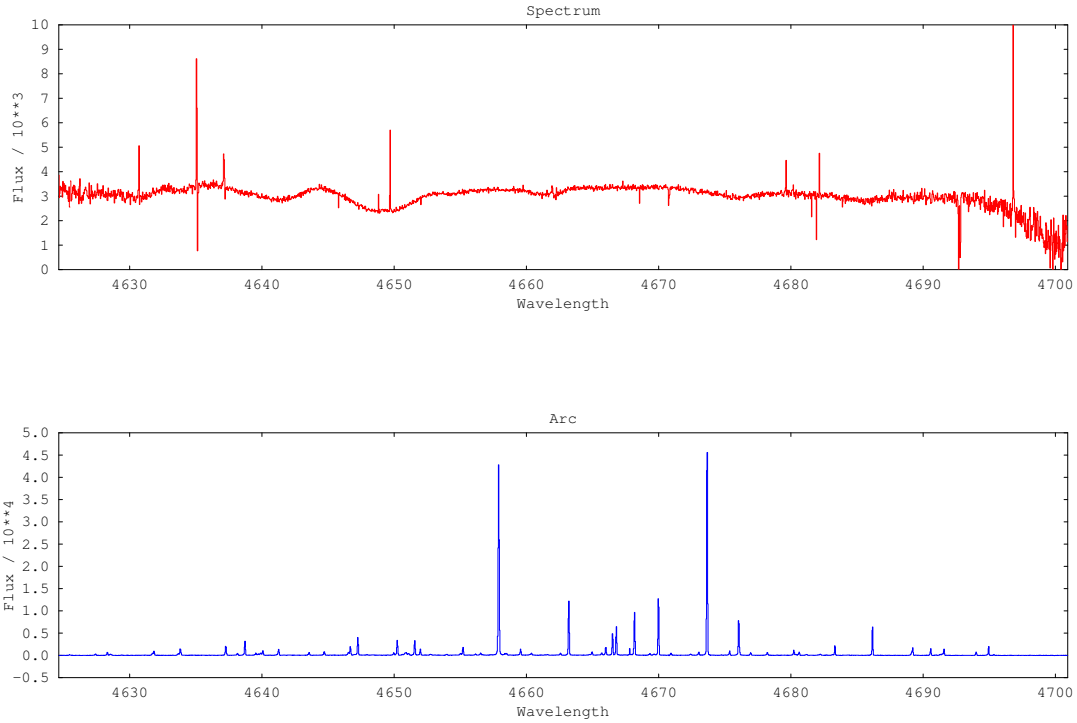
Starlink Project
Starlink Cookbook 7.2

Martin Clayton and Anthony Holloway

15 June 1998

Simple Spectroscopy Reductions



An extracted spectrum and its corresponding arc.
Graphic created using DIPSO[11].

Abstract

This Cookbook describes the basic concepts and methods used in optical astronomical spectroscopy; it is aimed at those new to the field. Complete worked example reductions for both one- and two-dimensional longslit spectra, using real datasets, are described. Common problems and their solutions are discussed. A section on related resources is included, as is a glossary of commonly used terms.

Contents

List of Figures

1 Introduction

This Cookbook describes the basic concepts and methods used in optical astronomical spectroscopy; it is aimed at those new to the field. The term 'spectroscopy' covers quite a few areas, each within itself worthy of a Cookbook. For the sake of brevity, the scope of this document is limited to one-dimensional (1-D) and longslit (2-D) spectra, in the optical waveband, recorded with CCD detectors.

The reductions described in later sections of this document are the most-common, simple, and straightforward that you might encounter. Problems arising when real-world data deviate from this ideal case, and how to overcome them, are described.

If you are starting your work with 2-D longslit spectra it is still a good idea to read the sections of this guide dealing with 1-D spectra as many of the ideas and techniques are common to both and easier to understand in the first instance for 1-D data.

This Cookbook does not describe the preparation of CCD data in detail; an outline of what's needed is given. Requirements for getting good extractions, in terms of calibration data, are also described.

Non-trivial spectroscopy—cross-dispersed, infrared *etc.*—are *not* covered by this Cookbook. See §1.2 (below) for some resources to help you in these areas.

1.1 How to Use this Cookbook

The most obvious thing to do is *read it.* But no one ever does that—so which bits do you *really* need to read?

§3 is the starting point if you're about to go observing for the first time.

The concepts used in spectrum extraction, problems and how to get around them, are covered in §4 for 1-D data and in §6 for longslit data.

Once you have an idea of what you're doing, you'll want to know how to do it; this is covered in §5 for 1-D data and in §7. These are worked examples of a complete reduction illustrating techniques described in §4 and §6

§8 gives some specific examples of recipes, and offers some problem-solving advice.

There is a glossary in this document. If you cannot find the term you want there, you might try the *NASA Thesaurus:*

- `http://data.nasa.gov/nasa-thesaurus/`

1.2 Other Sources of Information

A significant part of the process of spectrum extraction is the preparation of the CCD data frames. This subject is covered in the Starlink Cookbook, *The 2-D CCD Reduction Cookbook* (SC/5). Those planning to use IRAF[1] should consult *A User's Guide to CCD Reductions with*

[1]IRAF documents can be found in your IRAF installation; you do not need to get them from Tucson or a mirror. Check with your system manager for details.

IRAF (UGCRI) by Philip Massey. UGCRI is useful to *any* user of CCD data, even those planning to use CCDPACK[9] or FIGARO[18], to do the preparation.

IRAF users should read the documents for spectral data reduction within IRAF: *A User's Guide to Reducing Slit Spectra With IRAF* and *Guide to the Slit Spectra Reduction Task DOSLIT*. These give a comprehensive description of the IRAF approach to spectral data reductions. There is also a hypertext tutorial for DOSLIT at

- `http://www.starlink.ac.uk/iraf/web/tutorials/doslit/doslit.html`

You may be able to access a local copy of this tutorial, consult your system manager.

This Cookbook is the most basic introduction to spectroscopy, some guides for more specialised types of reductions are available:

- Starlink *Introduction to Echelle Spectroscopy* (SG/9) is a good first read for those new to échelle work.
- Starlink *Echelle Data Reduction Cookbook* (SC/3) which contains examples of data-reduction scripts, including templates for automated data reductions with ECHOMOP[16].

2 Simple 1-D and 2-D Spectra

If we divide up the light collected from an astronomical object by wavelength the result is a spectrum. What use is this? We can derive a great deal of information about the object we are looking at: chemical composition, velocities, temperatures, and densities.

Figure 1 shows some of the features of a typical spectrum. These data were recorded in the manner most commonly used: with a CCD. You can see some cosmic-ray features (bright spots caused by cosmic rays hitting the detector), as indicated by the label 'CR', and there are some night-sky lines (bright features in the spectrum of the light coming from the sky) running vertically across the spectrum.

Longslit spectroscopy produces CCD image frames which have essentially the same features as are present in 1-D spectra (see Figure 2. The difference arises when the slit is placed over an extended object and then an individual spectrum for each pixel row along the slit length is obtained.

For consistency, all spectra discussed in this document are arranged as in the above figures; the dispersion direction is oriented roughly parallel to the X-axis, the spatial direction parallel to the Y-axis.

3 Preparing for Observation

In this section some notes on what to be aware of prior to an observing run are given. In outline: to successfully extract and calibrate spectra a complete set of reference frames should

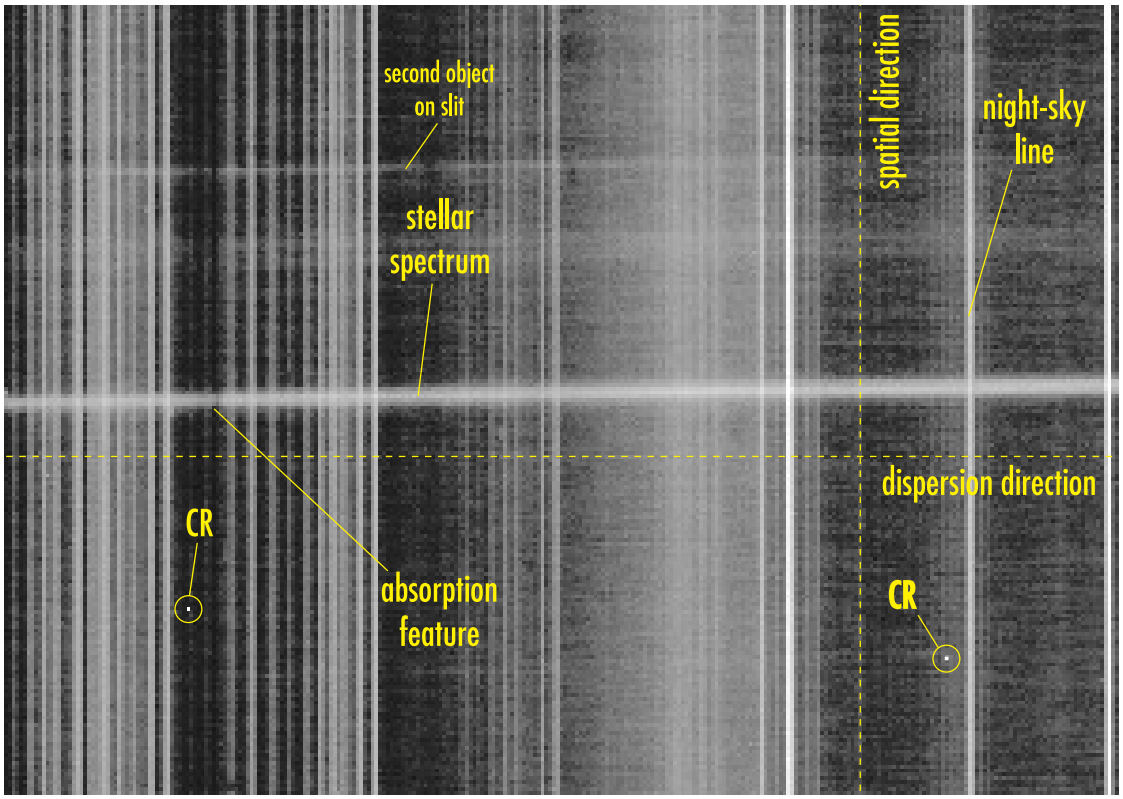


Figure 1: Basic characteristics of a simple 1-D spectrum.

be obtained at the telescope; it's always easier to take more calibrations at the telescope than to cope with poor data at the terminal.

You might want to refer to the more extensive discussion of CCD data calibration in UGCRI[14], the section entitled, 'How Many and What Calibration Frames Do You Need?'

3.1 What CCD Data Do You Need?

This is what you need to attempt 'textbook' data preparation:

Bias frames Zero-second exposures taken with no signal light entering the instrument, but with any pre-flash used for the object exposures.

Dark frames Long exposures taken with the shutter closed. Typically, the exposure time used is similar to that selected for the object frames.

Flat-field frames Exposures taken with a suitable continuum lamp (usually Tungsten) as light source.

Arc frames Exposures taken with an arc lamp (usually Thorium-Argon) as light source, to be used for wavelength reference.

Object frames Exposures taken with a target object or reference object as the 'light source'.

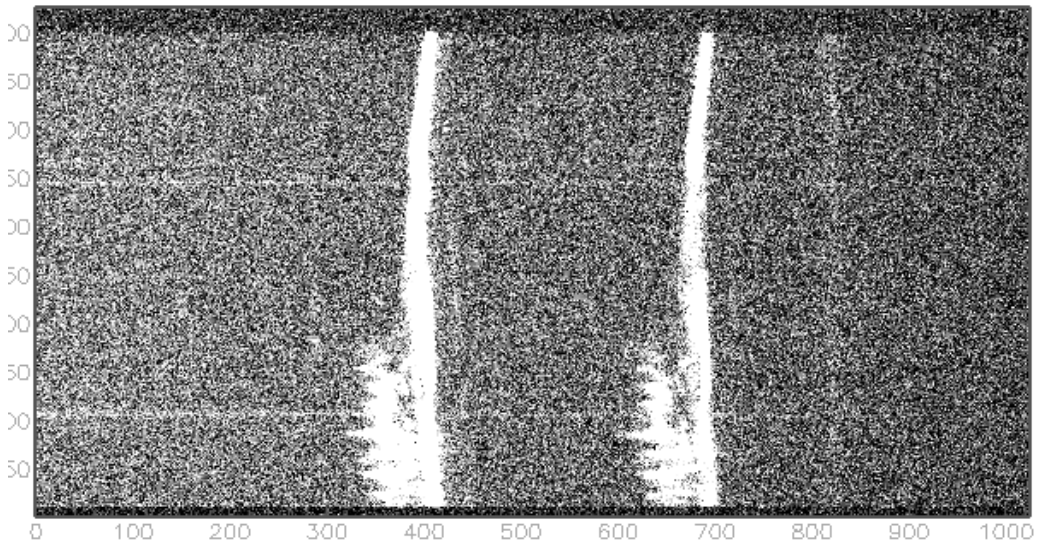


Figure 2: A longslit 2-D spectrum produces a number of spectra for positions along the slit length.

The arc and object frames will be processed by the data reduction software. The bias, dark, and flat-field frames are used in the preparation of the CCD data.

Whether you actually need all these data frames, and how many of each you need is to a large extent determined by what you hope to achieve. In general, you will need a set of calibration frames for *each night* of observing.

3.2 Detector Pre-processing

In order to remove detector-related effects a complete set of bias, dark, and flat-field frames should be obtained. It is important to *bracket* the science data exposures with sets of CCD reference frames. A post-observation review of these will reveal any image shifts. Having bracket frames allows the data to be processed even in the event that some time-dependent variation is found (as long as it is a small, slowly varying effect).

See §4.1 for outline details of CCD data preparation.

In some cases it may be possible to not use any bias frames. Instead, a median value for the bias level is obtained by inspecting the overscan region in some or all of the object/arc frames. You can use fewer bias frames when you have a high signal-to-detector-noise ratio.

For exposure times limited by cosmic-ray event counts, the dark current in most CCD cameras is not a significant factor. The simplest way to decide whether to take dark frames is to take one of exposure time similar to that you are using for object frames, and check the signal level.

3.3 Flat Fields

You may not need to flat field; flats taken with modern CCDs can be uniform in response to only a few percent. Flat fields are only needed if the signal-to-noise ratio you require is high. The precise figures will depend upon several factors, and should be estimated for each observation.

You should try to match the configuration of telescope and instrument in the flat-field and object exposures as closely as possible. If there might be an image shift between flat fields taken before observing and the science data, the best procedure is to take flats bracketing each science exposure in the same way as you would take arcs. Consult the instrument manual for your spectrograph to decide which strategy to use.

See §4.5 for details of the purpose and use of flat-field frames.

3.4 1-D Spectrum Tracing or 2-D Frame De-rotation

If the CCD is not perfectly aligned with the dispersion axis of the spectrometer then the data will be recorded at an angle to the grid of dectector pixels. This can be corrected by tracing the spectrum (for 1-D data) or de-rotating the frame (for 2-D data).

It may be worth having *several* possible frames for 1-D spectrum tracing—in case some of them are badly contaminated by cosmic rays and so difficult to use.

See §4.4 for more details on 1-D spectrum tracing.

3.5 Wavelength Calibration

As with the CCD characterisation frames, two wavelength-scale reference (arc) frames should be taken bracketing the science exposures *if* wavelength scales are required (*e.g.* for radial-velocity measurements). Normally you should find no significant difference between the two extracted arc spectra; however, if you only take one arc exposure and some shift does occur you won't be able to correct for it. Using both arc spectra, a time-weighted mean wavelength scale can be produced and applied to the science data.

See §4.7 for more details.

3.6 Flux Calibration

To get the best results when flux-calibrated spectra are wanted, select reference stars as close as possible on the sky to your target objects. You want the conditions of the target and reference exposures—both instrument configuration and air mass through which the observations are made—to be as similar as possible.

Refer to §4.8 for more information on flux calibration.

4 Basic Steps in 1-D Data Reduction

In this section each of the basic steps in the spectral data reduction procedure are described. Examples of practical techniques using these ideas are to be found in §5.

4.1 Image Preparation

Most spectral data will be obtained using a CCD. Careful preparation of CCD data prior to attempting the extraction of science data is essential. In this Cookbook the procedure is outlined (very!) briefly. Those points important to spectral data reduction are included.

The basic steps are:

Generate master bias frame Typically done by finding the median of several bias frames.

Subtract master bias frame The median bias frame is subtracted from each arc frame, object frame, and flat-field frame.

Generate master flat-field frame(s) Again, done by finding the median of several (flat-field) frames.

Crop images Regions—such as the overscan—are not used in the extraction process and should be removed (the same as cropping a photograph) as they may confuse the algorithms in the data reduction software. Often CCD images will have 'rough' edges, *i.e.* non-useable data, which should also be cropped.

Depending on your source data and choice of reduction software you may need to:

Rotate and orient the image All the major packages constrain the orientation of the spectra in some way. The images may have to be adjusted to meet these constraints. In general, if the spectra in your data run roughly horizontally (parallel to the X-axis) and the wavelength increases from left to right you will be fine. In other situations, you may need to use some utilities to re-orient the data. This may involve rotating and perhaps reflecting the images.

You may have data which contain dead columns or few-pixel hot-spots. Handling of these is discussed in the documentation for the CCD data preparation packages.

Once a set of arc and object images have been prepared, the data reduction process can begin.

4.1.1 Software for CCD Data Preparation

There are several packages for preparing CCD data. All of them offer similar functionality. You'll find that it's easiest to use the package which complements the spectral reduction software you choose. There are two popular Starlink packages which you might use: CCDPACK and FIGARO. CCDPACK includes some tools for conveniently managing the preparation of many frames and supports the propagation of errors. It also has a GUI-based interface for setting up automated reductions. CCDPACK also has better facilities for combination of images, offering several estimators. In IRAF the `noao.imred.ccdred` package should be used for CCD data preparation.

4.2 Finding the Spectrum in the Image

A spectrum running parallel to the X-axis of an image can be located by taking a cross-section (sometimes called a *slice* or *cut*) parallel to the Y-axis. For a simple spectrum, a plot of such a section will appear as in Figure 3. If the spectrum is faint, or has strong absorption features, it may be better to take the median of several columns and use this to investigate the spatial profile of the spectrum. This will avoid the case where a single-column section passes through the spectrum and there is no signal as, for example, in a strong absorption feature.

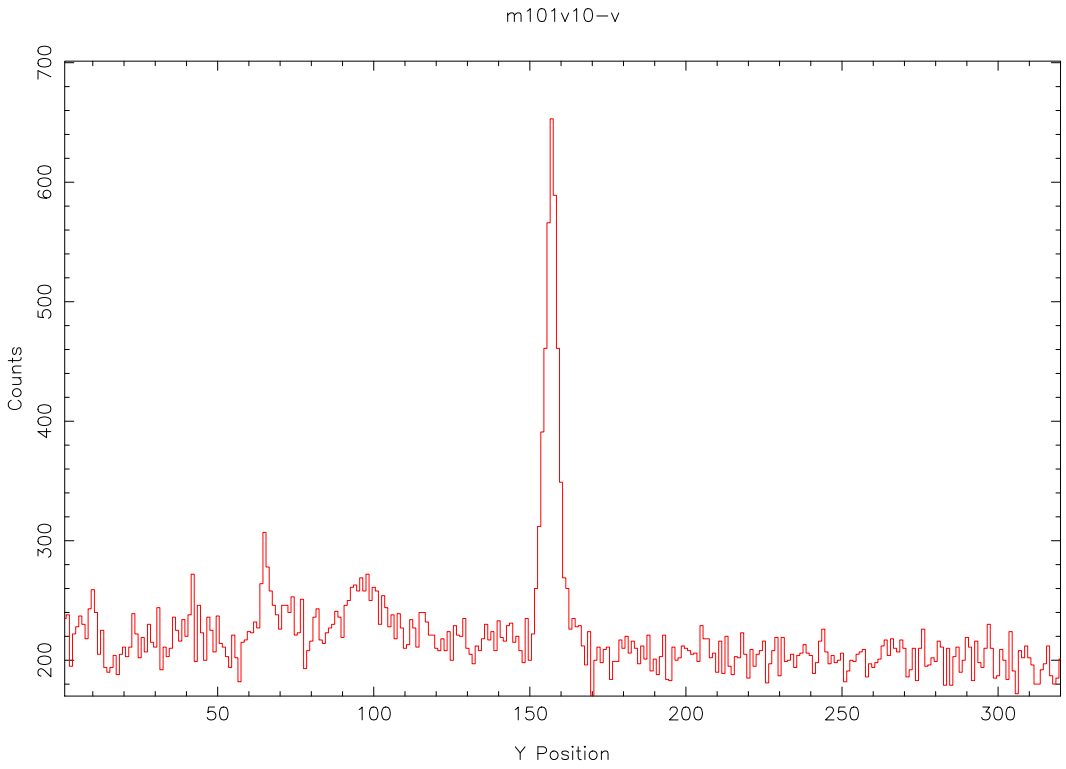


Figure 3: A cross-section through a simple spectrum.

4.3 Choosing the Object and Background Channels

Once we know which row of the image the spectrum is centered on, we can proceed to investigate the spatial profile by re-plotting the cross-section 'zoomed in', centred on that row. Figure 4 (left) shows such a plot. The object spectrum has a width at the base of about 18 pixels. We want to sum the signal from 9 pixels to the left of profile centre, to 8 pixels to the right of profile centre to give the gross spectrum for this object. On either side of the spectrum are flatter areas which we have chosen as background channels. We could take the median value of these areas to give a measure of the background. The background level in the object channel can then be estimated by taking the median of these values. In Figure 4, the left-hand background channel runs from 10- to 16-pixels to the left of the profile, and the right-hand channel from 9- to 14-pixels to the right of the profile.

Figure 4 (right) shows a more awkward case, the background is infected, possibly by a cosmic-ray hit, or more likely, by a second, faint object on the slit. In this case the right-hand background channel has been made smaller to avoid the feature. Use as many 'clean' background pixels as available—as long as this does not degrade resolution in the dispersion direction (see §4.6). A larger number of pixels gives both a better background estimate, and better rejection of cosmic rays. We might use a linear fit to the backgrounds to allow for the fact that the channels are asymmetrical.

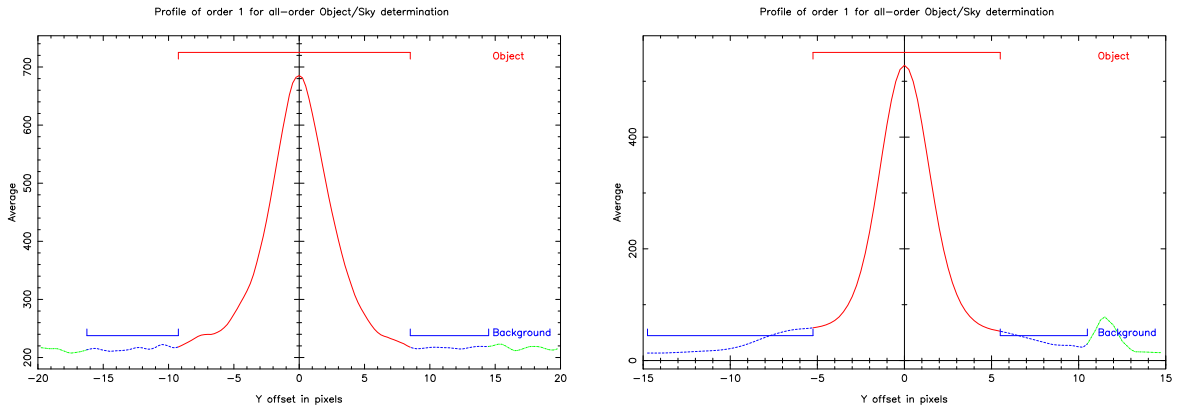


Figure 4: Selecting object and background channels (left), and a similar case, with the right-hand background channel adjusted to avoid a feature on the slit (right).

4.4 Tracing the Spectrum

In practice, the spectra produced by an instrument are not aligned precisely parallel to the pixels in the detector used to record them. There are many reasons for this, not least that the effect of atmospheric refraction at the blue and red ends of the spectrum is different. As the position of the centre of the spectrum is subject to shifts in the spatial direction, we need to find the centre of the spatial profile of the spectrum at points along the dispersion direction, and fit a curve to these points. This process is known as *tracing* the spectrum. Figure 5 shows the curve fitted to a trace overlaid on the image of the spectrum.

When summing the signal from each sample in the dispersion direction, the positions of the object and background channels are re-centered relative to the trace. This ensures that the 'software slit' used for the extraction correctly samples the spectrum.

For a bright spectrum with a continuum, tracing is done easily; however, if the spectrum has strong absorption features or is very faint, it can be difficult to find the trace centre along the whole length of the spectrum. There is no perfect way to overcome this problem. The strategy you adopt in these cases will depend upon which frames you have available; it may be adequate to use a flat-field with the dekker stopped down; it may be adequate to use a bright reference star image. As long as you have one frame which you can trace you can use this as a *template* for the extraction. The template trace may have to be shifted to register with the science frames. You can determine if this is needed by over-plotting the trace on the science frame and inspecting the fit. In the worst case of no frame proving suitable for tracing, the trace path may be defined manually; try to avoid this if at all possible.

4.5 Flat Fielding

A flat-field frame is needed to correct for the pixel-to-pixel sensitivity variations in the detector (and the small-scale variations in the throughput of the instrument optics). When imaging, an evenly illuminated scene is used as the flat field. For spectroscopy we would want to use a light source which is 'white', *i.e.*, the same brightness at all wavelengths. In practice this is not possible and the lamp used for the flat field has some wavelength-dependent variation in brightness. (The effect becomes greater as dispersion increases.) To remove the response of the lamp, a

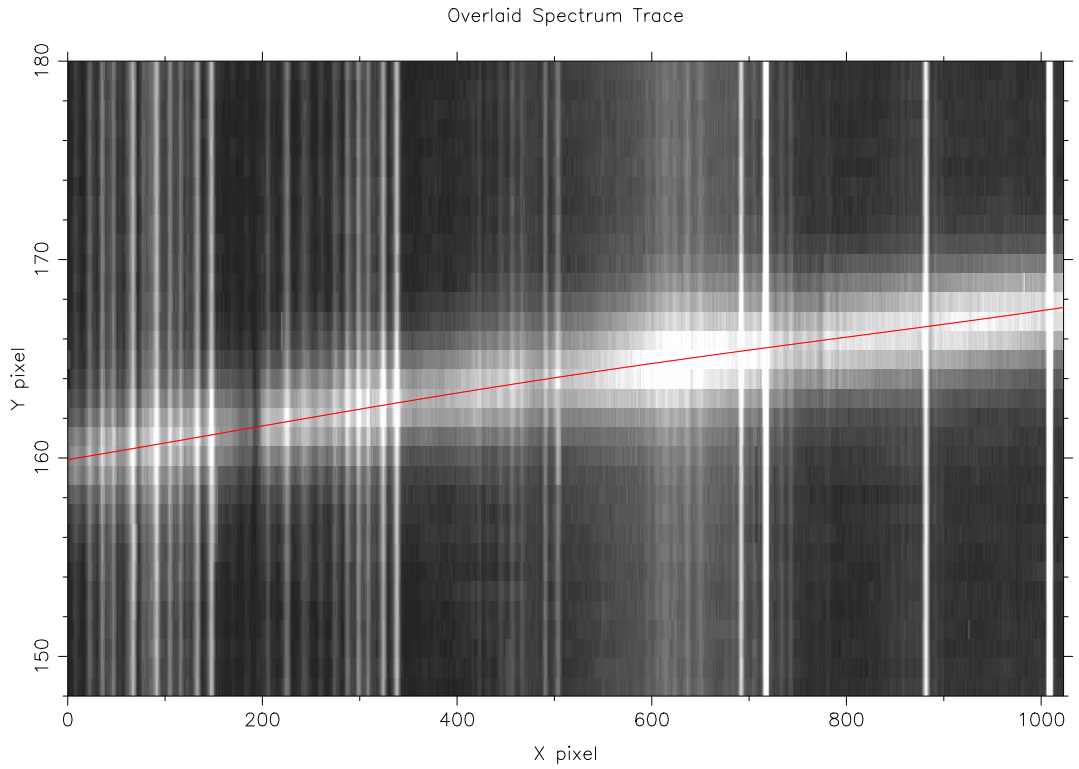


Figure 5: Trace of a spectrum overlaid on the image.

low-order curve is fitted to the response—assuming the spectrum runs with dispersion parallel to the X-axis, we collapse the flat-field frame by summing columns, producing a 'spectrum' for the lamp, then fit a curve to this spectrum. Figure 6 shows how one of these 'spectra' might appear. The spectrum has a simple, continuous shape with small-scale noise superimposed. Having fitted a curve, we divide the flat-field by it so as to leave the small-scale sensitivity variations only.

4.6 Extracting the Spectrum

In the previous steps nothing has been done to the data; instead, models of the data have been produced. We have:

- a model of the path of the spectrum in the image,
- a model of the object and background channels.

At this point we are ready to extract the spectrum. The procedure can be very simple, at each sample point along the dispersion direction we sum the signal from all the pixels selected as object; we then subtract a value based upon the pixels in the background channels. The same extraction is applied to both the target and reference star images (if any), as well as to the arc images.

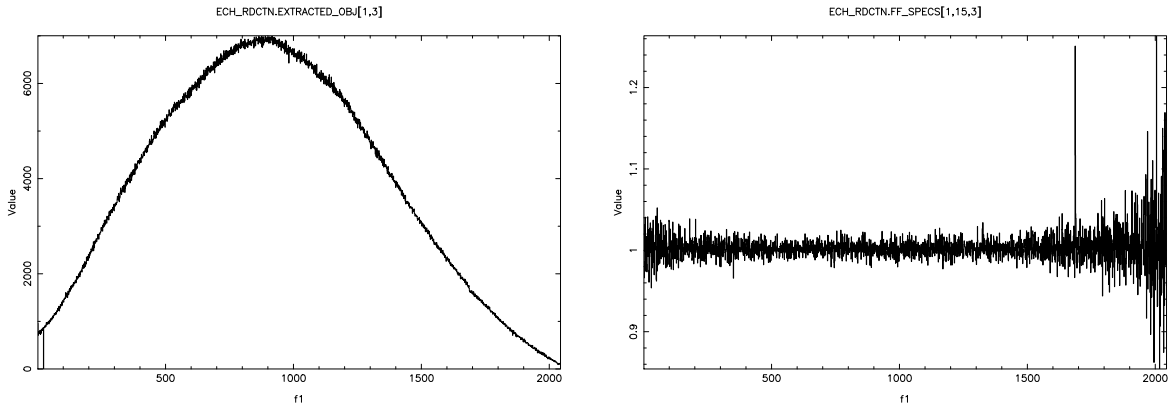


Figure 6: A section through a typical flat field for spectroscopy (left) and the same flat field after the large-scale curve has been divided out (right). Produced using ECHOMOP.

The procedure outlined above is known as a 'simple' or 'linear' extraction. In many cases such an extraction is adequate; however, this method does not make the most of the data available. A profile curve can be fitted to the spectrum *in the spatial direction*. At the centre of the profile the signal is greatest; on the outside, the signal falls off to the background level. If we sum these pixels in an unweighted manner we are ignoring the fact that the central pixels have a better signal-to-noise ratio as compared to those at the outside. To overcome this problem we use the so-called 'optimal' extraction scheme.

Optimal extraction is suitable for CCD data when we know the readout noise and gain of the CCD camera. The CCD readout noise is needed to calculate pixel weights. The gain is used to convert the data in the input images, which are in the 'arbitrary' units from the camera ADC, to units of recorded photons. Once we have the data in these units, we can apply an error-based weighting to the summation of data in each sample along the dispersion direction.

Optimal extraction assumes that a good model for the noise sources in our data is available; this is a fair assumption as the noise sources in a CCD camera system are well understood (well, at least at the sort of level we are interested in anyway). The main noise 'sources' are: the camera electronics, the CCD output node, and the shot noise of the electronic charge stored in a pixel (which represents the light signal). There are other sources—see Horne[10] and Marsh[13] for more details of the theory of optimal extraction.

In cases where the CCD parameters are not available, a profile-weighted extraction might be used. This weights the summation of the object pixels based upon their relative brightnesses. This should give a better signal-to-noise ratio than the simple extraction.

Which extraction scheme you select will depend on the nature of—and what you want from—your data. For bright, high signal-to-noise data there is little to be gained by going for an optimal extraction (little may be lost by doing one though…). Optimal-extraction algorithms require that the spatial profile of the object is a smooth function of wavelength. This means that optimal extraction is unlikely to be useful if spatial resolution is required and/or the spatial profile of the object varies rapidly with wavelength, as for objects with spatially-extended emission-line regions. For suitable data, optimal extraction also acts as a cosmic-ray filter: any pixel which deviates strongly from the profile model is likely to be contaminated, and can be rejected.

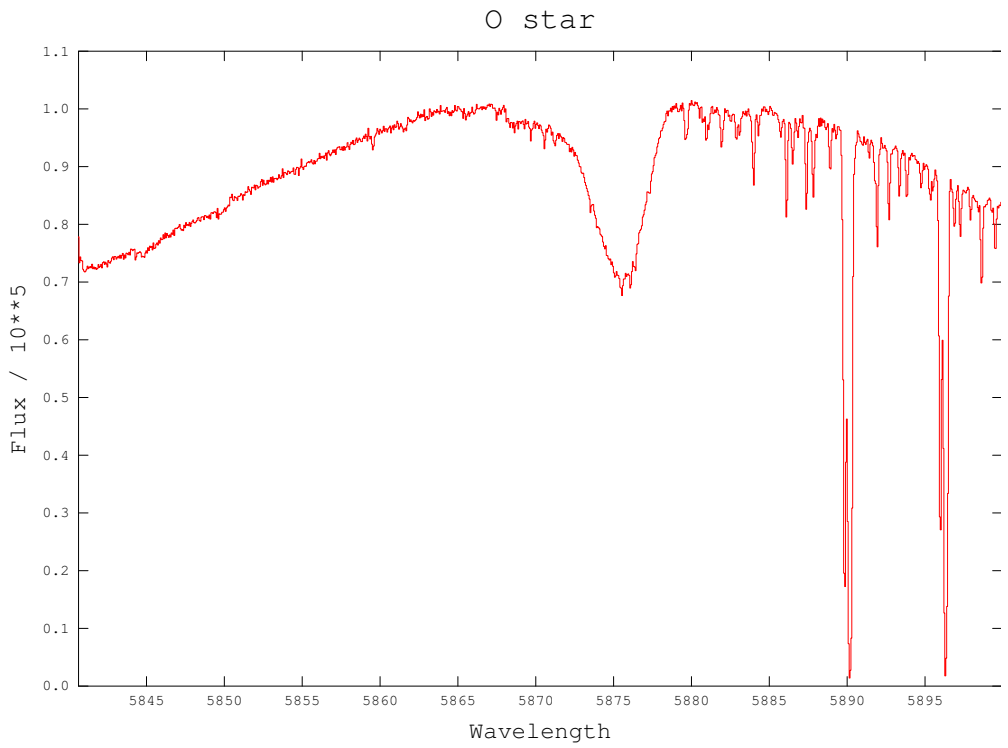


Figure 7: An extracted spectrum.

4.7 Wavelength Calibration

Once the spectrum extraction is complete we have a two-dimensional dataset: sample number and intensity. (You may also have error data for each sample.) Sample number is simply an index for each integration bin along the spectrum (*e.g.*, the X-axis in Figure 8). For some purposes this spectrum may be enough, but usually the next step in the reduction process is to find the relationship between wavelength and sample number.

The basic steps here are:

- Look at the arc (or comparison) spectrum and identify features of known wavelength.
- Fit a low-order polynomial to the arc wavelength-sample relationship.
- Paste the wavelength-sample relationship on to the object spectrum.

You should find that a list of spectral-feature wavelengths for common arc reference lamps is available on-line (see §5.9 for more details). You may also be able to lay your hands on a hardcopy of a 'mapped' comparison spectrum for your selected arc lamp, perhaps obtained using the same spectrograph as your data, this will consist of a plot (or series of plots) of the spectrum annotated with rest wavelengths. For example, Bessell & Pettini[4] should be available at most UK Starlink sites. Some people prefer the ESO arc-line atlas[3] in which the line wavelengths are easier to read.

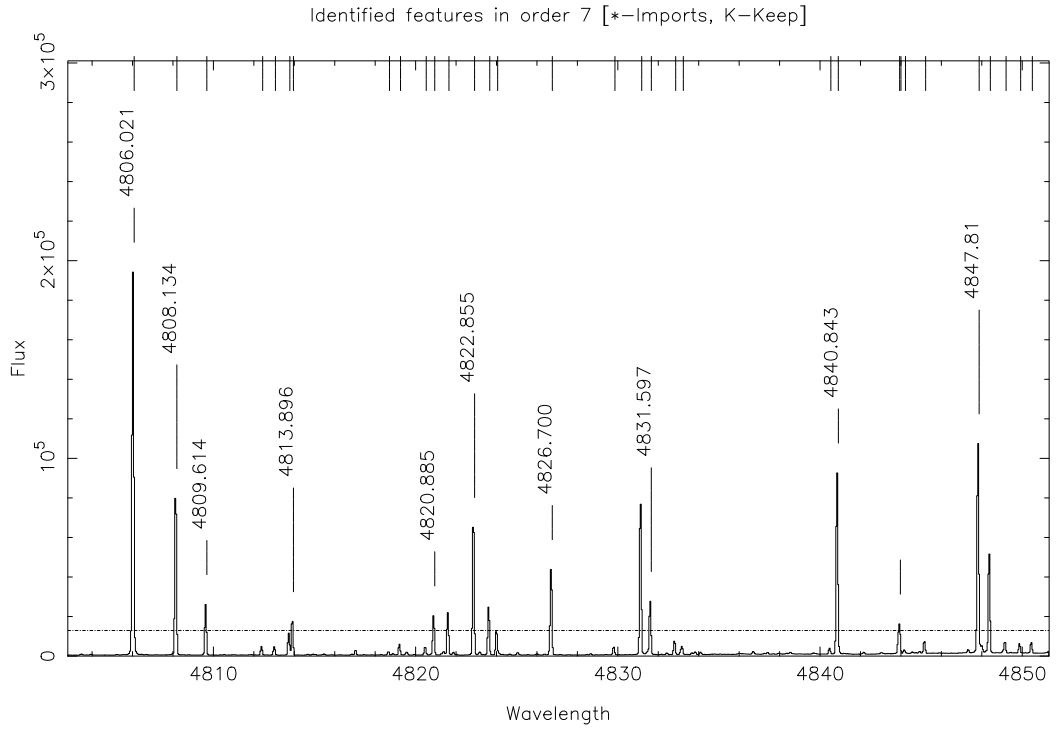


Figure 8: Line Identification: a typical plot during interactive fitting with **echmenu**.

Ideally an arc spectrum should contain at least three or four identifiable spectral lines, preferably with one close to each end of the spectrum, and one or more spread in the middle. In some cases it may be useful to refer to the object and/or reference star spectra to look for strong features of known, or approximately known, wavelength. These can be used to help you 'home-in' on features in the arc spectrum. When a fit is made to these features you will be advised of the goodness-of-fit, usually in the form of a plot of line versus deviation-from-fit or RMS deviation values for each line. You will be able to adjust the fit and reject any lines which seem to have been mis-identified, then re-fit the data.

Figure 8 shows a plot of a single order during interactive line-identification using **echmenu**.

Once you have a complete wavelength-calibrated comparison spectrum you can 'copy' the wavelength scale onto your object spectra. It may be useful to calibrate *two* arcs which bracket the object exposure in time. This will show any time-dependent variation in the wavelength scale. If there is some change (and it is reasonably small) you can take a time-weighted mean of the two bracketing wavelength scales and use this for the object spectrum.

4.8 Finishing Reduction

Once you have a wavelength-calibrated spectrum you may (again) be happy enough and not need to go any further. It is more likely that you will want to do one of two things, either flux calibrate, or normalise the spectrum.

4.8.1 Flux Calibration & Extinction Correction

If, as part of your observing programme, you take spectra of standard stars then you will be able to attempt to flux calibrate your data.

The flux-calibration process is simple: you compare the extracted standard-star spectrum to a published spectrum of the same star. At each 'wavelength' (I use the term very loosely here—see below for a proper explanation) you find the ratio of the observed to published spectrum. The result is a conversion factor at each 'wavelength'. To flux calibrate your target objects you multiply the spectra by these per-wavelength conversion factors. The effect of this is to remove instrumental-response wavelength-dependency. At least that's the idea… in practice, you must be careful to understand how the published standards are tabulated, and how this might relate to your data.

How Standards are Tabulated There are two ways in which the standard-star data are tabulated, both are tables of fluxes versus wavelengths: in one, each point has a corresponding pass-band width; in the other, each point is a fit to a continuum curve. In the first case, the published standard and your observed standard should look the same once the flux in the appropriate pass bands has been summed—even when absorption features are present in the spectrum. In the second case you must either fit a continuum to your data or, if there are no absorption features present in the wavelength range covered, you might just interpolate between the points in the published standard.

Air mass & Extinction The Earth's atmosphere absorbs and reddens light from astronomical sources. The effect is dependent upon many factors: weather, observing site, time of year, time of day (or night), and so on. The process is known as *atmospheric extinction*. The depth of atmosphere through which an object is observed is another factor determining the extinction. This depth is known as *air mass*.

When an object lies at the zenith the depth of atmosphere through which it is observed is one air mass. As the zenith angle of the object increases, so does the air mass.

The effect of extinction is wavelength-dependent and so a table of correction factors is required. Each observatory should have such a table available for its site.

You should observe your standard star through as similar an air mass to the target as possible. Unless the standard is very close to the target, and both are at a fairly high altitude, you may still need to compensate for differential extinction, rather than simply assuming the atmosphere has the same effect on both observations. It may be a good strategy to observe your standards at a range of different altitudes throughout a night, then fit a curve to these and interpolate to calibrate the science observations.

If you are fortunate, the air mass for a particular observation will automatically be present in the FITS headers of your data—otherwise you will have to calculate it.

4.8.2 Normalisation

If you are not flux calibrating your data then you will probably want to remove the low-frequency instrument profile from the data. This process is known as normalisation or blaze correction. Figure 9 shows a plot of a spectrum and the same spectrum after normalisation. The continuum

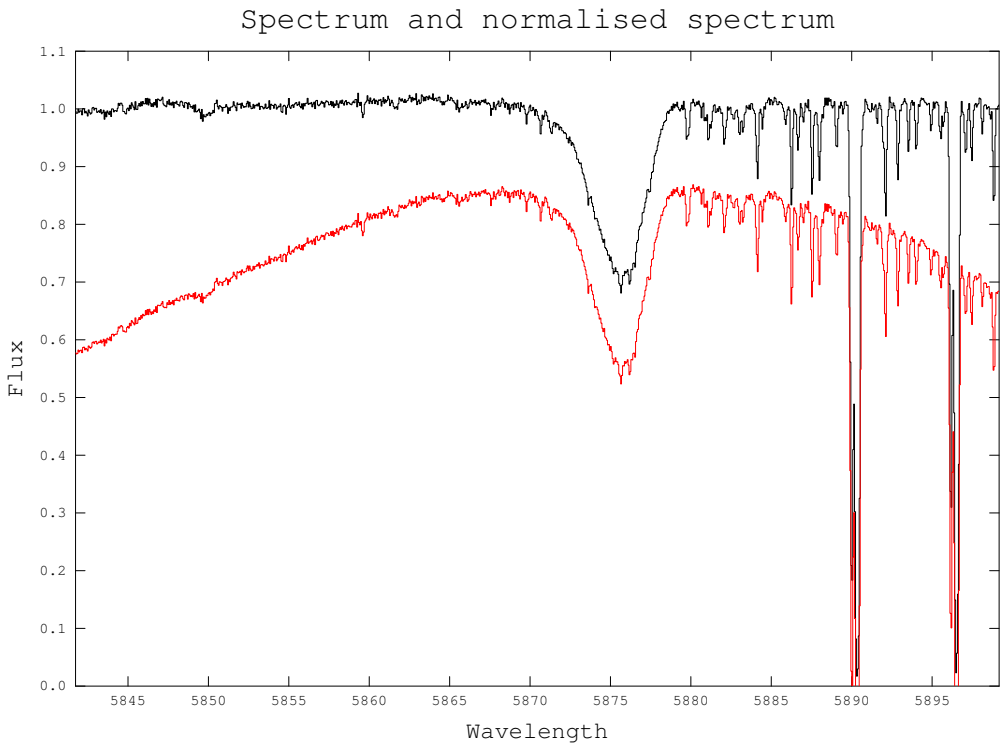


Figure 9: Normalisation: the top spectrum is the normalised version of the lower spectrum. Note that the flux values in the uncorrected spectrum have been scaled and shifted for this plot.

is more or less flat in the corrected spectrum. Normalisation can be useful when you want to look at absorption line profiles, model their shapes, or determine their widths.

There are several approaches to normalising the data. One method is to fit a polynomial or other curve to the spectrum and then divide it by the curve. This often works; however, it may well be necessary to manually select which parts of the spectrum to fit as strong spectral features will lead to a poor fit to the continuum. Another method is to 'draw' points of the continuum on to a plot of the spectrum and fit a curve to these points. This is usually an entirely manual process.

Another method is to fit a curve to the extracted 'spectrum' of a flat field and use that for normalisation. As for the object spectra, this method will only work if the flat-field is devoid of strong spectral features (which it should be, otherwise it isn't much use for flat-fielding).

5 A Worked Example

In this section, a simple run-through of the extraction process using Starlink software is given. §8 gives answers to specific questions and solutions to common problems.

Before you start the extraction you will have to do the detector-specific preparation of your data (most likely to remove CCD-related effects). If you have not done this, refer to §4.1 for an outline

of the procedure, and §1.2 for pointers to documentation of the process.

5.1 Setting Up

The first thing you need to do (if you've managed to prepare your CCD data you will most likely know this already, but...) is to run the Starlink setup. Normally Starlink software is installed in the `/star` directory and the commands you must execute are:

```
% source /star/etc/cshrc
% source /star/etc/login
```

If your Starlink software is installed somewhere other than `/star`—one place to look is `STARLINK_DIR`—then modify the commands appropriately. Most people include these lines in their shell login file (`.login` in your home directory).

Once you have done the Starlink 'login' you can initialise for any of the major packages simply by typing their names. For example, we are going to use KAPPA[8] and ECHOMOP[16] and so get a display something like:

```
% kappa    # Only needed once per session.

  KAPPA commands are now available -- (Version 0.9-3)
  KAPPA uses NAG routines, by permission of NAG ltd.

  Type kaphelp for help on KAPPA commands
  Type "showme sun95" to browse the hypertext documentation

% echomop  # Only needed once per session.
 ----------- Initialising for ECHOMOP ------------
              Echelle data reduction
         Version 3.2-0  9th December 1996

          Type "echhelp echomop" for help
    or "echhelp news" for news on changes

 Type "echwww" to start Mosaic documentation browser.
 Type "echmenu" to start the monolith.
```

We can now use KAPPA commands and **echmenu**, the menu-driven front-end for ECHOMOP.

At this point you might like to take a copy of the example data which comes with this document when installed as part of the Starlink document set. You will find the files in the directory

```
/star/examples/sc7/
```

Create an empty directory and enter it using **cd**. To copy the test data type the command

```
% /star/examples/sc7/copydata
```

You will then find you have these datafiles in your directory:

File	Description
`object.sdf`	Frame with the object spectrum
`flat.sdf`	Frame with a flat field
`arc.sdf`	Frame with a wavelength-reference arc
`object1.sdf`	Frame with a rotated spectrum

5.2 Inspecting the Images

The first thing you need to do is ensure that the images are oriented correctly. To do this use KAPPA **display** to look at the image:

```
% idset xwindows
% display object mode=pe accept
% lutspec
```

This will display the image `object` using a false-colour look-up table. You may have to use the **xdisplay**[2] command to ensure that the image is sent to the correct (*i.e.* your) screen. **display** has many options, for the moment we just need to check the orientation of the image, the `accept` keyword tells KAPPA to use default values for the parameters we have not given. The example data are correctly oriented, with the spectrum dispersion from left-to-right.

Try

```
% display object1 clear mode=pe accept
```

The `clear` parameter indicates that we want the previous display to be erased before `object1` appears. You'll see that this spectrum looks similar to the previous one, but it runs from top-to-bottom of the image—it needs to be rotated. To do this, use the KAPPA command **rotate**.

```
% rotate object1 object1r 90
```

This rotates the data in `object1` clockwise by 90 degrees and places the result in `object1r`. If you find that your real data need to be rotated, you will have to apply the change to all the data files (including arcs and flat fields). Try this to inspect the rotated data:

```
% display object1r clear accept
```

Jumping ahead (quite a way) you may want to ensure that the now correctly rotated data are arranged so that the spectrum is dispersed with wavelength increasing from left-to-right. Don't worry if you don't know how to do this—it isn't strictly necessary; however, you may find it easier to recognise the arc spectrum in an atlas if it is arranged with the wavelength increasing left-to-right as this is the way the atlases are printed. You can always come back later and flip the images if you find the arcs hard to recognise. There's no single way to spot which way the spectra are dispersed. You have to know a little about the spectra you are working with; where the strong or large features are. You might look at the arc as well for some familiar pattern in the reference spectrum.

If you do find that you need to reverse the dispersion direction, you can use KAPPA **flip** to do this:

```
% flip object1r object1rf 1
```

This tells KAPPA to flip the first (X-) axis in the data and put the resulting image in the dataset `object1rf`. You can see here the common practice of appending letters to the dataset names to indicate what processing they have undergone—'r' for rotate, 'f' for flip. This can be helpful if you have many images and are going to work over several sessions.

We can now proceed to the first stage in the actual spectrum extraction.

5.3 Locate the Spectrum

The spectrum extraction is done using the ECHOMOP package. At this stage, you will find it easiest to work through the example extraction here, rather than trying to understand all the facilities and parameters of ECHOMOP.

The first stage is to start **echmenu**:

```
% echmenu ech_rdctn=rdf1 display=yes soft=xwindows tune_mxskypix=31
This is ECHOMOP Version 3.2-0.
 ... setup messages ...
 Main menu options:
  0. HELP/HYPER (ASCII or hypertext help).
  1. Start a reduction.                  16. Check trace consistency.
  2. Trace orders.                       17. Post-trace Cosmic Ray locate.
 ... other options ...
 14. Save reduced data.                  29. System () commands.
 15. Plot order traces.                  30. Output balance-factor frame.
                                         31. EXIT (alias Q/E/QUIT/EXIT/99).
 Use -nn to edit/view option parameters.
 - Option number /'or Y for default=1'/ >
```

Each reduction is described by a database file specified by the `ech_rdctn` parameter, in this case we have chosen `rdf1`, which the program creates. The `display` parameter tells **echmenu** to overlay trace plots on the image itself so we can see how well the trace is doing. The `soft` parameter points output graphs to our existing `xwindows` screen. Don't worry about `tune_mxskypix` for now, it is needed (and explained) later on.

echmenu displays its complete menu—most of the options are specialised and we don't need them at this stage. **echmenu** works through the extraction process in a series of numbered steps (1, 2, 3, etc...). You can exit **echmenu** between stages and return later to pick up the reduction at the same point. **echmenu** tends to be quite verbose when displaying messages—don't be put off by this, most of the information is not useful at this stage and can simply be ignored. For brevity, most of these messages are not included in future screen display listings.

The prompt at the end of the above display is suggesting Option 1 *Start a reduction* which seems like a good place to begin:

```
 - Option number /'or Y for default=1'/ > 1
TRACIM - Frame for order tracing /''/ > object
 ... messages ...
INPTIM - Frame to extract data from /''/ > object
 ... messages ...
```

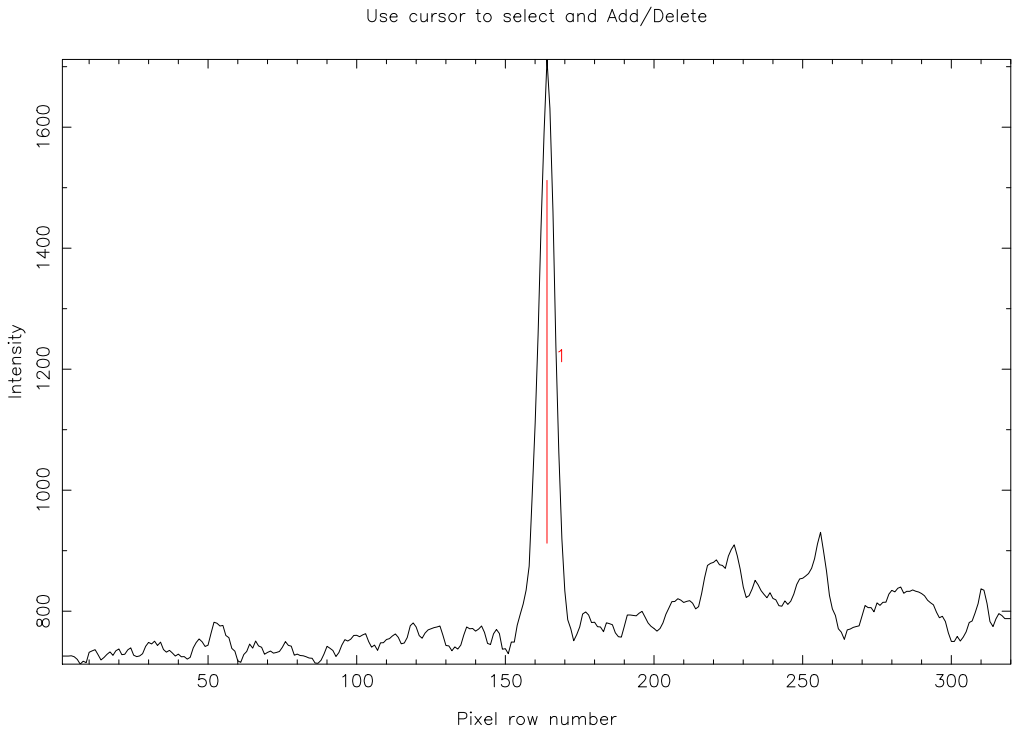


Figure 10: Order location: a cross-dispersion section produced using **echmenu** Option 1 *Start a reduction.*

```
ARC - Name(s) of reference (arc) lamp image(s) /''/ > arc
 ... messages ...
 Current number of orders: 1.
 Options:
   A - Add an order.
   D - Delete an order.
   C - Clear (delete all orders).
   R - Re-plot.
   E - Exit.
   M - Full menu display.
```

There are three prompts which appear: first, `tracim` is asking which image to use to trace the path of the spectrum—in this case we are able to use the object itself; `inptim` is asking for the name of the file from which the spectrum will be extracted, again this is `object`; lastly, `arc` is the name of the wavelength-reference image, which is called `arc` in the example data.

echmenu now displays a slice across the image `object` and you will see that it automatically locates features which look like they may be spectra. Figure 10 shows such a plot.

A short menu of options is displayed, if (as in this case) all is well, you can just hit the `E` key to accept the spectrum as correctly found. You may find that **echmenu** incorrectly guesses where the spectrum is; you can use the `C` key to remove the automatically located spectra and the `A` key to add a spectrum at the point indicated by the graphics cursor.

You will notice that **echmenu** appears to be able to handle more than one spectrum in one image;

this is the case—it can handle multiple separate spectra (*e.g.* from a fibre spectrograph) and multi-order spectra from échelle spectrographs. But we don't need to know about that at the moment.

5.4 Tracing the Spectrum

Having located the spectrum, we now get another menu; this is in fact a shortened version of the full menu with the 'most-likely' choices at the top of the list. We want the second option *Trace orders*; we only have one order but it works the same way.

```
 - Option number /'or Y for default=2'/ > 2
 ------------------------------------------------------------------------
 Starting processing task:     Trace orders.                  (ECH_TRACE)

TRCFIT - Function for trace fitting /'POLY'/ >
TRACE_MODE - Type of order tracing to use /'C'/ >
TRC_NPOLY - Number of coeffs of trace-fit function /4/ >
 Processing order 1...
 Data range (user-specified) from     10.00 to  25005.00.
 Order 1 traced from X=0 to 1025, with a sample success rate of 100%
```

The parameters here are: `trcfit` is the type of fit function to use, `poly` is often fine, `spline` works in most other cases; `trace_mode` determines which algorithm is used to find the spectrum at each X-sample (*i.e.* along the wavelength/dispersion axis), `C` means centroid and usually works fine, other options include `G` for a Gaussian fit, and `B` for a simple centre-of-gravity fit; `trc_npoly` selects the number of fit parameters (*i.e.* the order of fit for a `poly`) for a `spline` you need more parameters (seven plus number-of-knots, times two).

You will see a line plotted overlaid on the image on the graphics device. (Much like Figure 5, see page 9.) This shows the path of the trace as determined by **echmenu**. Usually the trace works well; to inspect the fit we use Option 3 *Clip fitted traces*. The only parameter you will be prompted for here is

```
TRC_INTERACT - YES for interactive order-fitting /YES/ >
```

Accept the default value (by hitting return) as we want to be able to improve the trace interactively. The first plot (see Figure 11) shows the deviation of each X-sample in image pixels from the trace path. Values less than half a pixel are fine. To see the trace points plotted on the fitted trace curve hit the `V` key (the full menu of options in the clipper can be displayed by hitting `M`). Hit `V` again to return to the 'deviations' plot. Deviant points can be clipped in a number of different ways: use the `.` key to delete the point closest to the graphics cursor; `N` clips points below the cursor Y-position; `P` clips points above the cursor Y-position. Option `R` is often useful if part of the trace has gone wrong; hit `R` to mark one end of the bad X-range and then hit the mouse (usually left) button to mark the other end. As points are clipped the display updates and the points disappear—you should see the extent of the deviations reduce.

When clipping, don't overclip—a nice 'cloud' of deviations of up to about plus or minus half of a pixel is fine. If, when using a `poly` curve you find that a very high-order fit (anything above 7th or, perhaps 8th, order) is needed to get a good fit then you probably should try a `spline` fit instead as you will start to 'over-fit' the data.

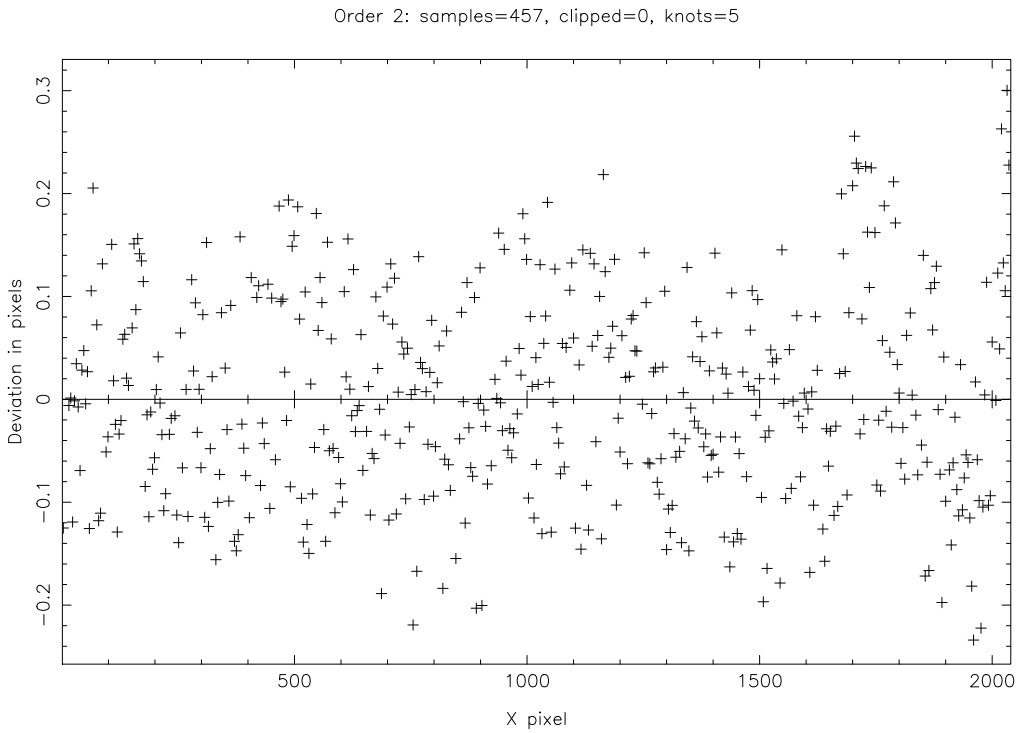


Figure 11: Trace fitting: a display of traced points versus fitted trace, produced using **echmenu** Option 3 *Clip fitted traces.* The plot is labelled 'Order 2' as this was the second order of an échelle spectrum.

5.5 Choosing the Object and Background Channels

The next step in the 'data-modelling' process is to take a section through the spectrum (cross dispersion) and decide which pixels contribute to the object signal and which to the background. The process is split into two stages: first, deciding where the edges of the dekker are; second, choosing the channels. Both stages are covered by **echmenu** Option 4 *Determine dekker/object extent*, they can be done separately as options 4.1 (dekker setup) and 4.2 (channel setup).

If you didn't specify `tune_mxskypix=31` when you started this **echmenu** session you should refer to §8.1 for details of how to do this within **echmenu**.

The first prompts in Option 4 are:

```
PFL_INTERACT - YES for interactive profiling /YES/ >
SLITIM - Frame for dekker measurement /''/ > flat
```

You should set `pfl_interact` to `yes` (or `true`, means the same thing) as you may want to adjust the dekker limits interactively. `slitim` is the name of the image you are going to use to find the edges of the slit; the object image will do, but the flat-field image is often better as the flat-field has steeper, and so clearer, edges. Specify `flat` if you are working with the example dataset.

As in previous options, you will be presented with a displayed graph and a short menu of options. The graph shows a section across the dispersion generated by averaging the cross-dispersion profile in some fraction of the X-samples. By default **echmenu** uses the middle 20%

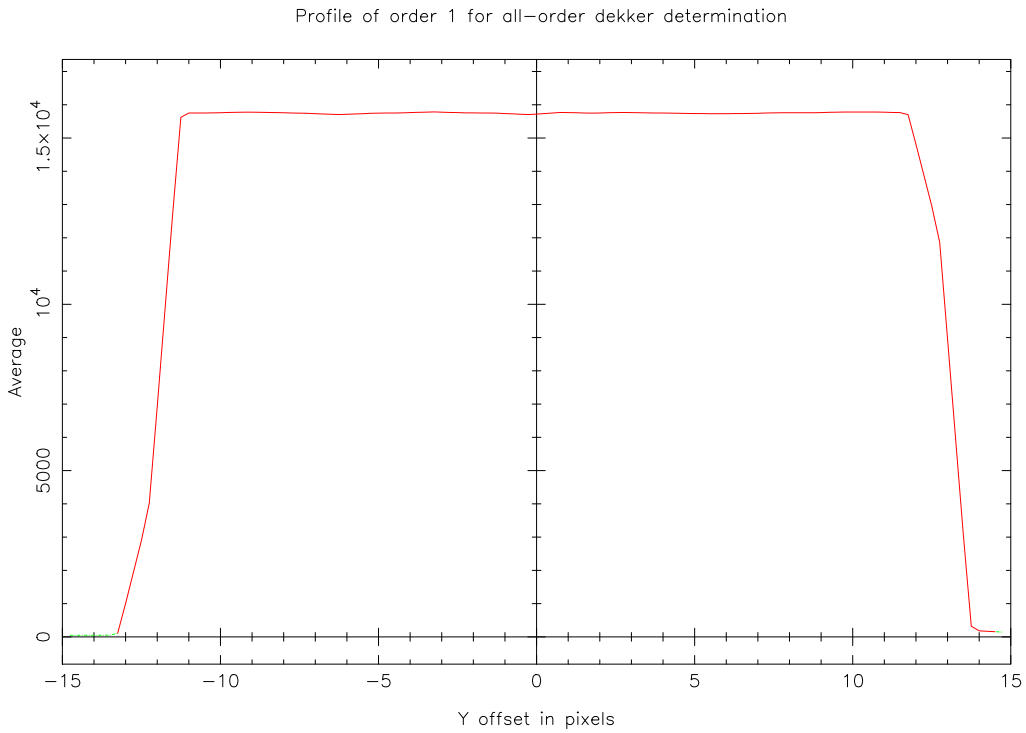


Figure 12: Finding the dekker-limits of the slit: a plot of a section through a flat field, produced using **echmenu** Option 4.1 *Determine dekker extent.*

of the pixels in the X-direction. This gives a smooth(er) curve than a single-pixel section and avoids the possibility of accidentally choosing a section across a strong absorption feature.

In the plot (see Figure 12) pixels which **echmenu** has guessed as inside the slit are displayed coloured red[2] and in a solid line; pixels outside the slit are in green and a dot-dash line style. To change the upper or lower boundary, position the graphics cursor at the new dekker edge position, and hit either the `U` key (upper limit) or `L` key (lower). The new selection will be displayed. If the profile appears to be wider than the displayed graph then click a `U` or `L` with the cursor outside the appropriate edge of the outline box and the display will expand. Once you are happy with the selection of the edges of the dekker hit the `E` key.

The next graph which pops up (such as Figure 4 (see page 8)) is slightly more complex. Once **echmenu** has the dekker limits it attempts to estimate which pixels within the slit are from the object, and which are background (called sky pixels in ECHOMOP). In the plot, pixels chosen as object are shown in red and a solid line; pixels selected as background are in blue and a dashed line; pixels outside the dekker are again shown in green and a dot-dash line. To help indicate the object and background channels, the pixels selected are also marked with horizontal 'brackets'; the object at the top (red); the background at the bottom (blue).

At this point you should examine the profile carefully, looking for suitable object and background channels. It may be the case that the default **echmenu** settings automatically select good channels; however, better results are often obtained by manually tweaking the selections. You

[2]The colours referred to will only be apparent if your medium/display supports colour.

can still adjust the dekker limits at this point, again using the `U` and `L` keys. Use the `S` key to mark a particular pixel in the profile as background (sky) and `O` to mark it as object. If you are working with the example datasets, try hitting `S` with the graphics cursor somewhere near the middle of the profile; you will see the display update and a pixel will change from object to background. Click `O` to change it back to object.

In the same way as in the trace-clipping option you can use the `R` key to start marking a range of pixels; press either `O` or `S` with the graphics cursor at the other end of the range to mark the pixels as object or background respectively.

You may have some pixels which appear to be neither object nor background—perhaps there is some other object contaminating the slit. Use the `I` key to mark these pixels to be ignored, such pixels appear in black, solid-line style on the plot.

At any point hitting the `M` key will display the full menu of options. Hit the `E` key when the channels look sensible. We now have the basic parts of the model describing the data—a trace, and object and background channels.

5.6 Flat Fielding

echmenu provides many options for flat fielding, these are handled by Option 5 *Model flat field*.

```
FFIELD - Name of flat-field image /''/ > flat
FLTFIT - Fitter for flat-field /'MEAN'/ > median
```

There are only two prompts which appear at this point. `ffield` is the name of our flat-field image (in this case `flat`) you can specify `none` if you don't need flat fielding or have no image to use. `fltfit` is the type of model to fit to the flat field. We have selected `median` as this gives better rejection of deviant data than `mean`, which simply uses the average value over nearby pixels to generate a smooth fit to the continuum lamp spectrum. There are many other options, and the fit can be under interactive control. A simple model is usually fine.

5.7 Extracting the Spectrum

The next stage in spectrum extraction is to model the background. **echmenu** Option 6 *Model sky* performs the modelling and you can accept the default values for the parameter prompts and get a reasonable model. If you intend to perform optimal extraction you will need to find the `photon_to_adu` (gain) value and `readout_noise`. You can do this using KAPPA **fitslist** to list the FITS extension in the input dataset, *e.g.*, for the `object` in the example data:

```
% kappa    # Only needed once per session.
... setup messages ...
% fitslist object
```

You should find the appropriate CCD parameters in the header, see §8.4 for more details. If the information you need is not present in the FITS headers you will have to consult the observatory's CCD manual.

The only other data required prior to extraction is the profile model, and this is only needed for optimal extractions. **echmenu** Option 7 *Model object profile* will generate a suitable model. If

you have been working through the example reduction data in a single session, **echmenu** will not prompt you for any further parameters when you run Option 7.

And so to Option 8, *Extract orders 1-D*. The only new parameter for this option is

```
EXTRACT_MODE - Extraction mode /'O'/ >
```

The default value, `O`, means optimal extraction. Other options are: `S` for a simple unweighted extraction, and `P` for a profile-weighted extraction.

5.8 Looking at the Extracted Spectrum

echmenu has an Option (number 27) for inspecting the contents of the current reduction database file. You can start the option by typing `P` (for 'plot') at the main-menu prompt. Once in the plotter, a menu appears and new prompt:

```
- Option /''/ >
```

If you type `obj` (for object) and then hit return, you will get a plot of the extracted object spectrum (see Figure 7, page 11, for an example). You can get a list of things you can plot by typing `D` (for directory). Here's some of the ones which might interest you at this stage:

You type	You see	You type	You see
`SKY`	Sky spectrum	`SKYV`	Sky spectrum errors
`OBJ`	Extracted object spectrum	`OBJV`	Extracted object errors
`ARC`	Extracted arc spectrum	`FWAV`	Fitted wavelengths
`FFLT`	Fitted flat-field balance factors		

We haven't generated the wavelength scale, so `fwav` isn't yet available.

Another useful plotter option is `L` (limit X- and/or Y-range of plot). You are prompted for the limiting values in both X- and Y-directions (Y as plotted, rather than in the source image—so for `OBJ`, Y will be flux; for `FWAV`, Y will be wavelength); enter zero if any limit is to be determined from the data.

Once you have finished inspecting the spectrum, hit `Q` to return to the main menu.

5.9 Wavelength Calibration

There are two stages to **echmenu** wavelength calibration; the first stage is simple; the second stage may be more complex. The first step is to run Option 9 *Locate arc line candidates*. This searches the arc spectrum for features which look like arc lines. The result is a small list of potential features which can be used in the second stage, wavelength calibration.

echmenu Option 10 *Identify features* performs wavelength calibration. There are a number of prompts:

```
ECH_FTRDB - Reference line list database /'$ARCDIRS/THAR'/ > $ARCDIRS/CUAR
ARC_TYPE - Type of arc lamp used /'$ARCDIRS/THAR.ARC'/ > $ARCDIRS/CUAR.ARC
WAVFIT - Function for wavelength fitting /'POLY'/ >
AUTO_ID - YES for fully automatic identification /FALSE/ >
HI_WAVE - Longest wavelength to search for arc lines /0/ >
LOW_WAVE - Shortest wavelength to search for arc lines /0/ >
MAX_DISPERSION - Max dispersion (Units per pixel) allowed /1/ >
MIN_DISPERSION - Min dispersion (Units per pixel) allowed /0.01/ >
W_NPOLY - Number of coeffs of wavelength fitting function /7/ >
```

Arc lamp database information for CuAr and ThAr lamps is available, if you use some other lamp you will need a list of wavelengths for features in the spectrum of the lamp. Often a ThAr lamp is used, however, in the example data the lamp used is CuAr, so you can should set `ech_ftrdb=$ARCDIRS/CUAR` and `arc_type=$ARCDIRS/CUAR.ARC`. Note that the names of these files are case-sensitive. `wavfit` is the type of fit to use, `poly` should be fine.

If you select `auto_id=true`, **echmenu** will attempt a fully automatic wavelength calibration. This will often work; however, it is better to inspect the fit manually, rather than just accepting it automatically; so we accept the default `auto_id=false`.

`hi_wave` and `low_wave` are limits on the wavelength range that might be covered by the spectrum; setting the limits both to zero (the default) indicates we have no idea of the wavelength range. However, if you do have a rough idea of either or both limits then enter them here. Constraining the wavelength range speeds up the process of feature identification. The units are Angstroms for the ThAr and CuAr databases.

`max_dispersion` and `min_dispersion` are the limits on the dispersion of the spectrum. Again, if you know the dispersion, set these values close to the value to help constrain automatic feature identification. Inspection of FITS headers may reveal the dispersion used in your data, you might otherwise look in the instrument handbook for the spectrograph used. The units here are Angstroms per detector pixel. The default values are set for common échelle spectrographs and so may be too small for your data. If in doubt, set `max_dispersion` to a large value and leave the default for `min_dispersion` as it is.

`w_npoly` is the number of coefficients of fit to use for the wavelength polynomial. The default value of 7 is fine. The fitter will adjust the order automatically the first time a fit is made if the value is unusable.

Once you have worked through all the prompts for Option 10 another mini-menu will appear offering a default of `I` (identify features manually). Accept the default by pressing return and (yet) another menu appears. The graphics display will also update showing the arc spectrum. Each potential feature identified in the arc spectrum will be marked by a short vertical dash above the feature. This will be similar-looking to Figure 8 (see page 12) except the features will not yet be labelled with wavelengths.

All these sub-menus may appear confusing, however, you can just accept the default to get to the interactive wavelength calibration. The other options are used when the dataset is multi-spectrum or multi-order. You should now have a menu like this one:

```
Option [Info,Del,Set,Thresh,Auto,New,Plot,Re-interp,Worst,BClip,Fit,+-=,
        XClip,Clear,Keep,List,Move,Zoom,Ozoom,>,<,Exit,Quit,Help,?]
```

As you can see, there are rather a large number of options. To apply an option, type its first letter with the graphics cursor on the displayed plot. Try an `A`, which will attempt automatic calibration. You will see some details of the fit displayed and then a plot with the wavelengths of the identified features overlaid. For the example dataset this is all you need to do—you now have a wavelength scale. If the display zooms on to only a small part of the spectrum hit `R` to get a full-spectrum plot.

At this point you may want to have one of the atlases mentioned earlier to hand to check the identifications manually. You can inspect the line-wavelength lists on-line, for example, the ThAr list by:

```
% more $ARCDIRS/THAR.ARC
```

The important point in deciding whether the fit is good is the RMS error of the fit. Below is an example of a good fit:

```
          Line    Wavelength  Calculated Discrepancy    RMS if
                              Wavelength                omitted

  1     297.366    4561.347    4561.349       0.001     0.00232
  2     451.861    4567.240    4567.238      -0.002     0.00243
... other lines ...
  9    1300.145    4598.763    4598.759      -0.004     0.00212
 10    1599.801    4609.567    4609.568       0.001     0.00207

RMS error: 0.00250.

Selected degree for fits: 4.
Number of features identified: 10.
```

You can see that the RMS error is quite small, also the 'RMS if omitted values' are all of similar values. If a mis-identified or badly recorded feature is included, then this will be indicated by a much lower 'RMS if omitted' than the other features. To remove such a feature: position the graphics cursor on the feature; hit `D` to delete the feature from the list of features to be fitted; hit `F` to apply the new fit.

It is important not to over-fit the feature list. In the above example ten points are fitted with a fourth-order curve; this is fine, seventh-order would be too high as the errors would then be fitted away. The plus and minus (`+` and `-`) keys change the order of fit. Press the `E` key when you are happy with the fit.

5.10 Normalising the Spectrum

Once you have got the extracted, wavelength-calibrated spectrum you have two options: you can output the data from ECHOMOP; or, you can normalise the spectrum to give a more-or-less flat continuum. The next section explains how to output data from ECHOMOP. This section outlines using Option 11 *Flatten order shape* to normalise the data.

Enter `11` at the main **echmenu** prompt and the following parameter prompts will appear:

```
FFIELD - Name of flat-field image /''/ > flat
BLZFIT - Function for blaze fitting /'POLY'/ > spline
BLZ_INTERACT - YES for interactive blaze-fitting /NO/ > Y
BLZ_NPOLY - Number of coeffs of blaze fitting function /7/ > 28
```

`ffield` is the name of the image to be used to model the instrument profile, if you have been working through the example data in one session you won't be prompted for this parameter again. You don't have to use a flat-field image; one option is to use the object spectrum image itself and fit a curve to the continuum. If you are following the example reduction, choose to use the flat-field image `flat`. For modelling the instrument response a `spline` curve is often better than a `poly`, this is simply because it is unusual for the response to be well represented by a simple polynomial, hence we have chosen `blzfit=spline`. `blz_interact=y` selects interactive adjustment of the fit. `blz_npoly` sets the number of parameters for the fit; we have chosen 28 as this is a `spline`. (Remember, for a spline we need seven plus number-of-knots, times two, parameters.)

After entering values for the parameter prompts, you will see a display very similar to that in the trace clipper; the same bit of code does both jobs. This means that you can use the same keys (*e.g.* `V`, to see the fitted curve versus the measured data) to view and improve the fit. Use `+` and `-` to increase or decrease the order of fit; `C` to clip points further from the X-axis than the current graphics cursor position. Once you are happy with the fit, hit `E` to accept it.

At this point you can re-run Option 27 (or `P`, the plotter) and view the flattened spectrum:

```
- Option number /'or Y for default=1'/ > P
... messages ...
- Option /''/ > OBJ
```

5.11 Output the Spectrum

echmenu Option 14 *Save reduced data* handles the output of data. You can produce NDFs which DIPSO[11], KAPPA and FIGARO can read, or you can output ASCII listings of the spectra.

```
ECH_RDUCD - Output spectrum data file /'ECH_RDUCD'/ > spectrum
RESULT_FORMAT - Output format required /'NDF'/ >
RESULT_TYPE - Type of result output required /'EXTOBJ'/ >
```

Here `ech_rducd` is the rather opaque-sounding prompt for 'output file name'; `result_format` is `NDF`, select `ASCII` for a listing instead; `result_type` should be set to `extobj` to output the extracted spectrum, `extarc` to output the extracted arc.

If you followed the example data reduction, you will now have a file `spectrum.sdf` in the working directory which contains a normalised, wavelength-calibrated spectrum which can be read by other Starlink software.

6 Longslit (2-D) Data Reduction

In this section each of the basic steps in the longslit spectral data reduction procedure are described. Many of the initial steps are similar to those for 1-D data and reference will be made to those sections with notes for any slight differences. Later steps do become 2-D data specific.

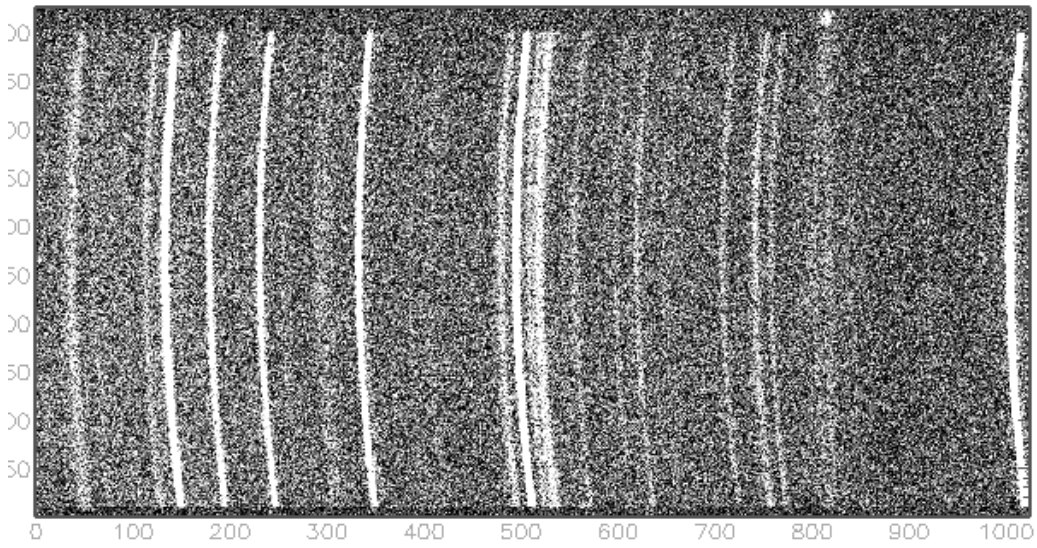


Figure 13: A 2-D arc line image showing curvature along the spatial axis.

Examples of practical techniques using these ideas are to be found in §7.

6.1 Image Preparation

The essential basic difference with 2-D data is that we are not looking to extract a region of the CCD frame into a single spectrum but wish to preserve the spatial direction in order to study changes in the spectrum with position.

Given this, the initial image preparation is the same as for 1-D data which is described in §4.1 and §4.5.

The order in which 2-D data is calibrated is different to that for 1-D data. Many of the tasks in processing the data (e.g. background sky emission line subtraction) require the 2-D frame to be correctly oriented with spatial and dispersion axes perpendicular.

As can be seen in the example arc line image (see Figure 13,page 27, for an example) the lines have a definite curvature along the spatial axis. This is due to the different path lengths the light takes through the optics along the slit. Correcting for this is carried out along with wavelength calibration.

6.2 Wavelength and curvature calibration

The process of wavelength calibration along the dispersion axis also allows us to correct for the geometric curvature of the frame described above.

The wavelength calibration works in the same way as described for 1-D spectra, matching the wavelengths of known arc lines to the pixel values for the centres of the lines and performing a polynomial fit to the data. Before this can be done we must calculate the curvature of the arc lines on the frame in order to straighten them.

Both of these tasks are typically done by the fitting of Gaussian functions to each arc line profile for each cross-section of the frame. These then give the centre of each arc line for each cross-section which can then be fitted by a polynomial along the arc line. A polynomial matching pixel and wavelength values along the dispersion axis can then be fitted.

For 2-D frames these fits are output in an .iar file, which contains the polynomial coefficients. These can then be used by other programs to 'scrunch' the data, in other words to resample the data correcting the distortions and calibrating the wavelength axis.

As with 1-D spectra it is important to use arc lines which enable a sensible polynomial fit to be made, i.e. well spaced over the whole of the CCD. Similarly, as low an order polynomial as sensibly fits the data should be used to prevent large errors being generated at the ends of the dispersion axis.

6.3 Scrunching

With the solution calculated for the calibration of the 2-D frame the next stage is to apply this to the data. This process is called scrunching.

The program (e.g. iscrunch in figaro) will take your .iar file containing the solution and a data frame and apply the corrections and calibration to it. This will involve re-binning the data and adding a new wavelength axis.

It can also be a useful test to scrunch the original arc frame, which should result in a set of perfectly straight arc lines.

A slightly more advanced program (iscruni in figaro) will allow you to use two .iar files and use the average values when calibrating the data. This is useful for long data exposures where arcs may show a shift from arc line images taken before and after the data.

6.4 Flux calibration

Flux calibration works in much the same way as for 1-D spectra, see §4.8.1.

The only difference comes when applying the calibration to the data when each row of the 2-D frame is calibrated rather than a simple spectrum.

7 A 2-D Worked Example

As there may be readers who have come to this section without reading the worked 1-D example I will reiterate a number of instructions. If you are familiar with the early steps please feel free to skip on.

Before you start the extraction you will have to do the detector-specific preparation of your data (most likely to remove CCD-related effects). If you have not done this, refer to §4.1 for an outline of the procedure, and §1.2 for pointers to documentation of the process.

7.1 Setting Up

The first thing you need to do (if you've managed to prepare your CCD data you will most likely know this already, but...) is to run the Starlink setup. Normally Starlink software is installed in the `/star` directory and the commands you must execute are:

```
% source /star/etc/cshrc
% source /star/etc/login
```

If your Starlink software is installed somewhere else, then modify the commands appropriately. Most people include these lines in their shell login file (`.login` in your home directory).

Once you have done the Starlink 'login' you can initialise for any of the major packages simply by typing their names. For example, we are going to use FIGARO[18], KAPPA[8] and TWODSPEC[23] and so get a display something like:

```
  % figaro     # Only needed once per session.

----------- Initialising for  Figaro ------------
             General data reduction
           Version 5.3-0  29 December 1997

         Type "fighelp figaro" for help
     or "fighelp news" for news on changes

Type "showme sun86" to browse HTML documentation

Use "abbrev" and "noabbrev" to turn parameter name
abbreviation on and off.


  % kappa    # Only needed once per session.

     KAPPA commands are now available -- (Version 0.9-3)
     KAPPA uses NAG routines, by permission of NAG ltd.

     Type kaphelp for help on KAPPA commands
     Type "showme sun95" to browse the hypertext documentation


  % twodspec   # Only needed once per session.

  TWODSPEC commands are now available -- (Version 0.9-0)

  Type "showme sun16" to browse the hypertext documentation.
```

We can now use FIGARO, KAPPA and TWODSPEC commands.

As with the 1-D example, you might like to take a copy of the example data which comes with this document when installed as part of the Starlink document set. You will find the files in the directory

```
/star/examples/sc7/
```

Create an empty directory and enter it using `cd`. To copy the test data type the command

```
% /star/examples/sc7/copy2ddata
```

You will then find you have these data files in your directory:

File	Description
`object2d.sdf`	Frame with the object spectrum
`arcframe2d.sdf`	Frame with a wavelength-reference arc
`quartz2dscrun.sdf`	Frame with a quartz lamp flat field

7.2 Inspecting the 2-D images

At this stage it is assumed that the image files have been de-biased and cosmic ray cleaned.

The first thing to do is to put the image frames on display for inspection. This can be done using KAPPA **display**:

```
% idset xwindows
% display object2d clear mode=pe accept
```

This shows the data frame with two emission lines, in this case from the Honeycomb Nebula in the Large Magellanic Cloud. (see Figure 2, page 4).

The arc line image can be shown using:

```
% display arc2d clear mode=pe accept
```

The curvature of the arc lines, which we wish to correct, is obvious (see Figure 13, page 27.

There are three key elements to the inspection of the images at this stage,

Check for rotation Typically done by visual inspection or taking cuts through a continuum source (e.g. a star on the slit) in the object frame.

Check the wavelength direction Wavelength must increase from left to right and is checked by spotting known arc or observed line patterns.

Noting the range of y-axis values over which the arc lines are seen This is so that we know the sensible range over which to try and fit the arc lines.

These frames are correctly rotated (i.e. have the wavelength axis parallel to the x-axis) and have wavelength increasing to the right. If this is not the case with your data, methods for correcting this were detailed previously in §5.2.

With the arc line image displayed we can carry out the third item mentioned above, noting the range of y-axis values the arc lines extend over. To do this we can use the KAPPA **cursor** command:

```
% cursor
```

The cursor should then be placed at each end of an arc line and the left mouse button clicked. This will report the coordinates of the point in the terminal window. When this has been done click the right mouse button to exit.

When I did this I obtained these values:

```
Picture comment: KAPPA_DISPLAY, name: DATA, reporting: PIXEL coordinates
 p1 = 144.6 (pixel)   p2 = 12.9 (pixel)
 p1 = 149.8           p2 = 492.7
```

So we have a range of 13 to 493 in the y-axis.

We can now go on to the fitting of these arc lines.

7.3 Arc line fitting

To carry out the fitting we use the TWODSPEC command `arc2d`. This program provides a number of menus and interactive displays and will be described in the text.

```
   % arc2d arcframe2d

arcframe2d[1024,525] THAR FOR PREV
Results structure present

=========< A r c _ O p t s >=========

New      : Set up line identifications from scratch
Repeat   : Use existing line identifications
Clone    : Use line identifications from another file
ARC_OPTS - Enter arc fit option /'ne'/ >
```

We wish to carry out a new identification of the arc lines so type `new` and press return. The program will ask for confirmation of this action, enter `y` to the questions.

```
MAXLINES - Maximum number of lines to allow room for /5/ >
```

This question asks for the number of arc lines you wish to use. Ideally you would like a number of lines spread evenly over the CCD array. As the resulting fits are typically 3rd order or lower then 5 lines is a reasonable default. Press return.

```
YSTART - analysis lower limit /20/ > 13
YEND - analysis upper limit /40/ > 493
```

In order to identify the arc lines we have to select them interactively. These question ask for the range of y-axis values over which to extract a spectrum for this identification. Simply enter the values which we found using the cursor earlier.

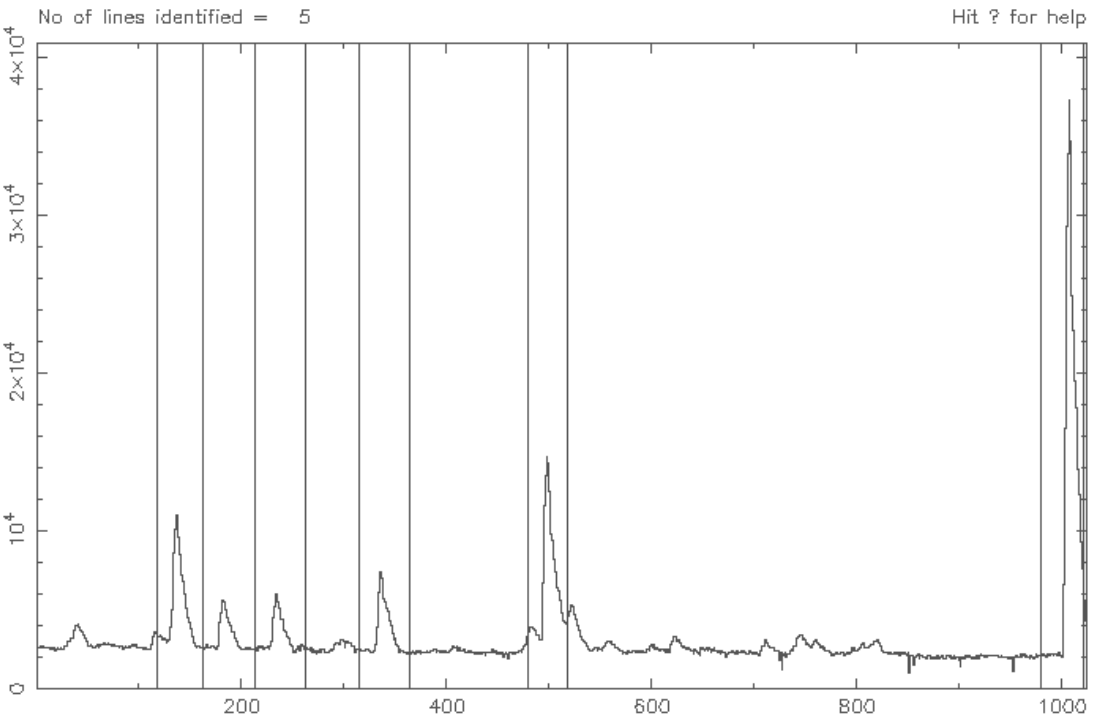


Figure 14: An arc line spectrum with lines selected between the tram-lines.

At this point the extracted spectrum will be displayed on the graphics window. You will see that the lines are quite broad as we have summed the data over a range in the y-axis and hence in curvature of the line.

The lines are then selected using the mouse, going from left to right across the spectrum in increasing wavelength. To select the lines to be used the left mouse button is clicked to the left and right of the arc line to mark tramlines. Only one arc line should lie within the tramlines and there should be some continuum left on each side of the line in order to fit a good baseline (see Figure 14, page 32)

7.4 Arc line identification

Once you have marked out each of the arc lines you wish to use press `e` to exit this stage. You are now given the option of selecting a line list, simply a text file of known lines for particular arc lamps or emission lines. These can be used as a reference when identifying the selected lines. In this case the arc was a Thorium Argon lamp for which a list is not available. The lines will then have to be identified using an arc line catalogue.

```
=========< L i n e   L i s t   M e n u >=========

Emission   : Emission lines
Absorption : Absorption lines
Neon       : Neon arc
```

```
Cuar       : Copper-argon arc
Helium     : Helium arc
Iron       : Iron arc
SKy        : Sky lines
STored file: User supplied list in file
Ok         : All tables read in
```

So we can type ok at this prompt.

The program now displays the first selected line in more detail and prompts you for an identification on the terminal screen.

```
L I N E   I D E N T I F I C A T I O N
-------------------------------------
Identify line number  1

=========< I d e n t i f i c a t i o n   M e n u >=========

[number]               : Line wavelength
Width [number]         : Redraw with different width
Next                   : Go to next line
Display                : Start/stop displaying line tables
Id [wavelength] [name] : Give line ID/wavelength
Quit                   : Exit leaving lines unidentified
```

As we have found the arc line pattern in the catalogue we can enter the known wavelengths for each arc line. The first line has a wavelength of 6708.97 angstroms and so we enter

```
id 6708.97 line1
```

and then confirm this is correct.

The second line is now displayed and we similarly enter the wavelength for this line. The wavelengths for the five lines marked in Figure 14) are:

```
id 6708.97 line1
id 6711.34 line2
id 6713.97 line3
id 6719.20 line4
id 6752.83 line5
```

After this the program will display the entered list of identifications for you to check. Enter no to 'edit line list' to reach the main menu:

```
=========< M a i n   M e n u >=========

Look : Look at values of data cube
Soft : Produce soft-copy plots of diagnostics
Hard : Produce hard-copy plots of diagnostics
Tols : Apply tolerances
Exit : Leave the program
```

```
Disp : Evaluate dispersion relation
Gaus : Fit Gaussians to line profiles
Add  : Add more lines
Poly : Fit polynomials in X-Sect direction
```

7.5 Curvature correction

So we now have the lines identified with their wavelengths. The next stage is to fit a Gaussian to each arc line for each row of the image. The centres of these Gaussians will then be used to calculate the corrections required.

Fitting these Gaussians is selected by entering `gaus` at the main menu prompt. Again we are asked for the limits over which rows the program should try and fit Gaussians to the data. These should be the same limits as you entered earlier, and the program will suggest these values as defaults. The next question regards the blocking factor. There is cost in processing time depending on how many Gaussians you fit and one way to reduce this is to add a block of rows together. Typically a blocking factor of 4 percent of the total number of rows is reasonable, so for our case a value of 18 pixels. This value is not crucial, so long as it is not larger then any obvious distortion in the arc line across the CCD. Blocking also brings a benefit in that it will increase the signal to noise ratio of the data and is often essential in getting a good fit to low signal arc lines.

Enter `18` and press return. The next question asks if you wish to see the Gaussian fits as they are carried out. If this is the first time you have done this you may want to say yes, but experienced users might say no as typically the fits are not very exciting and do take some time to go through. An example Gaussian fit is shown in Figure 15, page 35.

After this has been completed you are returned to the main menu. With the Gaussians fitted the next stage is to calculate the straightening of the arc lines. This is done by fitting a polynomial to the centres of the Gaussians for each arc line in the y-axis direction.

Enter `poly` and press return. Enter `no` at the next prompt regarding errors and `no` to the next question regarding weights as we wish to fit the polynomials solely on the centres of the Gaussians.

Now displayed is a plot showing the residual sum of squares of the polynomial fit to the fitted centres for increasing polynomial order. As can be seen in Figure 16, page 36, the third order fit looks to be a good one.

If you reply `yes` to the request to go on to next plot you will be shown the residuals for a first order polynomial. Press return again will show the second order fit and so on. Looking at the residuals to the third order fit we can see they are less than ± 0.1 pixels. At this display enter `no` to the request to see further orders.

The program then asks for the order you wish to use:

```
ORDER - order for polynomial fitting /2/ >
```

Enter `3` here and press return.

Now the data points and polynomial fit are displayed for the first arc line (or tooth as it is described here). Again we can continue to look at the data and fit for each of the arc lines.

When you reach the last line or type `no` earlier you are asked if you wish to produce a hard copy of the plots. For now select `no` and the program will ask if you want to accept the fitted polynomials:

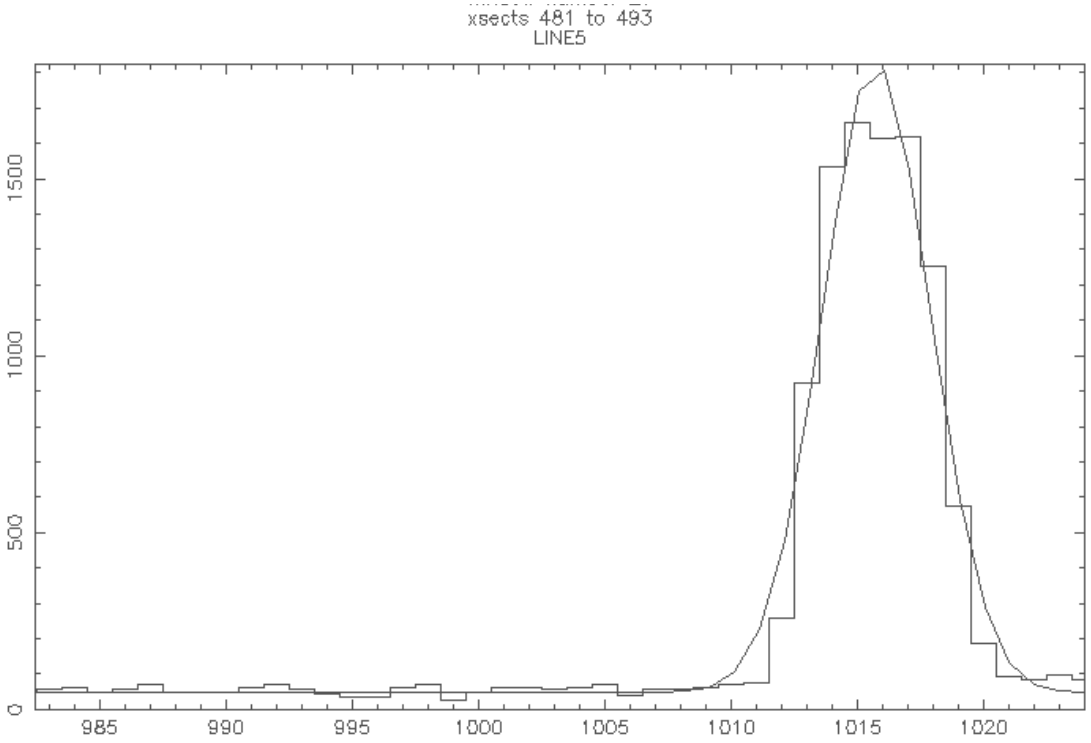


Figure 15: An example Gaussian fit to an arc line spectrum.

```
=========< C o n t i n u i t y   F i t s >=========

Accept : Accept fits
Retry  : Try again
Quit   : Give up

CHARACTER_VALUE - Continuity Fits /'ACCEPT'/ >
```

Enter `accept` to return to the main menu.

7.6 Calibrating the wavelength axis

With the straightening of the arc lines calculated the next step is to calibrate the wavelength dispersion axis. From the main menu enter `disp` to select the 'Evaluate dispersion relation' option.

The dispersion screen is then shown on the xwindows display (see Figure 17, page 37,). The arc lines are listed with boxes to toggle the use of that line in the calculations.

On the far right of the display is a selector for the order of fit to be used and then the y-axis (or X-section) number to perform the fit at.

In this case we shall use a 3rd order fit (which should already be selected) and to perform the fit at a y-axis value of around 300. You will need to click next to 300 in the slider at the right hand side of the display. The marker should then be shown at 300.

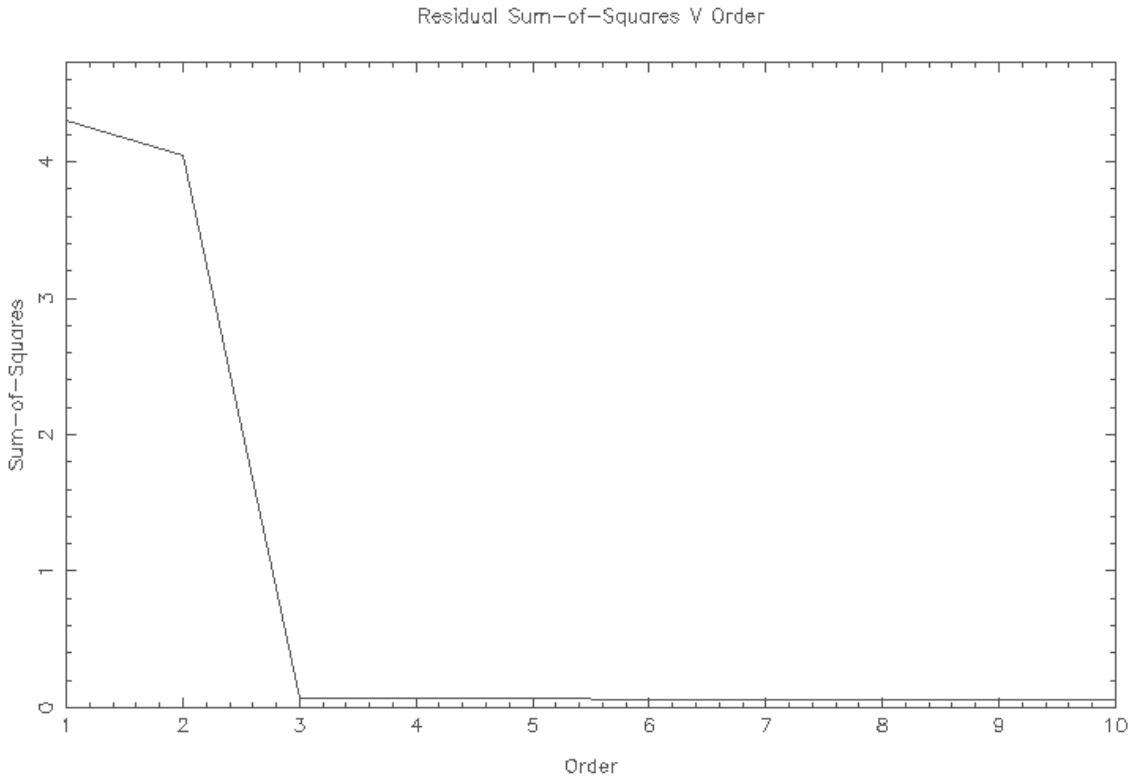


Figure 16: The residuals with increasing order for a polynomial fit to the arc line curvature.

Then carry out the fit, which solves the non-linear pixel value to wavelength value relation, press `f` while the cursor is in the main panel.

The program then displays the polynomial used for the fit in the upper panel and the residuals in the lower panel. Residuals of less than ± 0.05 are usual for a good fit. See Figure 18, page 38, for an example fit.

We could then go on to try a 2nd order polynomial fit - select order two by clicking the order selector at the right of the screen and press `f` again. You can see then residuals are now around ± 0.5.

Select order 3 again and refit the data (press `f` again).

When you are happy you press `a` to accept the fit. Control now reverts to the terminal window where you can opt to see tables of the fit results. Enter `no` to this and the next question regarding copying fits from one line to another. You will be asked to confirm the fits are ok.

When the fits are completed a line looking like:

```
Minimum start wavelength = 6705.248, maximum end wavelength = 6754.836
```

will have been displayed. These numbers are important and are used in the scrunching step to come.

The program then returns to the main menu and reports the writing of an .iar file which contains our polynomial fits.

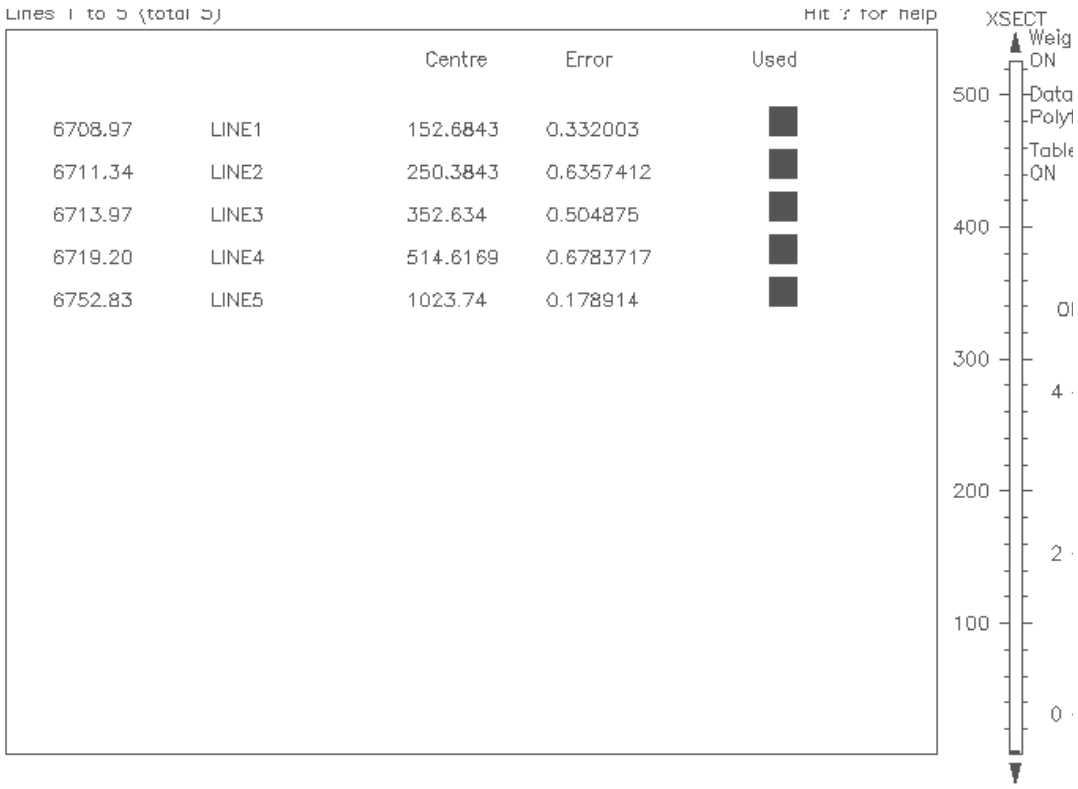


Figure 17: The dispersion calibration control screen.

We can now select exit to leave the program.

The .iar file generated is a simple ascii file and can be looked at with more:

```
 % more arcframe2d.iar

 2d fit to data in image arcframe2d

image dimensions  1024 by   525
number of rows that could not be fitted =     0
maximum chi-squared error =       0.00
maximum degree polynomial used =   3
             1            0.3543959884354487D-07 -0.1650474022720115D-04
  0.2625885040194589D-01  0.6705221506967161D+04
             2            0.3544256762805665D-07 -0.1649068558149317D-04
  0.2625527448138038D-01  0.6705225714745435D+04
             3            0.3544542125840266D-07 -0.1647696602714036D-04
  0.2625177126897877D-01  0.6705229808939712D+04

....
```

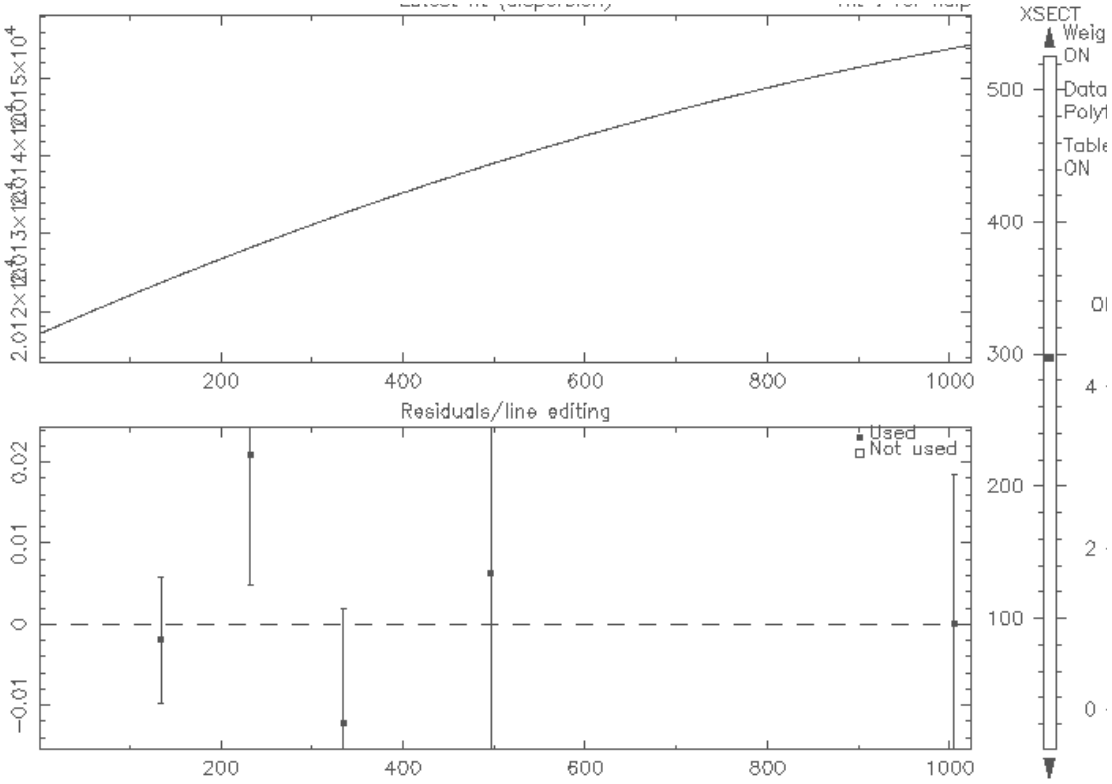


Figure 18: An example dispersion fit, displaying the polynomial used for the fit in the upper panel and the residuals in the lower panel.

7.7 Scrunching

The .iar file can be used by the FIGARO **iscrunch** program in order to correct and calibrate your data files:

```
% iscrunch object2d
```

You will then be asked for the name of the .iar file:

```
FILE - (FIle) File containing results of 2D arc fit /''/ >
```

enter `arcframe2d`.

The next question asks how many bins (pixels) you would like your new image to have. The simplest answer is to keep the same number of pixels that you had in your original image, hence for this file enter `1024`

The next question ask if you want to sample your data in logarithmic bins, enter `false`.

You will now be asked for the start and end wavelengths of your data, use the numbers noted earlier after the dispersion had been calculated.

```
WSTART - (WStart) Wavelength of center of first bin /6693.583/ > 6705.248
WEND - (WEnd) Wavelength of center of last bin (or increment) /6759.628/ > 6754.836
```

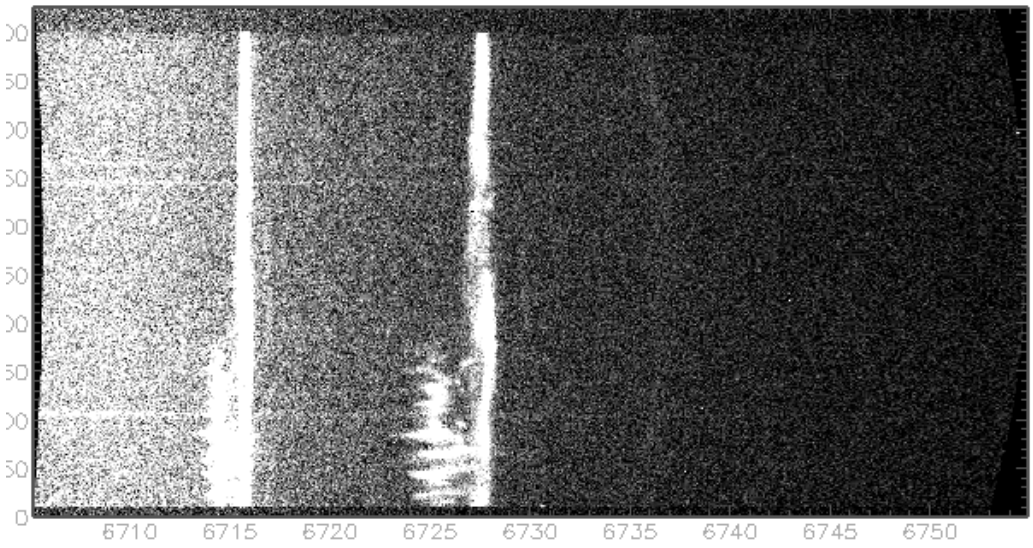


Figure 19: The data frame, now calibrated and curvature corrected.

The next question asks if the data should be treated as a flux per unit wavelength. Typically for data from spectrometers the answer to this is `false`.

Reply `true` to use quadratic interpolation for the data at the next question.

The final question asks for the output filename, e.g. object2dscrunch:

```
OUTPUT - (OUtput) Name of resulting scrunched image /''/ > object2dscrunch
```

We can now view this scrunched file using:

```
% display object2dscrunch clear mode=pe accept
```

which should look something like Figure 19, page 39.

A further check on the process is to scrunch the arc frame, in order to check that the arc lines do indeed come out perfectly straight.

7.8 Blaze correction

To remove the wavelength dependent sensitivity or blaze from the data we need to use a quartz flat. The file quartz2dscrunch.sdf is a ready scrunched quartz lamp image.

```
% display object2dscrunch mode=pe accept
% lutgrey
```

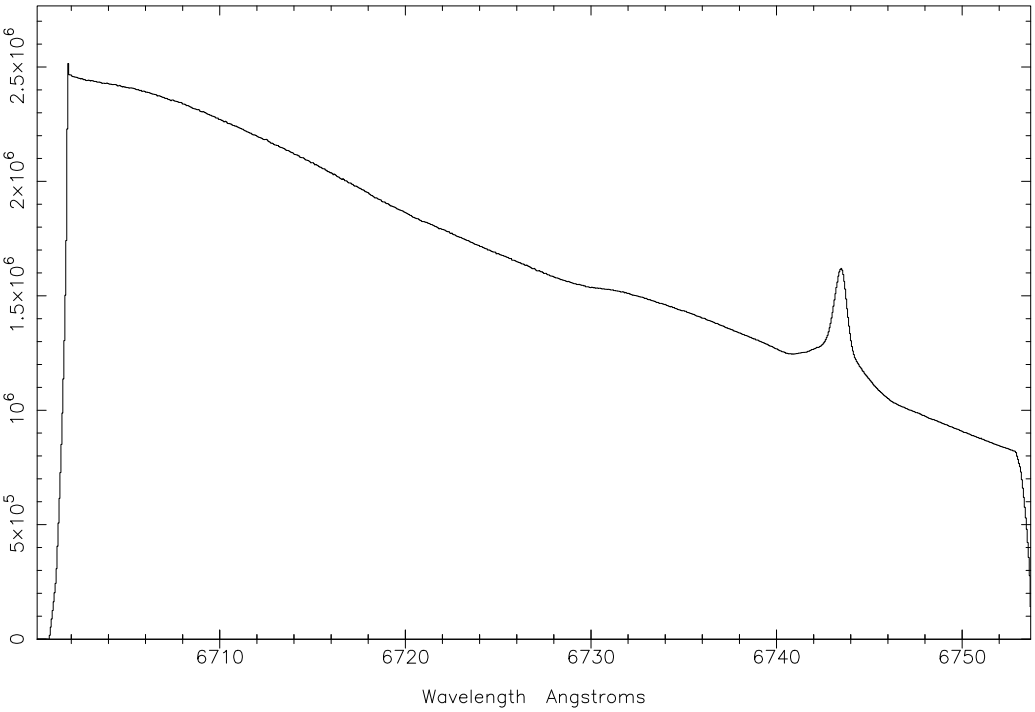


Figure 20: The extracted quartz spectrum.

The bright line towards the right of the image is a ghost. We wish to generate a smooth fit to the quartz lamp and the easiest way of doing this is to add together the rows to form a single spectrum. We use the values for the range in y-axis obtained earlier from measuring the arc lines.

```
% extract quartz2dscrunch 13 493 quartzspec
```

This generates a spectrum quartzspec.sdf which we can view with

```
% splot quartzspec whole=true autoscale=true hardcopy=false label=quartz
```

As can be seen in Figure 20, page 40, this spectrum has zero values at each end and the ghost line appears superimposed on the quartz lamp continuum. In order to remove these effects and obtain just the continuum we fit a polynomial curve to the data. We use the FIGARO **cfit** command:

```
% cfit output=quartzfit
```

Use the cursor and mark points (using the a key) along the spectrum, including points at the start and end where we have no data and ignoring the ghost line. When you have finished this press x the exit. The fitted line is then drawn over the spectrum.

We can view this on its own with

```
% splot quartzfit whole=true autoscale=true hardcopy=false label=quartzfit
```

We now have a smooth spectrum to divide into our data but before we do this one further stage is to normalize this curve. To do this we need to find the mean value for the quartzfit spectrum:

```
% stats quartzfit
```

For my fitted spectrum this gave these results:

```
   Pixel statistics for the NDF structure
/home/mips/star/starlink/cookbook/example/quartzfit

      Title                    : QUARTZ SII FOR HONEYCOMB
      NDF array analysed       : DATA

         Pixel sum             : 1.7037235E9
         Pixel mean            : 1663793
         Standard deviation    : 511041.5
         Minimum pixel value   : 794998.6
            At pixel           : (1024)
            Co-ordinate        : (6753.693)
         Maximum pixel value   : 2492099
            At pixel           : (1)
            Co-ordinate        : (6700.157)
         Total number of pixels : 1024
         Number of pixels used  : 1024
```

The key number is the pixel mean value of 1663793, which we will divide the spectrum with to give a mean value of 1.

```
% icdiv quartzfit 1663793 quartzfitnorm
```

We can now finally use this normalised spectrum to correct our observed data using the FIGARO **isxdiv** command:

```
% isxdiv object2dscrunch quartzfitnorm object2dscrunchnorm
```

This command divides the quartz spectrum into each row of our data frame to produce the quartz corrected file `object2dscrunchnorm.sdf` (see Figure 21, page 42.

) with:

```
% display object2dscrunchnorm clear mode=sc low=6 high=25 accept
% lutgrey
```

With the data calibrated you have now completed the data reduction of a 2-D longslit spectral array. For analysis of this data you may want to use the TWODSPEC[23] LONGSLIT package or DIPSO[11].

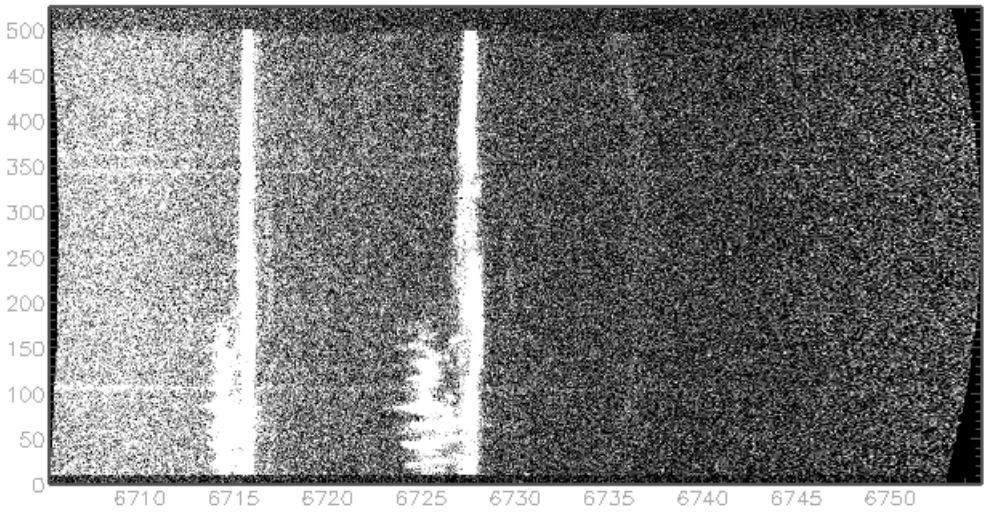


Figure 21: The calibrated data frame, corrected for the instrumental response.

8 Cookies

8.1 How to Use the echmenu Parameter Editor

In the worked example **echmenu** was extensively used. ECHOMOP is very flexible; there are a large number of parameters available which can be tuned to suit your data. Parameter values can be supplied on the command line, for example:

```
% echmenu tune_mxskypix=31
```

This is useful as normally `tune_mxskypix` will simply take its default value (21) without ever prompting.

Another method for checking and setting parameter values is to use the parameter editor built in to **echmenu**. If, for example, you want to inspect the parameters for Option 4 *Determine dekker/object extent*, you type `-4` at the main **echmenu** prompt:

```
- Option number /'or Y for default=1'/ > -4
```

A display like this will appear:

```
Parameters for: Determine dekker/object extent. (ECH_SPATIAL)
A * means the parameter must be set; the displayed value is the default.
Select number of parameter to change:

 0. Exit
 1. INPTIM          =*''            9. TUNE_MAXPOLY    = 50
 2. PFL_INTERACT    =*TRUE         10. TUNE_MXSKYPIX   = 21
 3. PFL_MODE        =*'A'          11. TUNE_OBJABV     = 0
 4. SLITIM          =*''           12. TUNE_OBJBLW     = 0
 5. TUNE_DIAGNOSE   = FALSE        13. TUNE_PFLSSAMP   = 301
 6. TUNE_DEKABV     = 0            14. TUNE_SKYHILIM   = 0.5
 7. TUNE_DEKBLW     = 0            15. TUNE_USE_NXF    = 0.2
 8. TUNE_DEKTHR     = 0.8

- Parameter number /1/ >
```

You can see that each of the task parameters is displayed with the current value, except for parameters marked with a `*`, for which the default is shown. To change a parameter you type its number (*e.g.* `10` for `tune_mxskypix`) and then enter the new value at the prompt.

When you are happy with the values set, enter `0` (zero) to exit the parameter editor.

8.2 Looking at FITS Header Cards: Air Mass

There are several tools available for looking at FITS headers in KAPPA. The simplest is `fitslist` which displays the complete list of FITS header cards. For example,

```
% kappa    # Only needed once per session.
... setup messages ...
% fitslist object
```

will display the FITS header for the example dataset included with this document. This can be used to retrieve information such as the air mass:

```
% fitslist object | grep -i air
AMEND   =             1.350784 / Airmass at approx. end of exposure
AIRMASS =             1.323036 / Airmass at approx. start of exposure
AMSTART =             1.323036 / Airmass at approx. start of exposure
```

As you can see, the above example has retrieved three cards matching the string `air`, the `-i` option for **grep** ensures that the search is not case-sensitive.

Having extracted these data from the FITS header, we can now estimate the air mass for the exposure—about 1.337 in the example (that's the mean of the start and end air masses).

8.3 Looking at FITS Header Cards: Wavelength Coverage

Another use for `fitslist` might be to track down details of the wavelength range covered by the spectrum. Four invocations of **fitslist** will get all the information needed:

```
% kappa    # Only needed once per session.
... setup messages ...
% fitslist object | grep -i wavelength
CENWAVE =                4499 /Approx. central wavelength (Angstroms)
% fitslist object | grep -i dispersion
DISPERSI=                  33 / Nomimal dispersion (Angstroms/mm)
% fitslist object | grep -i pixel
CCDXPIXE=           24.000000 / Size of unbinned pixels in x (micron)
CCDYPIXE=           24.000000 / Size of unbinned pixels in y (micron)
% fitslist object | grep -i ccd.size
CCDXSIZE=                1124 / X dimension of digitised CCD frame
CCDYSIZE=                1124 / Y dimension of digitised CCD frame
```

In this example, we can ignore the `CCDY` cards as these refer to the cross-dispersion direction of the image of the spectrum (I'm assuming that this is the example dataset `object` again here). If you have rotated your data frames then the `CCDY` cards should be used instead.

Using this simple formula we can determine the approximate wavelength range covered:

$$\lambda\lambda = \texttt{CCDXSIZE} \times \texttt{CCDXPIXE} \times \texttt{DISPERSI}$$

Appropriate units should be used, for the example above this comes to a coverage of about 890Å, centred at 4499Å. This information can be useful when trying to identify lines in an arc spectrum, and for constraining the automatic arc-line identifier in **echmenu**.

If the information you need is not present in the FITS headers you will have to consult the observatory's instrument manual.

8.4 Looking at FITS Header Cards: CCD Characteristics

Another common use for **fitslist** is to get the details of the CCD output transfer characteristic and readout noise. Two invocations of **fitslist** will get all the information needed:

```
% kappa    # Only needed once per session.
... setup messages ...
% fitslist object | grep -i noise
READNOIS=             5.300000 / Readout noise (electrons)
% fitslist object | grep -i -e gain -e adu
GAIN    =             1.300000 / Electrons/ADU conversion factor
```

The second **grep** searches for both 'gain' and 'adu' at the same time—it so happens that both are present in the record we are after.

If the information you need is not present in the FITS headers you will have to consult the observatory's CCD manual.

8.5 Problems Finding the Spectrum

The first thing to check if you can't see the spectrum clearly when running **echmenu** Option 1, as in Figure 10 (see page 18) is that the image is correctly oriented with the spectrum dispersed from left-to-right.

See §5.2 for details of how to check the orientation. Perhaps the 'zeroth' thing to check is that you have chosen the correct image for the parameter `tracim`.

If you find that **echmenu** Option 1 does not automatically find the spectrum, and you can't see it in the displayed plot, the next thing to try is a simple plot of a sum of all the columns of the image. This can be done with a series of FIGARO commands:

```
% figaro   # Only needed once per session.
... setup messages ...
% ystract object min max slice
% splot slice soft=xw accept
```

You should see the spectrum somewhere in this section plot. Inspect the X-axis and redisplay the section, but this time zoom-in on the spectrum (in the example data the spectrum is at about Y=183).

```
% splot slice xstart=170 xend=200 soft=xw whole=f accept
```

It should now be possible to decide where the centre of the spectrum is by eye, or using the FIGARO command **icur** to measure the display.

Once you have decided where the spectrum is, you can write this into the ECHOMOP reduction structure file using FIGARO **creobj** and **setobj**:

```
% creobj _integer 1 rdf1.more.echelle.order_ypos
% setobj 183 'rdf1.more.echelle.order_ypos(1)'
% setobj 1 rdf1.more.echelle.no_of_orders
```

This above assumes the reduction structure file is called `rdf1`. Notice that you have to mark the fact that one 'order' (*i.e.* spectrum) has been found. Once you have added this information you can proceed to try **echmenu** Option 2.

8.6 Tracing Strategies

Usually the default value `C`, 'centroid' for `trace_mode` works reasonably well. Running **echmenu** Option 3 *Clip fitted traces*, and performing some small interactive adjustments, results in a good, clean spectrum trace. Sometimes, however, the object spectrum cannot be used for tracing—there may be significant absorption features present, the object frame may be severely contaminated with cosmic-ray defects.

If a flat-field frame is available this can sometimes be used for tracing. Such a frame must have been exposed with the dekker not completely open; we need to be able to find the edges of the dekker on the frame. To trace a flat field, proceed in the normal way, but set `trace_mode=e` which stands for 'edges':

```
TRACE_MODE - Type of order tracing to use /'C'/ > E
```

The trace will then be determined as the point mid-way between the edges of the dekker.

Another alternative to tracing the target object spectrum is to trace the spectrum of another target taken in the same instrument configuration. How the spectra taken with an instrument

shift from exposure to exposure varies from instrument to instrument. For stable instruments you may find that the spectrum will have the same shape over the whole night; for a less stable instrument the spectra may shift depending on the zenith angle of the telescope. You should be aware of the characteristics for the instrument you are using. When you come to select the object and background channels for the spectrum extraction you may find that you have to position the channels off centre to compensate for a shift between the trace and target spectrum exposures.

8.7 What if the Slit is More Than 21-Pixels Wide?

When modelling a spectrum ECHOMOP assumes by default that the longest slit required will be 21 detector-pixels in extent. This is often fine, however, there are many occasions when the models need to be extended over a wider range of pixels in order to get a good background (sky) channel. The number of pixels used is controlled by the parameter `tune_mxskypix`.

You can use FIGARO to display a slice across an image of the spectrum, and then estimate the value of `tune_mxskypix`:

```
% figaro   # Only needed once per session.
... setup messages ...
% ystract object min max slice
% splot slice soft=xw accept
```

A good value would be about twice the width of the spectrum at its base (`i.e.` where the signal falls off to background level).

See §8.1 for details of how to set parameter values within **echmenu**.

8.8 Better Flat-Field Models

Using a median filter for the flat-field model, by setting `fltfit=median` in **echmenu** is generally fine, however, sometimes this may not give the best result. For example, if the flat-field lamp has some a feature in its spectrum in an area which you want to use (not the best situation in the first place…but) then taking the median over a range of a few pixels (the default is 10 pixels) will not work well if the size of the feature is similar.

There are alternative fitting schemes, the most useful probably being `spline`. It may be useful to hone the flat-field model interactively, this can be done by setting `tune_ffinter=yes`. The order of fit can then be adjusted interactively.

8.9 Reviewing the Flat Field

Although you can use the **echmenu** plotter to inspect the flat-field model for your spectrum, it is not always the easiest, or best, way to find problems. Another way is to output the model to an image file and then use KAPPA **display** to inspect that image. Within the ECHOMOP package there is a stand-alone task for extracting the 'flattened' field `ech_genflat`.

```
% echomop  # Only needed once per session.
... setup messages ...
% kappa    # Only needed once per session.
```

```
... setup messages ...
% ech_genflat ech_rdctn=rdf1 ech_rducd=flattened
% display 'flattened(,170:200)' mode=pe accept
```

The above is, again, suitable for the example data supplied with the on-line version of this document.

8.10 I Can't Recognise the Arc Spectrum

What do you do if the automatic wavelength calibration in **echmenu** doesn't work, and you haven't been able to recognise any arc features?

The first thing you might try is to check for a reversed arc—one in which the wavelength decreases from left-to-right. **echmenu** doesn't check for reversed arcs by default as that increases the processing time required. To try this, set the parameter `tune_revchk=true`, either on the command line:

```
% echmenu tune_revchk=yes
```

or using the **echmenu** parameter editor to change the parameters for Option 10:

```
 - Option number /'or Y for default=10'/ > -10
```

The next thing to try is to check the FITS headers to get an idea of the wavelength range covered.

The last item which might be wrong is the lamp type. Make sure you are looking at the right set of reference lines. This is another thing which you can check in FITS headers:

```
% kappa    # Only needed once per session.
... setup messages ...
% fitslist arc | grep -i lamp
CAGLAMPS= 'CuNe+CuAr         ' / Cass. A&G box comparison lamps
ISICELL = 'ON                ' / State of grating-cell clamps
```

You can see that for the `arc` image in the example data, a CuNe and/or CuAr line list is needed.

8.11 Flux Calibration

For details of methods of flux calibration, refer to the documentation for FIGARO (SUN/86)[18], Section 4.5.

9 Glossary

- **ADC**
  Analogue-to-digital converter. An electronic device which produces a digital representation of some analogue input signal.
- **ADU**
  Literally, Analogue-to-Digital Units. These are the raw numbers which emerge from a digitiser—the 'counts' per pixel read out from a CCD.
- **Arc lamp**
  A lamp which burns with a characteristic spectrum which is used as a reference or comparison for the wavelength scale of a spectrum.
- **AAO/AAT**
  Anglo-Australian Observatory/Anglo-Australian Telescope.
- **Bias frame**
  An image generated from several raw CCD frames taken with no light incident upon the detector and of 'zero' exposure time.
- **Blaze, blaze angle**
  Literally, *to cut* in the context of gratings. Arises from the nature of some gratings where the grooves are non-symmetrical in profile in order to concentrate the incident light in one or several orders on one side of the zero order of the image.
- **Blaze correction**
  Process of normalising a spectrum to remove the brightness variation due to the blaze angle. Sometimes called ripple removal or simply normalisation.
- **Bracketing**
  A term from photography. Simply means taking reference exposures before and after the 'main' exposure **bracketing** it in time. Can be used to apply to a pair of series of exposures taken before and after science data. For example, arc frames, flat-field frames *etc.*, are usually collected both before and after observing to allow any time dependency to be found and, at least to a first order, compensated for.
- **Centroiding**
  Process of estimating the true position of the centre of a spectral order in the spatial direction, where the shape of the profile of the order can be predicted and the profile is under-sampled.

  A similar process occurs in IPCS cameras to locate photon 'events' (usually with sub-pixel accuracy).
- **Collimator**
  Optical element which produces a light beam in which the rays are (at least very nearly) parallel.
- **Comparison Spectrum**
  A spectrum from a known source, typically an arc lamp, used as a reference for the modelling of the wavelength scale of spectra.
- **Continuum**
  The characteristic spectrum of an object with no absorption or emission features. For some objects this spectrum will approximate closely to a black-body spectrum, at least over a short range of wavelength.
- **Cosmic-ray hit**
  Extra signal present in CCD images due to the incidence of a cosmic ray on the detector during an integration. Cosmic-ray hits appear as bright spots, usually occupying only a few pixels on the detector. (Unless the ray is travelling nearly parallel to the surface of the detector in which case a

streak may be produced.) In spectroscopy cosmic-ray identification is a particular problem as real features in a spectrum can similarly occupy only a few pixels in the image.

The most effective method of cosmic-ray detection is to take two or more exposures of the same spectrum in the same instrument configuration and compare or take a median of the images.

- **Cross-dispersion**
  The direction perpendicular to that in which a spectrum is dispersed. In an échelle spectrograph a *cross-dispersing* optical element is used to separate orders in the direction perpendicular to the dispersion.
- **CCD**
  Charge-Coupled Device. For astronomy, the most commonly used optical imaging sensor.
- **CCDPACK**
  A Starlink package for the preparation of CCD data for reduction. Includes tools for managing the processing of large numbers of images. Described in SUN/139.
- **CONVERT**
  A Starlink utility package for converting between different image formats. Described in SUN/55[7].
- **Dark current**
  Electrons released in a detector (often a CCD) by the action of the thermal energy of the body of the detector.
- **Dark Frame**
  An exposure taken with the shutter closed. Typically, the exposure time used is similar to that selected for the object frames in an observing run. Dark frames give an estimate of the background level due to dark current in a CCD.
- **Dead column**
  Sometimes the interface between the vertical (parallel) and horizontal (serial) registers of a CCD is defective. As a result, the transfer of charge between the two registers does not work correctly. This kind of defect manifests its self as a column of pixels in the output image which are either all 'zeros' or all saturated, or a very high value. A dead column is not useable for imaging.
- **Dekker**
  A fork-shaped part of the slit assembly of a spectrograph which sets the length of the slit. This limits the size of the light beam in the direction perpendicular to the spectrograph dispersion.
- **Dispersion**
  A measure of the 'power' of a spectrograph. A dimensionless number, typically given in Å mm^{-1}. This number arises by dividing the true length of a section of an order in the output image (in the dispersion direction) by the wavelength range covered.

  Also the act of splitting light into its components by wavelength.
- **DIPSO**
  A self-styled 'friendly spectral analysis program' in widespread use in the community. Described in SUN/50.
- **DST**
  A data format used by some versions of FIGARO. The CONVERT utility provides facilities for translating DST format to and from NDF.
- **Echelle**
  Literally, from the French, *Ladder*. A grating in which the lines are ruled much further apart than those of an ordinary diffraction grating. This gives the échelle a very high resolution over a short wavelength range when the high orders are used.
- **Echellogram**
  Image of the spectral orders produced by an échelle spectrograph.

- **ESO**
  European Southern Observatory.
- **FIGARO**
  A general astronomical data reduction package. Available in several flavours. The Starlink version is described in SUN/86.
- **FITS**
  Flexible Image Transport System. The most commonly used format format for astronomical image data storage.
- **Flat field, flat fielding**
  A **flat field** is one illuminated with some uniform source. Used to determine the relative sensitivity of the elements (pixels) in a system.

  Flat fielding is the process of dividing by a normalised flat-field to remove the sensitivity variations of a system.
- **Free Spectral Range (FSR)**
  In a single-order instrument: the wavelength range covered by the instrument.

  In an échelle instrument: the part of an order spectrum which 'belongs' to that order, *i.e.*, the wavelength range over which this order is the brightest of the orders in the échellogram.
- **Gain, CCD output**
  The output amplifier of a CCD converts the stored signal, which is in the form of a small electronic charge, into a voltage which can then be sampled and digitised. The result is a number stored in computer memory which represents the signal recorded for a particular pixel. The conversion factor to translate this number into the number of photons recorded (actually, the number of electrons) is often called the **gain** or **output transfer function** of the camera. The units are usually electrons per ADU.
- **Grating, diffraction grating**
  Optical element ruled with (usually) thousands of fine parallel lines which produce interference patterns when light is incident upon them. Can be used as the main dispersing element in a spectrograph.

  The equation $m\lambda = d \sin\theta$ describes the diffraction pattern produced by the grating. Where: m is the order number, λ is a selected wavelength, d is the rule spacing, and θ is the angle of incidence of light.
- **GHRS**
  Goddard High-Resolution Spectrograph. An instrument on the Hubble Space Telescope.
- **Halation**
  A term originally used in photography to denote the process by which the image in a developed emulsion is spread beyond the bounds of the incident light. Is used to describe the spreading of light from one order to the next in an échelle spectrogram. Sometimes used to describe the spreading of light from the object channel into the background channel.
- **HDS**
  Hierarchical Data System. See NDF.
- **Hot-spot**
  Some pixels in the main image area of a CCD may be defective in manufacture. Such defects can manifest themselves as bright single- or few-pixel areas in an image from a CCD. These can appear similar to cosmic-ray defects, however, their position remains constant from exposure to exposure.
- **HST**
  Hubble Space Telescope.
- **IDS**
  Intermediate Dispersion Spectrograph. An instrument at the ING.

- **IHAP**
  An image format used by MIDAS. This format is available for input to MIDAS for backward-compatibility with some of the data acquisition systems at the La Silla Observatory.
- **ING**
  The Isaac Newton Group of telescopes at the La Palma Observatory.
- **INT**
  Isaac Newton Telescope at the La Palma Observatory.
- **IPCS**
  Image Photon Counting System. A common optical image sensor, has zero readout noise and good blue response.
- **IRAF**
  Image Reduction and Analysis Facility. A software package applicable to many areas of astronomical data reduction.
- **ISIS**
  A twin spectrograph at the WHT. The two 'arms' are optimised for response in the red and blue regions of the optical waveband.
- **IUE**
  International Ultraviolet Explorer.
- **JKT**
  Jacobus Kapteyn Telescope at the La Palma Observatory.
- **KAPPA**
  The Starlink Kernel Application Package. A suite of facilities for processing and viewing astronomical images. Described in SUN/95.
- **MIDAS**
  Munich Image Data Analysis System. A complete package for the handling of astronomical data. It is written and maintained by a team at ESO.
- **NDF**
  The Standard Starlink data storage format. An hierarchical format for multi-dimensional data storage. Accessed using libraries supported by Starlink. Use of NDF is described in Starlink Document SUN/33[22].
- **NOAO**
  National Optical Astronomical Observatories.
- **Order separation**
  The gap between adjacent orders in an échelle image. There is a compromise between the spectral range covered and the distance between orders. (If the orders are close together more fit on the detector and so a larger spectral range is covered.) When working with non-starlike objects a larger order separation is desirable otherwise the signal from adjacent orders may overlap.
- **Overscan, overscan region**
  The action of clocking a raster sensor (*e.g.*, CCD) for more cycles than the number of signal collection sites in the detector line. This leads to additional 'empty' pixels in the row as read out from the detector. On an image display this will appear as a band along the edge of the image, the **overscan region.** Used to determine the zero-point of the analogue circuit of the camera, *i.e.*, for no signal input to the system from the detector.
- **Periscope(s)**
  Optical arrangement which feeds light (usually from the sky background) into the slit of a spectrograph. These can be used when the object being observed would otherwise fill the slit and so no sky signal would be recorded.

- **Prism**
  Usually, a wedge-shaped optical element which disperses light passing through it. The name arises from the Greek *prisma prismatos*, 'thing sawn' (well that's what it says in the dictionary anyway…)
- **Quantum Efficiency, QE**
  The ratio of the number of photoelectrons produced to the number of photons incident upon a detector. CCDs have QEs of about 50% or greater at optical wavelengths.
- **RAL**
  Rutherford Appleton Laboratory. The Starlink project is run from RAL.
- **Readout noise**
  In this context, usually means the signal measured for no input signal for a detector such as a CCD.
- **Resolution**
  The difference in wavelength between two (notional) features which can be just distinguished in the spectrum.
- **Resolving power**
  The value $\lambda/\Delta\lambda$ where λ is the wavelength at some point in a spectrum and $\Delta\lambda$ is the resolution at that wavelength.
- **Scan, scanning**
  Process of determining the approximate position of orders in a spectral image. In the case of échelle spectra this allows you to select which orders you wish to extract.
- **Scrunch, scrunching**
  The process of correcting a raw 2-D spectral image for curvature along the slit length and calibrating the wavelength axis.
- **Slit**
  Usually narrow entry point for light to a spectrograph. The slit is often made from a pair of ordinary razor blades which can be machined to achieve very straight edges. This gives a precisely determined light source for the instrument.
- **Spectrograph**
  An instrument for separating and recording the spectral components of light. Contemporary instruments use electronic cameras to record the spectra.
- **Starlink**
  UK national network of computers for astronomical data reduction and the organisation which manages the network.
- **Stray light**
  Light which arises within an instrument due to reflections from surfaces not intended to act as optical elements.
- **SDF**
  Starlink Data File. Usually, a file with the extension `.sdf` is accessible via Starlink software and/or libraries. Most `.sdf` files you encounter will be in NDF format and so easily readable. An NDF is constructed using the *Hierarchical Data System* (HDS) which is described in SUN/92[21]. Non-NDF, HDS files can also be stored in files with the `.sdf` extension.
- **STSDAS**
  Space Telescope Science Data Analysis System. A package written for HST data reduction, closely integrated with IRAF.
- **Template, order**
  A description of the position of spectral orders in an image as determined by tracing the orders. The traced orders in one image being used to predict the position of the orders in a second image taken with the same instrumental configuration.

- **Template, reduction**
  A set of commands and/or parameter values which are appropriate for a general type of data reduction operation. Usually in the form of a *data reduction script* which can be tailored quickly for a particular reduction task.
- **Throughput**
  A measure of the overall efficiency of an optical system. For optical telescope/spectrograph combinations this will be of the order of a few to tens of percent.
- **Tracing**
  The process of finding the path of a spectrum or order of a spectrum across an image frame.
- **TWODSPEC**
  A 2-D spectral data reduction and analysis package. It is described in SUN/16.
- **UCLES**
  University College London Echelle Spectrograph. A medium-resolution instrument in the coudé room at the AAT.
- **UES**
  Utrecht Echelle Spectrograph. Northern hemisphere 'twin' of the UCLES at the WHT, has a different control system but similar optical design.
- **UHRF**
  Ultra-High Resolution Facility of the UCLES. An (up to) diffraction-limited resolution spectrograph for the AAT. Uses some of the optics of the UCLES.
- **VICAR**
  Literally *Video Image Communication and Retrieval.* A format used for some images notably those for most data from the IUE satellite.
- **VLT**
  Very Large Telescope. Usually refers to the ESO VLT, but can also refer to very-large telescopes in the general sense.
- **Wavelength scale**
  A spectrum extracted using some software package will consist of a series of samples of the spectral intensity along the dispersion direction. Often the samples are related to the arrangement of the pixels in the detector used. Each sample covers some small range of wavelength in the spectrum.

  A **wavelength scale** which allows us to calculate the approximate central wavelength for each sample can be generated by fitting curves to the observed positions of spectral features (of known wavelength) in a reference spectrum.
- **WHT**
  William Herschel Telescope. 4.2-m telescope at the La Palma Observatory.
- **Zero subtraction**
  Process of the removal of the instrument zero-signal level as determined by measuring the signal in the overscan region of a CCD image.

References

[1] L. Achmad and L. Pasquini,
CASPEC Thorium-Argon Atlas in the 3050–3650Å Region,
ESO Document ASD-MAN-SCI-000-001, May 1995.

[2] M. J. Bly & P. M. Allen,
XDISPLAY Setting Remote Xwindows,
Starlink User Note 129.4, March 1996. 5.2

[3] S. D'Odorico, M. Ghigo and D. Ponz,
An Atlas of the Thorium-Argon Spectrum for the ESO Echelle Spectrograph in the λλ 3400–9000Å Region,
ESO, 1987. 4.7

[4] Michael Bessell and Max Pettini,
UCLES Spectrum of the Thorium-Argon Hollow-Cathode Lamp,
AAO UM 28.1, January 1991. 4.7

[5] Martin Clayton,
Introduction to Echelle Spectroscopy,
Starlink Guide 9.2, October 1996.

[6] Martin Clayton,
Echelle Data Reduction Cookbook,
Starlink Cookbook 3.1, August 1996.

[7] Malcolm J. Currie, G. J. Privett & A. J. Chipperfield,
CONVERT, A Format-conversion Package,
Starlink User Note 55.6, December 1995. 9

[8] Malcolm J. Currie,
KAPPA—Kernel Application Package,
Starlink User Note 95.8, August 1992. 5.1, 7.1

[9] Peter W. Draper,
CCDPACK—CCD Data Reduction Package,
Starlink User Note 139.4, November 1995. 1.2

[10] Keith Horne,
An Optimal Extraction Algorithm for CCD Spectroscopy,
pp. 609–617, Pub. A. S. P. Vol. 98, June 1986. 4.6

[11] I. D. Howarth, J. Murray, D. Mills & D. S. Berry,
DIPSO (V3.3)—A Friendly Spectrum Analysis Program,
Starlink User Note 50.19, January 1996. (document), 5.11, 7.8

[12] Image Processing Group, ESO,
MIDAS Users Guide,
Volumes B and C, November 1994.

[13] Tom Marsh,
The Extraction of Highly Distorted Spectra,
pp. 1032–1037, Pub. A. S. P. Vol. 101, November 1989. 4.6

[14] Philip Massey,
A User's Guide to CCD Reductions with IRAF,
Central Computer Services, NOAO, June 1992. 3

[15] Ian S. McLean,
Electronic and Computer-Aided Astronomy,
Ellis Horwood Ltd., 1989.

[16] Dave Mills, John Webb and Martin Clayton,
ECHOMOP—Echelle Data Reduction Package,
Starlink User Note 152.2, December 1995. 1.2, 5.1

[17] J. G. Robertson,
Optimal Extraction of Single-Object Spectra from Observations with Two-Dimensional Detectors,
pp. 1220–1231, Pub. A. S. P. Vol. 98, November 1986.

[18] Keith Shortridge, Horst Meyerdierks, Malcolm Currie and Martin Clayton,
FIGARO—A General Data Reduction System,
Starlink User Note 86.12, October 1996. 1.2, 7.1, 8.11

[19] Francisco Valdes,
Guide to the Slit Spectra Reduction Task DOSLIT,
Central Computer Services, NOAO, April 1992.

[20] Francisco Valdes,
IRAF Tutorials: Extraction of Stellar Long Slit Spectra using DOSLIT,
`http://www.starlink.ac.uk/iraf/web/tutorials/doslit/doslit.html`,
April 1994.

[21] R. F. Warren-Smith & M. .D. Lawden,
HDS, Hierarchical Data System,
Starlink User Note 92.10, July 1995. 9

[22] R. F. Warren-Smith,
NDF, Routines for Accessing the Extensible N-Dimensional Data Format,
Starlink User Note 33.4, March 1995. 9

[23] T. N. Wilkins, D. J. Axon, J. W. Palmer & A. J. Holloway,
TWODPSEC, Some Additions to Figaro,
Starlink User Note 16.5, December 1997. 7.1, 7.8