background noise contribution to the variance weighting. This method is only approximate since no correction is made for the number of degrees of freedom and correlations when using the fitting method of background estimation.

If removal of cosmic rays and other deviant pixels is desired (called cleaning) they are iteratively rejected based on the estimated variance and excluded from the weighted sum. Unlike the previous version, a cleaned extraction is always variance weighted. This makes sense since the detector noise parameters must be specified and the spectrum profile computed, so all of the computational effort must be done anyway, and the variance weighting is as good or superior to a simple unweighted extraction.

The detection and removal of deviant pixels is straightforward. Based on the noise model, pixels deviating by more than a specified number of sigma (square root of the variance) above or below the model are removed from the weighted sum. A new spectrum estimate is made and the rejection is repeated. The rejections are made one at a time starting with the most deviant and up to half the pixels in the aperture may be rejected.

3. Spectrum Profile Determination: apsum, apall, apflatten, apfit

The foundation of variance weighted or optimal extraction, cosmic ray detection and removal, two dimensional flat field normalization, and spectrum fitting and modeling is the accurate determination of the spectrum profile across the dispersion as a function of wavelength. The previous version of the **apextract** package accomplished this by averaging a specified number of profiles in the vicinity of each wavelength after correcting for shifts in the center of the profile. This technique was sensitive to perturbations from cosmic rays and the exact choice of averaging parameters. The current version of the package uses a different algorithm, actually a combination of two algorithms, which is much more stable.

The basic idea is to normalize each profile along the dispersion to unit flux and then fit a low order function to sets of unsaturated points at nearly the same point in the profile parallel to the dispersion. The important point here is that points at the same distance from the profile center should have the nearly the same values once the continuum shape and spectral features have been divided out. Any variations are due to slow changes in the shape of the profile with wavelength, differences in the exact point on the profile, pixel binning or sampling, and noise. Except for the noise, the variations should be slow and a low order function smoothing over many points will minimize the noise and be relatively insensitive to bad pixels such as cosmic rays. Effects from bad pixels may be further eliminated by chi-squared iteration and clipping. Since there will be many points per degree of freedom in the fitting function the clipping may even be quite aggressive without significantly affecting the profile estimates. Effects from saturated pixels are minimized by excluding from the profile determination any profiles containing one or more saturated pixels.

The normalization is, in fact, the one dimensional spectrum. Initially this is the simple sum across the aperture which is then updated by the variance weighted sum with deviant pixels possibly removed. This updated one dimensional spectrum is what is meant by the profile normalization factor in the discussion below. The two dimensional spectrum model or estimate is the product of the normalization factor and the profile. This model is used for estimating the pixel intensities and, thence, the variances.

There are two important requirements that must be met by the profile fitting algorithm. First it is essential that the image data not be interpolated. Any interpolation introduces correlated errors and broadens cosmic rays to an extent that they may be confused with the spectrum profile, particularly when the profile is narrow. This was one of the problems limiting the shift and average method used previously. The second requirement is that data fit by the smoothing function vary slowly with wavelength. This is what precludes, for instance, fitting profile functions across the dispersion since narrow, marginally sampled profiles require a high order function using only a very few points. One exception to this, which is sometimes useful but of less generality, is methods which assume a model for the profile shape such as a gaussian. In the methods used here there is no assumption made about the underlying profile other than it vary smoothly with wavelength.

These requirements lead to two fitting algorithms based on how well the dispersion axis is aligned with the image columns or lines. When the spectra are well aligned with the image axes one dimensional functions are fit to the image columns or lines. Small excursions of a few pixels over the length of the spectrum can be adequately fit in this way. When the spectra become strongly tilted then single lines or columns may cross the actual profile relatively quickly causing the requirement of a slow variation to be violated. One thought is to use interpolation to fit points always at the same distance from the profile. This is ruled out by the problems introduced by image interpolation. However, there is a clever method which, in effect, fits low order polynomials parallel to the direction defined by tracing the spectrum but which does not interpolate the image data. Instead it weights and couples polynomial coefficients. This method was developed by Tom Marsh and is described in detail in the paper, *The Extraction of Highly Distorted Spectra*, **PASP 101**, 1032, Nov. 1989. Here we refer to this method as the Marsh algorithm and do not attempt to explain it further.

Both fitting algorithms weight the pixels by their variance as computed from the background and background variance if background subtraction is specified, the spectrum estimate from the profile and the spectrum normalization, and the detector noise parameters. The noise model is that described earlier.

The profile fitting can be iterated to remove deviant pixels. This is done by rejecting pixels greater than a specified number of sigmas above or below the expected value based on the profile, the normalization factor, the background, the detector noise parameters, and the overall chi square of the residuals. Rejected points are removed from the profile normalization and from the fits.

4. New Extraction Task: apall

All of the functions of the **apextract** package are actually part of one master program. The organization of the package into tasks by function with parameters to allow selection of some of the other functions, for example the aperture editor may be entered from virtually every task, was done to highlight the logic and organize the parameters into small sets. However, there was often confusion about which parameters were being used and the need to set parameters in one task, say **aptrace**, in order to use the trace option in another task, say **apsum**. In practice, for the most common function of extraction of two dimensional spectra to one dimension most users end up using **apsum** for all the functions.

In the new version, the old organization is retained (with the addition of new functions and some changes in parameters) but a new task, **apall**, is also available. This task contains all of the parameters needed for extraction with a parameter organization which is nicely formated for use with **eparam**. The parameters in **apall** are independent of the those in the other tasks. It is expected that many, if not most users will opt to use this task for spectrum extraction in preference to the individual functions.

The organization by function is still used in the documentation. This is still the best way to organize the descriptions of the various algorithms and parameters. As an example, the profile tracing algorithm is described in most detail under the topic **aptrace**.

5. Extraction Output Formats: apsum and apall

The extracted spectra are recorded in one, two, or three dimensional images depending on the *format* and *extras* parameters. If the *extras* parameter is selected the formats are three dimensional with each plane in the third dimension containing associated information for the spectra in the first plane. This information includes the unweighted spectrum and a sigma spectrum (estimated from the variances and weights of the pixels extracted) when using variance weighting, and the background spectrum when background subtraction is used. When *extras* is not selected only the extracted spectra are output.

The formats are basically the same as in the previous version; onedspec, multispec, and echelle. In addition, the function of the task **apstrip** in the previous version has been transferred to the extraction tasks by simply specifying "strip" for the format.

There are some additions to the header parameters in multispec and echelle format. Two additional fields

have been added to the aperture number parameter giving the aperture limits (at the reference dispersion point). Besides being informative it may be used for interpolating dispersion solutions spatially. A second, optional keyword per spectrum has been added to contain a title. This is useful for multiobject spectra.

6. Easier and Extended Aperture Identifications: apfind and apedit

When dealing with a large number of apertures, such as occur with multifiber and multiaperture data, the burden of making and maintaining useful aperture identifications becomes large. Several very useful improvements were made in this area. These improvements generally apply equally to aperture identifications made by the automated **apfind** algorithm and those made interactively using **apedit**. In the simplest usage of defining apertures interactively or with the aperture finding algorithm, aperture numbers are assigned sequentially beginning with 1. In the new version the parameter "order" allows the direction of increasing aperture numbers with respect to the direction of increasing pixel coordinate (either column or line) to be set. An "increasing" order parameter value numbers the apertures from left to right (the direction naturally plotted) in the same sense as the pixel coordinates. A "decreasing" order reverses this sense.

Some instruments, particularly multifiber instruments, produce nearly equally spaced spectra for which one wants to maintain a consistent numbering sequence. However, at times some spectra may be missing due to broken or unassigned fibers and one would like to skip an aperture identification number to maintain the same fiber assignments. To do this automatically, a new parameter called *maxsep* has been added. This parameter defines the maximum separation between two apertures beyond which a jump in the aperture sequence is made. In other words the sequence increment is given by rounding down the separation divided by this parameter. How accurately this value has to be specified depends on how large the gaps may be and the natural variability in the aperture positions. In conjunction with the minimum separation parameter this algorithm works quite well in accounting for missing spectra.

One flaw in this scheme is when the first spectrum is missing causing the identifications will be off. In this case the modified interactive aperture editing command 'o' asks for the aperture identification number of the aperture pointed at by the cursor and then automatically renumbers the other apertures relative to that aperture. The other possible flaw is identification of noise as a spetrum but this is controlled by the *threshold* parameter and, provided the actual number of spectra is known, say by counting off a graph, then the *nfind* parameter generally limits this type of problem.

A new attribute of an aperture is a title. If present this title is propagated through the extraction into the image headers. The title may be set interactively but normally the titles are supplied in another new feature, an "aperture identification" file specified by the parameter *apidtable*. This file provides the most flexibility in making aperture identification assignments. The file consists of lines with three fields, a unique aperture number, a beam or aperture type number which may be repeated, and the aperture title. The aperture identification lines from the file are assigned sequentially in the same order as would be done if using the default indexing including skipping of missing spectra based on the maximum separation.

By default the beam number is the same as the aperture number. When using an aperture identification file the beam number can be used to assign spectrum types which other software may use. For example, some of the specialized fiber reduction packages use the beam number to identify sky fibers and embedded arc fibers.

7. New Aperture Recentering Task: aprecenter

An automated recentering algorithm has been added. It may be called through the new **aprecenter** command, from any of the tasks containing the *recenter* parameter, or from the aperture editor. The purpose of this new feature is to allow automatically adjusting the aperture centers to follow small changes in the positions of spectra expected to be at essentially the same position, such as with fiber fed spectrographs. This does not change the shape of the trace but simply adds a shift across the dispersion axis.

Typically, one uses a strong image to define reference apertures and then for subsequent objects uses the reference positions with a recentering to correct for flexure effects. However, it may be inappropriate to base a new center on weak spectra or to have multiple spectra recentered independently. The recentering options provide for selecting specific apertures to be recentered, selecting only a fraction of the strongest (highest peak data level) spectra and averaging the shifts determined (possible from only a subset of the spectra) and applying the average shift to all the apertures. Note that one may still specify the dispersion line and number of dispersion lines to sum in order to improve the signal for centering.

8. New Aperture Resizing Task: apresize

An automated resizing algorithm has been added. It may be called through the new **apresize** command, from any of the tasks containing the *resize* parameter, or from the aperture editor with the new key 'z' (the y cursor level command is still available with the adjusting the aperture widths to follow changes in seeing and to provide a greater variety of global aperture sizing methods.

In all the methods the aperture limits are set at the pixel positions relative to the center which intersect the linearly interpolated data at some data value. The methods differ in how the data level is determined. The methods are:

- Set size at a specified absolute data level
- Set size at a specified data level above a background
- Set size at a specified fraction of the peak pixel value
- Set size at a specified fraction of the peak pixel value above a background

The automatic background is quite simple; a line connecting the first local minima from the aperture center.

The limits determined by one of the above methods may be further adjusted. The limits may be increased or decreased by a specified fraction. This allows setting wider limits based on more accurately determined limits from the stronger part of the profile; for example doubling the limits obtained from the half intensity point. A maximum extent may be imposed. Finally, if there is more than one aperture and one wants to maintain the same aperture size, the apertures sizes determined individually may be averaged and substituted for all the apertures.

9. New Aperture Mask Output: apmask

A new task, **apmask**, has been added to produce a mask file/image of 1's and 0's defined by the aperture definitions. This is based on the new IRAF mask facilities. The output is a compact binary file which may be used directly as an image in most applications. In particular the mask file can be used with tasks such as **imarith**, **imreplace**, and **display**. Because the mask facility is new, there is little that can be done with masks other than using it as an image. However, eventually many tasks will be able to use mask images. The aperture mask will be particularly well suited to work with **imsurfit** for fitting a surface to the data outside the apertures. This would be an alternative for scattered light modeling to the **apscatter** tasks.

10. Aperture Flat Fields and Normalization: apflatten and apnormalize

Slitlet, echelle, and fiber spectra have the characteristic that the signal falls off to near zero values outside regions of the image containing spectra. Also fiber profiles are usually undersampled causing problems with gradients across the pixels. Directly dividing by a flat field produces high noise (if not division by zero) where the signal is low, introduces the spectrum of the flat field light, and changes the profile shape.

One method for modifying the flat field to avoid these problems is provided by the task **imred.generic.flat1d**. However, this task does not use any knowledge of where the spectra are. There are two tasks in the **apextract** package which can be used to modify flat field images. **apnormalize** is not new. It divides the spectra within specified apertures by a one dimensional spectrum, either a constant for simple throughput normalization or some smoothed version of the spectrum in the aperture to remove the spectral shape. Pixels outside specified apertures are set to unity to avoid division effects. This task has the effect of preserving the profile shape in the flat field which may be desired for attempts to remove slit profiles.

Retaining the profile shape of the flat field can give very bad edge effects, however, if there is image flexure. A new task similar to **flat1d** but which uses aperture information is **apflatten**. It uses the spectrum profile model described earlier. For nearly image axes aligned spectra this amounts very nearly to the line or column fitting of **flat1d**. As with **apnormalize** there is an option to fit the one dimensional spectrum to remove the large scale shape of the spectrum while preserving small scale sensitivity variations. The smoothed spectrum is multiplied by the normalized profile and divided into the data in each aperture. Pixels outside the aperture are set to 1. Pixels with model values below a threshold are also set to 1. This produces output images which have the small scale sensitivity variations, a normalized mean, and the spectrum profile removed.

11. Two Dimensional Spectrum Fitting: apfit

The profile and spectrum fitting used for cleaning and variance weighted extraction may be used and output in the new task **apfit**. The task **apfit** is similar in structure to **fit1d**. One may output the fit, difference, or ratio. The fit may be used to examine the spectrum model used for the cleaning and variance weighted extraction. The difference and ratio may used to display small variations and any deviant pixels. While specific uses are not given this task will

probably be used in interesting ways not anticipated by the author.

12. I/O and Dispersion Axis Parameters: apextract and apio

The general parameters, primarily concerning input and output devices and files, were previously in the parameter set **apio**. This "pset" task has been removed and those parameters are now found as part of the package parameters, i.e. **apextract**. There is one new parameter in the **apextract** package parameters, dispaxis. In the previous version of the package one needed to run the task **setdisp** to insert information in the image header identifying the dispersion direction of the spectra in the image. Often people would forget this step and receive an error message to that effect. The new parameter allows skipping this step. If the DISPAXIS image header parameter is missing the package parameter value is inserted into the image header as part of the processing. Note that if the parameter is present in the image header either because **setdisp** was used or the image creation process inserted it (a future ideal case) then that value is used in preference to the package parameter.

13. Strip Extraction: apstrip

The task **apstrip** from the previous version has been removed. However, it is possible to obtain two dimensional strips aligned with the image axes by specifying a format of "strip" when using **apsum** or **apall**. While the author doesn't anticipate a good scientific use for this feature others may find it useful.