# Guide to the Fiber Optic Echelle Reduction Task DOFOE

*Francisco Valdes*

IRAF Group - Central Computer Services
National Optical Astronomy Observatories††
April 1992

*ABSTRACT*

The **dofoe** reduction task is specialized for scattered light subtraction, extraction, flat fielding, and wavelength calibration of Fiber Optic Echelle (FOE) spectra. It is a command language script which collects and combines the functions and parameters of many general purpose tasks to provide a single complete data reduction path. The task provides a degree of guidance, automation, and record keeping necessary when dealing with the complexities of reducing this type of data.

## 1. Introductions

The **dofoe** reduction task is specialized for scattered light subtraction, extraction, flat fielding, and wavelength calibration of Fiber Optic Echelle (FOE) spectra. It is a command language script which collects and combines the functions and parameters of many general purpose tasks to provide a single complete data reduction path. The task provides a degree of guidance, automation, and record keeping necessary when dealing with the complexities of reducing this type of data.

The general organization of the task is to do the interactive setup steps first using representative calibration data and then perform the majority of the reductions automatically, possibly as a background process, with reference to the setup data. In addition, the task determines which setup and processing operations have been completed in previous executions of the task and, contingent on the `redo` and `update` options, skip or repeat some or all the steps.

The description is divided into a quick usage outline followed by details of the parameters and algorithms. The usage outline is provided as a checklist and a refresher for those familiar with this task and the component tasks. It presents only the default or recommended usage. Since **dofoe** combines many separate, general purpose tasks the description given here refers to these tasks and leaves some of the details to their help documentation.

## 2. Usage Outline

[1] The images must first be processed with **ccdproc** for overscan, bias, and dark corrections.

[2] Set the **dofoe** parameters with **eparam**. Specify the object images to be processed, the flat field image as the aperture reference and the flat field, and one or more arc images. If there are many object or arc spectra per setup you might want to prepare "@ files". Verify and set the format parameters, particularly the number of orders to be extracted and processed. The processing parameters are set for simple extraction and dispersion correction but dispersion correction can be turned off for quicklook or background subtraction and cleaning may be added.

[3] Run the task. This may be repeated multiple times with different observations and the task will generally only do the setup steps once and only process new images. Queries presented during the execution for various interactive operations may be answered with "yes", "no", "YES", or "NO". The lower case responses apply just to that query while the upper case responses apply to all further such queries during the execution and no further queries of that type will be made.

---

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

[4] The apertures are defined using the specified aperture reference image which is usually a flat field in which both the object and arc fibers are illuminated. The specified number of orders are found automatically and sequential apertures assigned. The resize option sets the aperture size to the widths of the profiles at a fixed fraction of the peak height.

[5] The automatic order identification and aperture assignment is based on peak height and may be incorrect. The interactive aperture editor is entered with a plot of the apertures. It is essential that the object and arc fiber orders are properly paired with the arc fibers having even aperture numbers and the object fibers having odd aperture numbers. It is also required that no orders be skipped in the region of interest. Missing orders are added with the 'm' key. Once all orders have been marked the aperture numbers are resequenced with 'o'. If local background subtraction is selected the background regions should be checked with the 'b' key. Preceding this with the 'a' key allows any changes to the background regions to be applied to all orders. To exit type 'q'.

[6] The order positions at a series of points along the dispersion are measured and a function is fit to these positions. This may be done interactively to adjust the fitting parameters. Not all orders need be examined and the "NO" response will quit the interactive fitting. To exit the interactive fitting type 'q'.

[7] If flat fielding is to be done the flat field spectra are extracted. A smooth function is fit to each flat field spectrum to remove the large scale spectral signature. The final response spectra are normalized to a unit mean over all fibers.

[8] If scattered light subtraction is selected the scattered light parameters are set using the aperture reference image and the task **apscatter**. The purpose of this is to interactively define the aperture buffer distance for the scattered light and the cross and parallel dispersion fitting parameters. The fitting parameters are taken from and recorded in the parameter sets **apscat1** and **apscat2**. All other scattered light subtractions are done noninteractively with these parameters. Note that the scattered light correction modifies the input images.

[9] If dispersion correction is selected the first arc in the arc list is extracted. One fiber is used to identify the arc lines and define the dispersion function using the task **ecidentify**. Identify a few arc lines in a few orders with 'm' and 'k' or 'o', use the 'l' line list identification command to automatically add additional lines and fit the dispersion function. Check the quality of the dispersion function fit with 'f'. When satisfied exit with 'q'.

[10] The other fiber dispersion function is automatically determined using the task **ecreidentify**.

[11] The arc reference spectrum is dispersion corrected. If the spectra are resampled to a linear dispersion system (which will be the same for all spectra) the dispersion parameters determined from the dispersion solution are printed.

[12] The object spectra are now automatically background subtracted (an alternative to scattered light subtraction), extracted, flat fielded, and dispersion corrected. Any new dispersion function reference arcs assigned to the object images are automatically extracted and dispersion functions determined. A zero point wavelength correction is computed from the simultaneous arc fiber spectrum and applied to the object spectrum.

[13] The final spectra will have the same name as the original 2D images with a ".ec" extension added.

### 3. Spectra and Data Files

The basic input consists of dual fiber FOE object and calibration spectra stored as IRAF images. The raw CCD images must be processed to remove overscan, bias, and dark count effects. This is generally done using the **ccdred** package. Flat fielding is generally not done at this stage but as part of **dofoe**. The calibration spectra are flat field observations in both fibers, comparison arc lamp spectra in both fibers, and arc spectra in one fiber while the second fiber observes the object. If for some reason the flat field or calibration arc spectra have separate exposures for the two fibers the separate exposures may simply be added.

The assignment of arc calibration exposures to object exposures is generally done by selecting the nearest in time and interpolating. However, the optional *arc assignment table* may be used to explicitly assign arc images to specific objects. The format of this file is described in the task **refspectra**.

The final reduced spectra are recorded in two or three dimensional IRAF images. The images have the same name as the original images with an added ".ec" extension. Each line in the reduced image is a one dimensional spectrum (an echelle order) with associated aperture and wavelength information. When the `extras` parameter is set the lines in the third dimension contain additional information (see **apsum** for further details). These spectral formats are accepted by the one dimensional spectroscopy tasks such as the plotting tasks **splot** and **specplot**. The special task **scopy** may be used to extract specific apertures or to change format to individual one dimensional images. The task **scombine** is used to combine or merge orders into a single spectrum.

**4. Package Parameters**

The **echelle** package parameters, shown in Figure 1, set parameters affecting all the tasks in the package. Some of the parameters are not applicable to the **dofoe** task.

Figure 1: Package Parameter Set for the ECHELLE Package

```
                          I R A F
            Image Reduction and Analysis Facility
PACKAGE = imred
   TASK = echelle

(extinct= onedstds$kpnoextinct.dat) Extinction file
(caldir = onedstds$spechayescal/) Standard star calibration directory
(observa=  observatory) Observatory of data
(interp =        poly5) Interpolation type
(dispaxi=            2) Image axis for 2D images
(nsum   =            1) Number of lines/columns to sum for 2D images

(databas=     database) Database
(verbose=           no) Verbose output?
(logfile=      logfile) Text log file
(plotfil=              ) Plot file

(records=                      ) Record number extensions
(version= ECHELLE V3: July 1991)
```

The observatory parameter is only required for data without an OBSERVAT header parameter (currently included in NOAO data). The spectrum interpolation type might be changed to "sinc" but with the cautions given in **onedspec.package**. The dispersion axis parameter is only needed if a DISPAXIS image header parameter is not defined. The other parameters define the standard I/O functions. The verbose parameter selects whether to print everything which goes into the log file on the terminal. It is useful for monitoring what the **dofoe** task does. The log and plot files are useful for keeping a record of the processing. A log file is highly recommended. A plot file provides a record of apertures, traces, and extracted spectra but can become quite large. The plotfile is most conveniently viewed and printed with **gkimosaic**.

**5. Processing Parameters**

The **dofoe** parameters are shown in Figure 2.

Figure 2: Parameters Set for DOFOE

```
                          I R A F
            Image Reduction and Analysis Facility
PACKAGE = echelle
   TASK = dofoe

objects =               List of object spectra
(apref  =             ) Aperture reference spectrum
(flat   =             ) Flat field spectrum
(arcs   =             ) List of arc spectra
(arctabl=             ) Arc assignment table (optional)

(readnoi=           0.) Read out noise sigma (photons)
(gain   =           1.) Photon gain (photons/data number)
(dispaxi=  )_.dispaxis) Dispersion axis (1=along lines, 2=along columns)
(norders=           12) Number of orders
(width  =           4.) Width of profiles (pixels)
(arcaps =          2x2) Arc apertures

(fitflat=          yes) Fit and ratio flat field spectrum?
```

```
(backgro=         none) Background to subtract
(clean  =           no) Detect and replace bad pixels?
(dispcor=          yes) Dispersion correct spectra?
(redo   =           no) Redo operations if previously done?
(update =           no) Update spectra if cal data changes?
(batch  =           no) Extract objects in batch?
(listonl=           no) List steps but don't process?

(params =             ) Algorithm parameters
```

The input images are specified by image lists. The lists may be a list of explicit, comma separate image names, @ files, or image templates using pattern matching against file names in the directory. The aperture reference spectrum is used to find the orders and trace them. Thus, this requires an image with good signal in both fibers which usually means a flat field spectrum. It is recommended that flat field correction be done using one dimensional extracted spectra rather than as two dimensional images. This is done if a flat field spectrum is specified. The arc assignment table is used to specifically assign arc spectra to particular object spectra and the format of the file is described in **refspectra**.

The detector read out noise and gain are used for cleaning and variance (optimal) extraction. The dispersion axis defines the wavelength direction of spectra in the image if not defined in the image header by the keyword DISPAXIS. The width parameter (in pixels) is used for the profile centering algorithm (**center1d**).

The number of orders selects the number of "pairs" of object and arc fiber profiles to be automatically found based on the strongest peaks. Because it is important that both elements of a pair be found, no orders be skipped, and the aperture numbers be sequential with arc profiles having even aperture numbers and object profiles having odd numbers in the region of interest, the automatic identification is just a starting point for the interactive review. The even/odd relationship between object and arc profiles is set by the `arcaps` parameter and so may be reversed if desired.

The next set of parameters select the processing steps and options. The flat fitting option allows fitting and removing the overall shape of the flat field spectra while preserving the pixel-to-pixel response corrections. This is useful for maintaining the approximate object count levels, including the blaze function, and not introducing the reciprocal of the flat field spectrum into the object spectra. If not selected the flat field will remove the blaze function from the observations and introduce some wavelength dependence from the flat field lamp spectrum.

The `background` option selects the type of correction for background or scattered light. If the type is "scattered" a global scattered light is fit to the data between the apertures and subtracted from the images. *Note that the input images are modified by this operation*. This option is slow. Alternatively, a local background may be subtracted using background regions defined for each aperture. The data in the regions may be averaged, medianed, or the minimum value used. Another choice is to fit the data in the background regions by a function and interpolate to the object aperture.

The `clean` option invokes a profile fitting and deviant point rejection algorithm as well as a variance weighting of points in the aperture. These options require knowing the effective (i.e. accounting for any image combining) read out noise and gain. For a discussion of cleaning and variance weighted extraction see **apvariance** and **approfiles**.

The dispersion correction option selects whether to extract arc spectra, determine a dispersion function, assign them to the object spectra, and, possibly, resample the spectra to a linear (or log-linear) wavelength scale.

Generally once a spectrum has been processed it will not be reprocessed if specified as an input spectrum. However, changes to the underlying calibration data can cause such spectra to be reprocessed if the `update` flag is set. The changes which will cause an update are a new reference image, new flat field, adding the scattered light option, and a new arc reference image. If all input spectra are to be processed regardless of previous processing the `redo` flag may be used. Note that reprocessing clobbers the previously processed output spectra.

The `batch` processing option allows object spectra to be processed as a background or batch job. The `listonly` option prints a summary of the processing steps which will be performed on the input spectra without actually doing anything. This is useful for verifying which spectra will be affected if the input list contains previously processed spectra. The listing does not include any arc spectra which may be extracted to dispersion calibrate an object spectrum.

The last parameter (excluding the task mode parameter) points to another parameter set for the algorithm parameters. The way **dofoe** works this may not have any value and the parameter set **params** is always used. The algorithm parameters are discussed further in the next section.

## 6. Algorithms and Algorithm Parameters

This section summarizes the various algorithms used by the **dofoe** task and the parameters which control and modify the algorithms. The algorithm parameters available to the user are collected in the parameter set **params**. These parameters are taken from the various general purpose tasks used by the **dofoe** processing task. Additional information about these parameters and algorithms may be found in the help for the actual task executed. These tasks are identified in the parameter section listing in parenthesis. The aim of this parameter set organization is to collect all the algorithm parameters in one place separate from the processing parameters and include only those which are relevant for FOE data. The parameter values can be changed from the defaults by using the parameter editor,

```
cl> epar params
```

or simple typing `params`. The parameter editor can also be entered when editing the **dofoe** parameters by typing `:e params` or simply `:e` if positioned at the `params` parameter. Figure 3 shows the parameter set.

Figure 3: Algorithm Parameter Set

```
                        I R A F
          Image Reduction and Analysis Facility
PACKAGE = echelle
   TASK = params

(line   =        INDEF) Default dispersion line
(nsum   =           10) Number of dispersion lines to sum
(extras =           no) Extract sky, sigma, etc.?

                        -- DEFAULT APERTURE LIMITS --
(lower  =          -3.) Lower aperture limit relative to center
(upper  =           3.) Upper aperture limit relative to center

                        -- AUTOMATIC APERTURE RESIZING PARAMETERS --
(ylevel =         0.05) Fraction of peak or intensity for resizing

                        -- TRACE PARAMETERS --
(t_step =           10) Tracing step
(t_funct=      spline3) Trace fitting function
(t_order=            2) Trace fitting function order
(t_niter=            1) Trace rejection iterations
(t_low  =           3.) Trace lower rejection sigma
(t_high =           3.) Trace upper rejection sigma

                        -- DEFAULT BACKGROUND PARAMETERS --
(buffer =           1.) Buffer distance from apertures
(apscat1=             ) Fitting parameters across the dispersion
(apscat2=             ) Fitting parameters along the dispersion
(b_funct=     legendre) Background function
(b_order=            2) Background function order
(b_sampl=  -10:-6,6:10) Background sample regions
(b_naver=           -3) Background average or median
(b_niter=            0) Background rejection iterations
(b_low  =           3.) Background lower rejection sigma
(b_high =           3.) Background upper rejection sigma
(b_grow =           0.) Background rejection growing radius
(b_smoot=           10) Background smoothing length
```

```
                        -- APERTURE EXTRACTION PARAMETERS --
(weights=         none) Extraction weights (none|variance)
(pfit   =        fit1d) Profile fitting algorithm (fit1d|fit2d)
(lsigma =           3.) Lower rejection threshold
(usigma =           3.) Upper rejection threshold

                        -- FLAT FIELD FUNCTION FITTING PARAMETERS --
(f_inter=           no) Fit flat field interactively?
(f_funct=      spline3) Fitting function
(f_order=           20) Fitting function order

                        -- ARC DISPERSION FUNCTION PARAMETERS --
(coordli= linelist$thorium.dat) Line list
(match  =           1.) Line list matching limit in Angstroms
(fwidth =           4.) Arc line widths in pixels
(cradius=           4.) Centering radius in pixels
(i_funct=    chebyshev) Echelle coordinate function
(i_xorde=            3) Order of coordinate function along dispersion
(i_yorde=            3) Order of coordinate function across dispersion
(i_niter=            3) Rejection iterations
(i_low  =           3.) Lower rejection sigma
(i_high =           3.) Upper rejection sigma
(refit  =          yes) Refit coordinate function when reidentifying?

                        -- AUTOMATIC ARC ASSIGNMENT PARAMETERS --
(select =       interp) Selection method for reference spectra
(sort   =           jd) Sort key
(group  =          ljd) Group key
(time   =           no) Is sort key a time?
(timewra=          17.) Time wrap point for time sorting

                        -- DISPERSION CORRECTION PARAMETERS --
(lineari=          yes) Linearize (interpolate) spectra?
(log    =           no) Logarithmic wavelength scale?
(flux   =          yes) Conserve flux?
```

### 6.1. Aperture Definitions

The first operation is to define the extraction apertures, which include the aperture width, background regions, and position dependence with wavelength, for the object and arc orders of interest. This is done on a reference spectrum which is usually a flat field taken through both fibers. Other spectra will inherit the reference apertures and apply a correction for any shift of the orders across the dispersion. The reference apertures are defined only once unless the `redo` option is set.

The selected number of orders are found automatically by selecting the highest peaks in a cut across the dispersion. Note that the specified number of orders is multiplied by two in defining the apertures. Apertures are assigned with a limits set by the `lower` and `upper` parameter and numbered sequentially. A query is then given allowing the aperture limits to be "resized" based on the profile itself (see **apresize**).

A cut across the orders is then shown with the apertures marked and an interactive aperture editing mode is entered (see **apedit**). For **dofoe** the aperture identifications and numbering is particularly critical. All "pairs" of object and arc orders in the region of interest must be defined without skipping any orders. The orders must also be numbered sequentially (though the direction does not matter) so that the arc apertures are either all even or all odd as defined by the `arcaps` parameter (the default is even numbers for the arc apertures). The 'o' key will provide the necessary reordering. If local background subtraction is used the background regions should also be checked with the 'b' key. Typically one adjusts all the background regions at the same time by selecting all apertures with the 'a' key first. To exit the background and aperture editing steps type

Next the positions of the orders at various points along the dispersion are measured and "trace functions" are fit. The user is asked whether to fit each trace function interactively. This is selected to adjust the fitting parameters such as function type and order. When interactively fitting a query is given for each aperture. After the first aperture one

may skip reviewing the other traces by responding with "NO". Queries made by **dofoe** generally may be answered with either lower case "yes" or "no" or with upper case "YES" or "NO". The upper case responses apply to all further queries and so are used to eliminate further queries of that kind.

The above steps are all performed using tasks from the **apextract** package and parameters from the **params** parameters. As a quick summary, the dispersion direction of the spectra are determined from the **dispaxis** parameter if not defined in the image header. The default line or column for finding the orders and the number of image lines or columns to sum are set by the `line` and `nsum` parameters. A line of INDEF (the default) selects the middle of the image. The automatic finding algorithm is described for the task **apfind** and basically finds the strongest peaks. The resizing is described in the task **apresize** and the parameters used are also described there and identified in the PARAMETERS section. The tracing is done as described in **aptrace** and consists of stepping along the image using the specified `t_step` parameter. The function fitting uses the **icfit** commands with the other parameters from the tracing section.

### 6.2. Background or Scattered Light Subtraction

In addition to not subtracting any background scattered light there are two approaches to subtracting this light. The first is to determine a smooth global scattered light component. The second is to subtract a locally determined background at each point along the dispersion and for each aperture. Note that background subtraction is only done for object images and not for arc images.

The global scattered light fitting and subtraction is done with the task **apscatter**. The function fitting parameters are set interactively using the aperture reference spectrum. All other subtractions are done noninteractively with the same set of parameters. The scattered light is subtracted from the input images, thus modifying them, and one might wish to first make backups of the original images.

The scattered light is measured between the apertures using a specified buffer distance from the aperture edges. The scattered light pixels are fit by a series of one dimensional functions across the dispersion. The independent fits are then smoothed along the dispersion by again fitting low order functions. These fits then define the smooth scattered light surface to be subtracted from the image. The fitting parameters are defined and recorded in the two parameter sets `apscat1` and `apscat2`. The scattered light algorithm is described more fully in **apscatter**. This algorithm is relatively slow.

Local background subtraction is done during extraction based on background regions and parameters defined by the default background parameters or changed during interactive review of the apertures. The background subtraction options are to subtract the average, median, or minimum of the pixels in the background regions, or to fit a function and subtract the function from under the extracted object pixels. The background regions are specified in pixels from the aperture center and follow changes in center of the spectrum along the dispersion. The syntax is colon separated ranges with multiple ranges separated by a comma or space. The background fitting uses the **icfit** routines which include medians, iterative rejection of deviant points, and a choice of function types and orders. Note that it is important to use a method which rejects cosmic rays such as using either medians over all the background regions (`background` = "median") or median samples during fitting (`b_naverage` < -1). The background smoothing parameter `b_smooth` is may be used to provide some additional local smoothing of the background light. The background subtraction algorithm and options are described in greater detail in **apsum** and **apbackground**.

### 6.3. Extraction

The actual extraction of the spectra is done by summing across the fixed width apertures at each point along the dispersion. The default is to simply sum the pixels using partial pixels at the ends. There is an option to weight the sum based on a Poisson noise model using the `readnoise` and `gain` detector parameters. Note that if the `clean` option is selected the variance weighted extraction is used regardless of the `weights` parameter. The sigma threshold for cleaning are also set in the **params** parameters.

The cleaning and variance weighting options require knowing the effective (i.e. accounting for any image combining) read out noise and gain. These numbers need to be adjusted if the image has been processed such that the intensity scale has a different origin (such as a scattered light subtraction) or scaling (such as caused by unnormalized flat fielding). These options also require using background subtraction if the profile does not go to zero. For optimal extraction and cleaning to work it is recommended that any scattered light be accounted for by local background subtraction rather than with the scattered light subtraction and the `fitflat` option be used. The `b_smooth` parameter is also appropriate in this application and improves the optimal extraction results by reducing noise in the background signal. For further discussion of cleaning and variance weighted extraction see **apvariance** and **approfiles** as well as **apsum**.

### 6.4. Flat Field Correction

Flat field corrections may be made during the basic CCD processing; i.e. direct division by the two dimensional flat field observation. In that case do not specify a flat field spectrum; use the null string "". The **dofoe** task provides an alternative flat field response correction based on division of the extracted object spectra by the extracted flat field spectra. A discussion of the theory and merits of flat fielding directly verses using the extracted spectra will not be made here. The **dofoe** flat fielding algorithm is the *recommended* method for flat fielding since it works well and is not subject to the many problems involved in two dimensional flat fielding.

The first step is extraction of the flat field spectrum, if one is specified, using the reference apertures. Only one flat field is allowed so if multiple flat fields are required the data must be reduced in groups. When the `fitflat` option is selected (the default) the extracted flat field spectra are fit by smooth functions and the ratio of the flat field spectra to the smooth functions define the response spectra. The default fitting function and order are given by the parameters `f_function` and `f_order`. If the parameter `f_interactive` is "yes" then the fitting is done interactively using the **fit1d** task which uses the **icfit** interactive fitting commands.

If the `fitflat` option is not selected the extracted and globally normalized flat field spectra are directly divided in the object spectra. This removes the blaze function, thus altering the data counts, and introduces the reciprocal of the flat field spectrum in the object spectra.

The final step is to normalize the flat field spectra by the mean counts over all the fibers. This normalization step is simply to preserve the average counts of the extracted object and arc spectra after division by the response spectra.

### 6.5. Dispersion Correction

If dispersion correction is not selected, `dispcor`=no, then the object spectra are simply extracted. If it is selected the arc spectra are used to dispersion calibrate the object spectra. There are four steps involved; determining the dispersion functions relating pixel position to wavelength, assigning the appropriate dispersion function to a particular observation, determining a zero point wavelength shift from the arc fiber to be applied to the object fiber dispersion function, and either storing the nonlinear dispersion function in the image headers or resampling the spectra to evenly spaced pixels in wavelength.

The first arc spectrum in the arc list is used to define the reference dispersion solution. It is extracted using the reference aperture definitions. Note extractions of arc spectra are not background or scattered light subtracted. The interactive task **ecidentify** is used to define the dispersion function in one fiber. The idea is to mark some lines in a few orders whose wavelengths are known (with the line list used to supply additional lines after the first few identifications define the approximate wavelengths) and to fit a function giving the wavelength from the aperture number and pixel position. The dispersion function for the second fiber is then determined automatically by reference to the first fiber using the task **ecreidentify**.

The arc dispersion function parameters are for **ecidentify** and it's related partner **ecreidentify**. The parameters define a line list for use in automatically assigning wavelengths to arc lines, a centering width (which should match the line widths at the base of the lines), the dispersion function type and orders, parameters to exclude bad lines from function fits, and defining whether to refit the dispersion function as opposed to simply determining a zero point shift. The defaults should generally be adequate and the dispersion function fitting parameters may be altered interactively. One should consult the help for the two tasks for additional details of these parameters and the interactive operation of **ecidentify**.

Once the reference dispersion functions are defined other arc spectra are extracted as they are assign to the object spectra. The assignment of arcs is done either explicitly with an arc assignment table (parameter `arctable`) or based on a header parameter such as a time. The assignments are made by the task **refspectra**. When two arcs are assigned to an object spectrum an interpolation is done between the two dispersion functions. This makes an approximate correction for steady drifts in the dispersion. Because the arc fiber monitors any zero point shifts in the dispersion functions it is probably only necessary to have one or two arc spectra, one at the beginning and/or one at the end of the night.

The tasks **setjd** and **setairmass** are automatically run on all spectra. This computes and adds the header parameters for the Julian date (JD), the local Julian day number (LJD), the universal time (UTMIDDLE), and the air mass at the middle of the exposure. The default arc assignment is to use the Julian date grouped by the local Julian day number. The grouping allows multiple nights of data to be correctly assigned at the same time.

Defining the dispersion function for a new arc extraction is done with the task **ecreidentify**. This is done noninteractively with log information recorded about the line reidentifications and the fit.

From the one or two arc spectra come two full dispersion function, one for the object fiber and one for the arc fiber. When an object spectrum is extracted so is the simultaneous arc spectrum. A zero point shift of the arc spectrum relative to the dispersion solution of the dual arc observation is computed using **ecreidentify** (`refit`=no). This zero

point shift is assumed to be the same for the object fiber and it is added to the dispersion function of the dual arc observation for the object fiber. Note that this does not assume that the object and arc fiber dispersion functions are the same or have the same wavelength origin, but only that the same shift in wavelength zero point applies to both fibers. Once the dispersion function correction is determined from the extracted arc fiber spectrum it is deleted leaving only the object spectrum.

The last step of dispersion correction is setting the dispersion of the object spectrum. There are two choices here. If the `linearize` parameter is not set the nonlinear dispersion function is stored in the image header. Other IRAF tasks interpret this information when dispersion coordinates are needed for plotting or analysis. This has the advantage of not requiring the spectra to be interpolated and the disadvantage that the dispersion information is only understood by IRAF tasks and cannot be readily exported to other analysis software.

If the `linearize` parameter is set then the spectra are resampled to a linear dispersion relation either in wavelength or the log of the wavelength. For echelle spectra each order is linearized independently so that the wavelength interval per pixel is different in different orders. This preserves most of the resolution and avoids over or under sampling of the highest or lowest dispersion orders. The wavelength limits are taken from the limits determined from the arc reference spectrum and the number of pixels is the same as the original images. The dispersion per pixel is then derived from these constraints.

The linearization algorithm parameters allow selecting the interpolation function type, whether to conserve flux per pixel by integrating across the extent of the final pixel, and whether to linearize to equal linear or logarithmic intervals. The latter may be appropriate for radial velocity studies. The default is to use a fifth order polynomial for interpolation, to conserve flux, and to not use logarithmic wavelength bins. These parameters are described fully in the help for the task **dispcor** which performs the correction.

## 7. References

### 7.1. IRAF Introductory References

Work is underway on a new introductory guide to IRAF. Currently, the work below is the primary introduction.

P. Shames and D. Tody, *A User's Introduction to the IRAF Command Language*, Central Computer Services, NOAO, 1986.

### 7.2. CCD Reductions

F. Valdes, *The IRAF CCD Reduction Package -- CCDRED*, Central Computer Services, NOAO, 1987.

F. Valdes, *User's Guide to the CCDRED Package*, Central Computer Services, NOAO, 1988. Also on-line as `help ccdred.guide`.

P. Massey, *A User's Guide to CCD Reductions with IRAF*, Central Computer Services, NOAO, 1989.

### 7.3. Aperture Extraction Package

F. Valdes, *The IRAF APEXTRACT Package*, Central Computer Services, NOAO, 1987 (out-of-date).

### 7.4. Task Help References

Each task in the **specred** packages and tasks used by **dofibers** have help pages describing the parameters and task in some detail. To get on-line help type

`cl> help` *taskname*

The output of this command can be piped to **lprint** to make a printed copy.

```
      apall - Extract 1D spectra (all parameters in one task)
  apdefault - Set the default aperture parameters and apidtable
     apedit - Edit apertures interactively
     apfind - Automatically find spectra and define apertures
      apfit - Fit 2D spectra and output the fit, difference, or ratio
  apflatten - Remove overall spectral and profile shapes from flat fields
     apmask - Create and IRAF pixel list mask of the apertures
apnormalize - Normalize 2D apertures by 1D functions
 aprecenter - Recenter apertures
   apresize - Resize apertures
```

```
  apscatter - Fit and subtract scattered light
      apsum - Extract 1D spectra
    aptrace - Trace positions of spectra

      bplot - Batch plots of spectra
  calibrate - Apply extinction and flux calibrations to spectra
  continuum - Fit the continuum in spectra
   deredden - Apply interstellar extinction corrections
    dispcor - Dispersion correct spectra
     dopcor - Doppler correct spectra
 ecidentify - Identify features in spectrum for dispersion solution
ecreidentify - Automatically identify features in spectra
 refspectra - Assign wavelength reference spectra to other spectra
     sarith - Spectrum arithmetic
   scombine - Combine spectra
      scopy - Select and copy apertures in different spectral formats
   sensfunc - Create sensitivity function
 setairmass - Compute effective airmass and middle UT for an exposure
      setjd - Compute and set Julian dates in images
      slist - List spectrum header parameters
   specplot - Stack and plot multiple spectra
      splot - Preliminary spectral plot/analysis
   standard - Identify standard stars to be used in sensitivity calc

      dofoe - Process Fiber Optic Echelle spectra
      demos - Demonstrations and tests

            Additional help topics

   onedspec.package - Package parameters and general description of package
  apextract.package - Package parameters and general description of package
 approfiles - Profile determination algorithms
 apvariance - Extractions, variance weighting, cleaning, and noise model
   center1d - One dimensional centering algorithm
      icfit - Interactive one dimensional curve fitting
```

**Appendix A: DOFOE Parameters**

objects

List of object spectra to be processed. Previously processed spectra are ignored unless the `redo` flag is set or the `update` flag is set and dependent calibration data has changed. Extracted spectra are ignored.

apref = ""

Aperture reference spectrum. This spectrum is used to define the basic extraction apertures and is typically a flat field spectrum.

flat = "" (optional)

Flat field spectrum. If specified the one dimensional flat field spectrum is extracted and used to make flat field calibrations.

arcs = "" (at least one if dispersion correcting)

List of arc spectra in which both fibers have arc spectra. These spectra are used to define the dispersion functions for each fiber apart from a zero point correction made with the arc fiber during an observation. One fiber from the first spectrum is used to mark lines and set the dispersion function interactively and dispersion functions for the other fiber and arc spectra are derived from it.

arctable = "" (optional) (refspectra)

Table defining arc spectra to be assigned to object spectra (see **refspectra**). If not specified an assignment based on a header parameter, `params.sort`, such as the observation time is made.

readnoise = "0." (apsum)

Read out noise in photons. This parameter defines the minimum noise sigma. It is defined in terms of photons (or electrons) and scales to the data values through the gain parameter. A image header keyword (case insensitive) may be specified to get the value from the image.

gain = "1." (apsum)
Detector gain or conversion factor between photons/electrons and data values. It is specified as the number of photons per data value. A image header keyword (case insensitive) may be specified to get the value from the image.

dispaxis = ")_.dispaxis" (apextract)
Default dispersion axis. The dispersion axis is 1 for dispersion running along image lines and 2 for dispersion running along image columns. If the image header parameter DISPAXIS is defined it has precedence over this parameter. The default value defers to the package parameter of the same name.

norders = 12 (apfind)
Number of orders to be found. This number is used during the automatic definition of the apertures from the aperture reference spectrum. Note that the number of apertures defined is twice this number, one set for the object fiber orders and one set for the arc fiber orders. The interactive review of the aperture assignments allows verification and adjustments to the automatic aperture definitions.

width = 4. (apedit)
Approximate base full width of the fiber profiles. This parameter is used for the profile centering algorithm.

arcaps = "2x2"
List of arc fiber aperture numbers. Since the object and arc fiber orders are paired the default setting expects the even number apertures to be the are apertures. This should be checked interactively.

fitflat = yes (flat1d)
Fit and divide the extracted flat field field orders by a smooth function in order to normalize the wavelength response? If not done the flat field spectral shape (which includes the blaze function) will be divided out of the object spectra, thus altering the object data values. If done only the small scale response variations are included in the flat field and the object spectra will retain their observed flux levels and blaze function.

background = "none" (apsum, apscatter)
Type of background light subtraction. The choices are "none" for no background subtraction, "scattered" for a global scattered light subtraction, "average" to average the background within background regions, "median" to use the median in background regions, "minimum" to use the minimum in background regions, or "fit" to fit across the dispersion using the background within background regions. The scattered light option fits and subtracts a smooth global background and modifies the input images. This is a slow operation and so is NOT performed in quicklook mode. The other background options are local to each aperture at each point along the dispersion. The "fit" option uses additional fitting parameters from **params** and the "scattered" option uses parameters from **apscat1** and **apscat2**.

clean = yes (apsum)
Detect and correct for bad pixels during extraction? This is the same as the clean option in the **apextract** package. If yes this also implies variance weighted extraction and requires reasonably good values for the readout noise and gain.

dispcor = yes
Dispersion correct spectra? Depending on the `params.linearize` parameter this may either resample the spectra or insert a dispersion function in the image header.

redo = no
Redo operations previously done? If no then previously processed spectra in the objects list will not be processed (unless they need to be updated).

update = no
Update processing of previously processed spectra if aperture, flat field, or dispersion reference definitions are changed?

batch = no
Process spectra as a background or batch job.

listonly = no
List processing steps but don’t process?

params = "" (pset)
Name of parameter set containing additional processing parameters. The default is parameter set **params**. The parameter set may be examined and modified in the usual ways (typically with "epar params" or ":e params" from the parameter editor). Note that using a different parameter file is not allowed. The parameters are described below.

## -- PACKAGE PARAMETERS

Package parameters are those which generally apply to all task in the package. This is also true of **dofoe**.

observatory = "observatory"
Observatory at which the spectra were obtained if not specified in the image header by the keyword OBSERVAT. For FOE data the image headers identify the observatory as "kpno" so this parameter is not used. For data from other observatories this parameter may be used as describe in **observatory**.

interp = "poly5" (nearest|linear|poly3|poly5|spline3|sinc)
Spectrum interpolation type used when spectra are resampled. The choices are:

```
    nearest - nearest neighbor
     linear - linear
      poly3 - 3rd order polynomial
      poly5 - 5th order polynomial
    spline3 - cubic spline
       sinc - sinc function
```

dispaxis = 2
Default dispersion axis. The dispersion axis is 1 for dispersion running along image lines and 2 for dispersion running along image columns. If the image header parameter DISPAXIS is defined it has precedence over this parameter.

database = "database"
Database (directory) used for storing aperture and dispersion information.

verbose = no
Print verbose information available with various tasks.

logfile = "logfile", plotfile = ""
Text and plot log files. If a filename is not specified then no log is kept. The plot file contains IRAF graphics metacode which may be examined in various ways such as with **gkimosaic**.

records = ""
Dummy parameter to be ignored.

version = "ECHELLE: ..."
Version of the package.

## PARAMS PARAMETERS

The following parameters are part of the **params** parameter set and define various algorithm parameters for **dofoe**.

### -- GENERAL PARAMETERS --

line = INDEF, nsum = 10
The dispersion line (line or column perpendicular to the dispersion axis) and number of adjacent lines (half before and half after unless at the end of the image) used in finding, recentering, resizing, editing, and tracing operations. A line of INDEF selects the middle of the image along the dispersion axis.

extras = no (apsum)
Include extra information in the output spectra? When cleaning or using variance weighting the cleaned and weighted spectra are recorded in the first 2D plane of a 3D image, the raw, simple sum spectra are recorded in the second plane, and the estimated sigmas are recorded in the third plane.

-- DEFAULT APERTURE LIMITS --

lower = -3., upper = 3. (apdefault)
Default lower and upper aperture limits relative to the aperture center. These limits are used when the apertures are first found and may be resized automatically or interactively.

-- AUTOMATIC APERTURE RESIZING PARAMETERS --

ylevel = 0.05 (apresize)
Data level at which to set aperture limits during automatic resizing. It is a fraction of the peak relative to a local background.

-- TRACE PARAMETERS --

t_step = 10 (aptrace)
Step along the dispersion axis between determination of the spectrum positions. Note the `nsum` parameter is also used to enhance the signal-to-noise at each step.

t_function = "spline3", t_order = 2 (aptrace)
Default trace fitting function and order. The fitting function types are "chebyshev" polynomial, "legendre" polynomial, "spline1" linear spline, and "spline3" cubic spline. The order refers to the number of terms in the polynomial functions or the number of spline pieces in the spline functions.

t_niterate = 1, t_low = 3., t_high = 3. (aptrace)
Default number of rejection iterations and rejection sigma thresholds.

-- DEFAULT BACKGROUND PARAMETERS --

buffer = 1. (apscatter)
Buffer distance from the edge of any aperture for data to be included in the scattered light determination. This parameter may be modified interactively.

apscat1 = "", apscat2 = "" (apscatter)
Parameter sets for the fitting functions across and along the dispersion. These parameters are those used by **icfit**. These parameters are usually set interactively.

b_function = "legendre", b_order = 1 (apsum)
Default background fitting function and order. The fitting function types are "chebyshev" polynomial, "legendre" polynomial, "spline1" linear spline, and "spline3" cubic spline. The order refers to the number of terms in the polynomial functions or the number of spline pieces in the spline functions.

b_naverage = -100 (apsum)
Default number of points to average or median. Positive numbers average that number of sequential points to form a fitting point. Negative numbers median that number, in absolute value, of sequential points. A value of 1 does no averaging and each data point is used in the fit.

b_niterate = 0 (apsum)
Default number of rejection iterations. If greater than zero the fit is used to detect deviant fitting points and reject them before repeating the fit. The number of iterations of this process is given by this parameter.

b_low_reject = 3., b_high_reject = 3. (apsum)
Default background lower and upper rejection sigmas. If greater than zero points deviating from the fit below and above the fit by more than this number of times the sigma of the residuals are rejected before refitting.

b_smooth = 10 (apsum)
Box car smoothing length for background when using background subtraction. Since the background noise is often the limiting factor for good extraction one may box car smooth the the background to improve the statistics.

-- APERTURE EXTRACTION PARAMETERS --

weights = "none" (apsum)
Type of extraction weighting. Note that if the `clean` parameter is set then the weights used are "variance" regardless of the weights specified by this parameter. The choices are:

"none"
The pixels are summed without weights except for partial pixels at the ends.

"variance"
The extraction is weighted by the variance based on the data values and a poisson/ccd model using the `gain` and `readnoise` parameters.

pfit = "fit1d" (apsum) (fit1d|fit2d)
Profile fitting algorithm for cleaning and variance weighted extractions. The default is generally appropriate for FOE data but users may try the other algorithm. See **approfiles** for further information.

lsigma = 3., usigma = 3. (apsum)
Lower and upper rejection thresholds, given as a number of times the estimated sigma of a pixel, for cleaning.

-- FLAT FIELD FUNCTION FITTING PARAMETERS --

f_interactive = no (fit1d)
Fit the one dimensional flat field order spectra interactively? This is used if `fitflat` is set and a two dimensional flat field spectrum is specified.

f_function = "spline3", f_order = 20 (fit1d)
Function and order used to fit the composite one dimensional flat field spectrum. The functions are "legendre", "chebyshev", "spline1", and "spline3". The spline functions are linear and cubic splines with the order specifying the number of pieces.

-- ARC DISPERSION FUNCTION PARAMETERS --

coordlist = "linelist$thorium.dat" (ecidentify)
Arc line list consisting of an ordered list of wavelengths. Some standard line lists are available in the directory "linelist$".

match = 1. (ecidentify)
The maximum difference for a match between the dispersion function computed value and a wavelength in the coordinate list.

fwidth = 4. (ecidentify)
Approximate full base width (in pixels) of arc lines.

cradius = 4. (reidentify)
Radius from previous position to reidentify arc line.

i_function = "chebyshev", i_xorder = 3, i_yorder = 3 (ecidentify)
The default function, function order for the pixel position dependence, and function order for the aperture number dependence to be fit to the arc wavelengths. The functions choices are "chebyshev" or "legendre".

i_niterate = 3, i_low = 3.0, i_high = 3.0 (ecidentify)
Number of rejection iterations and sigma thresholds for rejecting arc lines from the dispersion function fits.

refit = yes (ecreidentify)
Refit the dispersion function? If yes and there is more than 1 line and a dispersion function was defined in the arc reference then a new dispersion function of the same type as in the reference image is fit using the new pixel positions. Otherwise only a zero point shift is determined for the revised fitted coordinates without changing the form of the dispersion function.

-- AUTOMATIC ARC ASSIGNMENT PARAMETERS --

select = "interp" (refspectra)
Selection method for assigning wavelength calibration spectra. Note that an arc assignment table may be used

to override the selection method and explicitly assign arc spectra to object spectra. The automatic selection methods are:

average
Average two reference spectra without regard to any sort parameter. If only one reference spectrum is specified then it is assigned with a warning. If more than two reference spectra are specified then only the first two are used and a warning is given. This option is used to assign two reference spectra, with equal weights, independent of any sorting parameter.

following
Select the nearest following spectrum in the reference list based on the sorting parameter. If there is no following spectrum use the nearest preceding spectrum.

interp
Interpolate between the preceding and following spectra in the reference list based on the sorting parameter. If there is no preceding and following spectrum use the nearest spectrum. The interpolation is weighted by the relative distances of the sorting parameter.

match
Match each input spectrum with the reference spectrum list in order. This overrides the reference aperture check.

nearest
Select the nearest spectrum in the reference list based on the sorting parameter.

preceding
Select the nearest preceding spectrum in the reference list based on the sorting parameter. If there is no preceding spectrum use the nearest following spectrum.

sort = "jd", group = "ljd" (refspectra)
Image header keywords to be used as the sorting parameter for selection based on order and to group spectra. A null string, "", or the word "none" may be use to disable the sorting or grouping parameters. The sorting parameter must be numeric but otherwise may be anything. The grouping parameter may be a string or number and must simply be the same for all spectra within the same group (say a single night). Common sorting parameters are times or positions. In **dofoe** the Julian date (JD) and the local Julian day number (LJD) at the middle of the exposure are automatically computed from the universal time at the beginning of the exposure and the exposure time. Also the parameter UTMIDDLE is computed.

time = no, timewrap = 17. (refspectra)
Is the sorting parameter a 24 hour time? If so then the time origin for the sorting is specified by the timewrap parameter. This time should precede the first observation and follow the last observation in a 24 hour cycle.

-- DISPERSION CORRECTION PARAMETERS --

linearize = yes (dispcor)
Interpolate the spectra to a linear dispersion sampling? If yes the spectra will be interpolated to a linear or log linear sampling If no the nonlinear dispersion function(s) from the dispersion function database are assigned to the input image world coordinate system and the spectral data are not interpolated.

log = no (dispcor)
Use linear logarithmic wavelength coordinates? Linear logarithmic wavelength coordinates have wavelength intervals which are constant in the logarithm of the wavelength.

flux = yes (dispcor)
Conserve the total flux during interpolation? If `no` the output spectrum is interpolated from the input spectrum at each output wavelength coordinate. If `yes` the input spectrum is integrated over the extent of each output pixel. This is slower than simple interpolation.