WYFFOS Data Reduction Manual

Royal Greenwich Observatory
Madingley Road,
Cambridge CB3 0HJ

J.R. Lewis
October 1996

Contents

1 Introduction

In this document we describe **wyf_red**, a data reduction task specialised for the extraction of the multifibre spectra generated by WYFFOS. This is an IRAF script task which is based on the generalised fibre reduction task **dofibers** which was written by Frank Valdes[1] at NOAO. It does differ from **dofibers** (and its variants) in a number of important respects:

CCD processing With tasks such as **dofibers** you are expected to have done the CCD processing *before* using the relevant extraction task. This is not the case with **wyf_red** which will do any requested CCD processing itself. You can, of course, do this yourself if you want.

Aperture identification One of the major headaches of multi-object spectroscopy is the book-keeping involved. The way that tasks like **dofibers** solve this problem is for the user to define an **aperture identification table** which cross references the spectra with a title and an object type (object, sky, *etc.*) The problem is solved similarly by **wyf_red** with the exception that the aperture identification table is generated automatically by information in the input file header. (This is especially useful for WYFFOS, since the header information does not form a one to one correspondence with the output spectra as a number of the fibres in the large fibre bundle were accidently jumbled before being cemented to the slit configuration.)

Header compression At the time of writing the files generated by WYFFOS and AUTOFIB2 contain almost 3000 FITS header items. This is massive overkill for most applications and most packages will be slowed down by the sheer weight of such a big header. For this reason **wyf_red** will rewrite the header of an input file into a 'compressed' format. No information is lost in the compression and the routine which does the compression can spot occasional mistakes in the header records which at the time of writing still manage to sneak into the files (although these are very few). Appendix E gives a detailed description of the compressed header format and how it relates to the raw header.

File matching With multi-object spectroscopy the onus is usually on the user to make sure that (for example) the object frames and the offset sky frames which are fed to the reduction task were done with the same fibre configuration. Again this is sorted out automatically by **wyf_red** using information contained in a standard WYFFOS header. This gives the user considerable freedom in how the files to be reduced are specified. This will be discussed in more detail in section 4.

[1]The author wishes to thank Frank Valdes for all his help and advice and for his permission to ransack is scripts

Sawtooth correction Most fibre optic spectrographs have the fibre ends in a colinear arrangement at the slit end. This means that (ignoring the distortion of the camera) a given position in the spectral direction will be at roughly the same wavelength for each spectrum. However for WYFFOS, in order to try and save space at the detector, the fibre ends were placed in three parallel rows, which means that a given point in the spectral direction can have three (approximate) wavelengths. **wyf_red** does this 'sawtooth' correction explicitly as each spectrum is being extracted by shifting its world coordinate system by an amount which approximately reflects the amount of offset at the slit configuration. N.B. The need to make this correction means that for the most part tasks such as **dofibers** will fail for WYFFOS data.

End finding WYFFOS can observe at resolutions so low that the spectral range of the spectrograph is actually smaller than the total spectral range which could be accommodated on the chip. While this isn't necessarily a 'problem' for extraction, it can be a problem when it comes to doing the spectral flat fielding – that is a sharp 'cliff' at either edge of the mean flat field spectrum is nearly impossible to fit correctly and will introduce numerous wiggles into the resulting spectra, no matter how high an order polynomial or spline is used. Once **wyf_red** has determined the positions of all of the spectra and their distortions in the spatial direction using the aperture reference image (see below) the spectra on that image can be (at the users discretion) extracted and added together to form one mean spectrum. A routine then would look for sharp edges and decides whether or not the spectra need to be trimmed. This can also be done by hand using a cursor. If the user decides not to search for these cliffs the spectral limits are defined automatically according to the sawtooth correction shifts. Since most users will *not* observe at such low resolutions the default is to skip looking for these cliffs.

Arc line identification The "sawtooth correction" means that **wyf_red** can't use the straightforward scheme which **dofibers** uses for the identification of arc lines. Therefore in **wyf_red** we have three different schemes to choose from, a completely automatic algorithm (everything is done for you), a semi-automatic algorithm (you identify the lines in one spectrum and all the lines in all the other spectra are done for you), and a fully manual algorithm (you identify the lines for each spectrum – rather tedious). These will be discussed more fully in section 4.

In the following section, I describe some of the basic aspects of multi-fibre spectroscopy. This is intended for people who are new to the subject and more experienced observers can probably skip over it. In Section 3, I describe what you need to do before getting started with **wyf_red** (this is recommended reading). Section 4 gives a full description of how to use **wyf_red**. Section 5 includes a list of other applications which are available in the WYFFOS package. These may turn out to be quite helpful and in particular it is highly recommended that you read about the application **wyfgeom** before you start. In Appendix A a description is given of all the parameters for **wyf_red** while in Appendix B a description is given of all the parameters in **wyfpars** (a pset of parameters which rarely need to be changed in **wyf_red**). Information on how to access the parameters in **wyfpars** for those who are unfamiliar with IRAF psets is also given. The package parameters for **wyffos** are included in Appendix C. Appendix D lists which header items are being accessed and why. Finally Appendix E explains the compressed header format.

2 Aspects of Multifibre Spectroscopy

2.1 Bias Subtraction

A "bias" level is introduced onto a CCD chip in order to ensure that the chip is working in a linear regime. A bias also guards against negative numbers at readout which have a statistical chance of occuring in regions of low emission. This bias level must be determined and subtracted out of an input CCD frame before any further analysis can be carried out. This can be done one of two ways. The first is to look at the overscan region of each frame and do a statistical analysis to find the mean bias and then to subtract this constant off the whole rest of the frame. The second is to average a sample of bias frames (frames taken with zero exposure time) pixel by pixel to produce a mean bias frame. This mean frame is then subtracted (pixel by pixel) from each input image. This is the method to choose if you believe there is structure in the bias of your chip. It is important to realise that if you want to use a mean bias frame you need to have at your disposal a good number of individual bias frames in order to form the mean frame (say at least 10, preferably 20). This is in order to beat down the readout noise in the individual bias frames to such a level that you are not adding a significant amount of noise to your program images by doing a frame subtraction.

2.2 Aperture Definition

One of the most critical things to get right in multiobject spectroscopy is the determination of the aperture information for each spectrum. That is the location of each spectrum, the spread of each spectrum in the spatial direction and the distortion of the spectrum in the dispersion direction. The latter is quite important for instruments such as WYFFOS where owing to the field distortions introduced by the optics of the system a spectrum won't necessarily line up along the rows/columns in the dispersion direction (just the opposite is usually the case and spectra can be very distorted, especially near the edge of a frame). Getting the aperture information is also crucial as the ancillary information for each object observed, such as the (α, δ), the object name, *etc.*, is stored in the header of the input frame and it is important for bookkeeping that each spectrum is cross-referenced with the correct header information.

Normally, objects observed with instruments like WYFFOS are quite faint and determining the aperture information for such a frame can be very difficult. This is where an "aperture reference frame" can come in quite handy. This is a high signal/noise frame from which the apertures can be defined and might typically be a flat field exposure. Note that any frame can be used as an aperture reference for another frame (a program frame), provided that the fibres used in the program frame are also illuminated on the aperture reference frame.

One final point to consider with respect to aperture definition is the order in which the fibres appear on the output frame. In a WYFFOS frame the spectra are vertical, so what needs to be considered is whether an object observed in the first available science fibre will appear on the right or the left hand side of the output frame. This depends upon which corner the CCD is being read from. If you don't know which way around the fibres are, then you must contact your support astronomer to find out. The application **wyfgeom** described in section 5 can help you set the appropriate defaults in **wyf_red** once you have this piece of information.

2.3 Scattered Light Correction

All spectrographs have scattered light and WYFFOS is no exception. It is important to get rid of this scattered light, especially if the objects you are observing are faint. **wyf_red** uses the IRAF routine **apscatter** to estimate and subtract out the scattered light component. This is done by fitting polynomials or splines to the emission along the lines and columns of the image, but ignoring the places where the aperture definition says there should be a part of a spectrum. Once **apscatter** has successfully modelled the excess light between the spectra it interpolates to the regions where the spectra are located and then subtracts that model off the frame. It is highly recommended that you fit at least one frame by hand since you will need to choose the order of polynomial/spline which fits your scattered light best.

2.4 Flat Fielding and Throughput Correction

This is an alias for the removal of the multiplicative effects that originate from instrumental signature, the throughput of the fibres, the vignetting (which depends upon the position of the object in the field), and the pixel to pixel variation in response within the chip itself. The instrumental signature and the pixel to pixel variation are found through flat field exposures (exposure of a uniform light source through the fibres). These should be very high signal to noise frames as the pixel to pixel variation is usually only of the order of a few percent. The fibre throughput and vignetting corrections are found through exposures of an 'offset sky' frame. These are frames of sky taken very near to a program region *with the fibres in the same configuration as the program region*, but only of sky. The signal/noise ratio needn't be very high as we need only consider the mean counts per fibre. It is good observing practice to take 2 or 3 offset skies per region as a precaution against starlight accidently falling down one or more of the fibres. The relative transmission of each fibre as seen from this total flux per fibre relative to the average flux of all the fibres will be the product of the field vignetting and the fibre throughput. A constant is used to normalize the results of the vignetting–throughput correction so that in the mean the response is 1. This ensures that the number of counts on a frame remains roughly the same after flat fielding. For **wyf_red** (as with **dofibers** and its relatives) the flat-field frame and the vignetting–throughput frames are combined to form one 'response' frame.

2.5 Extraction

By 'extraction' we mean the summing up of the spectral pixels along the spatial direction into a final one-dimensional spectrum. Admittedly there is a bit more to it than that, especially when low signal/noise data are being extracted. The WYFFOS package uses a method called 'optimal extraction' which attempts to give lower weight to regions of relatively low signal/noise than to those regions with relatively high signal/noise. The net result is that in low signal/noise regions where the signal is of the same order as the readout noise of the chip, the lower weights mean that the contribution to the final spectrum of the readout noise is significantly reduced. In fact the term 'optimal extraction' refers to methods which attempt to maximise the possible signal/noise in the output spectrum.

Because of the way that optimal extraction works, it can predict roughly the number of counts it expects to find in each pixel of a spectrum depending upon the wavelength and its position in the point-spread function. Thus pixel values which are significantly higher than the predicted value can be removed as these are likely to be cosmic ray events. Optimal extraction can cause the removal of many such spurious pixels and result in a much 'cleaner' final spectrum (although it may not still be completely free from cosmic ray events). It is, however, possible to switch off this cleaning algorithm.

2.6 Arc line Identification and Dispersion Correction

Arc line identification is one of the most boring aspects of spectroscopic data reduction. It is rather important though if you want to wavelength calibrate your spectra and sky subtract. For WYFFOS there are three different ways in which arc lines can be identified:

Automatic line identification A routine which automatically identifies arc lines has been included for WYFFOS. This works by a simple projection method and uses information in the headers of the files as a starting point. For arc spectra with a reasonable number of lines (say greater than about 5) this works very well and it is the preferred way to do arcline identification.

Semi-automatic line identification The automatic algorithm may fail for some arcs. Either the number of lines is too small or perhaps the header information is sadly off the mark. Semi-automatic line identification can save the day in a rather painless way. This entails the user identifying lines for one spectrum and the results of this being used to identify the lines in the rest of the arc spectra automatically. This is currently the default setting.

Fully manual line identification If the above two methods fail, there is always the option to identify the lines for each arc spectrum manually. This should not be necessary very often and the reasons for possibly having to use it will be explained later on.

Once the arc lines have been identified, it has to be decided which arc frame applies to which program frame. The assigning of reference arcs is done using automatic criteria which the user can change. Generally what happens is that if there are two arcs available, one on either side of the program frame (chronologically speaking) then the mean solution from both of these frames is taken for the program frame. If only one arc is available, then that arc frame's solution is applied directly to the program frame. 'Dispersion correction' is the act of taking the dispersion solutions found in the reference arcs, applying them to program frames and then using those solutions to rebin the input spectra so that they are on a linear or log linear wavelength scale.

2.7 Sky Subtraction

There are two ways normally to subtract the sky contribution from the fibre spectra. Note that if you want to do a sky subtraction, then you must also do a wavelength calibration!

Simultaneous sky exposure When an object field is configured it is quite normal to use some of the fibres to monitor the sky background. Under most circumstances using these sky fibres to form a mean sky spectrum is the preferred way to do sky subtraction. It is however essential to have at least 10 to 20 dedicated sky fibres in order to get a meaningful mean sky spectrum. Note also that in regions where the sky brightness is very non-uniform (for example, for observations of stars in external galaxies), it is essential that the sky fibres be placed in representative places and that the sky subtraction is *not* done with a mean sky spectrum.

Separate sky frames These are generally already being taken with each program frame, as having an offset sky exposure is essential for the determination of the throughput and vignetting corrections. It is possible to extract the sky frames as an 'object' frame and thus do a spectrum by spectrum subtraction in order to take out the sky contribution from the program spectra. This can be a dangerous way to do sky subtraction if the offset sky is of low signal or in circumstances where the sky background is varying on reasonably short time scales (*e.g.* bright of moon or in conditions of patchy cloud). It can be however quite a useful way of checking the sky subtraction as done with the simultaneous sky fibres. It is also the only way of doing sky subtraction in the event that the observer has not allocated any sky fibres at the configuration stage.

2.8 Residual emission line removal

Sky subtraction is never perfect. Reasons for what appear to be bad sky subtraction include the fact that the vignetting or dispersion corrections may not have been done very well (perhaps there are hardly any lines?). The line spread function may also depend upon the fibre's position relative to the centre of the field. Any of these could lead to sky emission lines being present as either residual emission (not enough sky subtracted off), absorption (too much sky subtracted) or as p-cygni type profiles (bad dispersion correction). Of course, just because you see these effects doesn't necessarily mean that you have bad sky subtraction – a certain amount of these effects may be present even in well sky-subtracted data. As such it is worthwhile getting rid of the residual emission from sky lines just to improve the presentation of the spectra. This is achieved by the user supplying a file with the wavelengths of the sky lines to be 'tippexed' out and the number of Angstroms on either side to define the region of the cleaned line. Note that this won't get rid of the rogue cosmic ray hits, as these don't occur in predictable locations.

3 Getting started

wyf_red is an IRAF task which is composed largely of scripts. This means that much of the functionality of **wyf_red** comes from other IRAF tasks from various other packages. The packages most used by **wyf_red** are **onedspec** and **apextract**. Looking at the help files for these two packages (especially the former) could be *very* beneficial.

Do you know anything about IRAF? Have you ever used it before? If the answer to either of these questions is 'no', then you might consider trying to find out a little more about IRAF before you proceed with **wyf_red**. A very useful document called *A Beginner's Guide to Using IRAF* is available (see the reference list at the end of this document) and a few hours spent reading it will certainly make the learning curve much more bearable. Even experienced IRAF uses may learn a trick or two, so it might be worth a skim even if you are pretty sure you know what you're doing.

Next check and see if you have **wyf_red** on your system. It is in a package called **rgo** which you should see when you first get into the IRAF **cl**. If you don't see **rgo** or if after loading **rgo** you don't see a package called **wyffos**, then you either don't have the **rgo** suite of packages, or you have an old version. In that case contact the author in Cambridge for an up to date version and get your local IRAF manager to install it for you (it should only take a few minutes).

Assuming now you have access to **wyf_red** you'll need to read your files to disc. One major hair in the WYFFOS pie is the enormous header which is generated at the telescope. A standard setup of IRAF will cause a large section of that header to be destroyed and as **wyf_red** really needs most of what is there, then you are likely to run into trouble unless you take a few precautions. The 'offending' IRAF environment variable is called 'min_lenuserarea'. This defines the maximum size of an IRAF file header. When you load the package **wyffos** the value of 'min_lenuserarea' is automatically bumped up to a level which will accommodate raw WYFFOS headers. So before you read your FITS tapes (using **rfits**) it will be worthwhile loading the **wyffos** package to be on the safe side. If you don't want to load the **wyffos** package, you could change 'min_lenuserarea' to a safe value, *e.g.* `cl> reset min_lenuserarea = 300000`, or you could change it in your *login.cl* file. Once you have sorted out the 'min_lenuserarea' problem in whatever way you want, you can then read in your files.

4 Using wyf_red

In this section we describe how to use **wyf_red** to reduce WYFFOS data. Just what you mean by 'reduce' will determine how much of this you need to read. Currently **wyf_red** will CCD process, aperture identify, scattered-light correct, flat field (including throughput and vignetting corrections), extract (including optional cosmic ray removal), dispersion correct, sky subtract. Optionally, residual sky line emission can also be removed ('tippexing'). These topics will all be discussed roughly in turn. If after reading this section you are still confused, have a look at Figure 1. This is a diagram which indicates roughly what gets done to your data and when it happens.

4.1 Input files

One of the first questions that anyone asks when using a new piece of software is "what files do I need?" There are four parameter prompts which need to be answered with the names of files.

The parameter *input* can specify a wide range of files. In order to reduce an object frame you need to have at least one object frame and an offset sky frame (for the throughput correction). If you want to do dispersion correction, you'll need an arc frame as well. At its most basic, *input* must specify all of the object data files. **wyf_red** looks in the headers of the files specified by *input* and decides which files are objects, offset skies and arcs. If, for example, you don't specify an offset sky for a particular object region in the *input* list, then **wyf_red** will look in your current directory for an appropriate offset sky frame. If it finds more than one, then it takes the one which is chronologically closest to the object frame. Similarly with arc frames – if none are specified, **wyf_red** will search for one. Conversely, if, for example, there are two offset sky frames which are appropriate for a given object frame and you want to make sure that one particular one is chosen, then specifying it in the *input* list will mean that the other one will be ignored. (**wyf_red** uses the header parameter *af2fldnm* to decide which offset skies are appropriate for which object frames. See Appendix D for more details.) You can even specify the input list as "*.imh". If you are doing bias subtraction using a bias frame, then you can specify the bias frames in the *input* list. If you don't specify bias frames and you want to do a bias frame subtraction, then **wyf_red** searches the current working directory for bias frames. The same applies for flat–field frames. They can be specified on the *input* list, but this isn't necessary.

The second file parameter is *apref*. This is the aperture reference frame, as described in section 2.2. As mentioned before, this can be almost any frame, so long as it has reasonably high signal to noise and so long as every fibre which has data in the object frames is illuminated in the aperture reference image. If, for example, you are reducing an object frame with reasonably bright objects, then the object frame itself could serve as an aperture reference. Otherwise, an aperture reference could be a twilight sky or flat field exposure.

The last two file parameters are *mbias* and *mflat*. If you want your bias estimate to be determined by the mean of a number of bias frames, then *mbias* is the name of the mean bias frame. If that file exists, then it will be used to do the bias subtraction. If it doesn't then it will be created by forming a mean of whatever bias frames have been specified in the *input* list. If none have been specified in the input list, then *mbias* will be formed from all available bias frames in the current working directory. If *mbias* is left blank, then the bias will be determined for each frame individually along the overscan region of the frame. **wyf_red** should have all the necessary information about the chip at its disposal, so there shouldn't be any problem with determining the location of the bias strip. Similarly, if *mflat* is specified, then any flat-field frames which have been either specified in the *input* list or (failing that) which reside in the current working directory will be combined to form a mean flat frame. If *mflat* is left blank then the flat-fielding will not be done. However the throughput and vignetting correction will still be done with the offset sky exposure.

4.2 Aperture definition

As mentioned before, aperture definition implies locating each spectrum and tracing its spatial distortion. Although you don't have to, it is highly recommended that for the first few regions you reduce, you should do this bit by hand (set the parameter *w_interact* to 'yes'). If you do, you will be presented with a graph which has a cut through the aperture reference at the middle of the frame. On top of each peak should be a number which has been preassigned to the aperture. This is where things could get a little tricky. If you are using a flat-field exposure then there is a pretty good chance that all of the fibres will be illuminated with the exception of the broken and disabled fibres. At the time of writing 18 fibres have been disabled from the large fibre bundle. Twelve of these are on either end of the chip, which means you shouldn't notice the fact that they are missing. There may be some apertures which the finding algorithm has missed. You should mark these with the cursor (using keystroke 'm'). If you don't like the scale of the plot, you can expand it by using the 'window' cursor function. This is done by typing the letter 'w' and then 'j', 'k', 't', 'b' or 'a'. 'j' and 'k' define the left and right of the window you want respectively while 't' and 'b' define the top and bottom of the window. 'a' requests that you go back to the autoscale mode which you started with. As with most cursor mode operations in IRAF you can get a full list of keystroke operations by typing '?' at the cursor prompt.

Getting the aperture number right is also rather important. This is because the amount of shift given to each spectrum for the sawtooth correction depends upon the aperture number. It's also important for making sure that the cross reference to the header information about that fibre is correct. The aperture location algorithm has information on the spacing between each spectrum, so it usually does a pretty good job of assigning aperture numbers. One thing you should be aware of is that the 'aperture number' **is not** the same as the spectrum number. For example, the first spectrum (namely the left most spectrum on the plot) will not be called aperture number 1 in general. At the time of writing the first nine program fibres in the large fibre bundle have been disabled. Thus the first fibre which can possibly be allocated is number 10. You shouldn't change this numbering system for the reasons mentioned above until after you have finished using **wyf_red**. You can then renumber the apertures using the routine **scopy** in the **onedspec** package. If you are in doubt about which fibres should be present in your data, you can always refer to the aperture identification table which **wyf_red** will have generated. This table will have a name which is the concatenation of the aperture reference file name, the field name (as specified in the header item 'af2fldnm') and the string 'apid'.

Once you have identified all of the apertures, then you type 'q' to get out. You'll then be asked if you want to fit the trace interactively. This shouldn't be necessary. As with most yes/no queries you can answer with either upper or lower case. An answer in lower case will mean that that answer will will only apply to the current spectrum and you will be asked again the next time around. Answering in upper case means you won't be prompted again.

4.3 Scattered light correction

If you are doing your reductions interactively, you will be asked if you want to fit the scattered light for each frame by hand. You should probably do so for the first frame at least in order to make sure you get it right. If you opt to do so, then you will first get a plot of a cut across the dispersion at the middle of the frame. The regions where the aperture definition says there should be a spectrum have been ignored so that which is left over should be representative of the scattered light on the image (this is why the cut won't look like what you see when you are doing the aperture definition). A fit will be done to the emission – the fitting function and order will be displayed at the top of the graph. If you wish to change the function or its order then you can do so with the commands `:func <functionname>` and `:order <order>` respectively. As usual the command '?' will give a listing of all the keystroke commands. Once you are happy with the fit, you can exit by typing 'q'. At this point you will be prompted with a question that asks whether you want to quit or whether you are interested in seeing another line. You might want to have a look at another line to make sure that the fitting parameters you decided upon for the first fit apply equally well to other lines. Once you are happy with the fits to the rows, there will be a brief pause and then you'll be presented with a plot of the modelled emission along the middle column. You can then define fitting parameters for the fits in the vertical direction in the same way as you did for fitting the rows.

Once you have defined the fitting parameters for the first file **apscatter** will go away and model the scattered light for the first file. You will then be prompted for each of the following files to see if you want to redefine the scattered light model. You may feel that this is not necessary, but you should be aware that the scattered light profile from an offset sky will most likely be very different from that of an object frame. For that reason **wyf_red** takes the precaution of asking.

It is possible to fit the scattered light to arc frames. In general this shouldn't be necessary. However in extreme cases the scattered light may affect the baseline profile of the arc spectra. The re-identification algorithm (which is a simple cross-correlation) that is used when **autoid** is 2 can then return spurious results.

4.4 Flat fielding and throughput correction

Although they are both multiplicative corrections and hence can be considered at the same time, flat fielding and throughput corrections address two very separate issues. Flat fielding will only be done if you specify a mean flat field with the parameter *mflat*. As was previously mentioned, this is generally done in order to remove the small pixel to pixel variations in sensitivity of the

CCD. You can, however, choose to remove the instrumental signature by using a flat field as well. If you plan ultimately to flux calibrate you spectra (this is **not** currently covered by **wyf_red**, by the way), then you probably don't want to remove the instrumental signature, since that will go away in the flux calibration. You are given the choice of fitting a function to the mean flat field and then dividing this function through the individual flatfield spectra (*fitflat* is 'yes'). If you choose to do this, then the instrumental signature will remain in your program objects, but the pixel to pixel variations will be removed. If *w_interact* is 'yes' then you can do this fit interactively. Otherwise the fit will be done automatically using a function as specified in the *wyfpars* pset (see Appendix B). The flat fields will be normalised so that the mean value of all the flat fields is unity. This ensures that in the mean the total number of counts on the frame remains the same after flat fielding as before.

As mentioned before, the throughput/vignetting correction is done using the mean counts in an offset sky exposure. It is quite important that an offset sky that was taken with the fibres in the same configuration as the program frame is used since the vignetting suffered by a given fibre will depend upon its position in the focal plane. A throughput/vignetting correction is computed for each fibre based on the mean value for the offset skies (again, normalised to unity). If a flat field is being used, these values are multiplied into the flat field frame before the program frames are flat fielded. If not, a throughput correction frame is created and this is divided into the program objects to correct for throughput/vignetting.

4.5 Extraction and cleaning

The object spectra will be extracted using a method commonly known as 'optimal extraction' or 'variance weighted extraction' (see reading list of references on optimal extraction). One of the advantages of optimal extraction is its ability to recognise possible cosmic ray events and delete them. This will be employed if the parameter *clean* has been set. Not all cosmic rays will be excluded in this way, but it does get rid of most of them and the extra expense in processing time is well worth it, especially for faint objects with long integration times.

4.6 Dispersion correction

If you want your spectra to be put on some sort of wavelength scale, then obviously you will need at least one arc reference image and you will need to set the parameter *dispcor* to 'yes'. You will also need a list of the available lines for the particular arc lamp that was used during the observations. There are a number of standard line lists in the directory **wyffos$linelists/**. If you don't find the one you need there, then you can create your own and give **wyf_red** its location and name using the parameter *coordlist*. You have to choose what level of automation you want for the arc line identification as well. If you have an arc with only a few lines in a whole spectrum, it is perhaps best to steer clear of the automatic line identification algorithm, since it works using a statistical method to decide on whether a solution is valid or not, and having only a few lines will not give a good statistical measure. You make the choice of algorithm by setting the parameter *autoid*. The values *autoid* can take are as follows:

1 Use the fully automatic line identification algorithm. This uses information in the header and a projection algorithm to define the dispersion and central wavelength. No interaction is required from you.

2 Use the semi-automatic line identification algorithm. For this you identify and fit an arc for one spectrum. You would be well advised to have a look at the help file for the task **identify** before attempting this. Then **wyf_red** goes away and finds all the lines in the rest of the spectra and taking the sawtooth correction into account does a fit for each.

3 Do each arc spectrum manually. This can be quite a bore, but may be necessary on some occasions. For example if you use an arc lamp with only 2 or 3 lines (which probably precludes your using the automatic algorithm) and if the number of fibres used is quite small, then the variation in the position of the arc lines caused by the optics of the system may be large enough such that they cannot be tracked with such a sparse sample of fibres, even when the sawtooth correction has already been applied. In this case you are required to identify and fit the lines in one spectrum as with the previous item. But then each of the following spectra will be presented to you on a rough wavelength scale based on the fit done for the first spectrum. It may then only be necessary to mark one arc line and allow **identify** to shift the spectrum (using the keystroke 'g'), so you should be able to go through the spectra reasonably quickly. Again you are advised to look at the documentation for **identify** before you attempt this.

Once the arc line identification is done, then **wyf_red** will attempt to wavelength calibrate the arc spectra. When it comes to dispersion correcting the object data, **wyf_red** makes a formal decision as to which arc frame should be used in reference to the object frame (see Appendix B for more details). All spectra in the object frame are then dispersion corrected.

4.7 Sky subtraction

If the objects you are observing are intrinsically faint, then getting the sky subtraction right is obviously a very high priority, even in dark time. For this reason **wyf_red** allows you a certain amount of latitude in this area. The "standard" way to do sky subtraction with multi-object spectroscopy is to combine all of the dedicated sky fibre spectra (the ones taken simultaneously with the object spectra) into one sky spectrum. The combination is done in such a way as to remove cosmetic defects such as cosmic rays. Then this mean sky spectrum is subtracted from each of the object spectra. There is a danger here that at least one of the sky fibres may have been accidently placed in the vicinity of a reasonably bright object. This is probably alright given that the algorithm that does the cleaning of the mean spectrum will probably reject most of the pixels from such a spectrum. If you really want to be sure, you can edit the sky spectra with the parameter *skyedit*. If this is set to 'yes' then before sky subtraction is done, the program will try and establish which spectra should be used to form the mean sky. It does this by combining all of the sky spectra into a mean and examining the results obtained by subtracting this mean from the individual sky spectra. If mean residuals are too high or two low, then the sky spectrum is rejected. If the mean residuals on the blue end are much different from those on the red end, the spectrum is rejected on the basis of curvature. All spectra with very low fibre responses are also rejected as using these will just add noise to the program spectra. If *w_interact* is 'yes' then you will get a list of all of the sky spectra and the reason that any of them might have been rejected. You will be asked to confirm the result. If you don't like the result, then you can always answer 'no' here and type in the aperture number of the spectra you want to reject.

Another way to do sky subtraction is to use the offset sky exposures. You can extract the offset skies as though they were objects by setting the parameter *skyobj* to 'yes'. You then need to scale the sky spectra by the ratio of the object and sky exposure times (these are in the header item 'exposed') using the task **sarith**. Finally use **sarith** again to subtract the scaled offset sky spectra from the object spectra. Note that when subtracting the mean sky spectrum **wyf_red** will put a flag into the header of the object frame which says that it has been sky subtracted. This is to protect you from accidently trying to sky subtract a frame twice. Thus the method just outlined whereby you subtract offset skies outside of **wyf_red** will *not* provide this sort of protection, so beware!

One last point is worth considering is whether the 'sky' you wish to subtract is not just 'sky' but also contains a contribution from something like the background light from an extended object. Such a case could arise if for example you were observing globular clusters in the halo of a nearby galaxy. In this case you will have to do your sky subtraction much more carefully and instead of using **wyf_red** you should consider much more basic tools like **scopy, sarith** and **splot** in order to make sure you get it right.

4.8 Tippexing

As mentioned before it is quite often desirable to remove remnants of sky emission lines after sky subtraction. This is done setting the *tippex* option to 'yes' and supplying the name of a 'tippex' file in the parameter *tfile*. A tippex file needs to have two items per line – the wavelength of the feature to be removed and the half width of the feature. If, for example, the file has the entry '6300.0 5.0' as one of its lines, then the region 6295 to 6305 will be tippexed. You can have as many entries in a tippex file as you want.

5 Other WYFFOS applications

The **wyffos** package contains several other applications apart from **wyf_red**. These are described here. All of these routines have help files which can be accessed from within IRAF.

5.1 wyfgeom

The orientation of the spectra on the output frame can vary depending upon the chip is read out at the end of an observation. Although it is highly unlikely that this will change during the course of a run, it may well change between runs. It is therefore vital that you know which way around the fibres are on the output frame. The more crucial piece of information is whether the fibres go from right to left or left to right on the output frame (in the numerical sense). For example, does the first fibre appear on the right or left hand side of the chip? This is critical since the numbering system for the fibres defined by **wyf_red** and the underlying IRAF routines will produce erroneous results if you get this backwards. That is it will assign object information to the wrong spectra and throw the bookkeeping system completely out the window. If you aren't sure which way around your spectra are, then contact your support astronomer and find out before you start working with **wyf_red** since it is a little piece of information which may save you a lot of time and heartache.

The other, less important question, is whether the spectra have a positive or negative dispersion. That is, if you plot one up, will its blue end be on the left (positive) or the right (negative)? This isn't so important unless you are using the automatic arcline identifier. You can restrict

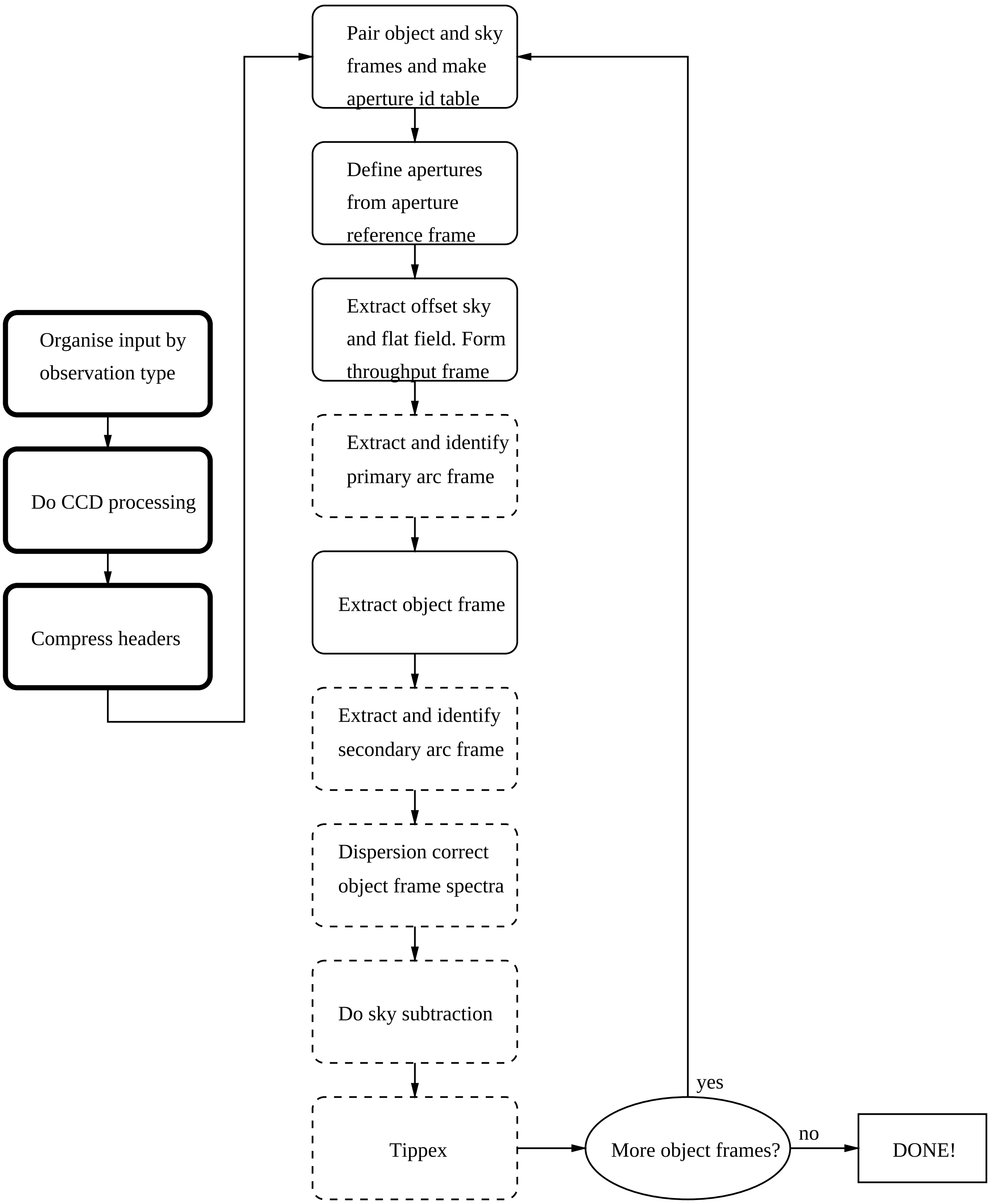

Figure 1: A rough flow diagram of **wyf_red**. The heavy boxes are tasks that are done on the input data as a whole. The dashed boxes are only done if the user wants to dispersion correct the data. The term 'extract' here includes correcting for scattered light if the user has opted to do so.

the solutions this algorithm considers to either positive or negative solutions only if you want. This will cut down the possibility of getting an erroneous solution. If you don't know 'which way is up' then you can say so and the algorithm will search both positive and negative solutions.

This routine sets the parameters *wyfpars.order* and *wyfpars.dflip*, so if you use this routine, you don't need to set these parameters yourself.

5.2 fibinfo

This is a routine which can help you to extract information from the header about an object which has been obesrved through a specific fibre. If you know which pivot number you want then this routine will unscramble the pivot and fibre numbering system for you. Depending upon whether you ask for verbose mode you can either just receive the bare essentials (pivot number, fibre number, object name, ra, dec) or all of the information which is available for that pivot.

5.3 fixhead

As mentioned before, one of the major pains of WYFFOS data is the enormous headers which are generated. Header compression does help, although the remaining header is still fairly big. Any file which is reduced through **wyf_red** will automatically have its header compressed. If you have a raw frame which you want to transport to another packge, then you can use this routine to cut down the header size and make it such that you don't have to have such an enormous value for *min_lenuserarea*.

5.4 transm

This routine allows you to calculate the relative transmissions of the fibres just from one flat field image. This can be particularly useful if the fibres are all placed in a circle for the exposure. This then takes out any relative throughput differences between the fibres which might arise as a result of vignetting in the field.

6 Recommended reading

6.1 IRAF documentation

The following files might be of interest to users. All of them are available by anonymous ftp from NOAO (140.252.1.1) in the directory /iraf/docs. There is a chance you'll find them on the Cambridge node in the directory /data/star/iraf/iraf/local/docs.

apex.ps.Z A description of the package APEXTRACT. This is the package which does the extraction and scattered light correction in **wyf_red**.

beguide.ps.Z A beginner's guide to IRAF. This is an excellent reference which rank beginners and experienced users alike will find invaluable. Very highly recommended!

ccduser2.ps.Z A users guide to CCD reductions. This might be useful for acquainting you with how the package CCDRED (which does the CCD processing for **wyf_red**) works.

dofibers.ps.Z A description of the task **dofibers**. The scripts which form **dofibers** were used as the basis for the scripts which form the bulk of **wyf_red**. (In a few cases they are stolen verbatim). Reading this might be useful especially if you want an alternative explanation for some of the things you might find confusing in this document.

The IRAF FAQ sheet is also a useful place to turn when you are confused. This is available on the WWW at http://iraf.noao.edu/faq/FAQ.html

Finally, the help files for the ONEDSPEC routines are quite useful. The most helpful of these are **sarith**, **scopy**, **specplot** and **splot**. Note that the ONEDSPEC package is loaded automatically by the WYFFOS package, so all of these applications should be available to you.

6.2 References to optimal extraction

Horne, K, 1986, *P.A.S.P*, **98**, 609.
Marsh, T.R., 1989, *P.A.S.P.*, **101**, 1032.
Mukai, K, 1990, *P.A.S.P.*, **102**, 183.

A Parameters for wyf_red

Figure 2 shows a listing of the parameters for **wyf_red** and their defaults. Here we also give a fuller description of each parameter.

input These are the names of all of the input files for **wyf_red**. As explained in the text the list must have at least one object frame. If no offset sky is given in this list for any one of the listed object frames, then the current working directory is searched for offset sky exposures. Similarly, if you are also processing bias and flat field frames and none are listed in the input list, then **wyf_red** will search the current working directory for such input files. The input list can also be used to *restrict* which files get used. For example if you have three arc frames and you want to make sure that a particular one is used to dispersion correct the object spectra in the list, then if that arc frame is specified in the input list, then it will be used and all others in the current working directory will be ignored.

apref This is the name of the aperture reference frame. An aperture reference frame can be any sort of frame, so long as it has reasonably high signal to noise. (You are, however, strongly discouraged from using arc frames for this purpose since their signal is so localised). This frame will serve to give **wyf_red** an idea of where the spectra are in the spatial direction and also of the curvature of each spectrum. You **must** specify an aperture reference!

batch If this is 'yes' then the whole of **wyf_red** will be run in the background. Note this means that none of the interactive tools in **wyf_red** can be used, so make sure that this is what you really want! If you want to run **wyf_red** in the background, then you should do it by setting this parameter and not simply by typing `wyf_red &` which will not set things up properly.

Figure 2: Parameters for **wyf_red** and their defaults

```
       input = ""              List of input program images
       apref = ""              Aperture reference image
       batch = no              Do processing in batch?
      (mbias = "")             Mean bias frame
      (mflat = "")             Mean flat frame
     (skyobj = no)             Reduce offset skys as objects as well?
   (override = no)             Override automatic aperture info?\n
  (scattered = yes)            Do scattered light correction?
    (arcscat = no)             Do scattered light correction for arcs?
    (fitflat = yes)            Fit and ratio flat field spectrum?
      (clean = yes)            Detect and replace bad pixels?
    (dispcor = yes)            Dispersion correct spectra?
(skysubtract = yes)            Subtract sky?
    (skyedit = yes)            Edit the sky spectra?
   (saveskys = yes)            Save sky spectra?
     (tippex = yes)            Tippex out sky line residuals
      (tfile = "")             Name of tippex file
      (splot = no)             Plot final spectra?
       (redo = no)             Redo operations if previously done?
     (update = yes)            update spectra if cal data changes?
   (listonly = no)             List steps but don't process?\n
 (w_interact = yes)            Do some of the procedures interactively?
     (autoid = 2)              What level of automation in arcline ids?
  (coordlist = "wyffos$linelists/argon.dat") Line list\n
    (wyfpars = "")             WYFFOS parameters\n
```

mbias This is the name of the mean bias frame. All input and aperture reference frames will be bias subtracted (unless this has already been done). If you want to subtract a mean bias frame rather than a constant then you inform **wyf_red** by specifying a mean bias frame. If this frame already exists it is used to do the bias subtraction. If it doesn't already exist, then the bias frames in the input list or in the current working directory (if there are none in the input list) are used first for form *mbias* and then to bias subtract the input list. If you wish to subtract a bias constant which is determined for each frame in the input list individually from the bias region, then leave *mbias* blank.

mflat This is the name of the mean flat field frame. If it exists, then it will be used for flat fielding the object data. If it doesn't exist, then any flat field frames in the input list or the current working directory will be combined first to form *mflat* and then to do the flat fielding of the object frames. If *mflat* is left blank, then no flat fielding will be done.

skyobj This switch allows you to include the offset sky exposures in the object frame list. That is, once they have been used to determine the throughput and vignetting correction, they will be extracted and calibrated as though they were object frames. This can be useful if you want to use these for sky subtraction or for diagnostics when things appear to have gone wrong.

override This switch allows you to override the automatic aperture identification algorithm. You are strongly advised against doing this, but in case you feel you need to, then the parameters you would then need to set are *wyfpars.nfibres, wyfpars.minsep, wyfpars.maxsep, wyfpars.width* and *wyfpars.apidtable.* See Appendix B for a fuller description of these parameters.

scattered If you want to do a scattered light correction then this should be set to 'yes'. If you are doing the reduction interactively (*w_interact* is 'yes') then you will be prompted for fitting the scattered light

arcscat If you want to do a scattered light correction on arc frames then this should be set to 'yes'. If you are doing the reduction interactively (*w_interact* is 'yes') then you will be prompted for fitting the scattered light. with a cursor. Otherwise the routine will use the parameters in the psets **apscat1** and **apscat2** to model the scattered light. In general removing scattered light from arc frames won't be necessary. But in extreme cases where scattered light affects the large scale baseline profile of the arc spectra the cross-correlation routine that identifies arc lines (**autoid**= 2) will return non-sense results.

fitflat If you want to fit a function to the mean flat field, then this should be set to 'yes'. If *w_interact* is 'yes' then you will be able to fit the function by hand with a cursor. If not, then a function described by the parameters *wyfpars.f_function* and *wyfpars.f_order* will be used. See Appendix B for more information on these parameters.

clean This switch allows the extraction algorithm to 'clean' the spectra of cosmetic defects, such as cosmic rays. This costs a little in processing time, but is well worth the price especially for faint objects.

dispcor If you wish to dispersion correct your data, then this should be set to 'yes'. Note that in order to dispersion correct you must have at least one arc frame available. If **wyf_red** can't find any arc frames, then it will automatically switch *dispcor* to 'no'.

skysubtract If you want to do sky subtraction by forming a mean sky from the available dedicated sky fibres, then this should be set to 'yes'. Note that the unskysubtracted data are **not** saved. So, if you are not sure about the sky subtraction, you should perhaps run the data through **wyf_red** once with no sky subtraction. Then copy the extracted spectrum frame to another file as a backup. Then rerun **wyf_red** with the skys subtraction on. **wyf_red** will know to go straight to the sky subtraction since the flat-fielding, dispersion correction, *etc.* will all be flagged in the header has having been done.

skyedit This causes the program automatically to select which sky apertures should be used in the final mean sky. If you don't agree with its recommendations, then you can change them so long as *w_interact* is 'yes'.

splot This allows you to review the extracted spectra at the end of the reduction process using the application **splot**. Again, this is a task with a large suite of keystroke commands available and the user is advised to familiarise himself with these (with the appropriate help file).

redo This switch is used to guard against redoing operations on files which have already been done. If, for example, the input list contains a file which has already been dispersion corrected and one which hasn't, then with *redo* set to 'no' **wyf_red** won't bother to redo the dispersion correction on the former. If this is set to 'yes' then everything will be redone on every file no matter how much or how little.

update This allows for redoing parts of the reduction, but only if the aperture, flat-field or dispersion information changes.

listonly If this is set to 'yes' then you will get the output you would expect to see on your screen from a proper run of **wyf_red**, but none of the reduction will actually be done. This allows for a 'dry run' in case you're not sure you've set things up correctly.

w_interact If you have set this to 'yes', then some of the setting up of the calibration information can be done interactively. Note that any of the parameters in **wyf_red** which imply interactive input cannot be done interactively unless *w_interact* is switched on!

autoid This is an integer parameter which specifies the level of interaction in the arc line identification procedure. (1) is fully automatic, (2) is semi-automatic and (3) is fully manual arc line identification. See Section 2.6 for a fuller description.

coordlist This is name of the line list for the arc lines. This must specify the full path to the file as well has the file name. There are several of the most popular available in **wyffos$linelists/** and the user is encouraged to look there first before creating a line list.

wyfpars This is the 'wyfpars' pset. See Appendix B for more details.

B Parameters in wyfpars

Figure 3 shows a listing of the pset *wyfpars*. The parameters in *wyfpars* are accessible through the usual methods for a pset, that is either scrolling down to the *wyfpars* parameter in the **wyf_red** parameter list and typing **:e** or through the command `epar wyfpars`. Below is a description of each of the parameters.

Figure 3: Parameters in the pset **wyfpars** and their defaults

```
        (line = INDEF)          Default dispersion line
        (nsum = -20)            Number of dispersion lines to sum or median
     (nfibres = 126)            Number of spectra to find
       (width = 6.)             Width of profiles
      (minsep = 8.)             Minimum separation between spectra
      (maxsep = 10.)            Maximum separation between spectra
    (apidtable = "")            Aperture ID table
       (dosaw = yes)            Do sawtooth correction?
       (sfile = "")             Name of file with sawtooth ranges and shifts
       (fends = no)             Do you want to locate spectral ends?
       (order = "decreasing")   Order of apertures
      (extras = no)             Extract sky, sigma, etc.?\n\n-- DEFAULT APERTUR
       (lower = -3.)            Lower aperture limit relative to center
       (upper = 3.)             Upper aperture limit relative to center\n\n-- A
      (ylevel = 0.05)           Fraction of peak or intensity for resizing
        (peak = yes)            Is ylevel a fraction of the peak?
         (bkg = yes)            Subtract background for resizing?
    (avglimits = no)            Average limits over all apertures?\n\n-- TRACE
      (t_step = 10)             Tracing step
  (t_function = "spline3")      Trace fitting function
     (t_order = 2)              Trace fitting function order
  (t_niterate = 2)              Trace rejection iterations
       (t_low = 3.)             Trace lower rejection sigma
      (t_high = 3.)             Trace upper rejection sigma\n\n-- SCATTERED LIG
      (buffer = 1.)             Buffer distance from apertures
     (apscat1 = "")             Fitting parameters across the dispersion
     (apscat2 = "")             Fitting parameters along the dispersion\n\n-- A
     (weights = "variance")     Extraction weights (none|variance)
        (pfit = "fit1d")        Profile fitting algorithm (fit1d|fit2d)
    (readnoise = "READNOIS")    Read out noise sigma (photons)
        (gain = "GAIN")         Photon gain (photons/data number)
      (lsigma = 3.)             Lower rejection threshold
      (usigma = 3.)             Upper rejection threshold
     (nsubaps = 1)              Number of subapertures\n\n-- FLAT FIELD FUNCTIO
(f_interactiv = no)             Fit flat field interactively?
  (f_function = "spline3")      Fitting function
     (f_order = 10)             Fitting function order\n\n-- ARC DISPERSION FUN
      (arclog = "autoarc.log")  Log file for automatic line id algorithm
       (dflip = -1)             Consider +ve, -ve or both dispersions in autoar
       (match = 2.)             Line list matching limit in Angstroms
         (sep = 2)              Minimum pixel separation
    (contrast = 0.1)            Maximum contrast between strongest and weakest
        (edge = 0)              Minimum distance from edge
  (maxfeatures = 50)            Maximum number of features for automatic identi
    (pkthresh = 10.)            Minimum threshold level for peaks
      (fwidth = 4.)             Arc line widths in pixels
```

Figure 3: *cont.*

```
     (cradius = 10.)            Centering radius in pixels
     (cthresh = 10.)            Feature threshold for centring
  (i_function = "chebyshev")    Coordinate function
     (i_order = 4)              Order of dispersion function
  (i_niterate = 3)              Rejection iterations
       (i_low = 3.)             Lower rejection sigma
      (i_high = 3.)             Upper rejection sigma
       (refit = yes)            Refit coordinate function when reidentifying?
 (addfeatures = no)             Add features when reidentifying?\n\n-- AUTOMATI
      (select = "interp")       Selection method for reference spectra
        (sort = "jd")           Sort key
       (group = "ljd")          Group key
        (time = no)             Is sort key a time?
    (timewrap = 17.)            Time wrap point for time sorting\n\n-- DISPERSI
   (linearize = yes)            Linearize (interpolate) spectra?
         (log = no)             Logarithmic wavelength scale?
        (flux = yes)            Conserve flux?\n\n-- SKY SUBTRACTION PARAMETERS
     (combine = "average")      Type of combine operation
      (reject = "avsigclip")    Sky rejection option
       (scale = "none")         Sky scaling option
```

line The aperture location algorithm sums up a number of rows across the dispersion in order to define the location of each spectrum. This is the central row of the summed block of rows.

nsum The aperture location algorithm sums up a number of rows across the dispersion in order to define the location of each spectrum. This is the total number of rows to be summed. If the number is negative, then the profiles will be medianed rather than summed.

nfibres This is the number of spectra which can be found on a given frame. This is usually set up automatically by **wyf_red** and anything which you might put here will be ignored unless the **wyf_red** parameter *override* has been set. This is not highly recommended.

width This is the width in pixels of each spectral profile. This is usually set up automatically by **wyf_red** and anything which you might put here will be ignored unless the **wyf_red** parameter *override* has been set. This is not highly recommended.

minsep Fibre spectra are reasonably regularly spaced and this is the minimum spacing between two fibres. This is used to determine whether two adjacent peaks can be both spectra, or whether perhaps one of them is spurious. This is usually set up automatically by **wyf_red** and anything which you might put here will be ignored unless the **wyf_red** parameter *override* has been set. This is not highly recommended.

maxsep This is the maximum spacing between two fibres and is used to determine whether a large blank space between two spectra can be as a result of one or more missing fibres. This is usually set up automatically by **wyf_red** and anything which you might put here will be ignored unless the **wyf_red** parameter *override* has been set. This is not highly recommended.

apidtable This is the aperture identification table. See the help file for **apfind** for a full description of aperture identification tables. This is usually set up automatically by **wyf_red** from information contained in the header and anything which you might put here will be ignored unless the **wyf_red** parameter *override* has been set. This is not highly recommended.

dosaw If this is set, then the sawtooth correction will be performed automatically. Normally this should always be set to 'yes' as if the sawtooth correction is not done, the most of the rest of the calibrations will fail as well.

sfile This is the name of the file with the sawtooth correction factors. This is automatically defined by **wyf_red**, so you needn't worry about this.

fends If **wyf_red** is supposed to locate the ends of the spectra then this parameter should be set. This may be important with low resolution observations where the spectra are smaller than the detector. See Section 1 under the heading 'End finding' for more information.

order This defines whether the fibre numbers increase or decrease from left to right on an output image. The current value is "decreasing" which means that the lowest numbered fibres will appear on the right hand side of the output image. You can set this parameter by using the application **wyfgeom**.

extras If this is set, then you will get "extra" information in the spectrum output file. Normally the output spectral format is a 2D file with each spectrum occupying a single row. If *extras* is 'yes' then the output files are 3D with the optimally extracted specta in the first plane, a simple sum spectrum in the second and the estimated sigma spectrum in the third.

lower, upper These are the upper and lower limits of the spectral aperture relative to the centre of the aperture profile. These are normally set by **wyf_red** automatically, so unless you have set the **wyf_red** parameter *override* then anything you set here will be ignored.

ylevel, peak, bkg, avglimits These are all parameters which are essentially ignored by **wyf_red**, but were included in case it is ever necessary to allow for the apertures to be resized (see the help for **apfind** to find out what I'm on about here).

t_step This is a step in the dispersion direction between determinations of the aperture position and is used to determine the aperture trace (*i.e.* its distortion in the spatial direction). Note, the parameter above *nsum* also gives the number of lines to sum for each of these position determinations.

t_function This is the function which is used to fit the spectral trace.

t_order This is the 'order' of the function used to fit the trace. As with the rest of IRAF, when a polynomial 'order' is requested, then what is meant is the true order plus one, *e.g.* fitting a constant means a value of *order* of one. If *t_function* is a spline, *order* refers to the number of spline pieces, e.g *t_function* is 'spline3' and *order* is 2, then you'll be fitting two cubic splines.

t_niterate, t_low, t_high These all refer to the fitting iterations done when tracing the spectra and are the number of iterations, the lower rejection threshold and the upper rejection threshold, respectively. The threshold parameters are in units of the fit sigma.

buffer When fitting the scattered light **apscatter** avoids the pixels which it believes to be part of a spectrum. In order that the scattered light estimate shouldn't be biassed by data from the fibres, the parameter *buffer* is used to define an extra space between the fibre pixels and pixels which can be used for estimating the scattered light. Note that the spectra are pretty closely packed and you run the risk of not having any pixels left over for scattered light estimation if you make this parameter too big.

apscat1, apscat2 These are the psets which determine the functions and parameters used to fit the observed scattered light. *apscat1* refers to the fits done across the dispersion and *apscat2* refers to the fits done along the dispersion. See **apscatter** for more details.

weights The spectra can be extracted in either a simple unweighted fashion (just a straight sum) or by variance weighting (optimal extraction).

pfit This is the algorithm used for the variance weighting. 'fit1d' refers to the method of Horne (1986) and 'fit2d' refers to the method of Marsh (1989).

readnoise, gain These are the noise properties of the CCD used. They can either be numbers of the names of header items. Only change these if you know what you are doing!

lsigma, usigma These are rejection thresholds used for cleaning cosmic rays out of spectra. They will only be employed if the **wyf_red** parameter *clean* has been set to 'yes'. The units are in sigma.

nsubaps **wyf_red** like **dofibers** has the ability to divide each fibre spectrum into a number of sub-apertures. This is done in order to correct for wavelength shifts across a single fibre. It should normally not be necessary for this parameter to be changed from 1.

f_interactiv If this is 'yes' and you are doing flat fielding, then this allows you to fit the flat field interactively.

f_function, f_order If flat fielding is done and the **wyf_red** parameter *fitflat* is 'yes' then these are the function to be fit and its order. Again 'order' refers either to the order of a polynomial plus one or the number of spline pieces.

arclog If you are using the automatic arcline identification algorithm (**wyf_red** parameter *autoid* is 1) then this is the name of a log file for the routine to use.

dflip If you are using the automatic arcline identification algorithm, you can restrict the solution space it searches by specifying whether the dispersion is positive (red end to the right on a plot) or negative (red end on the left). *dflip* takes the value of 1 for a positive dispersion, and a value of -1 for a negative dispersion. If you are not sure, then you can set the value to 0 and both negative and positive dispersions will be examined. If you use the routine **wyfgeom** then this parameter will be set for you.

match This is the maximum number of Angstroms which a wavelength in a line list is allowed differ from the value predicted by a dispersion function. If this value is not exceeded then the feature at the predicted position is said to match the wavelength in the line list.

sep This is the minimum separation in pixels allowed between two spectral features.

contrast This is a maximum 'contrast' between the strongest and weakest features in the arc spectrum. Having a contrast which is zero will mean that large numbers of possibly spurious local maxima will be identified as arc lines.

edge This is the minimum buffer distance between any arc line and the edge of the spectrum.

maxfeatures This is the maximum number of spectral features allowed in an arc spectrum.

pkthresh This is a minimum number of counts for a peak to be considered an arc line.

fwidth This is an estimate of the arc line width and is used in the centring algorithm.

cradius, cthresh These parameters are currently unused.

i_function, i_order These are the fitting function and its order used to fit the dispersion solution. As usual the order refers to the true order of the polynomial or the number of spline pieces.

i_niterate, i_low, i_high These are the number of fitting iterations and the clipping thresholds (in sigma) for the dispersion solution fits.

refit, addfeatures These parameters are not currently in use.

select This is the selection method used for defining which arc spectrum will refer to which object spectrum. See the help file for **refspectra** for more information.

sort, group, timewrap, time The first two are image header keywords used for sorting and grouping the arc frames. The second two define whether the sorting is done according to time or some other way. Again see the help file for **refspectra** for a complete description here.

linearize If this is 'yes' then the spectra will be resampled (interpolated) onto either a linear or a log-linear system. If not, then the wcs of the image will be assigned a non-linear dispersion function.

log If this is 'yes' then the output spectra will be on a log wavelength scale.

flux If this is 'yes' then the output pixels are integrated over the input pixel extent, thus conserving the flux. Otherwise the output spectrum is a straight interpolation of the input.

combine, reject These are the method used to combine the dedicated sky fibres into one master sky and the rejection algorithm used to remove cosmetic defects from the input spectra.

scale This is a type of scaling which can be applied to each of the spectra. This should not be necessary to use if the flat-fielding and throughput correction have been applied.

Figure 4: Parameters for the WYFFOS package

```
 (observatory = "lapalma")      Observatory of data
      (interp = "spline3")      Interpolation type
     (cenwave = "cenwave")      Initial estimate of central wavelength
        (disp = "disp_app")     Initial estimate of the dispersion in Angtroms/pix
      (backup = "backup/")      Backup directory or prefix
    (database = "database")     Database
    (dispaxis = 2)              Dispersion axis
     (verbose = no)             Verbose output?
     (logfile = "logfile")      Log file
    (plotfile = "")             Plot file\n
     (records = "")             Record number extensions
     (version = "WYFFOS V1.0: Oct 1996")
```

C Parameters for the wyffos package

Figure 4 gives a list of parameters for the **wyffos** package. These are parameters which in general needn't ever be changed by the user with a few exceptions. The **wyffos** parameters are as follows:

observatory This is the observatory at which the spectra were taken if that information is not in the header keyword OBSERVAT.

interp This is the interpolant used when spectra are being resampled.

cenwave, disp These are the header parameters where the rough central wavelength and the dispersion (in A/pix) are stored.

backup When the CCD data reduction is done (bias subtraction, *etc.*) the original data will be copied to the directory specified by this parameter. This is a fail-safe in case something goes wrong.

database This is the name of the directory where aperture and arcline identification information will be stored.

dispaxis This is the axis which is parallel to the dispersion of the spectra. *e.g.* if *dispaxis* is 2, then the spectra are vertical.

verbose If this is set to 'yes' then you will get a bit of extra info as the reduction progresses.

logfile This is the name of the log file to which **wyf_red** will all its logging information.

plotfile This is the name of a file to which to redirect plot. This is for the most part unused.

records Dummy parameter not to be used.

version This is the version of the package you have.

D Header items used by wyf_red

Below is a list of the header items which should be available in a standard WYFFOS data file and which **wyf_red** makes use of. The information for the individual AF2 pivots is also presented in Appendix E.

wyffeed This is the name of the fibre feed used in the observations. At the time of writing, only one feed is available (called LARGE).

ccdypixe This is the physical size of the CCD pixel in microns in the Y (dispersion) direction and is used to convert the dispersion in the header from Angstroms/mm to Angstroms/pixel. It is also used to convert standard feed information such as the spacing between spectra into pixel units.

cenwave This is a rough estimate of the central wavelength of the observation. Sometimes this is not terribly accurate, but it should at least be within a few hundred Angstroms.

dispersi This is an estimate of the number of Angstroms/mm for the spectral dispersion and is usually pretty accurate (within about 5-10%).

af2fldnm AF2 has a name for each configured field which is written to this parameter. This is used as a unique identifier for a field so that object frames and offset sky frames can be paired off. This is important as the offset sky frame must be taken with the fibres in the same configuration as the object frame in order to get the vignetting correction right.

obstype This is the 'type' of file which is defined at the telescope. The ones currently in use for **wyf_red** are: BIAS (bias frames), FLAT (dome flats), SKY (offset sky exposures), TARGET (object frames) and ARC (arc frames). Occasionally because of mistakes made at the observing console, the wrong value of OBSTYPE is written to the header. This can be corrected by using the task **hedit** to insert the correct value. The original values of OBSTYPE are placed in the headers by the telescope control software. If the arclamp is switched on and the observing command is 'arc' then the value ARC will be placed in the OBSTYPE header item. If, instead the observing command 'run' is used, then OBSTYPE will take the value TARGET and **wyf_red** will then assume that this is a program frame. It is therefore important to make sure that the correct commands are used at the observing console and if they haven't then to correct the values of OBSTYPE *before* attempting to use **wyf_red**.

date-obs This is the civil date of the observation.

utobs This is the UT of the observation

exposed This is the exposure time of the observation in seconds

ra, dec These are the RA and DEC of the telescope for a particular exposure.

epoch This is the epoch of the coordinates.

st This the local sidereal time of the observation

ccdchip This is the name of the chip used in the observations and is used to define the data region and the bias strip.

If the headers of your files have not been compressed, then you will find the following items of data in the headers for each AF2 pivot. Note, this is not a list of *all* the header information for each pivot, just the bits which **wyf_red** eventually uses (albeit that **wyf_red** only uses them once they've been compressed). The format for the header items is "F" followed by a three digit number (with leading zeros) which denotes the pivot number, followed by the header item name (which is what will appear in the list below). For example the fibre type for pivot 30 will be stored in item "F030TYPE".

type This is the fibre type. There are four possible values: PR – program object, SK – pivot has a dedicated sky fibre, FI – pivot has a fiducial fibre, UN – pivot has a fibre which for one reason or another is not currently being used.

name This is the name given to the object by the user in the configuration file.

ra This is the RA of the fibre.

dec This is the DEC of the fibre.

mag This is the object magnitude

cmnt This is a comment string which may have been added at the configuration stage.

E Compressed header format

In order to cut down the amount of space taken by WYFFOS headers we have introduced a compressed header format. Raw WYFFOS files have 16 header cards per pivot. The compressed format puts this data on at most three cards (usually on two). In a compressed header the header items are called "F" followed by a three digit number (with leading zeros) denoting the pivot number, followed by "DAT" and then followed by 1, 2 or 3. In table E we give a listing of the header objects and where they can be found in the compressed format. For example, the fibre type for pivot 30 which is stored in "F030TYPE" in the raw WYFFOS format is stored in columns 44:45 in "F030DAT1" in compressed format. Note that if the comment header item is blank in the raw WYFFOS format, then the third compressed header card is not created.

Table 1: Location of WYFFOS header items in compressed format

Old Item	Card no.	Columns	Remarks
NAME	1	1:18	Object name
RA	1	20:30	Object RA
DEC	1	32:42	Object DEC
TYPE	1	44:45	Fibre "type" (PR, SK, FI, UN)
MAG	1	47:55	Object magnitude
ID	1	57:62	Object identification code
FIBX	2	1:7	Fibre X position
PEX	2	9:14	Fibre X position error
FIBY	2	16:22	Fibre Y position
PEY	2	24:29	Fibre Y position error
FIBT	2	31:35	Fibre position angle
FPVX	2	37:43	Pivot X position
FPVY	2	45:51	Pivot Y position
STAT	2	53:53	Pivot status code
WT	2	55:60	Pivot weight
CMNT	3	1:63	Any comments in configuration file