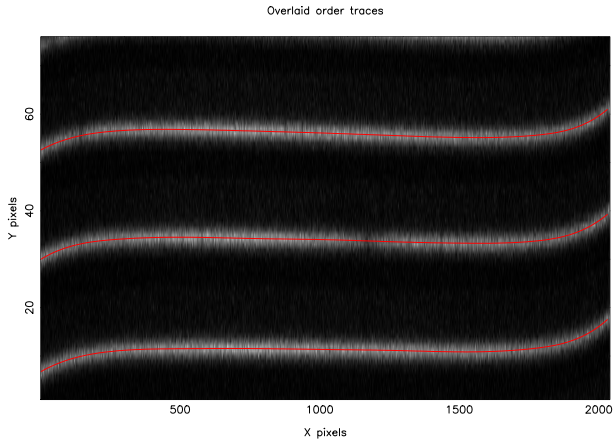
SUN/152.4

Starlink Project
Starlink User Note 152.4

Dave Mills, John Webb & Martin Clayton

2003 July 14

ECHOMOP—Echelle Data Reduction Package
Version 3.3-7
Users' Manual



Abstract

ECHOMOP provides facilities for the extraction of spectra from 2-D data frames. These data can be single-order spectra, or multi-order échelle spectra.

A substantial degree of automation is provided, particularly in the traditionally manual functions for cosmic-ray detection and wavelength calibration; manual overrides are available.

Features include; robust and flexible order tracing, optimal extraction, support for variance arrays, and 2-D distortion fitting and extraction.

Contents

1 INTRODUCTION

ECHOMOP provides a set of tasks to expedite the reduction of échelle spectra data frames. The options available range from full-scale automated reduction to step-by-step, order-by-order manually-assisted processing. The tasks were written originally for the reduction of data from the University College London Echelle Spectrograph (UCLES); however, the algorithms are sufficiently flexible to permit reduction of spectra from a wide variety of sources.

1.1 Summary of Facilities

Below is a brief summary of what can be done with ECHOMOP. It should help you to decide whether ECHOMOP can help you with your data reduction.

- Spectral order location.
- Cosmic-ray removal.
- Detection of bad image rows and columns, and saturated pixels.
- Spectral order tracing.
- Determination of object channels.
- Generation of flat-field balance models.
- Modelling of scattered light.
- Optimal spectrum extraction.
- Echelle blaze correction.
- Quick-look spectral extraction.
- Automated location of lines in arc spectra.
- Wavelength calibration of arc spectra.
- Distortion fitting for spectral orders.
- Extraction of distorted orders.
- Scrunching of extracted spectral orders.
- Production of distortion-free image.
- Plotting of data used for a reduction.
- Output of products.

All ECHOMOP facilities can be accessed from a single task, ECHMENU, which guides you through a reduction.

1.2 How to Use this Document

There is an *Introduction to Echelle Spectroscopy* (SG/9) which is a good starting point for those new to échelle data reduction.

If you are new to ECHOMOP, you should refer to the demonstration described in the next sub-section and become familiar with the basic steps used to perform a reduction in Standard Steps. It is a good idea to read the Notes for Observers prior to observing.

More experienced users will want to refer to the detailed task and parameter information under Tasks and Parameters.

The remainder of this document contains the following information:

Using ECHOMOP describes the basic use of the main ECHOMOP task ECHMENU and the single-option tasks available from the shell.

Tasks & Options gives more detailed information on each task with lists of all the parameters used by each task.

Parameters describes all ECHOMOP parameters in full.

Automatic Reductions and Cloning briefly outline the important considerations for reducing multiple datasets automatically using a 'template' reduction performed manually.

The Tips section provides advice on handling particular types of data and specific processing techniques.

The sections on Input Data and Output Data describe ECHOMOP input and output data formats.

ECHARC describes the inter-operability of ECHOMOP with the FIGARO ECHARC program.

The Notes for Observers gives advice on how to go about preparing for an échelle observing run and what needs to be done with your data prior to using ECHOMOP for the spectral extraction.

The Hints are based on information in the on-line HELP text. Each sub-section describes the cause of an ECHOMOP problem and offers possible solutions.

The Release Notes are the change log for ECHOMOP. Information about the current release is included.

1.3 Demonstration

You can obtain an on-line introduction to the use of ECHOMOP, after you have setup ECHOMOP as described in section 2.1, by typing the command

```
% ech_demo -graphics-device-
```

where `-graphics-device-` is available for plotting the graphs during the reduction (*e.g.,* `xw`). This runs a demonstration reduction using a set of example data frames. It demonstrates the main functions of ECHOMOP and follows a simple reduction from the start to the production of an extracted, wavelength-calibrated spectrum. The text of the demonstration is also available for printing in the file

```
$ECHOMOP_DEMO/ech_demo.txt
```

for those who prefer to read from a paper copy whilst running the demo.

2 USING ECHOMOP

ECHOMOP may be used in a variety of ways depending upon your expertise.

The most convenient method for normal use is to start the main ECHMENU task. This task provides a guided path through the processes of astandard reduction.

2.1 Getting Started

ECHOMOP is prepared for use from the shell by typing the command:

```
% echomop
```

To start a reduction type:

```
% echmenu
```

at the command-line prompt.

ECHOMOP uses a disc-based 'reduction database' to store objects which may need to be passed between various reduction tasks. You will initially be asked to provide a name for this file, which will then be created in the current directory. This file is associated with a particular set of data frames (object, arc, flat *etc.*) thereby providing the ability to replay any aspect of the reduction at a later date. Reduction files can occupy a large amount of disk space and you may wish to delete them once the reduced spectrum has been obtained.

When running, ECHMENU will provide a default for the next task to be performed; this will usually be the correct one for a standard reduction. To accept the recommended next step, it is only necessary to press one of the accept characters followed by a carriage return. Recognised accept characters are 'Y', '+', and '/'.

2.2 ECHMENU

The simplest method of using ECHOMOP is via the task ECHMENU which has access to all the processing modules, along with a set of utilities for interactive tuning of parameters, and plotting of graphs of intermediate results. As a reduction progresses, the details and results of each task are stored in a disk file allowing the repetition or alteration of any aspect of the reduction at any stage. ECHMENU prompts for the name of this 'reduction database' file when it starts. If the name of an existing reduction-database is given then this file will be opened, otherwise the package will create a new file using the name provided. The name of the file may also be provided on the command line, *e.g.*:

```
% echmenu ech_rdctn=filename
```

Once the reduction-database has been opened, the main menu is presented. This menu consists of all the major steps in the reduction process, along with a set of utility options, and some more specialised options (*e.g.*, 2-D distortion correction). An option is selected by typing its number followed by carriage return. Some options may also be selected by entering a keyword (or its first character) followed by carriage return, *e.g.*,

```
PLOT
```

or

```
P
```

would select the graph plotting utility. Some options consist of sets of different tasks which usually need to be run together. It is possible to run these sub-options individually by typing the number and sub-option number in the form main.sub-option (*e.g.*, 1.2). ECHOMOP attempts to provide the most sensible option (for a standard reduction) as the default, and you can accept this simply by typing one of the accept characters (*e.g.*, 'Y'), plus a carriage return. Option 0 always provides HELP information on the default option, and then allows you to browse the HELP library for more detailed information.

Most options invoke processing modules. These modules will usually require the specification of a number of parameters which will be automatically prompted for by the program. It is possible to preview/edit all the parameters (including any hidden ones) for any option by entering its number prefixed by a minus sign. *e.g.*, -3 would allow preview/edit of the parameters used by option 3.

> **NOTE:** *In general, once a parameter has been provided it will NOT be re-prompted for by any module which requires it. The modules will simply re-use the value already supplied. The -option strategy overrides this behaviour and allows you to change any parameter value at any point.*

Any parameter can be provided on the startup command-line, in this case the parameter will not be prompted for.

2.3 Graphic Display

Graphics display is controlled by the parameter:

```
soft=/device/
```

The FIGARO SOFT command

```
% soft options
```

may be used to get a list of available graphics devices.

ECHOMOP also supports both `SOFT=NONE` and `SOFT=NULL` which disable all interactive graphics facilities and save CPU time when such facilities are not required.

Hardcopy graphics is directed to `/device/` using the parameter:

```
hard=/device/
```

and enabled (in preference to on-screen graphics) using the hidden parameter `HARDCOPY` when starting ECHOMOP tasks, *e.g.*:

```
% echmenu hardcopy=y /any-other-parameters/
```

2.4 Image Display

Before proceeding with a reduction it is recommended that the data frame be examined on an imaging display to ensure that the orders are orientated as ECHOMOP expects (See Orientation of Spectra).

If an imaging graphics display is available then you can use the DISPLAY command-line parameter when invoking ECHOMOP tasks, *e.g*:

```
% echmenu display=yes /other-parameters/
```

The image display will automatically be used to:

- over-plot order trace paths in Options 2 and 15.
- over-plot extracted object limits in Option 19.
- determine positions for 'browse' option in the plotter.

2.5 Parameters and Prompts

Most of the user-input information required by ECHOMOP is obtained using the ADAM parameter system. This standard method for getting values leads to a consistency of user-interface between all ADAM-based tasks. However, in some cases it is more convenient to present the user with a menu and this approach is used for the top-level selection of processing steps, and also during interactive graphic steps. When an ADAM parameter is prompted for, the default value (used if the response is carriage-return only) is indicated enclosed in single quotes and back slashes (`/' '/`). Instead of entering a value you may also use the following special responses which are processed by the ADAM parameter system:

`?` — provides information on parameter type, minimum and maximum allowed values *etc.*

`??` — provides same information as above and then allows you to browse the HELP library.

`!!` — requests an abort of the task.

`\` — causes all subsequent parameters to adopt their default values **without** further prompts. (May also be appended to a value response, *e.g.*, `10\` would set the current parameter to 10 and switch off any further prompting).

2.6 Tuning Parameters

ECHOMOP tasks use a number of 'hidden' parameters. These will not normally need to be changed, but provide flexibility when processing problem data. They are also used by ECHMENU to configure itself for 'quick-look' extraction. ECHMENU Option 23 allows you to interactively examine and alter these parameters. Parameter values may also be supplied on task command-lines. ECHOMOP tasks always report any `TUNE_` parameters which have been set to non-default values.

2.7 Processing Steps

The reduction cycle has been split into a set of steps, most of which perform some form of image analysis. The results of these analysis steps provide various function fits and data describing the features of the image. All these steps take place before any extraction of data from the raw image. The extraction is performed in a single step taking into account the previously established data characteristics. This provides for the efficient processing of multiple data frames where the same object has been observed, as many of the processing steps need only be done once, for the first frame of the series.

All the discrete processing steps are available as both individual tasks (providing efficient execution of single steps), and as options from a menu-driven control task (providing automatic processing and context-sensitive assistance).

To run a particular step you can:

- select the appropriate option from the main ECHMENU menu.
- type the task name at the command line, *e.g.*:

```
% ech_linloc
```

NOTE: *that when using the individual tasks the strategy for selecting single order/all order operation is different. Individual tasks which can operate on single orders all utilise the parameter:*

```
IDX_NUM_ORDERS
```

with which you should specify the number of the order to process, or a zero to indicate that all orders are to be processed in turn.

2.8 Repeating Steps

In many cases the automatic sequence of steps provided by the ECHMENU main menu defaults will be enough to complete the reduction. In some cases however, it may be necessary to restart the reduction at a particular stage, maybe using a different raw image, different degree of polynomial *etc*.

ECHMENU allows the selection of any valid step of a reduction at any stage. If the step requested needs results from steps which have not yet been performed then you will be informed of this and the default next step will be set back to the step which generates the required information. If a single step task is invoked before all its input requirements are available (*e.g.*, order tracing before order location) then the task will abort with a message indicating which task needs to be run in order to provide the necessary pre-requisites.

2.9 Standard Steps

The following steps constitute a standard reduction as implemented by the monolithic ECHMENU task. You can follow this sequence by hitting Y and carriage-return at each main menu prompt, and supplying data frame names and parameters values when prompted.

- 1 — Start a reduction (`ech_locate`).
  - 1.1 — Check frame dimensions (`ech_fcheck`).
  - 1.2 — Check trace frame for cosmic-ray hits/bad rows and columns (`ech_decos1`).
  - 1.3 — Determine order slope (`ech_slope`).
  - 1.4 — Count orders (`ech_count`).
  - 1.5 — Locate orders (`ech_locate`).
- 2 — Trace the orders and fit polynomials (`ech_trace`).
- 3 — Clip trace polynomial fits (`ech_fitord`).
- 4 — Determine dekker limits and sky/object channels (`ech_spatial`).
  - 4.1 — Determine dekker limits (`ech_dekker`).
  - 4.2 — Determine object and sky channels (`ech_object`).
- 5 — Model flat-field per-pixel balance factors (`ech_ffield`).
- 6 - Model sky background (`ech_sky`).
- 7 — Model object profile (`ech_profile`).
- 8 — Extract object and arc spectra (`ech_extrct`).
- 9 — Locate arc lines and fit Gaussian profiles (`ech_linloc`).
  - 9.1 — Determine average arc line FWHM (`ech_fwhm`).
  - 9.2 — Locate arc line candidates (`ech_lines`).
- 10 — Identify arc lines and fit wavelength scales (`ech_idwave`).
- 11 — Model blaze profile and apply to object spectrum (`ech_blaze`).
  - 11.1 — Model order ripple (`ech_fitblz`).
  - 11.2 — Correct order ripple (`ech_doblz`).
- 12 — Scrunch object spectrum (`ech_scrunch`).
  - 12.1 — Fit arc-line FWHMs as a function of wavelength (`ech_fitfwhm`).
  - 12.2 — Calculate wavelength scale (`ech_wscale`).
  - 12.3 — Scrunch extracted object orders (`ech_scrobj`).
  - 12.4 — Scrunch extracted arc orders (`ech_scrarc`).
- 14 — Write results file (`ech_result`).

The demonstration provided by `ech_demo` follows this overall sequence and the reader is encouraged to try it before using ECHOMOP in earnest. Steps 9 through 12 would be omitted if no wavelength calibration was required. Step 5 would be omitted if no flat field was available. In cases where multiple object frames have been taken, it would be efficient to use only steps 6, 7, 8, 11, 12 for additional frames. A quick-look extraction may be performed as soon as step 4 has been completed (using Option 19).

2.10 Accessing Single Function Sub-options

Many of the ECHMENU main menu tasks consist of a number of independent functions which are automatically invoked. For example Option 12 consists of 4 sub-options. The ECHMENU program allows the selection of any sub-option directly, *e.g.*:

```
12.3
```

would select Option 12.3 (Scrunch extracted object order) explicitly.

To check if sub-options are available, the context specific HELP for the option should be viewed (type 0 at the main menu prompt). Some sub-options are individually accessible when using the single purpose tasks (*e.g.*, `ech_scrarc`).

2.11 Automation

ECHOMOP provides automation to many aspects of échelle data reduction which have traditionally required substantial interactive assistance:

- Order location
- Cosmic-ray pixel identification
- Object/Sky pixel classification
- Arc line identification

The provision of these features permits the completely automatic reduction of échelle data frames. To help ensure the success of automatic reduction runs, special attention should be made to ensure that the following conditions are satisfied:

- A bright continuous spectrum image is provided for order tracing.
- The trace/arc/object images supplied are the correct ones. For instance supplying the wrong arc image will cause problems if the orders are in different places on the detector.
- Constraints placed on the expected range of wavelength and dispersion are wide enough. *e.g.*, if in doubt about the dispersion being 0.2 or 0.3 Angstroms per-pixel, provide limits of `MIN_DISPERSION=0.1`, `MAX_DISPERSION=0.5`.

2.12 Quick-look Mode

Invoking ECHMENU with the `TUNE_QUICK=Y` will cause the automatic setting of tuning parameters to values designed for fastest processing.

The following algorithmic optimizations will be used:

- order tracing will use centre-of-gravity to locate the centre of the order.
- a maximum of 200 trace sample points per order will be measured.

- criteria for clipping points from the trace fit will be set at 1-pixel maximum deviation, and up to 5 pixels may be clipped per iteration.
- extraction will use the simple version of the 1-D algorithm which is faster than optimal/weighted extraction.
- arc-line identification will assume the data is arranged with wavelength increasing with X-pixel number.

2.13 Error Arrays

ECHOMOP can deal with errors from the following variety of sources:

- Errors on the flat-field frame which should be provided in the flat-field frame error array. If no error array is provided then estimates using root-n statistics are used.
- Errors due to the readout-noise on CCD type detectors. The readout noise value is supplied as a parameter and is applied to the whole frame when calculating the variances on extracted pixels.
- Linear error sources determined may be treated by providing an error array associated with the object data frame. The values should represent an error source that does not vary with the intensity of the particular pixel involved. An example of this type of error source would be the bias subtraction. If many bias frames are available, then the error for each detector pixel can be determined. Supplying these errors as the object error array would then cause them to be correctly included in the variance calculations during extraction.
- Multiplicative errors such as those due to errors on the balance factors (from the flat field) may be taken into account by incorporating the appropriate values in the error array supplied with the flat-field frame. It is up to you to ensure these variances correctly represent the combined effect of both the flat-field noise and any additional multiplicative error source(s).
- Errors due to sky modelling and subtraction are calculated and processed internally by the extraction task.
- Errors due to detected cosmic-ray hits are treated by ignoring the contaminated pixel and rescaling the variance on the remaining pixels in that (wavelength) increment.
- All output spectra have an error array associated with them, these error arrays always contain the variance for the parallel entry in the data arrays.

3 ECHOMOP TASKS & ECHMENU OPTIONS

The ECHMENU monolith main menu options are described in the following text. Many of these options are also available as tasks which can be accessed from the shell. Where a task is available its name is given in the section heading.

In each of the following sections a description is given of the task, it's purpose, the parameters it uses and the reduction file objects it accesses. Parameters used by the tasks are described in detail under Parameters.

0:`echhelp` **– HELP**

Provides entry into the HELP library for browsing. Upon entry the first topic displayed will depend upon the context and will usually be appropriate to the next default option at the time.

To leave the HELP type a CTRL-Z, or hit carriage-return until the ECHMENU menu re-appears.

The stand-alone HELP browser `echhelp` can be invoked from the shell or the hypertext version of the text can be accessed using

```
% echwww
```

1: `ech_locate` **– Start a Reduction**

This is usually the first option selected and will cause the following operations to be performed:

- 1.1 (`ech_fcheck`)

  This option is enabled by setting the parameter `TUNE_FCHECK=YES`, if not set no frame checking is performed.

  The following checks are made:

  - Checks frame dimensions. This simply checks that the trace and input frames have the same dimensions.
  - Checks for bad rows or columns specified in the reduction database.
  - Checks for non-specified bad rows or columns. The task checks along both data frame rows and columns detecting series of pixels of uniform intensity. These pixels are assumed to be 'bad' and are flagged as such.
  - Checks for saturated pixels (threshold level set by).
  - Checks for BAD values in the input and trace frames.

  The trace task will ignore any pixels flagged during frame checking. This can aid the tracing of frames containing saturated pixels, bad rows or columns, *etc.* which might upset the tracing process.

- 1.2 (`ech_decos1`)

  Optionally check the 'trace' frame for cosmic rays. This option is normally switched off. It is enabled by setting the parameter `TUNE_CRTRC=YES`. The approach is to apply two median filters to the data frame. One in the X-direction (along rows) and the other in the Y-direction (along columns). Both of these median images are then divided into the original and the resulting image histogrammed. You then have to select a threshold point on the displayed histogram. All pixels generating samples above the clip threshold are flagged as cosmic rays. This method does not rely on the échelle nature of the image and may be used on data frames of non-spectral type with some success.

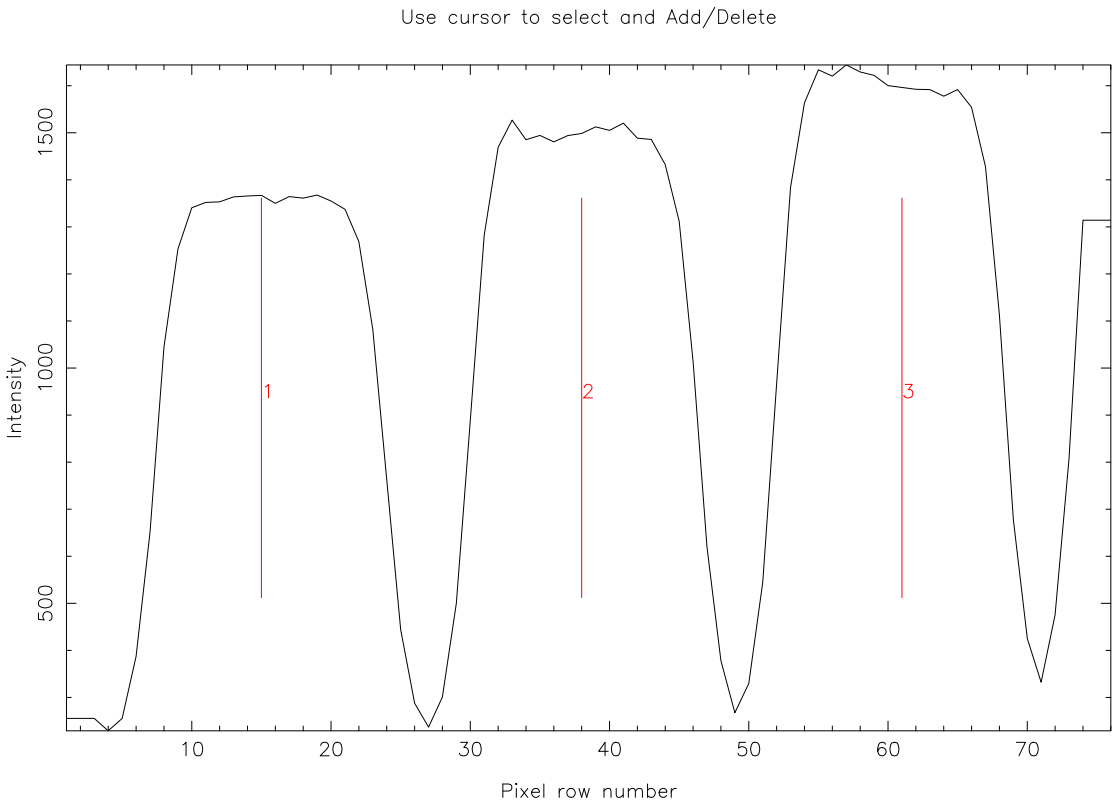


Figure 1: The graph displayed during order location. The frame used was a continuum image created with the slit jaws wide open. This is a recommended procedure for creating reliable trace frames.

- 1.3 (`ech_slope`)

  Determine approximate slope of orders across the frame. A common problem when extracting échelle orders is that they are significantly sloped with respect to the pixel rows/ columns. ECHOMOP can cope with highly sloping orders because it works by first locating a single point in each order and then using this as the start point for tracing the order path. ECHOMOP also calculates the approximate slope of the orders prior to tracing, and uses this information to predict the order position when tracing fails (usually due to contaminated pixels).

  The estimate obtained by this method is however, an average value over all orders. It is therefore not applicable when the orders are highly distorted, or when each order has a different slope.

  The slope as calculated by the program may be overridden manually using the FIGARO SETOBJ command:

  ```
  % setobj object=-rdctn-file-name-.MORE.ECHELLE.ORDER_SLOPE value=n.nnnn
  ```

- 1.4 (`ech_count`)

Count the orders and record their positions in Y at the centre (X=NX/2) of the frame. The orders can usually be located completely automatically by ECHOMOP. The algorithm takes the central few columns of a the trace frame and cross correlates them to provide an estimate of the average order separation. It then uses this estimate to size a sampling box. The central columns are then re-sampled using averaging and a triangle filter to enhance the central peaks. The resulting data is then searched to find the peaks representing the centre of each order. This technique can be severely affected by cosmic rays and other high energy contaminants in the image. It is recommended that only flat-field (or bright object) frames are used to perform order location and tracing).

In cases where the algorithm cannot locate all the orders which are apparent by visual inspection, you may help locate the orders manually. To control this the parameter `TUNE_AUTLOC=NO` should be used (default). In which case ECHOMOP will display a graph of the central portion of the frame and wait for interactive verification of, or correction to, the located orders positions. For cases where you are confident that the located orders will be correct, `TUNE_AUTLOC=YES` can be specified.

In many cases the location of the orders (at the frame center) can be completed automatically. Cosmic-ray cleaning of the trace frame can be enabled using `TUNE_CRTRC=YES`, and bad row/column checking by using . The located orders are plotted on the graphics device (specified using the `SOFT` parameter), and may be edited using the graphics cursor.

When `TUNE_AUTLOC=NO` you will be presented with the following menu options:

The Figure above shows an example of the order-location plot with 3 located orders.

- **A**(dd an order)
  Marks the current cursor position as your guess at an order centre.
- **D**(elete an order)
  Deletes the order *nearest* to the cursor position.
- **C**(lear all orders)
  Deletes all the current order selections.
- **R**(eplot the display)
  Replots the graph. The displayed orders are renumbered from left-to-right.
- **E**(xit)
  Exit the order-location editor.
- **M**(enu)
  Displays the full menu of options. Normally a one-line menu is displayed summarising options.

Parameters:

- **INPTIM** - Frame to extract data from.
- **TRACIM** - Frame for order tracing.
- **ARC** - Name(s) of reference (arc) lamp image(s).
- **TUNE_AUTLOC** - YES for automatic location of orders.

- **TUNE_CRINTER** - YES if interpolation is required.
- **TUNE_CRMAX** - Maximum number of cosmic-ray pixels.
- **TUNE_CRTRC** - YES for trace frame cosmic-ray checking.
- **TUNE_CRXBOX** - X-size of box for median calculations.
- **TUNE_CRYBOX** - Y-size of box for median calculations.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_FCHECK** - YES for frame checking enabled.
- **TUNE_MINCR** - Minimum Cosmic-Ray intensity.
- **TUNE_PARTORD** - YES if checking for partial orders.
- **TUNE_SATRTN** - Saturated pixel intensity.
- **TUNE_TWTHR** - Trace width threshold.
- **TUNE_XBOX** - Size of X-sampling box for order location.
- **USE_MEDIAN** - YES if median is to be used.

Reduction File Objects:

- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ/WRITE.
- **NREF_FRAME** - *type:* _INTEGER, *access:* READ/WRITE.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ/WRITE.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ/WRITE.
- **ORDER_SLOPE** - *type:* _REAL, *access:* READ/WRITE.
- **ORDER_YPOS** - *type:* _INTEGER, *access:* READ/WRITE.
- **TRACE_WIDTH** - *type:* _INTEGER, *access:* READ/WRITE.

2:`ech_trace` – Trace the Orders

The tracing of the paths of the orders across the data frame is often a source of difficulty as it is fairly easy for blemishes in the frame to fatally deflect order tracing algorithms from the actual path of the order. ECHOMOP provides a variety of options to help combat these problems. ECHOMOP order tracing first locates the positions of the orders at the centre of the frame, and estimates the average order slope. It uses this information to predict the existence of any partial orders at the top/bottom of the frame which may have been missed by the examination of the central columns during order location.

Tracing then proceeds outwards from the centre of each order. At each step outwards a variable size sampling box is used to gather a set of averages for the rows near the expected order centre. The centre of this data is then evaluated by one of the following methods:

- Gaussian attempts to fit a Gaussian profile across the data. Works well for bright object frames.
- Centroid calculates the centroid of the data. Most generally applicable method.
- Edge Detects the upper and lower 'edges of the data and interpolates. Is less accurate but works well for difficult flat fields (*e.g.*, saturated)
- Balance Calculates the centre of gravity (balance point). Works well for difficult data when G and C methods are having problems.
- Re-trace Uses a previous trace as a template to predict the trace whenever it cannot be centred. Will normally be used in conjunction with automatic trace consistency checking to improve poorly traced orders.
- Triangle filter In addition it is possible to apply a triangle filter during tracing to help enhance the order peaks and improve the trace. This is done by prefixing the selected trace specifier with a 'T', thus TC would use triangle-filtered centroids.

The trace algorithm will loop increasing its sampling box size automatically when it fails to find a good centre. The sample box can increase up to a size governed by the measured average order separation.

When a set of centres have been obtained for an order, a polynomial is fitted to their coordinates. The degree is selectable. For ideal data, these polynomials will represent an accurate reflection of the path of the order across the frame. For real data it is usually helpful to refine these polynomials by clipping the most deviant points, and re-fitting. Options are provided to do this automatically or manually.

When dealing with distorted data it is often necessary to use a high degree polynomial to accurately fit the order traces. This in turn can lead to problems at the edges of the frame when the order is often faint.

Typically the polynomial will 'run away' from the required path. The simplest solution is of course to re-fit with a lower order polynomial, however, this may not be satisfactory if the high degree is necessary to obtain a good fit over the rest of the order.

In these circumstances, and others where one or more orders polynomials have 'run away', ECHOMOP provides an automatic consistency checker. The consistency checking task works by fitting polynomials to order-number and Y-coordinate at a selection of positions across the frame. The predicted order centres from both sets of polynomials are then compared with each other and then mean and sigma differences calculated. The 'worst' order is then corrected by re-calculating its trace polynomial using the remaining orders (but excluding its own contribution). This process is repeated until the mean deviation between the polynomials falls below a tunable threshold value. The consistency checker will also cope with the 'bad' polynomials which can result when partial orders have been automatically fitted.

Viewing traced paths

The tracing of the échelle order paths is central to the entire extraction process and care should be taken to ensure the best traces possible. ECHOMOP provides a large number of processing alternatives to help ensure this can be done. Most of these provide information such as RMS deviations *etc.*, when run. In general however, the best way of evaluating the success or failure of the tracing process is to visually examine the paths of the trace fitted polynomials. Three methods of viewing the traced paths are provided.

- Viewing the fitted paths overlaid on an image of the trace frame as tracing is done (Set parameter `DISPLAY=YES`).
- Using a graphics device to plot the paths of all orders (Task `ech_trplt`/ECHMENU Option 15).

For a single order, a more detailed examination of the relation of a trace polynomial to the points it fits, can be obtained using the V(view) command in the task `ech_fitord`/ECHMENU Option 3.

Parameters:

- **TRACE_MODE** - Type of order tracing to use.
- **TRACIM** - Frame for order tracing.
- **TRCFIT** - Function for trace fitting.
- **TRC_NPOLY** - Number of coeffs of trace-fit function.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_IUE** - Non-zero if IUE type data frame.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MXBADSMP** - Maximum consecutive number of bad samples.
- **TUNE_MXSMP** - Maximum number of X-samples to trace.
- **TUNE_XBOX** - Size of X-sampling box for order location.
- **USE_MEDIAN** - YES if median is to be used.

Reduction File Objects:

- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **ORDER_SLOPE** - *type:* _REAL, *access:* READ.
- **ORDER_YPOS** - *type:* _INTEGER, *access:* READ.

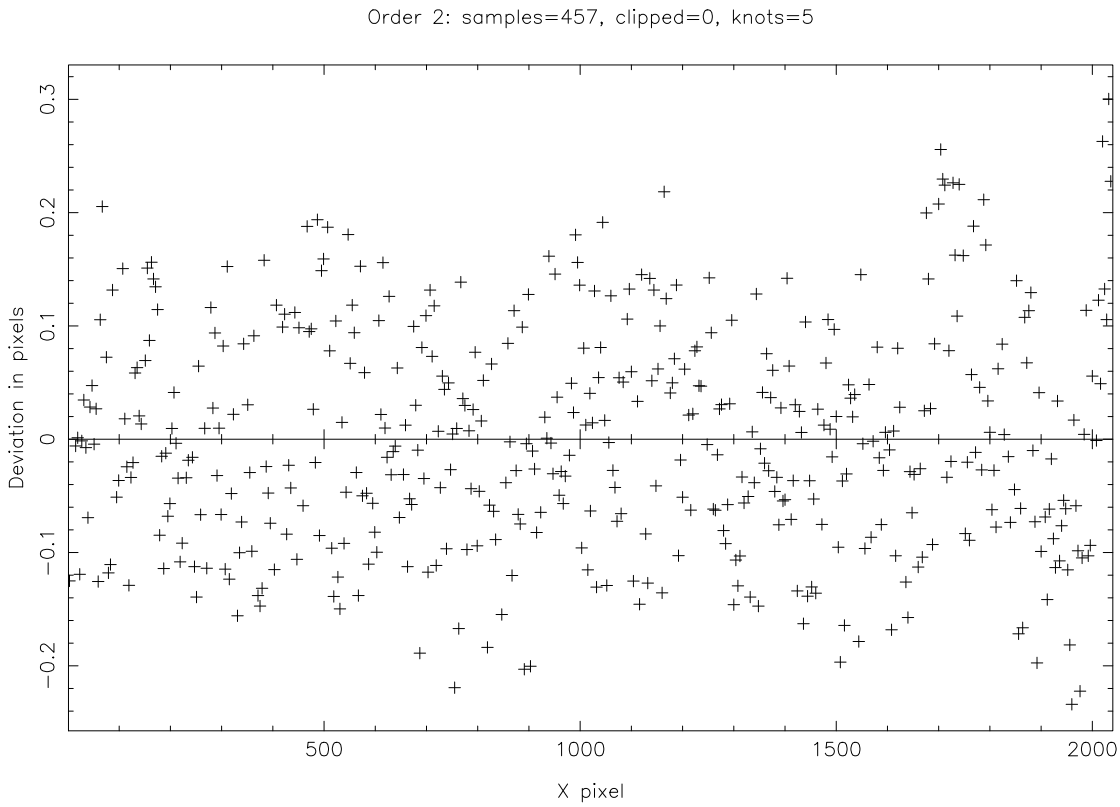


Figure 2: A typical plot of the order trace deviations as show during the trace clipping stage (Option 3). The points represent the deviation of the order from the fitted polynomial/spline at a set of sample column positions.

- **TRACE** - *type:* _REAL, *access:* READ/WRITE.
- **TRACE_PATH** - *type:* _REAL, *access:* READ/WRITE.
- **TRACE_WIDTH** - *type:* _INTEGER, *access:* READ.
- **TRC_IN_DEV** - *type:* _REAL, *access:* READ/WRITE.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ/WRITE.

3:`ech_fitord` – Clip Trace Polynomials

This routine performs automatic or manual clipping of points from the set of samples representing the path of an order across the frame. A variety of methods of manually clipping points is available, and the degree of the polynomial used may also be altered. The relationship of the fit to the trace samples, and the deviations, may be examined graphically.

Orders may be repeatedly re-fitted once they have been initially traced. In general an automatic fit, clipping to a maximum deviation of third of a pixel will yield good trace polynomials. The routine may also be used to remove sets of points which are distorting a fit, for example a run of bad centre estimates caused by a dodgy column/row.

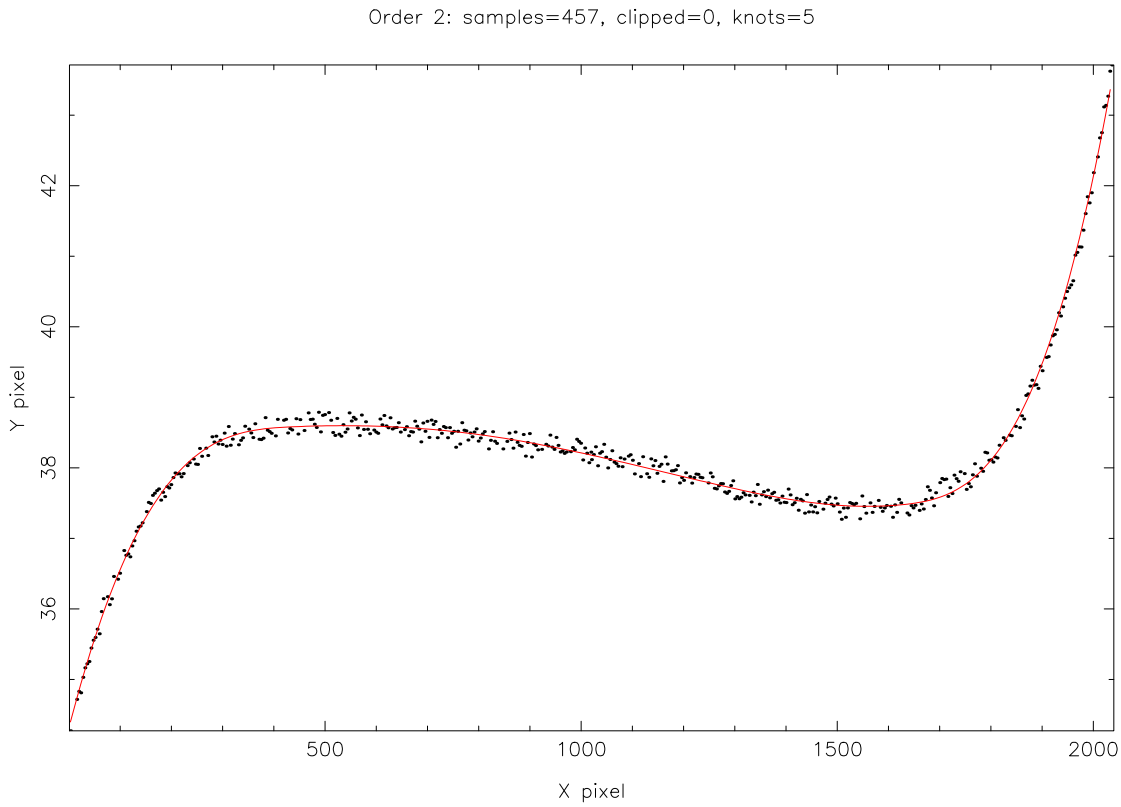


Figure 3: A Plot of the fitted order trace overlaid on the trace data. This plot was made using the option "V" in ECHMENU Option 3.

This routine should be used manually on single orders when they are clearly not correctly fitted. This will usually be seen by viewing the paths of the fitted order traces using `ech_trplt`. In many cases it will be possible to get a good trace fit by clipping a set of obviously bad samples, and re-fitting (perhaps with a lower degree of polynomial).

The other main option for coping with such problems is to rely on the trace consistency checking routine. This will produce a set of consistent trace fits in most cases. If a particular order is beyond recovery and you do not wish to extract it at all, then it may be disabled by clipping away all of its trace samples with this routine.

The figure above shows an example of the deviations plot after automatic clipping has been performed on an order.

If you set `TRC_INTERACT=YES` then a multi-option menu will be displayed. The options are:

- **.**

  Clips the sample nearest to the cursor when the key is pressed.

- **N**(egative threshold)

  Clips all samples with deviations which are **more negative** than the current Y-position of the cursor.

- **P**(ositive threshold)

  Clips all samples with deviations which are **more positive** than the current Y-position of the cursor.

- **C**(lip)

  Clips all points with absolute deviations greater than the current Y-position of the cursor.

- **R**(ange clip)

  Allows the clipping of a set of samples denoted by a range of values along the X-axis of the trace. This will generally be used when the order is only partially present, or perhaps to remove the contributions of a swath of bad columns. The starting point for the range to be clipped is the current cursor X-position, and the end point is selected by moving the cursor and hitting any key.

- **B**(ox clip)

  Like option `R` but allows you to specify the corners of a box in which all samples will be clipped. Any two corners which define the area to be clipped can be used. The first corner of the box to be clipped is the current cursor X-position; the second (opposite) corner is selected by moving the cursor and hitting any key.

- **A**(utoclip)

  Selects automatic clipping in which points will be iteratively clipped and the trace re-fitted. The number of points to be clipped is set by `TUNE_MXCLP`. If this is zero, points are clipped until the absolute deviation of the worst sample has fallen below the threshold set by `TUNE_CLPMAXDEV`.

- **G**(o)

  Used to switch from interactive clipping to entirely automatic mode. Thus after manually clipping a couple of orders it may be observed that the orders are well traced and that few points need be clipped, this can be left safely to auto-clipping. The Go option selects auto-clipping for the current, and any subsequent orders.

- **D**(isable)

  Used when it is clear that the traced samples do not follow an order at all and you wish to prevent any subsequent processing of the order. Orders may be automatically disabled if a small enough fraction of samples remain after auto-clipping has been done.

- **O**(ff)

  Toggles plotting of the fits; normally a replot occurs for every key hit.

- **F**(unction)

  Rotates the type of fit used. Currently available types are polynomial and spline.

- **V**(iew)

  The graph used for clipping shows the deviations of each sampled trace point. The View option shows the actual coordinates of the trace samples, providing an easy reference as to the agreement with the order path as expected from visual examination of the raw data. **Note** that View mode is mutually exclusive of all other operations and must be exited (type any key) before clipping can be resumed.

- **Q**(uit)

  Leaves the trace fitting for this order without saving the trace polynomial in the reduction database. It is therefore lost.

- **E**(xit)

  Leave trace fitting for the current order and save the latest trace polynomial in the reduction database.

- **!**

  Leaves the trace fitting for this order and all subsequent orders.

- **+**

  Increments the degree of polynomial used to fit the trace samples. It may be increased up to the maximum specified by `TUNE_MAXPOLY`.

- −

  Decrements the degree of polynomial used to fit trace samples.

Parameters:

- **TRCFIT** - Function for trace fitting.
- **TRC_INTERACT** - YES for interactive order-fitting.
- **TRC_NPOLY** - Number of coeffs of trace-fit function.
- **TUNE_CLPBY** - Number of points autoclip before re-fit.
- **TUNE_CLPMXDEV** - Maximum deviation from polynomial.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.

Reduction File Objects:

- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **TRACE** - *type:* _REAL, *access:* READ.
- **TRC_CLIPPED** - *type:* _INTEGER, *access:* READ/WRITE.
- **TRC_IN_DEV** - *type:* _REAL, *access:* READ/WRITE.
- **TRC_OUT_DEV** - *type:* _REAL, *access:* READ/WRITE.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ/WRITE.

4:`ech_spatial` – Determine Dekker/Object Limits

The determination of the position of the object data within the slit proceeds by first locating the slit jaws. To do this either an ARC frame or (ideally) a flat-field frame may be used. The profile is calculated along the slit and the edges are then located by determining the points where the profile intensity drops below a tunable threshold. For problem cases the dekker positions may also be indicated manually on a graph of the ARC/Flat field profile. Once the dekker limits have been determined, the object profile is measured. The object is sampled by averaging the profile over all orders (using the central columns of the frame only). The median intensity of the profile inside the dekker limits is then calculated and used to set an expected sky threshold. The profile is then examined by stepping outwards from its peak until the profile intensity falls to the expected sky threshold. Masks are then created in which each pixel along the slit is flagged as sky or object. You may also interactively edit these masks and flag particular sections of the profile as sky or object. Only pixels flagged as 'object' will contribute to an extraction. Therefore the profile editing provides a (tedious) simple mechanism for producing spatially resolved spectra (each spatial increment in turn is flagged as the only object pixel in the profile, and extracted).

The default behaviour is to average all the orders together thus generating a composite profile. In certain circumstances it may be necessary to derive a separate profile for each order (for instance for multi-fibre spectra). To select this option the parameter `TUNE_USE_NXF=1` must be set.

This option must be used before any extraction of the orders can take place. It consists of two steps:

- 4.1 (`ech_dekker`)

  Analyse the Arc/Flat-field frame to determine the positions of the edges of the dekker.

- 4.2 (`ech_object`)

  Analyse the Object frame to determine the position of the object within the slit, and its extent above and below the traced order.

In each case a plot is produced on the graphics device (specified using the `SOFT` parameter), showing the status of pixels in relation to their position relative to the path of the order across the frame.

The Figure above shows an example of the dekker plot. The regions indicated by a solid line are inside the dekker. Pixels in positions corresponding to the dot/dash line are outside the dekker and will not be used during processing.

The Figure above shows an example of the object limits plot. The regions indicated by a solid line are 'object' pixels and will contribute to the extraction. Regions shown by the dashed line are 'sky' pixels which will be used to calculate the sky model. Pixels in the dot/dash region are outside the dekker limits and will not be used during processing.

The pixels' status as set by this option, determine what part they play when the extraction takes place. That is, it determines if a pixel is part of the sky, object, outside the slit, or to be ignored completely by the extraction routine. If the parameter `PFL_INTERACT=YES` is set then you are also provided with the opportunity to edit these quantities on a profile plot.

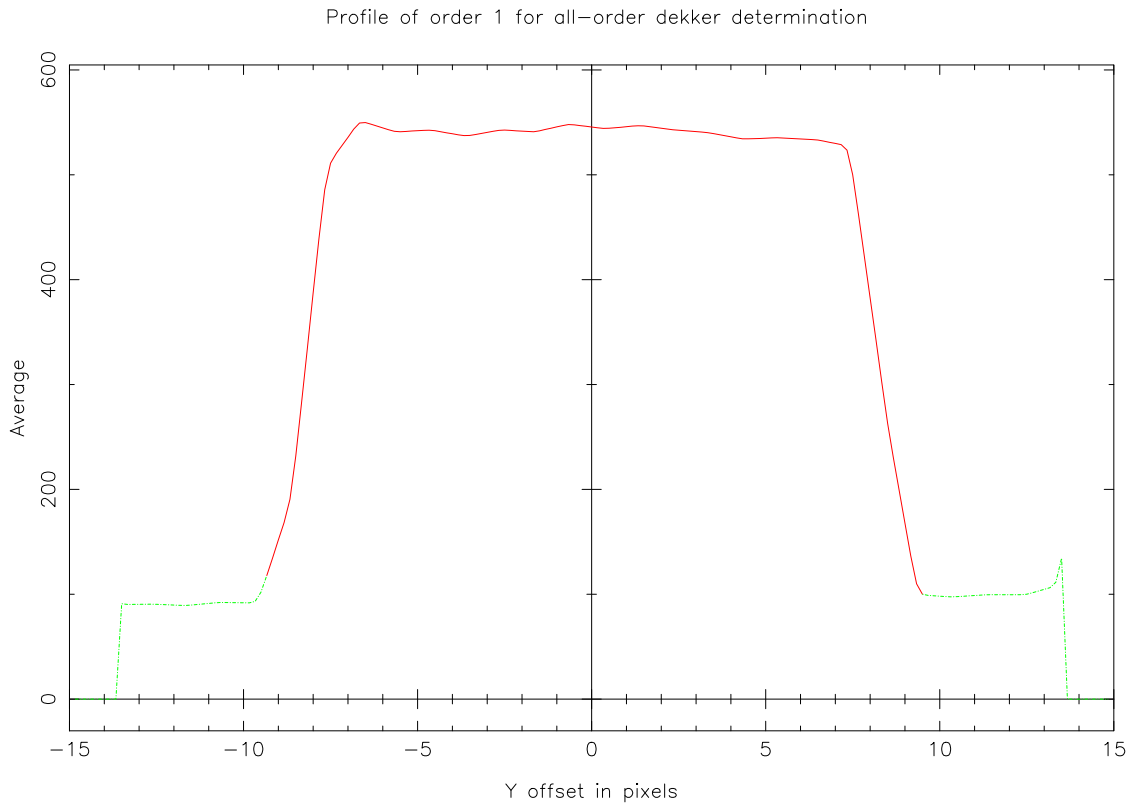


Figure 4: The plot used for dekker limit setting. In interactive use the cursor is moved to the desired X-position and then either `l` (for lower) or `u` (for upper) key is pressed to set a limit. If the scale of the graph is too small then it can be extended by pressing `l` beyond the left-hand vertical axis, or `u` beyond the right-hand vertical axis.

In addition the limits may be specified by using parameters, which will over-ride the values calculated by the modules.

- `TUNE_DEKBLW` to set the lower dekker.
- `TUNE_DEKABV` to set the upper dekker.
- `TUNE_OBJBLW` to set the lower object limit.
- `TUNE_OBJABV` to set the upper object limit.

After running Option 4, the post-trace cosmic-ray locator may be run if required (Option 17). It cannot be used before as it uses the object limits derived in Option 4.

Parameters:

- **INPTIM** - Frame to extract data from.
- **PFL_INTERACT** - YES for interactive profiling.

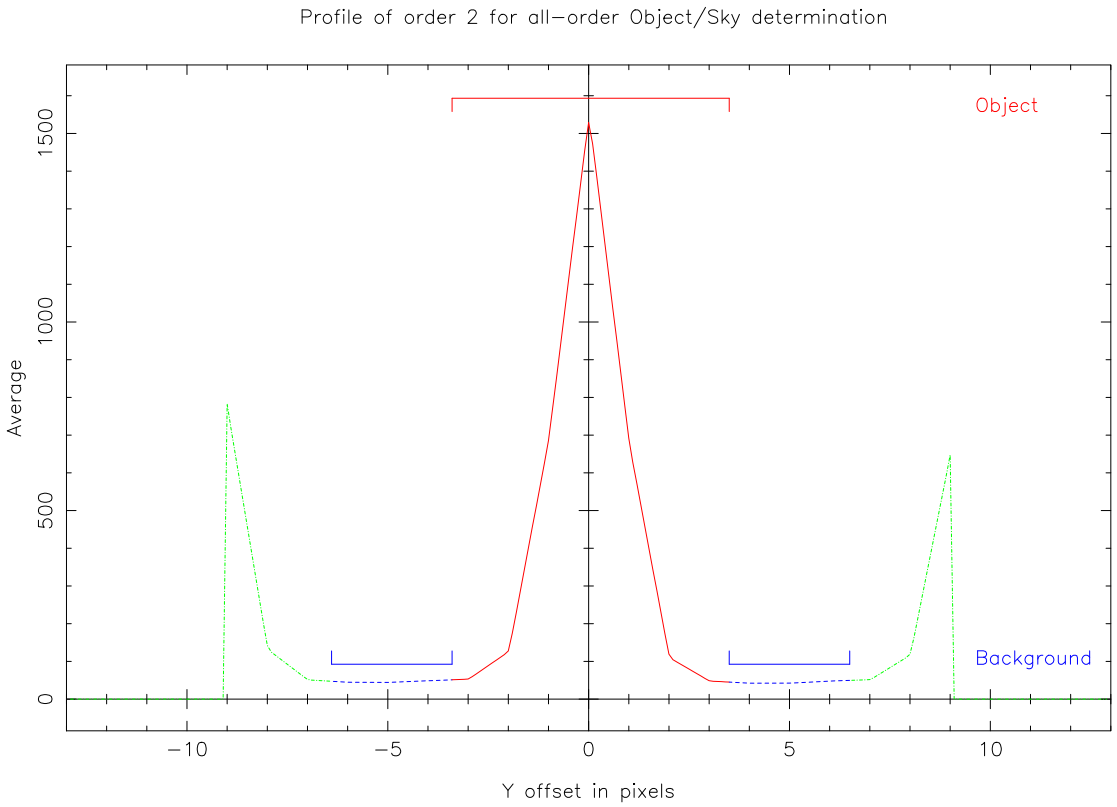


Figure 5: The type of plot used for object limit setting. The cursor is placed at the X-position of the region of interest and then one of the `o` (object), `s` (sky), `i` (ignore), `b` (both), keys pressed. Dekker limits may also be changed using the `u` and `l` keys.

- **PFL_MODE** - Profiling mode (D, S, O, A).
- **SLITIM** - Frame for dekker measurement.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_DEKABV** - Dekker extent in pixels above trace.
- **TUNE_DEKBLW** - Dekker edge in pixels below trace.
- **TUNE_DEKTHR** - Threshold for dekker location.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MXSKYPIX** - Maximum number of sky pixels.
- **TUNE_OBJABV** - Number of object pixels above trace.
- **TUNE_OBJBLW** - Number of object pixels below trace.
- **TUNE_PFLSSAMP** - Maximum number of subsamples in profile.
- **TUNE_SKYHILIM** - Upper threshold for sky intensity.

- **TUNE_USE_NXF** - Fraction of X-samples to use in profile.

Reduction File Objects:

- **CONTIN_PROFILE** - *type:* _REAL, *access:* READ/WRITE.
- **DEK_ABOVE** - *type:* _INTEGER, *access:* READ/WRITE.
- **DEK_BELOW** - *type:* _INTEGER, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **OBJECT_PROFILE** - *type:* _REAL, *access:* READ/WRITE.
- **OBJ_MASK** - *type:* _INTEGER, *access:* READ/WRITE.
- **SKY_MASK** - *type:* _INTEGER, *access:* READ/WRITE.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.

5:`ech_ffield` – Model Flat-field Balance Factors

The 'balance factors' are the per-pixel values which are multiplied into the raw data to perform the photometric correction required to correct for differing pixel-to-pixel responses of some detectors (mostly CCDs).

ECHOMOP will use a flat-field frame if one is available. The flat-field frame should be produced using a continuum lamp exposure with the instrument in an identical configuration to that used for the object exposure (to ensure that any wavelength dependent behaviour of pixel response is taken into account). The exposure should be of sufficient duration to attain high counts in the brightest parts of the image.

ECHOMOP fits functions in two directions; along the traces, and along the image columns. The degree of polynomials fitted is tunable, but the (low) default degree will normally be perfectly reasonable. Each flat-field pixel in an order is then used to calculate a 'balance' factor. This is a number close to 1 which represents the factor by which a given pixel exceeds its expected value (predicted by the polynomial).

Note that this technique requires that the flat-field orders vary slowly and smoothly both along and across each order.

If required many flat-field frames (with identical instrument configuration) may be co-added prior to ECHOMOP, and the high S/N flat field used by ECHOMOP. At present no special facilities are provided for calculating the actual error on such a co-added flat field; the expected error (derived from root-N statistics) is what is used to calculate the error on the balance factors **unless** appropriate variances are provides in the flat-field frame error array. Other modes of operation are triggered by setting the parameter `FLTFIT`. If the parameter is set to `NONE` then no modelling in the X-direction will be performed. If the parameter is set to `MEAN`, then no polynomials are used, but the balance factors are calculated using the local mean value based on a 5-pixel sample.

This will normally be used when the flat field at the dekker limits cannot be modelled because its intensity changes too rapidly on a scale of 1 pixel due to under-sampling of the profile.

The full set of modelling options is:

- `MEAN` a local mean parallel to trace.
- `MEDIAN` a local median parallel to trace.
- `POLY` a polynomial function parallel to trace.
- `SPLINE` bi-cubic spline parallel to trace.
- `SMOOTH` Gaussian smoothing along pixel rows.
- `SLOPE` local slope along pixel rows.
- `NONE` all balance factors set to unity.

If you produce your own balance-factor frame, then this may be used by ECHOMOP. The parameter `TUNE_PREBAL` should then be set to `YES`. In this case no modelling takes place and the balance factors are simply copied from the frame supplied. This should be used if the ECHOMOP models do not generate an appropriate flat-field. In cases where no flat-field frame is available then the parameter `TUNE_NOFLAT=YES` can be specified; or, alternatively you can reply `NONE` when prompted for the name of the flat-field frame. In either case, the balance factors will be set to unity.

Parameters:

- **FFIELD** - Name of flat-field image.
- **FLTFIT** - Fitter for flat-field.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_FFINTER** - YES for interaction with flat field.
- **TUNE_FFLMED** - YES for local median, NO for mean.
- **TUNE_FFLSMP** - Number of local pixels to median/mean.
- **TUNE_FFNXREJ** - Reject cycles for X-fits.
- **TUNE_FFNXPLY** - Number of X-coefficients.
- **TUNE_FFNYREJ** - Reject cycles for Y-fits.
- **TUNE_FFNYPLY** - Number of Y-coefficients.
- **TUNE_FFSUBSMP** - YES for subsampling.
- **TUNE_FFTHRESH** - Reject threshold in sigma.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.

- **TUNE_MXSKYPIX** - Maximum number of sky-pixels.
- **TUNE_NOFLAT** - YES if no flat-field frame is available.
- **TUNE_PREBAL** - YES for pre-balanced flat field.

Reduction File Objects:

- **DEK_ABOVE** - *type:* _INTEGER, *access:* READ.
- **DEK_BELOW** - *type:* _INTEGER, *access:* READ.
- **FITTED_FLAT** - *type:* _REAL, *access:* READ/WRITE.
- **FLAT_ERRORS** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.

6:`ech_sky` – Model Sky Background

The sky intensity is modelled at each increment along the order. The degree of polynomial fitted is adjustable, by default it is set to zero to obtain the 'average' sky behavior.

The use of polynomials or splines of higher degree is advisable when there is a significant gradient to the sky intensity along the slit, as the polynomials are used to predict the sky intensity at each object pixel in the order independently. Note that the meaning of 'increment' differs between regular and 2-D distortion-corrected extractions. For a simple extraction an increment is a single-pixel column. For a 2-D extraction each increment is a scrunched wavelength-scale unit, thus ensuring the accurate modelling of distorted bright emission lines in the sky.

It is also possible to model the sky intensity in the wavelength direction using polynomials. In this case there are parameters available to define the threshold for possible sky lines which will be excluded from the fit (`TUNE_SKYLINW` and `TUNE_SKYLTHR`). When a wavelength-dependent model is used it is also possible to request an extra simulation step which allows the accurate evaluation of the errors on the fitted model (using a monte-carlo simulation). This procedure can improve the variances used during an 'optimal' extraction, particularly in cases where the object is only fractionally brighter than the sky. The simulation is enabled using the hidden parameter `TUNE_SKYSIM=YES`.

The determination of which pixels are sky is done using the masks set by the profiling task or ECHMENU option. These masks can be freely edited to cope with any special requirements as to which regions of the slit are to be used for the sky. This facility is of particular use when processing frames where 'periscopes' have been used to add in 'sky' regions when observing an extended source. In such cases, ECHOMOP currently provides no special treatment and the periscope sky-pixel positions will have to be edited into the sky mask using the task `ech_spatial`/ECHMENU Option 4.

It is also possible to use a separate sky frame by flagging all pixels as sky, modelling the sky, and then resetting the requisite object pixels (using Option 4.2) before extracting using the object frame.

In cases where there is significant contamination of the background due to scattered light, it is possible to use a global model of the background intensity instead. `ech_mdlbck` (Option 22) performs this process and should be used **instead** of the sky modelling option (the two processes are mutually exclusive).

Parameters:

- **FFIELD** - Name of flat-field image.
- **INPTIM** - Frame to extract data from.
- **PHOTON_TO_ADU** - Conversion factor for photons.
- **READOUT_NOISE** - Detector readout noise in counts.
- **SKYFIT** - Function for sky fitting.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MXSKYPIX** - Maximum number of sky-pixels.
- **TUNE_NOFLAT** - YES if no flat-field frame is available.
- **TUNE_SKYINTER** - YES if interactive sky-modelling.
- **TUNE_SKYLINW** - Maximum expected sky-line width.
- **TUNE_SKYLTHR** - Sigma threshold for sky-lines.
- **TUNE_SKYPOLY** - Degree of polynomial to use for sky.
- **TUNE_SKYREJ** - Number of reject cycles.
- **TUNE_SKYRTHR** - Reject threshold in sigma.
- **TUNE_SKYSIM** - YES for sky simulation to be used.
- **TUNE_SKYXPLY** - Degree of X-polynomial to use for sky.

Reduction File Objects:

- **DEK_ABOVE** - *type:* _INTEGER, *access:* READ.
- **DEK_BELOW** - *type:* _INTEGER, *access:* READ.
- **FITTED_FLAT** - *type:* _REAL, *access:* READ.
- **FITTED_SKY** - *type:* _REAL, *access:* READ/WRITE.
- **FLAT_ERRORS** - *type:* _REAL, *access:* READ.

- **FSKY_ERRORS** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **SKY_MASK** - *type:* _INTEGER, *access:* READ.
- **SKY_SPECTRUM** - *type:* _REAL, *access:* READ/WRITE.
- **SKY_VARIANCE** - *type:* _REAL, *access:* READ/WRITE.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.

7:`ech_profile` – Model Object Profile

The object profile model is constructed by subsampling the profile and may be an all order average, or independently calculated for each order (enabled by setting `TUNE_USE_NXF=1`). There are also facilities for modelling profiles which vary slowly with wavelength by fitting polynomials in the wavelength direction (set `TUNE_OBJPOLY>0`).

The degree of subsampling is controlled using the parameter `TUNE_PFLSSAMP` which sets the number of subsamples across the spatial profile.

Parameters:

- **INPTIM** - Frame to extract data from.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MXSKYPIX** - Maximum number of sky-pixels.
- **TUNE_OBJPOLY** - Degree of polynomial to use for object.
- **TUNE_PFLSSAMP** - Maximum number of subsamples in profile.
- **TUNE_USE_NXF** - Fraction of X-samples to use in profile.

Reduction File Objects:

- **DEK_ABOVE** - *type:* _INTEGER, *access:* READ.
- **DEK_BELOW** - *type:* _INTEGER, *access:* READ.
- **FITTED_SKY** - *type:* _REAL, *access:* READ.
- **MODEL_PROFILE** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.

- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **OBJ_MASK** - *type:* _INTEGER, *access:* READ.
- **SKY_MASK** - *type:* _INTEGER, *access:* READ.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.

8:`ech_extrct` – Extract Object and Arc Order Spectra

The extraction of both object and arc orders proceeds in parallel to ensure that the same weights are used in both cases. There are three possible weighting schemes implemented currently. All methods maintain variances and allow individual pixels to be excluded from the extraction process by referring to the object frame quality array. Simple extraction weights all object pixels equally and is much less computationally demanding than the other methods. The object intensity is calculated by summing all the object pixels in each column for each order.

Profile weighted extraction weights each pixel by a factor $P(i,j)^2$ where $P(i,j)$ is the calculated normalised profile at spatial offset j (sub-sampled) from the trace centre and i is the column number.

Optimally weighted (or Variance weighted) extraction weights each pixel by the product of the calculated profile $P(i,j)$ and an estimate of the uncertainty of the pixel intensity.

This estimate is based on the calculated variance following the scheme described by Horne in *An Optimal Extraction Algorithm for CCD spectroscopy* (P.A.S.P. 1986), modified to cope with profile subsampling associated with sloping and/or distorted orders.

In addition, the rejection of cosmic-ray-contaminated pixels has been made available as a separate function in ECHOMOP as the package is not dedicated to CCD-only data reduction. The original cosmic-ray rejection described by Horne has also been retained and can be enabled using the parameter `TUNE_CRCLEAN`, although the dedicated CR module seems to perform better in most cases. Optimally weighted extraction has been shown to improve S/N in the extracted spectra by factors corresponding to up to 20% increases in exposure time, and its use is therefore to be encouraged in most cases. The provision of sky variance modelling helps to ensure that the optimal extraction can still perform 'optimally' even with very low S/N data.

> **NOTE:** *Option 19—Quick-look extraction is provided primarily for at-the-telescope use to permit the observer to quickly check that decent data are being obtained. Quick-look does not use the sky model or flat-field model and should not be used to produce spectra for further analysis.*

Parameters:

- **ARC** - Name(s) of reference (arc) lamp image(s).
- **EXTRACT_MODE** - Extraction mode.
- **FFIELD** - Name of flat-field image.
- **INPTIM** - Frame to extract data from.
- **PHOTON_TO_ADU** - Conversion factor for photons.

- **READOUT_NOISE** - Detector readout noise in counts.
- **TUNE_CRCLEAN** - YES if Cosmic-Ray clean needed.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MXSKYPIX** - Maximum number of sky-pixels.
- **TUNE_NOARC** - YES if no arc frame is available.
- **TUNE_OBJPOLY** - Degree of polynomial to use for object.
- **TUNE_PFLSSAMP** - Maximum number of subsamples in profile.
- **TUNE_SKVRCORR** - YES to apply sky variance correction.
- **TUNE_SKYPOLY** - Degree of polynomial to use for sky.
- **TUNE_SKYREJ** - Number of reject cycles.
- **TUNE_SKYRTHR** - Reject threshold in sigma.

Reduction File Objects:

- **BLAZE_SPECT** - *type:* _REAL, *access:* READ/WRITE.
- **DEK_ABOVE** - *type:* _INTEGER, *access:* READ.
- **DEK_BELOW** - *type:* _INTEGER, *access:* READ.
- **EXTRACTED_ARC** - *type:* _REAL, *access:* READ/WRITE.
- **EXTRACTED_OBJ** - *type:* _REAL, *access:* READ/WRITE.
- **EXTR_ARC_VAR** - *type:* _REAL, *access:* READ/WRITE.
- **EXTR_OBJ_VAR** - *type:* _REAL, *access:* READ/WRITE.
- **FITTED_FLAT** - *type:* _REAL, *access:* READ/WRITE.
- **FITTED_PFL** - *type:* _DOUBLE, *access:* READ.
- **FITTED_SKY** - *type:* _REAL, *access:* READ/WRITE.
- **FLAT_ERRORS** - *type:* _REAL, *access:* READ/WRITE.
- **FSKY_ERRORS** - *type:* _REAL, *access:* READ/WRITE.
- **MODEL_PROFILE** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.

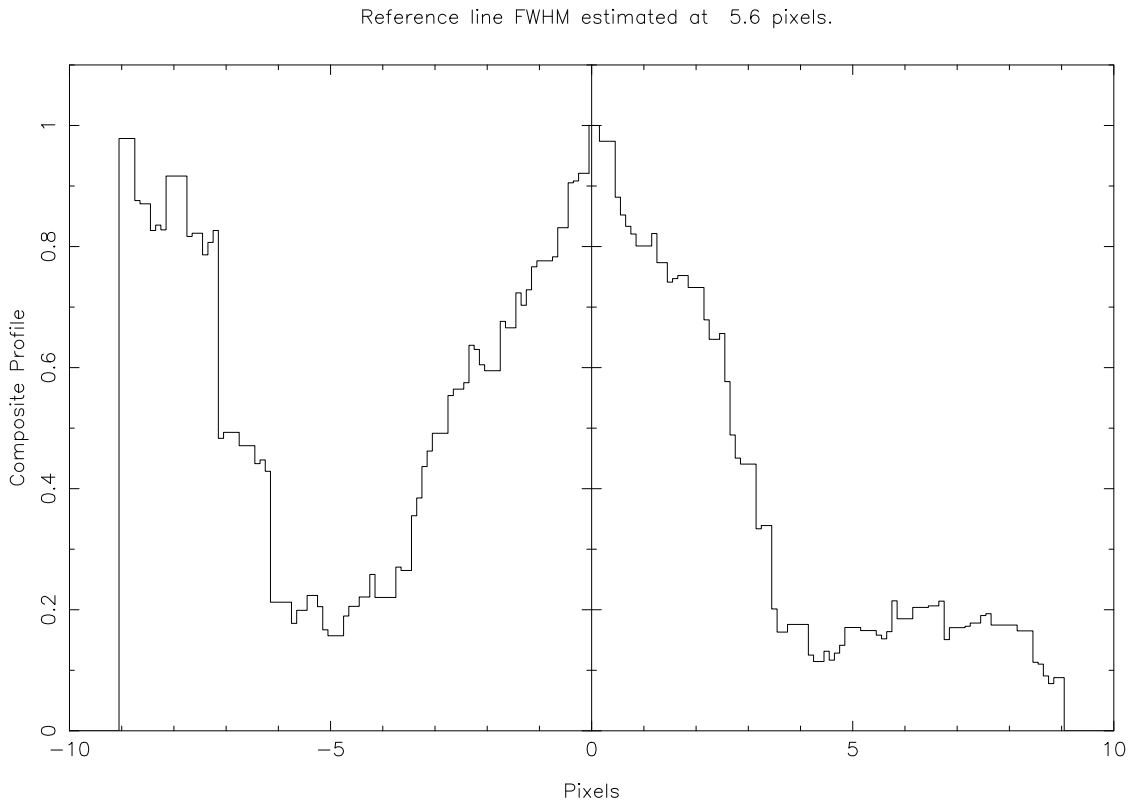


Figure 6: The "average" line-profile for an arc. This was produced using ECHMENU Option 9.

- **OBJ_MASK** - *type:* _INTEGER, *access:* READ.
- **SKY_MASK** - *type:* _INTEGER, *access:* READ.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.

9:`ech_linloc` – Locate Arc Line Candidates

This option is used to locate arc line features for later identification. It consists of two steps:

- 9.1 (`ech_fwhm`)
  Determine the average arc line FWHM.
- 9.2 (`ech_lines`)
  Locate all possible arc lines.

The FWHM is evaluated by co-adding all possible arc line features in the arc frame, and then averaging the resulting profile, and calculating its FWHM. The value is used to scale the Gaussian-s which are fitted to each arc line in order to obtain an estimate of its center position (in X).

Possible arc lines are denoted by any region of an order in which 5 consecutive pixels (P1-5) obey the following relation:

$$P_1 < P_2 < P_3 > P_4 > P_5$$

and are amenable to a Gaussian fit with the FWHM calculated.

This ensures that even very faint features are put forward for possible identification (useful when there are no bright lines in an entire order).

Parameters:

- **ARC** - Name(s) of reference (arc) lamp image(s).
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MAXRFLN** - Maximum number of reference lines.
- **TUNE_NOARC** - YES if no arc frame is available.
- **TUNE_RFLNTHR** - Reference line lower threshold.

Reduction File Objects:

- **EXTRACTED_ARC** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NREF_FRAME** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **OBS_INTEN** - *type:* _REAL, *access:* READ/WRITE.
- **OBS_LINES** - *type:* _REAL, *access:* READ/WRITE.
- **REF_LINE_FWHM** - *type:* _REAL, *access:* READ/WRITE.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.

10:`ech_idwave` – Wavelength Calibrate

The wavelength calibration is done using a reference feature list, usually provided by an ARC-lamp exposure. The routine allows any candidate features to be identified and used as 'knowns' for the calibration (as position/intensity pairs). These features may then be manually identified using a reference lamp atlas. Facilities are provided for adding/deleting lines and altering the degree of polynomial fit performed.

An automatic line-identifier is included which operates by searching the supplied line list for multi-line 'features'. You may optimise the search by constraining the space to be searched in terms of permissible wavelength and/or dispersions (in Angstroms/pixel).

In addition the program will automatically constrain the search range further if it can determine the central order number and wavelength (by looking in the data frame header)

As soon as three orders have been successfully calibrated, the search range for the remaining orders is re-evaluated to take this into account. In general, the automatic method will be most useful when you are unsure of the exact wavelength range covered. When the level of doubt is such that the wavelength scale may decrease from left-to-right across the frame, then the software may be instructed to automatically check for this 'reversed' condition. Set parameter `TUNE_REVCHK` to `YES` to check for a reversed arc; the parameter defaults to `NO` to avoid wasting CPU time. If the wavelength scale is reversed then you should use FIGARO IREVX to flip all the relevant images, and then re-start the reduction.

Finally the software is flexible as to the vertical orientation of the orders, *i.e.*, higher wavelength orders may be at the top or bottom of the frame (for échelle data). Calibration may be performed using either 1 or 2 (before and after) ARC frames at present. See Arc Frames for details.

Options are presented in a menu form and selected by typing a one or two character string, followed by carriage return. The following options are supported:

- **AI** (automatic identification)

  Initiates a search and match of the feature database. Any preset limitations on the wavelength and dispersion range will be taken into account. When a solution is found, a fit report includes a probable status. You may then refine the solution manually, or search for further solutions by hitting the A key.

- **E**(xit)

  Leaves the line identification menu and updates the reduction database copy of the wavelength polynomial to reflect the latest calculated fit.

- **H**(elp)

  Provides interactive browsing of the relevant sections of the help library for line identification.

- **IM** (import ECHARC data) Provides an interface to the ECHARC arc-line identification program. In general the XP option should first be used to export data for ECHARC processing. ECHARC would then be run, and finally the data IMported back into ECHOMOP.

- **O**(rder selection)

  Used to change the currently selected order when operating manually. The order number which is to be selected will be prompted for.

- **P**(olynomial degree)

  Used to alter the degree of polynomial to be used for the wavelength fitting. This may vary up to the maximum set by `TUNE_MAXPOLY`.

- **XP** (export data for ECHARC)

  Used to provide an interface to the ECHARC arc line identification program. In general the XP option should first be used to export data for ECHARC processing. ECHARC would then be run, and finally the data IMported back into ECHOMOP.

- **I**(nteractive identification)

  Enters the interactive line specification, examination section. This section has a menu which provides features for addition/deletion/re-fitting/listing *etc.*, of identified lines.

  This sub-menu includes a set of interactive options to assist in identifying arc lines and fitting the wavelength polynomial to describe the variation along an order. The options provided are as follows and are all selected using a single character. Note that the carriage-return key is not necessary for option selection, care is therefore needed to ensure that the cursor is correctly positioned **before** a cursor dependent option is selected.

  - **A**(uto)
    Initiates an automatic search and match of the reference feature database. Any preset limits on the wavelength and dispersion search range are taken into account. Each possible solution is reported upon with an indication of its probable accuracy. You may then choose to reject it, or examine it in detail for verification or modification.
  - **B**(Clip) Automatically clips any potential blends from the set of identified lines, and re-fits the wavelength polynomial. Possible blends are flagged using the identifier B on the graph, and by a trailing B on the line details output during fits.
  - **C**(lear all identifications)
    Removes all identified line information for the order. The polynomial fit remains though, and any good lines could be recovered by using the R(e-interpolate) option.
  - **D**(elete a line)
    Removes an identified line from the set of identified lines. This will usually be used to remove a suspect line which has been incorrectly automatically identified. The identified line nearest to the cursors X-position when the D is pressed, will be the one which is deleted.
  - **E**(xit)
    Leaves the interactive identification/fitting routine and returns to the main line id menu. The current set of fitted lines, and the polynomial are saved in the reduction database.
  - **F**(it polynomial)
    Performs a polynomial fit to the positions/wavelengths of all currently identified lines. Reports on the deviation of each line from the fit, and the improvement possible by deleting each line and re-fitting. No check is made against the database to see if further lines may now be identified, use the R option if this is required.
  - **I**(nformation)
    Reports information known about the nearest located line (relative to X-position of cursor when I is pressed). Details include position, wavelength *etc.*
  - **K**(eep)
    Flags the nearest identified-line to be retained during re-interpolation even when it would otherwise be rejected. Any manually identified lines will be automatically flagged in this way. Kept lines are indicated by a K on the graph.
  - **L**(ist known lines)
    Examines the feature database to find the nearest features to the current X-position of the cursor, assuming the current polynomial fits' predictions about the corresponding wavelengths.

– **M**(ove to coordinate)

When viewing a zoomed graph of lines, allows the central point of the plot to be moved to any point along the X-axis. A prompt is made for the exact co-ordinate required. Once a wavelength scale has been fitted, Move operates on wavelengths, otherwise it operates on X-coordinates.

– **N**(ew line)

Used to specify a completely new feature. Should be used when you wish to identify a line which has not been located at all by the arc line location algorithm. All located lines brighter than the threshold (see Threshold option) are indicated by a | appearing above their peak. Identified lines additionally have the wavelength shown.

– **O**(utwards zoom)

Reverses any currently selected zoom factor on the displayed graph of lines.

– **P**(lot)

Refreshes the plotted graph.

– **Q**(uit)

Leaves the manual identification/fitting routine **without** updating the polynomial in the reduction database.

> **NOTE:** *any newly identified lines will be saved. Only the polynomial (which represents the outcome of an identification) is left unchanged.*

– **R**(e-interpolate)

A polynomial fit is made to the currently identified lines. This polynomial is then used to search the database for any new line candidates. All new candidates are then added into the fit and the polynomial iteratively re-fitted and clipped until a stable solution is obtained.

– **S**(et line wavelength)

Used to set the wavelength of a line. The line whose wavelength is to be set should be the one nearest to the X-position of the cursor and may be an identified or unidentified line. Use the N(ew) option to create a new line where none had been auto-located.

– **T**(hreshold)

Used to set the threshold for identifiable lines. Only identifiable lines can be automatically identified by the Auto or Re-interpolate options. The threshold is plotted on the graph as a horizontal dotted line and is initially set according to the value of the parameter `TUNE_IDSTRNG`.

– **XCLIP**

Used to toggle automatic clipping of poorly fitted lines. It is checked every time a fit is performed and when set will allow the fitting module to reject lines if doing so will significantly improve the RMS for the fit.

– **Z**(oom plot)

Increases the magnification of the graph used to plot the line positions. To reverse the effect use the O(utwards zoom) option.

– **+** (Increment polynomial order)

Increments the degree of polynomial used for the wavelength fitting. May be increased up to the maximum specified by the parameter `TUNE_MAXPOLY`.

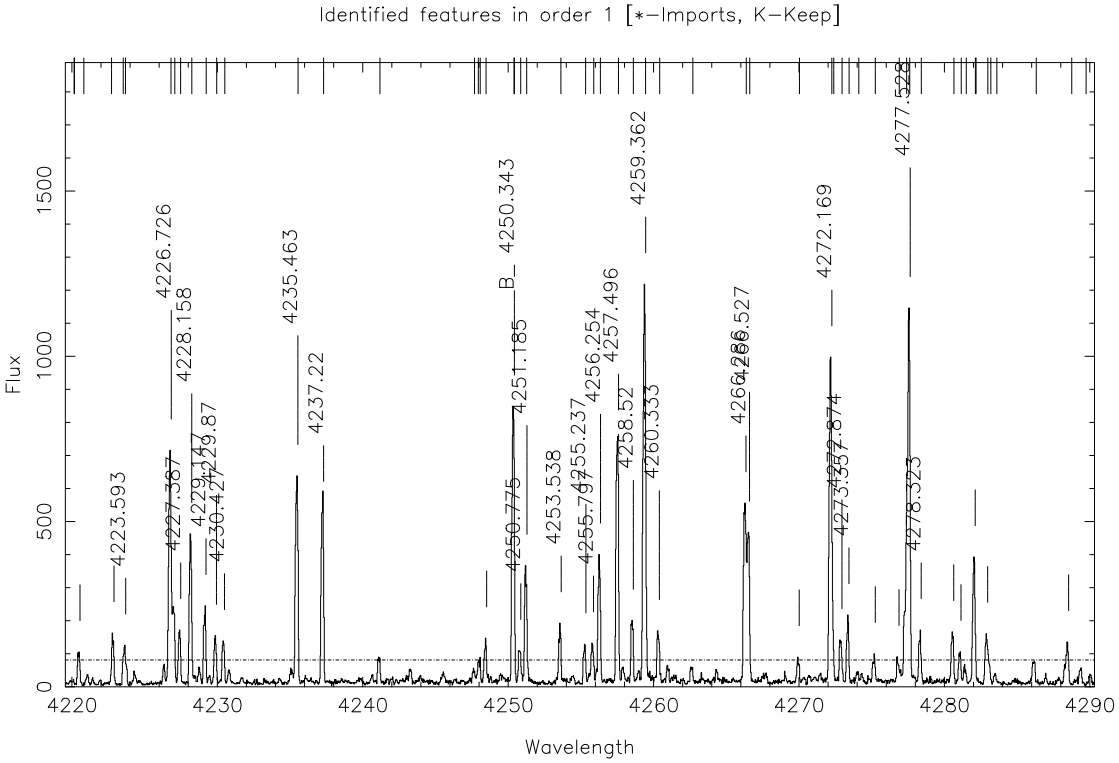


Figure 7: A typical plot during interactive line-identification.

- **-** (Decrement polynomial order)
  Decrements the degree of polynomial used for the wavelength fitting.
- > (Shift display right)
  When a zoomed graph is plotted, will shift the viewpoint along to the right (plus X) by an amount sufficient to show the adjacent section of the plot.
- < (Shift display left)
  When a zoomed graph is plotted, will shift the viewpoint along to the left (minus X) by an amount sufficient to show the adjacent section of the plot. (any other key), provides available information on the line nearest to the current X-position of the cursor.
- **=**
  Switches between fitting functions. Currently supported are polynomials and splines.
- **^**
  Sends a copy of the current plotted graph to the hardcopy device. The appropriate device must have been supplied at program startup (*e.g.* `echmenu hard=ps_l`).

The Figure above shows an example of the plot displayed during interactive line-identification. The following points should be noted:

- Identified lines have their wavelength shown directly above the peak

- Possible blends are flagged with a B
- The threshold intensity for identifiable lines is shown as a horizontal dot/dash line
- The positions of line-list lines are shown along the top X-axis

Parameters:

- **AUTO_ID** - YES for fully automatic identification.
- **CENTRAL_ONUM** - Hidden parameter.
- **CENTRAL_WAVE** - Hidden parameter.
- **ECH_FTRDB** - Reference line list database.
- **HI_WAVE** - Longest wavelength to search for arc lines.
- **LOW_WAVE** - Shortest wavelength to search for arc lines.
- **MAX_DISPERSION** - Max dispersion (Units per-pixel) allowed.
- **MIN_DISPERSION** - Min dispersion (Units per-pixel) allowed.
- **TUNE_DB_SCOPE** - Scope of database neighbour scanning.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_IDINMN** - Minimum number of features used.
- **TUNE_IDINMX** - Maximum number of features used.
- **TUNE_IDMDLT** - Maximum neighbours to check.
- **TUNE_IDMXDIF** - Maximum ratio difference.
- **TUNE_IDSDLT** - Starting number of neighbours to check.
- **TUNE_IDSTRNG** - Minimum strength of identified lines.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MAXRFLN** - Maximum number of reference lines.
- **TUNE_REVCHK** - Maximum number of reference lines.
- **W_NPOLY** - Number of coeffs of wavelength fitting function.
- **WAVFIT** - Function for wavelength fitting.

Reduction File Objects:

- **EXTRACTED_ARC** - *type:* _REAL, *access:* READ.
- **FITTED_WAVES** - *type:* _DOUBLE, *access:* READ/WRITE.

- **ID_COUNT** - *type:* _INTEGER, *access:* READ/WRITE.
- **ID_LINES** - *type:* _REAL, *access:* READ/WRITE.
- **ID_STATUS** - *type:* _INTEGER, *access:* READ/WRITE.
- **ID_WAVES** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NREF_FRAME** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **OBS_INTEN** - *type:* _REAL, *access:* READ/WRITE.
- **OBS_LINES** - *type:* _REAL, *access:* READ/WRITE.
- **ORDER_IDNUM** - *type:* _INTEGER, *access:* READ/WRITE.
- **REF_LINE_FWHM** - *type:* _REAL, *access:* READ/WRITE.
- **WSEAR_END** - *type:* _REAL, *access:* READ/WRITE.
- **WSEAR_START** - *type:* _REAL, *access:* READ/WRITE.
- **W_POLY** - *type:* _DOUBLE, *access:* READ/WRITE.

11:`ech_blaze` – Fit and Apply Ripple Correction

This option consists of two steps as follows:

- 11.1 (`ech_fitblz`) Model blaze function using a polynomial.
- 11.2 (`ech_doblz`) Apply blaze function to extracted spectrum.

If flux calibration is not being performed it is sometimes desirable to remove the 'blaze' function from the extracted spectrum to assist in fitting line profiles *etc.* during data analysis.

A task is provided for this purpose which operates by fitting curves to the flat-field orders. The curves can be polynomials, splines or simple fits based on local-median values. The fits may be automatically or interactively clipped and the resulting blaze spectrum is normalised such that its median intensity is unity.

The normalised blaze is then divided into the extracted spectrum. It is important to remember that this operation is performed upon the 'extracted' spectrum.

After a blaze function has been applied to the extracted order all its values may be reset to unity to ensure that the order(s) cannot be re-flattened in error (`TUNE_BLZRSET=YES`). If the blaze is to be re-applied then the correct procedure is to first re-extract the order(s) concerned and then re-fit the blaze.

Parameters:

- **BLZFIT** - Function for blaze fitting.

- **BLZ_INTERACT** - YES for interactive blaze-fitting.
- **BLZ_NPOLY** - Number of coeffs of blaze fitting function.
- **FFIELD** - Name of flat-field image.
- **TUNE_BLZRSET** - YES if blaze function to be reset after use.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_NOFLAT** - YES if no flat-field frame is available.
- **TUNE_YBLAZE** - YES for Y-blaze correction.

Reduction File Objects:

- **BLAZE_MEDIANS** - *type:* _REAL, *access:* READ/WRITE.
- **BLAZE_SPECT** - *type:* _REAL, *access:* READ/WRITE.
- **DEK_ABOVE** - *type:* _INTEGER, *access:* READ.
- **DEK_BELOW** - *type:* _INTEGER, *access:* READ.
- **EXTRACTED_OBJ** - *type:* _REAL, *access:* READ/WRITE.
- **EXTR_OBJ_VAR** - *type:* _REAL, *access:* READ/WRITE.
- **FITTED_WAVES** - *type:* _DOUBLE, *access:* READ.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NREF_FRAME** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **TRC_OUT_DEV** - *type:* _REAL, *access:* READ.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.

12:`ech_scrunch` – Scrunch

This option is used to scrunch the extracted order spectra (and arc order spectra) into a (usually) linear wavelength scale.

- 12.1 (`ech_fitfwhm`)
  Fit reference line FWHM as a function of wavelength
- 12.2 (`ech_wscale`)
  Calculate the wavelength scale

- 12.3 (`ech_scrobj`)

  Scrunch the extracted object order

- 12.4 (`ech_scrarc`)

  Scrunch the extracted arc orders

The scrunching of spectra into a linear wavelength scale provides exactly the same facilities available using the FIGARO SCRUNCH program, except that it works on an order-by-order basis.

ECHOMOP provides both global (bin size constant for all orders) and per-order scrunching options. The global option would normally be used when it is necessary to co-add the extracted orders from multiple data frames, and a standard bin-size is required.

Scrunching results in both a 2-D array of scrunched individual orders, and a merged 1-D array of the whole wavelength range. A utility (Option 21) is provided to assist in the co-adding of spectra from many frames together. This option assumes that the first frame in the reduction has been scrunched with the required wavelength scale. It then reads a list of additional reduction database names (or EXTOBJ result file names) from an ASCII file called `NAMES.LIS`. The extracted spectra from each of these reduction files are then scrunched to the same scale and co-added into the scrunched spectra in the current reduction file. The type of weighting during addition is controlled using the parameter `TUNE_MRGWGHT`.

Parameters:

- **BIN_SIZE** - Bin size for global scrunch.
- **SCRUNCH_TYPE** - Type of spectrum to scrunch.
- **SET_WSCALE** - YES to scrunch to a global bin size.
- **START_WAVE** - Start wavelength for rebinned scale.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_FLUX** - YES if flux is to be conserved.
- **TUNE_INTR** - YES if linear interpolation required.
- **TUNE_LOG** - YES if output scale logarithmic.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MAXRFLN** - Maximum number of reference lines.
- **TUNE_MERGE** - YES for merging multiple frame data.
- **TUNE_QUAD** - YES if quadratic interpolation required.
- **TUNE_SCFRACT** - Fractional ratio for twin scales.
- **TUNE_SCRADD** - Number of bins to add together.
- **TUNE_SCRMODE** - Scrunching mode control.

- **TUNE_SKEW** - Skew shift in bins.
- **TUNE_YBLAZE** - YES for Y-blaze correction.

Reduction File Objects:

- **BLAZE_SPECT** - *type:* _REAL, *access:* READ.
- **ERR_SPECTRUM** - *type:* _REAL, *access:* READ/WRITE.
- **EXTRACTED_ARC** - *type:* _REAL, *access:* READ.
- **EXTRACTED_OBJ** - *type:* _REAL, *access:* READ.
- **EXTR_ARC_VAR** - *type:* _REAL, *access:* READ.
- **EXTR_OBJ_VAR** - *type:* _REAL, *access:* READ.
- **ID_COUNT** - *type:* _INTEGER, *access:* READ.
- **ID_LINES** - *type:* _REAL, *access:* READ.
- **ID_WAVES** - *type:* _REAL, *access:* READ.
- **ID_WIDTHS** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_BINS** - *type:* _INTEGER, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NREF_FRAME** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NX_REBIN** - *type:* _INTEGER, *access:* READ/WRITE.
- **REF_LINE_FWHM** - *type:* _REAL, *access:* READ.
- **SCRNCHD_ARC** - *type:* _REAL, *access:* READ/WRITE.
- **SCRNCHD_ARCV** - *type:* _REAL, *access:* READ/WRITE.
- **SCRNCHD_OBJ** - *type:* _REAL, *access:* READ/WRITE.
- **SCRNCHD_OBJV** - *type:* _REAL, *access:* READ/WRITE.
- **SCRNCHD_WAVES** - *type:* _DOUBLE, *access:* READ/WRITE.
- **WAVELENGTH** - *type:* _DOUBLE, *access:* READ/WRITE.
- **WID_POLY** - *type:* _DOUBLE, *access:* READ/WRITE.
- **W_POLY** - *type:* _DOUBLE, *access:* READ.
- **1D_SPECTRUM** - *type:* _REAL, *access:* READ/WRITE.

13:ech_ext2d – 2-D Distortion Correction

Detectors such as the IPCS often cause major geometric distortions in the image created using them. ECHOMOP provides a mechanism for modelling such distortion, and using that model to provide corrections during the extraction process. It is also possible to generate a 'corrected' version of each order, for visual examination, or processing by other (single spectra) software.

The distortion model uses a coordinate system based on X = calibrated wavelength at trace, Y = pixel offset from trace, and is thus performed independently for each order in turn.

The ARC frame is used to locate the positions of each identified arc line at a variety of offsets from the trace centre. The difference between its wavelength (as identified) and that predicted by the wavelength polynomial for its observed position is then calculated. These differences are modelled using a Chebyshev polynomial.

Once a 2-D fit has been obtained, it is refined by either manual or automatic clipping of deviant points. When done manually the positions of all the points being fitted (*i.e.*, arc line centers) may be plotted in a highly exaggerated form, in which systematic distortions of sub-pixel magnitude are readily apparent.

As the wavelength scale produced by the distortion fitting leads inevitably to some re-binning when the extraction takes place, it is normal to extract into a scrunched wavelength scale (*e.g.*, constant bin size) and this is the default behavior of the 2-D extraction task/ECHMENU Option 13.

This option is used to perform a full 2-D distortion-corrected extraction and is provided for cases where the distortion of the frame is significant. The option consists of four steps as follows:

- 13.1

  Calculate wavelength scale.

- 13.2

  Fit 2-D polynomials to the distortion.

- 13.3

  Re-bin the order into a corrected 2-D form.

- 13.4

  Extract from the re-binned form.

Distortion correction is done on a per-order basis, each order having its own distortions mapped independently. The distortion is modelled by using a tie-point data set composing of the positions and wavelength of all identified arc lines in the order. Thus, a wavelength calibration is a pre-requisite to the distortion corrected extraction operation. A 2-D Chebyshev polynomial is then fitted to the wavelength deviations of each arc-line pixel, relative to the wavelength at the trace/arc line intersection. The polynomial is used to generate delta-wavelength values at pairs of (X,Y-offset) coordinates, *i.e.*;

delta-wavelength = 2dPoly(X-pixel, Y-offset from trace)

for all X- and all Y-offsets within the dekker.

This map of wavelength delta values is then used to drive a 2-D scrunch of the order into a form where each column (X=nn) corresponds to consistent wavelength increment.

The final step is to extract the data from this re-binned form. The extraction algorithm used is identical to the 1-D case from this point on.

Parameters:

- **ARC** - Name(s) of reference (arc) lamp image(s).
- **BIN_SIZE** - Bin size for global scrunch.
- **EXTRACT_MODE** - Extraction mode.
- **FFIELD** - Name of flat-field image.
- **INPTIM** - Frame to extract data from.
- **PHOTON_TO_ADU** - Conversion factor for photons.
- **READOUT_NOISE** - Detector readout noise in counts.
- **SET_WSCALE** - YES to scrunch to a global bin size.
- **SKYFIT** - Function for sky fitting.
- **START_WAVE** - Start wavelength for rebinned scale.
- **TUNE_CRCLEAN** - YES if Cosmic-Ray clean needed.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MAXRFLN** - Maximum number of reference lines.
- **TUNE_MXSKYPIX** - Maximum number of sky-pixels.
- **TUNE_NOARC** - YES if no arc frame is available.
- **TUNE_OBJPOLY** - Degree of polynomial to use for object profile.
- **TUNE_PFLSSAMP** - Maximum number of subsamples in profile.
- **TUNE_SKVRCORR** - YES to apply sky variance correction.
- **TUNE_SKYLINW** - Maximum expected sky-line width.
- **TUNE_SKYLTHR** - Sigma threshold for sky-lines.
- **TUNE_SKYPOLY** - Degree of polynomial to use for sky.
- **TUNE_SKYREJ** - Number of reject cycles.
- **TUNE_SKYRTHR** - Reject threshold in sigma.

- **TUNE_SKYSIM** - YES for sky simulation to be used.
- **TUNE_SKYXPLY** - Degree of X-polynomial to use for sky.
- **W2_NX_POLY** - Maximum order of 2-D X-axis polynomial.
- **W2_NY_POLY** - Maximum order of 2-D Y-axis polynomial.
- **2D_INTERACT** - YES for interactive 2-D polynomial fitting.

Reduction File Objects:

- **DEK_ABOVE** - *type:* _INTEGER, *access:* READ.
- **DEK_BELOW** - *type:* _INTEGER, *access:* READ.
- **FITTED_FLAT** - *type:* _REAL, *access:* READ.
- **FITTED_PFL** - *type:* _DOUBLE, *access:* READ.
- **FITTED_SSKY** - *type:* _REAL, *access:* READ/WRITE.
- **FLAT_ERRORS** - *type:* _REAL, *access:* READ.
- **FSSKY_ERRORS** - *type:* _REAL, *access:* READ/WRITE.
- **ID_COUNT** - *type:* _INTEGER, *access:* READ.
- **ID_LINES** - *type:* _REAL, *access:* READ.
- **ID_WAVES** - *type:* _REAL, *access:* READ.
- **MODEL_PROFILE** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_BINS** - *type:* _INTEGER, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NREF_FRAME** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NX_REBIN** - *type:* _INTEGER, *access:* READ/WRITE.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **OBJ_MASK** - *type:* _INTEGER, *access:* READ.
- **REBIN_ARC** - *type:* _REAL, *access:* READ/WRITE.
- **REBIN_EARC** - *type:* _REAL, *access:* READ/WRITE.
- **REBIN_EORDER** - *type:* _REAL, *access:* READ/WRITE.
- **REBIN_ORDER** - *type:* _REAL, *access:* READ/WRITE.
- **REBIN_QUALITY** - *type:* _BYTE, *access:* READ/WRITE.

- **REF_LINE_FWHM** - *type:* _REAL, *access:* READ.
- **SCRNCHD_ARC** - *type:* _REAL, *access:* READ/WRITE.
- **SCRNCHD_ARCV** - *type:* _REAL, *access:* READ/WRITE.
- **SCRNCHD_OBJ** - *type:* _REAL, *access:* READ/WRITE.
- **SCRNCHD_OBJV** - *type:* _REAL, *access:* READ/WRITE.
- **SCRNCHD_WAVES** - *type:* _DOUBLE, *access:* READ.
- **SKY_MASK** - *type:* _INTEGER, *access:* READ.
- **SSKY_SPECTRUM** - *type:* _REAL, *access:* READ/WRITE.
- **SSKY_VARIANCE** - *type:* _REAL, *access:* READ/WRITE.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.
- **W_POLY** - *type:* _DOUBLE, *access:* READ.
- **W_POLY_2D** - *type:* _DOUBLE, *access:* READ/WRITE.

14:`ech_result` – Write Results File

This option provides three output formats for data reduced within ECHOMOP.

The supported formats are NDF, ASCII, and DIPSO stack. Many other file formats can be accessed by use of the Starlink utility CONVERT. Where applicable to the data format, errors will be included. For example, DIPSO stacks can not handle error data; NDFs can.

Object or arc spectra data may be output. Data for any of: extracted orders, scrunched orders, or merged spectra may be used. A single order may be selected for output using the task `ech_single`/ECHMENU Option 24 otherwise all-order data are output.

Parameters:

- **RESULT_FORMAT** - Output format required.
- **RESULT_TYPE** - Type of result output required.
- **ASCII_FILE** - ASCII file to list data to.
- **ECH_RDUCD** - Output spectrum data file.
- **STACK** - DIPSO stack to store data to
- **TUNE_AAACODE** - AAA category code.
- **TUNE_AIRTOVAC** - YES to correct wavelengths.
- **TUNE_ARCHIVE** - YES to enable automatic archiving.
- **TUNE_USEAAA** - YES if Abstracts object category codes used.

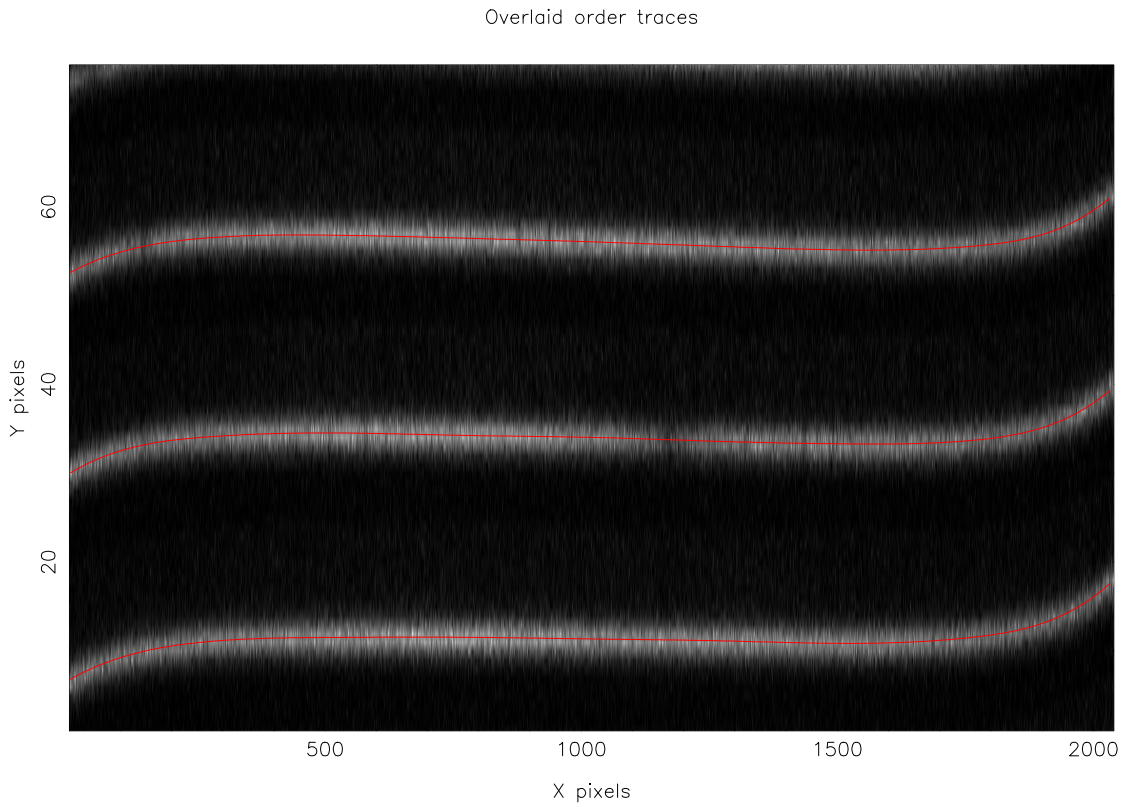


Figure 8: A typical set of order paths as plotted using ECHMENU Option 15. The distortions at the order extremities are due to the IPCS detector used.

15:`ech_trplt` **– Plot Order Traces**

This option simply plots a graph with the same dimensions as the raw data frames. The graph shows the paths of the traced order polynomials across the frame. The option should be used after Option 2 or 3 to check that the traces are appropriate.

The Figure below shows an example of the order trace paths plot.

Parameters:

- **TRACIM** - Frame for order tracing.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.

16:`ech_trcsis` **– Check Trace Consistency**

This option may be used to check the consistency of the order traces with each other. The task predicts the path of each trace by fitting a function to the positions of the other orders at each X-position.

The order whose trace deviates the most from the prediction is flagged as the 'worst' and the option to update the path using the predictions is offered. This will allow the easy correction of

common tracing problems which can occur at the frame edges and with very faint or partial orders.

In general it will only be effective when there are more than half a dozen orders in the frame. The degree of variation and the consistency threshold may be set using the parameters `TUNE_CNSDEV` and `TUNE_TRCNS`.

Parameters:

- **TRACIM** - Frame for order tracing.
- **TRC_INTERACT** - YES for interactive order-fitting.
- **TRC_NPOLY** - Number of coeffs of trace-fit function.
- **TUNE_CNSDEV** - Maximum consistent deviation.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MXSMP** - Maximum number of X-samples to trace.
- **TUNE_TRCNS** - Consistency requirement.

Reduction File Objects:

- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ/WRITE.

17:`ech_decos2` – Post-trace Cosmic-Ray Locate

This utility option should be run immediately after `ech_spatial`/ECHMENU Option 4 has been used to define the dekker and object limits.

It uses information about the order paths and the spatial profile in order to do a more effective cosmic-ray location. The sky and object pixels are processed separately in two passes.

Each order in turn is processed by evaluating the degree to which its pixels exceed their expected intensities (based upon profile and total intensity in the increment).

A cumulative distribution function is then constructed and clipped at a pre-determined sigma level. Clipping and re-fitting continues until a Kolmogorov-Smirnov test indicates convergence or until the number of clipped points falls to one per iteration. Located cosmic-ray pixels are flagged in the quality array. Both this and the pre-trace-locator are automatically followed by a routine to do sky-line checking. This routine can restore any pixels it judges to be possible sky line pixels (rather than cosmic-ray hits). If there are many frames of the same object available then it is possible to use coincidence checking to enhance cosmic-ray detection. The script `decos_many` will take list of input frames and perform this checking, and flags the cosmic-ray pixels in each frame. To use it type:

```
% $ECHOMOP_EXEC/decos_many
```

Parameters:

- **INPTIM** - Frame to extract data from.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_DPRBTHR** - Maximum probability for K-S test.
- **TUNE_DSGMTHR** - Sigma threshold for clip.
- **TUNE_FINCPLY** - Number of coefficients.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MINCR** - Minimum Cosmic-Ray intensity.
- **TUNE_MXSKYPIX** - Maximum number of sky pixels.

Reduction File Objects:

- **DEK_ABOVE** - *type:* _INTEGER, *access:* READ.
- **DEK_BELOW** - *type:* _INTEGER, *access:* READ.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **OBJ_MASK** - *type:* _INTEGER, *access:* READ.
- **SKY_MASK** - *type:* _INTEGER, *access:* READ.
- **SKY_SPECTRUM** - *type:* _REAL, *access:* READ/WRITE.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.

18:`ech_decimg` – Image Cosmic-Ray Pixels

This option uses the quality array to determine which pixels have been flagged as contaminated by cosmic-ray hits. It then takes the original object frame and makes a copy in which all the hit pixels are replaced by zero values. This frame should then be blinked with the original to visually assess the success of the cosmic-ray location process.

Parameters:

- **INPTIM** - Frame to extract data from.
- **OUTPUT_IMAGE** - Output image frame.
- **TRACIM** - Frame for order tracing.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.

19:`ech_qextr` – Quick-look Extraction

This option allows quick extraction of an order (or all orders) once `ech_spatial`/ECHMENU Option 4 has been completed (object- and sky-pixel selection). The extraction method used is simple sum of pixels in the dekker and the sky subtraction is done by calculating the average value over all sky pixels in the increment. No flat-field balance factors are used. This option should only be used to get a quick-look at the data, the spectra produced should not be used for further analysis as much better results will be obtained by using Option 8 for a full extraction.

Parameters:

- **ARC** - Name(s) of reference (arc) lamp image(s).
- **INPTIM** - Frame to extract data from.
- **PHOTON_TO_ADU** - Conversion factor for photons.
- **READOUT_NOISE** - Detector readout noise in counts.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MXSKYPIX** - Maximum number of sky-pixels.
- **TUNE_NOARC** - YES if no arc frame is available.

Reduction File Objects:

- **DEK_ABOVE** - *type:* _INTEGER, *access:* READ.
- **DEK_BELOW** - *type:* _INTEGER, *access:* READ.
- **EXTRACTED_ARC** - *type:* _REAL, *access:* READ/WRITE.
- **EXTRACTED_OBJ** - *type:* _REAL, *access:* READ/WRITE.
- **EXTR_ARC_VAR** - *type:* _REAL, *access:* READ/WRITE.
- **EXTR_OBJ_VAR** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **OBJ_MASK** - *type:* _INTEGER, *access:* READ.
- **SKY_MASK** - *type:* _INTEGER, *access:* READ.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.

20:`ech_wvcsis` – Check Wavelength Scales

This function performs a function analogous to that performed by Option 16 (order traces), only operating upon the wavelength fits.

It is thus used after Option 10 has been used to calculate the wavelength scales.

The wavelength consistency check is confined to those areas beyond the range within which lines have been identified. It therefore only corrects the very ends of the orders wavelength scales.

These are the regions where problems are most likely to occur, as the polynomial fits can become unstable when a high number of coefficients has been used, and there are no fitted points for a substantial fraction of the order (*e.g.*, first 20%).

Parameters:

- **AUTO_ID** - YES for fully automatic identification.
- **ECH_FTRDB** - Reference line list database.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MAXRFLN** - Maximum number of reference lines.
- **TUNE_MXSMP** - Maximum number of X-samples to trace.
- **W_NPOLY** - Number of coeffs of wavelength fitting function.

Reduction File Objects:

- **EFTRDB_WAVELENGT** - *type:* _REAL, *access:* READ.
- **ID_COUNT** - *type:* _INTEGER, *access:* READ/WRITE.
- **ID_LINES** - *type:* _REAL, *access:* READ/WRITE.
- **ID_STATUS** - *type:* _INTEGER, *access:* READ/WRITE.
- **ID_WAVES** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **OBS_INTEN** - *type:* _REAL, *access:* READ.
- **OBS_LINES** - *type:* _REAL, *access:* READ.
- **ORDER_IDNUM** - *type:* _INTEGER, *access:* READ.
- **WSEAR_END** - *type:* _REAL, *access:* READ.
- **WSEAR_START** - *type:* _REAL, *access:* READ.
- **W_POLY** - *type:* _DOUBLE, *access:* READ/WRITE.

21:`ech_mulmrg` **– Merge Multiple Spectra**

This utility option is provided to assist in co-adding spectra from many frames together. This option assumes that the first frame in the reduction has been scrunched with the required wavelength scale.

It then reads a list of additional reduction database names from an ASCII file called `NAMES.LIS`. The extracted spectra from each of these reduction files is then scrunched to the same scale and co-added into the scrunched spectra in the current reduction file.

The parameter `TUNE_MRGWGHT` controls the type of weighting used during addition.

Parameters:

- **SET_WSCALE** - YES to scrunch to a global bin size.
- **TUNE_AIRTOVAC** - YES to correct wavelengths.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_FLUX** - YES if flux is to be conserved.
- **TUNE_INTR** - YES if linear interpolation required.
- **TUNE_LOG** - YES if output scale logarithmic.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MRGMAXX** - Number of rhs pixels to ignore.
- **TUNE_MRGMINX** - Number of lhs pixels to ignore.
- **TUNE_MRGWGHT** - Type of weighting to apply.
- **TUNE_QUAD** - YES if quadratic interpolation required.
- **TUNE_SCFRACT** - Fractional ratio for twin scales.
- **TUNE_SCRADD** - Number of bins to add together.
- **TUNE_SCRMODE** - Scrunching mode control.
- **TUNE_SKEW** - Skew shift in bins.

Reduction File Objects:

- **ERR_SPECTRUM** - *type:* _REAL, *access:* READ/WRITE.
- **EXTRACTED_OBJ** - *type:* _REAL, *access:* READ.
- **EXTR_OBJ_VAR** - *type:* _REAL, *access:* READ.
- **NO_OF_BINS** - *type:* _INTEGER, *access:* READ.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NREF_FRAME** - *type:* _INTEGER, *access:* READ.

- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NX_REBIN** - *type:* _INTEGER, *access:* READ.
- **SCRNCHD_OBJ** - *type:* _REAL, *access:* READ/WRITE.
- **SCRNCHD_OBJV** - *type:* _REAL, *access:* READ/WRITE.
- **SCRNCHD_WAVES** - *type:* _DOUBLE, *access:* READ/WRITE.
- **WAVELENGTH** - *type:* _DOUBLE, *access:* READ/WRITE.
- **W_POLY** - *type:* _DOUBLE, *access:* READ.
- **1D_SPECTRUM** - *type:* _REAL, *access:* READ/WRITE.

22:`ech_mdlbck` **– Model Scattered Light**

This option is used in place of Option 6 (Model sky) in cases where there is severe scattered light contamination. It works by fitting independent polynomials/splines to each image column (actually only the inter-order pixels). Once the column fits have been done the results are used as input to a second round of fits which proceeds parallel to the the order traces.

The final fitted values are saved in the sky model arrays.

This process is very CPU intensive and should not be used unless is it needed.

Parameters:

- **FFIELD** - Name of flat-field image.
- **INPTIM** - Frame to extract data from.
- **PHOTON_TO_ADU** - Conversion factor for photons.
- **READOUT_NOISE** - Detector readout noise in counts.
- **SKYFIT** - Function for sky fitting.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MXSKYPIX** - Maximum number of sky-pixels.
- **TUNE_NOFLAT** - YES if no flat-field frame is available.
- **TUNE_SKYPOLY** - Degree of polynomial to use for sky.
- **TUNE_SKYREJ** - Number of reject cycles.
- **TUNE_SKYRTHR** - Reject threshold in sigma.
- **TUNE_SKYSIM** - YES for sky simulation to be used.

Reduction File Objects:

- **DEK_ABOVE** - *type:* _INTEGER, *access:* READ.
- **DEK_BELOW** - *type:* _INTEGER, *access:* READ.
- **FITTED_FLAT** - *type:* _REAL, *access:* READ.
- **FITTED_SKY** - *type:* _REAL, *access:* READ/WRITE.
- **FLAT_ERRORS** - *type:* _REAL, *access:* READ.
- **FSKY_ERRORS** - *type:* _REAL, *access:* READ/WRITE.
- **NO_OF_ORDERS** - *type:* _INTEGER, *access:* READ.
- **NX_PIXELS** - *type:* _INTEGER, *access:* READ.
- **NY_PIXELS** - *type:* _INTEGER, *access:* READ.
- **SKY_MASK** - *type:* _INTEGER, *access:* READ.
- **SKY_SPECTRUM** - *type:* _REAL, *access:* READ/WRITE.
- **SKY_VARIANCE** - *type:* _REAL, *access:* READ/WRITE.
- **TRC_POLY** - *type:* _DOUBLE, *access:* READ.

23:`ech_tuner` – Adjust Tuning Parameters

This option simply provides a centralised mechanism for viewing and editing the values of all the tuning parameters known to the system.

Most of the time it is more convenient to use the -option syntax from the main menu, as this only lists parameters used by the current default option.

In Option 23, any parameters used by the current default option are flagged with an asterix.

If, when used, a tuning parameter has a non-default value, an informational message is displayed.

24:`ech_single` – Set Single-order Processing

Allows the selection of a single order for all tasks which operate on an order-by-order basis, *e.g.*, to re-fit the order trace for order 3, leaving all other orders unchanged, you would first use this option to change the selected order to number 3, and then invoke the `ech_fitord` task/ECHMENU Option 3.

Note that any options which operate on all orders at once (*e.g.*, trace consistency checking) will still operate correctly when a single order is selected, as they ignore the selection and use all the orders anyway).

Note that when using the individual tasks the strategy for selecting single order/all order operation is different. Individual function tasks which can operate on single orders all utilise the parameter `IDX_NUM_ORDERS`.

which you should set to the number of the order to process, or to zero to indicate that all orders are to be processed in turn. *e.g.*:

```
% ech_trace idx_num_orders=4
```

would just trace order number 4.

25:`ech_all` – Set All-order Processing

Selects automatic looping through all available orders for all tasks which are performed on an order by order basis. This is the default method of operation.

26:`ech_disable` – Disable an Order

This option disables an order from any further processing. The mechanism for doing this is to remove the order trace. If you need to re-enable an order then it should be re-traced by using Option 24 (select single order) and then Option 2 (trace an order).

Parameters:

- **BAD_ORDER** - Number of the order to disable.
- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.

27:`ech_plot` – Run Plot Utility

Many of the temporary results arrays stored in the reduction database can be of assistance when tracking down problems during a reduction. All of these can be graphically examined using the `ech_plot` task/ECHMENU Option 27.

This utility prompts for object names for the Y-axis (and optionally the X-axis separated by a comma). If a null object name is returned then that axis will be automatically generated using monotonically increasing integer values.

The normal usage will be to supply only the name of the Y-axis object and leave the X-axis to be auto generated. An exception is when plotting wavelength objects along the X-axis.

Note that unless you are interested in the first order of a multi-order array then the array indices to start plotting from **must** be supplied.

e.g.: OBJ would denote the first orders' extracted object spectra. OBJ[1,4] would denote the fourth orders' extracted object spectra. ARC[100,13,2] would denote the region of the second arc frames thirteenth order starting at X-sample 100.

Note that in this last case unless the N(umber) of samples to plot has been set to less than the array X-dimension then some samples from the fourteenth order would also be plotted.

Also provided are the following facilities most of which are selected by typing the single character followed by carriage-return.

- **B**(rowse) Used in conjunction with an imaging display. The object or arc data frame should be specified or it will be prompted for (*e.g.*: B MYOBJ)

ECHOMOP will display the image and then put a cursor on the display, you can then position the cursor and type a key indicating which type of data you wish to select for plotting.

Options include:

- O — for object order
- A — for arc order
- S — for scrunched object order
- F — for flat-field model balance factors
- S — for sky model

All arrays are plotted using the cursor position to determine which order, subsample *etc.* is required. By default the full X-axis dimension is used, unless the N command is used to explicitly set the number of samples to plot.

For example, setting N to 20 and imaging the arc frame would allow the extracted profile of an arc line to be plotted simply by positioning the cursor on the line on the image and hitting A, plotting a 20-sample section of the extracted arc from the requisite order.

- **D**(irectory)/ **FD** (full directory)

  Lists a directory of reduction database objects. These are not all arrays and therefore not all plottable. The most common objects are listed by D(ir) and are all arrays.

  The names are such that it is easy to recognise which arrays contain the information required. For example the object FFLT contains the fitted flat-field balance factors for each order and trace offset.

  Specifying an object name without any dimensional specifications will plot the first N elements starting from the beginning of the array, *i.e.*, ARRAY[1,1,1....] to ARRAY[N,1,1....]

- **E**(xit)

  Returns to the main ECHMENU menu or exits from the `ech_plot` single activity task.

- **G**(raphics/Grayscale)

  Toggles the plotting mode. The grayscale mode is useful for plotting swathes of 2-D and 3-D objects. *e.g.*:

  - in Graphics mode — FSKY[1,1,1] plots the first spatial increment of the sky model for order 1.
  - in Grayscale mode — FSKY[1,1,1] images the entire sky model for order 1. Thus in most cases the first two indices will always be equal to 1 when using grayscale plotting.

- **H**(elp)

  Accesses the HELP Facility.

- **I**(nteractive cursor)

  Sets the cursor display such that the cursor may be moved around the next graph to be plotted and the exact data values plotted can be examined (by pressing the space character).

- **N**(umber of samples)

  Sets the number of data samples plotted from the arrays. Unless specified in the object name, plotting always commences from the beginning of the array and the N is set to the maximum number of elements in the array. Setting a smaller value effectively allows you to zoom in on a small subsection of any array.

- **L**(imit setting)

  Allows the X- and Y-limits of the plot to be set. The limits are normally calculated automatically according to the data in the array. To resume auto scaling just set all the X,Y limits back to 0.

- **R**(ebin factor)

  Sets the degree of re-binning to be performed on the data before plotting it. It remains active until reset to 1 (indicating no rebinning). Specify a positive factor for simple summed bins, and a negative factor to request full smoothing.

- **S**(tyle)

  Allows the specification of various style parameters for the plots. The line styles and colours may be specified, as may the plot style. The following keywords are recognised and may be appended to the S to save time. *e.g.*, `S RED` sets the plot colour to red.

  Keywords:

  - Colours RED,WHITE,BLACK,BLUE,GREEN,YELLOW,CYAN,MAGENTA
  - Line types DOTD,DASH,LINE
  - Plot types POINTS,LINES,*,+,x,BINS
  - Text fonts ROMAN,ITALIC
  - Misc PROMPT,NOPROMPT

- **U**(ser-defined window)

  Toggles user-windowing. When this is activated each subsequent plot will prompt the user to select two limits on the display surface. A box based on these limits will then be used for the plot.

  This allows complete freedom in producing a set of plots. Plots may be partially overlaid, stacked *etc.* Once you have produced the display required the easiest way to get a hardcopy of it is to use the built in facilities on your workstation/X-terminal to grab it from the screen (this normally results in a Postscript file).

- **W**(indowing)

  Permits the division of the plotting surface into panes into which subsequent graphs are plotted. Two factors are requested, the subdivision factor in the horizontal, and in the vertical directions. *e.g.*, 3 and 2 would divide the surface into two rows of three graphs. Setting both factors to 1 restores normal single graph behaviour.

- **+** (Overplot)

  May be used either on its own or be used as a prefix to any object name. Causes the next graph to be plotted over the previous one without re-drawing the axes *etc.*

- **?** (Help)

  The same as H(elp).

- Name of object(s) to plot

  A single object name indicates that the object array is to be plotted along the Y-axis using a default linear scale as the X-axis.

  Two object names separated by a comma indicates that the two arrays are to be plotted against each other.

28:`ech_menu` – Display Full Menu

This option selects the display of the full menu of options available in ECHMENU top-level menu. By default only the utility options and the currently most likely selections will appear on the menu.

29:`ech_system` – Do System Command

This option allows the execution of one or more system level commands without leaving and re-starting the ECHMENU shell task.

The command is prompted for with:

```
- System_$ /''/ >
```

you should then enter a command. If more than one command is required then the `csh` command should be given to initiate a fully independent process.

This process must be terminated by a CTRL-D in order to return to the ECHMENU shell task. This option can be used to perform system commands like `ls` and also FIGARO commands such as IMAGE *etc.* However, ECHOMOP commands which change parameter values will not operate perfectly because the monolith already has the ECHOMOP parameter file open.

System commands can also be entered directly at the Option: prompt by making the first character a `$`. Thus:

```
$ls
```

would execute the system 'directory' command, and return to the main menu.

30:`ech_genflat` – Output Flattened-field

This option was introduced at ECHOMOP version 3.2-0. The flat-field balance factors produced by `ech_ffield` are written to an image which can then be inspected, for example using KAPPA DISPLAY.

Parameters:

- **ECH_RDUCD** - Output spectrum data file.

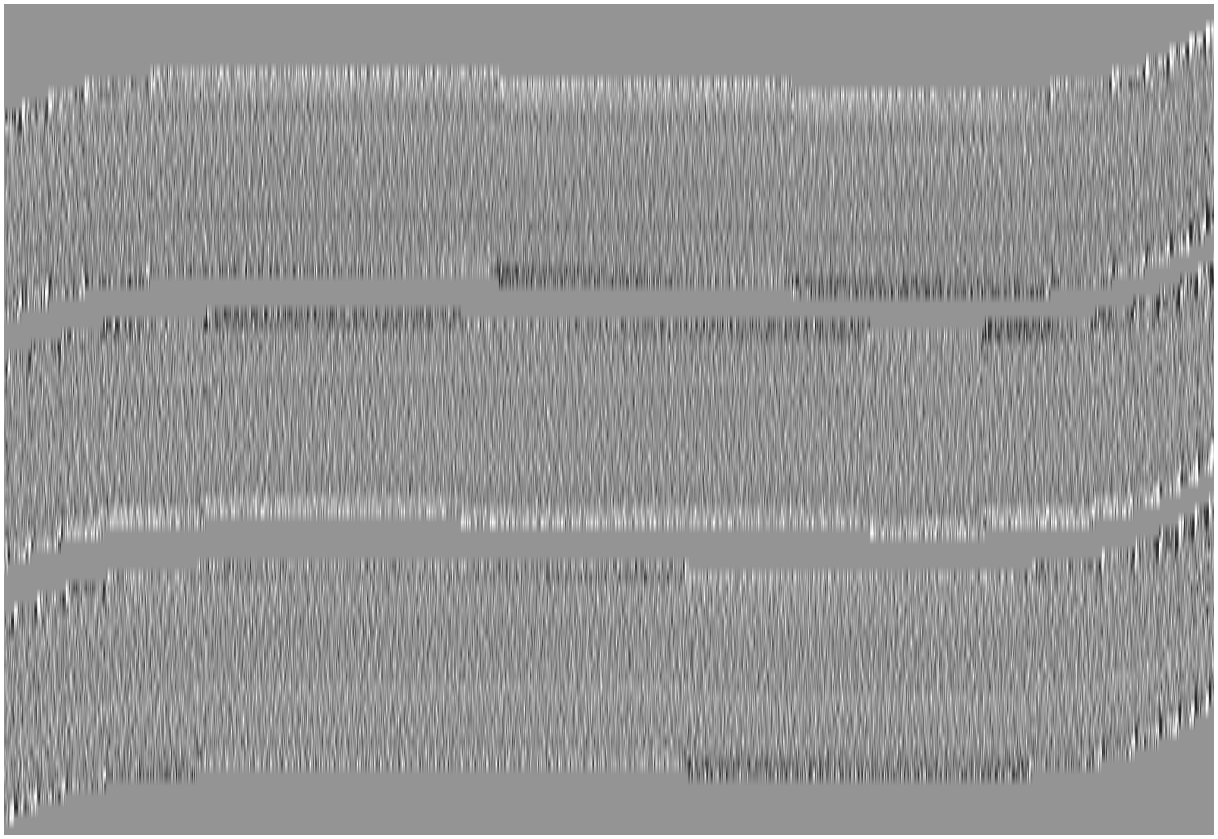


Figure 9: A flattened-field. This was produced using ECHMENU Option 30.

- **TUNE_DIAGNOSE** - YES to log activity to debugging file.
- **TUNE_MAXPOLY** - Maximum coefficients for fits.
- **TUNE_MXSKYPIX** - Maximum number of sky pixels.
- **TUNE_NOFLAT** - YES if no flat-field frame is available.

31:`ech_exit` – Exit ECHMENU

This option exits the ECHMENU shell task. It is also possible to select this option by typing either EXIT, QUIT, a single 'E' or 'Q', or 99, followed by carriage return.

4 PARAMETERS

This is a complete list of all the parameters used by ECHOMOP tasks.

ARC = _CHAR

The names of all reference images to be used for wavelength calibration must be supplied here. The number of images should be equivalent to the number specified for parameter NREF_FRAME. The image names should be separated by commas, for example: reffrm1, reffrm2.

Note that in most circumstances there will only be two reference frames. One taken prior to the object frame, and one subsequently. If you do not wish to provide an arc frame at all then you may reply NONE is response to this prompt or, alternatively set the parameter TUNE_NOARC=YES. If no arc is available then you must provide your own sets of wavelength feature positions, or wavelength scales, **if** wavelength calibration is required.

ARC_TYPE = _CHAR

As of ECHOMOP *v3.3-0, this parameter is ignored and should be removed from any scripts using it. The reference-line database should be specified by the parameter* ECH_FTRDB. *The parameter will be completely removed at the next release of* ECHOMOP.

ASCII_FILE = _CHAR

The name of the ASCII data file to be produced.

The default is echomop_output.txt.

AUTO = _LOGICAL

If AUTO=YES then ECHOMOP will use all the options available to automate the processes it invokes. Currently the following parameters will be affected:

- TRC_INTERACT set to NO.
- TRC_VETO set to NO.
- ID_INTERACTIVE set to NO.
- CR_INTER set to NO.
- 2D_INTERACT set to NO.
- PFL_INTERACT set to NO.

AUTO_ID = _LOGICAL

Specifies that automation is required when arc line identification is being done. Interactively, options are available to allow you to tune the fitted wavelength polynomial, add/delete features *etc.* If AUTO_ID is set to YES then the package will try to identify the arc line features automatically. For orders with sufficient features this will usually be successful ($>$12 lines), and a useful strategy is probably to do an automatic run first, and then manually examine the fitted features to verify their suitability.

BAD_ORDER = _INTEGER

Specifies an order which is to be disabled from further processing.

The default value is zero which is ignored, in case of a mistake.

Orders are disabled by writing the 'bad value' quantity into the first coefficient of the trace polynomial. This ensures that they also appear disabled to other tasks (almost all of which will need the trace polynomials).

To re-enable an order, re-fit a polynomial to its traced path, using ech_fitord/echmenu option 3.

BIN_SIZE = _REAL

Specifies the bin size in wavelength units. This bin size is only used when scrunching to a global wavelength scale. By default the program will calculate an appropriate bin size for you, but this parameter allows you to override this by supplying a non-zero value.

Initial suggested value: 0.0.

BLZFIT = _CHAR

This parameter selects the type of line fit for blaze function fitting.

The recognised functions are POLY (the default), MEDIAN and SPLINE.

Note that the required maximum number of coefficients is different in each case. For POLY the maximum number of coefficients corresponds to the number of degrees plus one, so for a cubic use coeffs=4. For a SPLINE the maximum number of coefficients corresponds to 2x(knots+7), so for 3 knots use coeffs=20. If you are using SPLINE fits then you should ensure that the value of TUNE_MAXPOLY is set accordingly, as well as using an appropriate value for BLZ_NPOLY.

When MEDIAN is selected local medians of X-extent BLZ_NPOLY-pixels are used as the blaze function.

Initial suggested value: POLY.

BLZ_INTERACT = _LOGICAL

Set to YES if the blaze fitting is to be done interactively. Automatic fitting is performed otherwise.

Initial suggested value: NO.

BLZ_NPOLY = _INTEGER

The default number of coefficients to be used when fitting functions to the order blaze.

The type of function to be used is set using the parameter BLZFIT (default is POLY for polynomial).

When BLZFIT=POLY then the number of coefficients should be set to degree of polynomial + 1.

When BLZFIT=SPLINE the number of coefficients should be set to 2 x (number of knots + 7).

When BLZFIT=MEDIAN the number of coefficients should be set to the number of pixels over which a local median value of the blaze function is to be found.

For example, setting BLZ_NPOLY to 4 would permit the fitting of polynomials of the form:

```
constant + ax + bx squared + cx cubed
```

Initial suggested value: 7.

CENTRAL_ONUM = _INTEGER

Parameter is 'hidden'

CENTRAL_WAVE = _REAL

Parameter is 'hidden'

CR_OUTPUT = _CHAR

The name of the output image containing a copy of the original with suspected cosmic-ray pixels replaced either by interpolation, or by flag values.

DECIMG = _CHAR
The name of the raw image in which cosmic-ray-pixel detection is to be performed.

Pixels may be either replaced with an interpolated value, or by a flag value.

DISPLAY = _LOGICAL
If `DISPLAY=YES` is used, then an imaging device is assumed to be available and some programs in the package will overlay plots on the image to help illustrate their results.

For example, the paths of the orders as fitted with polynomials will be overlaid on a part of the traced image.

Initial suggested value: `NO`.

ECH_ECHAR = _CHAR
The name of the file to use to exchange data with the ECHARC program.

ECH_FTRDB = _CHAR
The name of the file containing the reference line list database.

The database is built from a `.ARC` file using the task `ech_ftrdb`.

The database should have the same file name as the `.ARC` file used for its creation. The default is to use `$ARCDIRS/THAR`. Note the use of the `$ARCDIRS` environment variable to specify the search path for arc databases. This should normally be used as it searches in the normal FIGARO order. If you wish to specify a private database then the full pathname may be provided, e.g., `/mydisk/mydir/sub/ANARC`.

If no arc database exists for the `.ARC` list you want to use then use the `ech_ftrdb` task to create a copy. You might like to check to see if anyone local already has the database you need.

Initial suggested value: `'$ARCDIRS/THAR'`.

ECH_RDCTN = _CHAR
The name of the file created to hold the data reduction history.

This file will normally be created by the first ECHOMOP task run. Any valid datafile name can be used.

By default the file is created in the current directory.

ECH_RDUCD = _CHAR
The name of the file to save data to.

ECH_STRUCT = _CHAR
The name of the file containing the definition used to create a reduction database.

EXTRACT_MODE = _CHAR
The type of extraction to perform.

This may be one of the following:

`O` — Optimal (default).
`S` — Simple.
`P` — Profile weighted.

FFIELD = _CHAR

The name of the flat-field frame.

The package uses this frame to calculate balance factors on a per-pixel basis. If a balance frame is already available, then the frame should be supplied instead of a flat field, and the parameter TUNE_PREBAL should be set.

If you do not wish to supply a flat field at all, then you may either reply NONE to this prompt (must be uppercase), or set the parameter TUNE_NOFLAT=YES.

If no flat-field frame is supplied then the balance factors will be set to unity for all pixels.

FLAG = _REAL

If interpolation is not required, the suspected cosmic-ray pixels will be replaced using the supplied flag value.

FLTFIT = _CHAR

The type of fit to be used for flat-field modelling.

The available functions are MEAN (the default), POLY, SPLINE, MEDIAN, SLOPE and SMOOTH.

All modes except POLY and SPLINE use local neighbourhoods delineated using a sample size (in pixels) defined by TUNE_FFLSMP.

Note that the required maximum number of coefficients for POLY and SPLINE are different. For POLY, the maximum number of coefficients corresponds to the number of degrees plus one, so for a cubic use coeffs=4. For a SPLINE the maximum number of coefficients corresponds to 2x(knots+7), so for 3 knots use coeffs=20. If you are using SPLINE fits then you should ensure that the value of TUNE_MAXPOLY is set accordingly, as well as using an appropriate value for TUNE_FFNXPLY and/or TUNE_FFNYPLY.

Initial suggested value: MEAN.

FRAME_CHECK = _LOGICAL

Set FRAME_CHECK if the processed frames are to be checked for bad rows and columns.

The trace frame is checked, and the other frames are assumed to have identical bad row and/or column features.

HARD = _CHAR

If a value for HARD is specified, the plot is written to a file in the format defined by the HARD parameter.

For example, ps_l would generate Landscape format PostScript files.

HELIO_COR = _REAL

The heliocentric correction is applied to the output wavelength-scale of any results files.

Initial suggested value: 0.0.

HI_WAVE = _REAL

The identification of lines proceeds faster for smaller search ranges. Specify the highest wavelength covered by the spectrum. The units in which the wavelength is expressed must be the same as those used in the .ARC line list file. This will normally be Angstroms but may be changed by supplying an appropriate .ARC file.

IDX_NREF_FRAME = _INTEGER
Specifies the number of the arc frame to be processed.

IDX_ parameters provide a method of indexing reduction database arrays by order number and frame number.

A non-zero value for IDX_NREF_FRAME causes processing of the selected frame only.

IDX_NREF_FRAME=0 causes automatic looping through all values for the 'index', *i.e.*, processing of all arc frames.

IDX_NUM_ORDERS = _INTEGER
Specifies which order is to be processed.

The IDX_ parameters provide a method of indexing reduction database arrays by order number and frame number.

A non-zero value for IDX_NREF_ORDERS causes processing of the selected order only.

IDX_NREF_ORDERS=0 causes automatic looping through all values for the 'index', *i.e.*, processing of all orders.

ID_INTERACTIVE = _LOGICAL
Specifies that interactive arc-line identification is required.

Many options are available in this mode to allow the tuning of the fitted wavelength polynomial, addition and deletion of features *etc.*

If ID_INTERACTIVE is not set then the ECHOMOP will try to identify arc-line features automatically. For orders with enough features (more than 12 lines) this will usually be successful.

A useful strategy is often to do an automatic run first, and then manually examine the fitted features to verify their suitability.

INDIRECT = _CHAR
Internal parameter used to indirect reduction objects to be copied from other reduction databases instead of being calculated (the < function in ECHOMOP).

INPTIM = _CHAR
The name of the file holding the frame to be extracted.

This will usually be the Object frame, however the Standard-Star frame may also be extracted. In cases where there are multiple frames of the same Object, then they may each be extracted by re-running this task and specifying a different frame each time (assuming the multiple data frames are perfectly registered).

LOW_WAVE = _REAL
The identification of lines proceeds faster for smaller search ranges. Specify the lowest wavelength covered by the spectrum. The units in which the wavelength is expressed must be the same as those used in the .ARC line list file. This will normally be Angstroms but may be changed by supplying an appropriate .ARC file.

MAX_DISPERSION = _REAL
The range of dispersions to be searched by the automatic line identifier is specified in terms of minimum and maximum Wavelength Unit per-pixel. Care should be taken to ensure that the specified range is wide enough. If in doubt use a larger range than is

probably needed. For example, if the actual dispersion is expected to be approx 0.5, then a range of 0.2 to 0.8 would not be inappropriate, whereas a range of 0.4 to 0.6 could possibly be too restrictive. The normal wavelength units will be Angstroms, but alternative units may be used by providing an appropriate .ARC line list.

MIN_DISPERSION = _REAL

The range of dispersions to be searched by the automatic line identifier is specified in terms of minimum and maximum Wavelength Unit per-pixel. Care should be taken to ensure that the specified range is wide enough. If in doubt use a larger range than is probably needed. For example if the actual dispersion is expected to be approx 0.5, then a range of 0.2 to 0.8 would not be inappropriate, whereas a range of 0.4 to 0.6 could possibly be too restrictive. The normal wavelength units will be Angstroms, but alternative units may be used by providing an appropriate .ARC line list.

NREF_FRAME = _INTEGER

The number of wavelength-reference frames to be used.

These will usually be arc spectra. The default is a single frame, but it is possible to use two frames which bracket the object exposure in time. In this case it is assumed that the shift between exposures has been due to smooth variation, an assumption which it is usually not possible to verify.

Note: the names of all frames must be supplied in response to the ARC prompt when locating the reference features. For example:

```
ARC=<frame1>,<frame2>
```

NUM_ORDERS = _INTEGER

Specifies the number of orders present in the frames to be reduced.

OBJFIT = _CHAR

This parameter selects the type of curve fit for object profiling.

The available functions are POLY (the default) and SPLINE.

Note that the required maximum number of coefficients is different in each case. For POLY the maximum number of coefficients corresponds to the number of degrees plus one, so for a cubic use coeffs=4. For a SPLINE the maximum number of coefficients corresponds to 2x(knots+7), so for 3 knots use coeffs=20. If you are using SPLINE fits then you should ensure that the value of TUNE_MAXPOLY is set accordingly, as well as using an appropriate value for TRC_NPOLY.

Initial suggested value: POLY.

OBJ_SKY_GAP = _INTEGER

Parameter is 'hidden'

OUTPUT_IMAGE = _CHAR

Specifies an output frame (container file name). The frame will contain a full 2-D image array and associated errors if appropriate.

PFL_INTERACT = _LOGICAL

When interactive profiling is selected you can use a graphics cursor to specify the required attributes of each pixel in the spatial profile. The characteristics may be globally applicable, or specified independently for each order. In addition, any polynomial-profile fitting is also

performed interactively, with the degree of polynomial and polynomial type adjustable. The fits and residuals at each sub-sample position can be reviewed.

PFL_MODE = _CHAR
The profiling mode is used to specify which profiles are to be calculated/edited.

The options are:

`D` — Calculate/edit Dekker limits.
`O` — Calculate/edit Object profile.
`S` — Calculate Flux standard profile.
`A` — All, do each of the above in turn.

Normally this parameter would be set to A, and the program run automatically. The profile and calculated limits examined as the program ran, and then re-run interactively with one of the above options.

PHOTON_TO_ADU = _REAL
Some detectors produce many electrons (counts) for each incident photon, or alternatively, one count for many photons. In order to be able to calculate using photon-events it is necessary to take account of this factor. The `PHOTON_TO_ADU` parameter should be set to the number of incident photons corresponding to each count recorded by the detector.

RBNOBJ = _CHAR
The rebinned image is used to store the distortion-corrected version of the data.

It is only needed in cases of severe geometric distortion and is organised such that pixel columns correspond to linear wavelength bins in a given order.

READOUT_NOISE = _REAL
Some electronic detectors produce a significant amount of noise during the readout process. The older CCDs are often thus afflicted. If a non-zero value is supplied then it will be used to adjust the variances during extraction to take this error into account.

RESULT_FORMAT = _CHAR
Selects the format of the output file.

Options are:

- NDF — Output to the NDF specified by parameter `ECH_RDCTN`
- ASCII — Output to the specified NDF and to the file specified by `ASCII_FILE`
- STACK — Output to the DIPSO stack specified by parameter `STACK`

RESULT_TYPE = _CHAR
Selects the type of data to be saved.

The following types are recognised:

— `EXTOBJ`, the unscrunched extracted object orders.
— `EXTARC`, the unscrunched extracted arc orders.
— `SCROBJ`, the scrunched extracted object orders.
— `SCRARC`, the scrunched extracted arc orders.
— `OSPECT`, the merged object spectrum.

The default is EXTOBJ.

SCRUNCH_TYPE = _CHAR

The type of spectrum to be scrunched. This may be one of the following:

— OBJ, the object.
— STAR, the reference star.
— ARC, the arc lamp.

SET_WSCALE = _LOGICAL

This parameter determines the overall type of scrunching to use when scrunching extracted orders. Set to YES if the current frames wavelength range is to be used (all orders) to generate a full range wavelength scale to scrunch into; *i.e.*, all orders are scrunched into the same bin size, but the no of X-axis bins in each order will vary. Set to NO if each order is to be scrunched to its own scale. This retains the X-dimension of the orders, but each order will be scrunched into a slightly different bin size.

SKYFIT = _CHAR

The type of fit to be used for sky-background modelling.

The available functions are MEAN (the default), POLY and SPLINE.

Note that the required maximum number of coefficients for POLY and SPLINE are different. For POLY, the maximum number of coefficients corresponds to the number of degrees plus one, so for a cubic use coeffs=4. For a SPLINE the maximum number of coefficients corresponds to (knots+7)*2, so for 3 knots use coeffs=20. If you are using SPLINE fits then you should ensure that the value of TUNE_MAXPOLY is set accordingly, as well as using an appropriate value for TUNE_SKYPOLY and/or TUNE_SKYXPLY.

Initial suggested value: MEAN.

SLITIM = _CHAR

The frame to be used for determining the limits of the dekker.

This should be a frame which clearly delineates the inter-order gap and the full extent of the slit. In general the arc-lamp frame is used, but a flat-field may also work.

SOFT = _CHAR

If SOFT is specified, the plot is written to the device defined as the current softcopy device.

STACK = _CHAR

Name of the DIPSO stack to save data to.

The '_STK.sdf' part of the filename should not be given.

The default value is ECHOMOP.

START_WAVE = _REAL

This parameter is only used when a user-specified wavelength scale is being generated (BIN_SIZE>0). It specifies the start wavelength for the re-binned wavelength scale. The default is to calculate the start wavelength from the fitted wavelength scale if no value is supplied using this parameter.

Initial suggested value: 0.0.

TRACE_MODE = _CHAR

The type of tracing method to be used.

The following options are available:

U — indicates (ICUR) coordinates are supplied.

R — selects re-tracing using old trace as baseline.

T — indicates that a triangle filter is to be applied before centre location.

A — indicates that in cases of failure, the program is to automatically fall-back to the next-least demanding centering mode.

Plus one of the following mandatory selections:

B — basic, simple-minded centre of gravity.

E — locate profile edges and interpolate.

C — find centroid of profile.

G — fit a Gaussian across profile.

For example:

TAG — would fit Gaussians to triangle-filtered data and automatically fall-back to try centroiding for problem samples.

B — would do simple centre-of-gravity calculations on the raw profile, with no automatic fallback being used.

The default mode is C which provides a reasonable balance of probability-of-a-good-trace vs. time-taken. The order of specification is significant and must obey the following rules:

— Optional U or R followed by
— Optional T and/or A followed by
— One of B, E, C or G.

TRACIM = _CHAR

The name of the file containing a frame to be used for the tracing of the orders. Tracing may be attempted using a wide variety of data frame types. If the object frame has a good signal-to-noise ratio then this should be used. In cases where the object is faint, or exhibits large-scale absorption features, then either the flat-field or flux-calibration reference frame should be used.

TRCFIT = _CHAR

This parameter selects the type of line fit for order tracing.

The available functions are POLY (the default) and SPLINE.

Note that the required maximum number of coefficients is different in each case. For POLY the maximum number of coefficients corresponds to the number of degrees plus one, so for a cubic use coeffs=4. For a SPLINE the maximum number of coefficients corresponds to 2x(knots+7), so for 3 knots use coeffs=20. If you are using SPLINE fits then you should ensure that the value of TUNE_MAXPOLY is set accordingly, as well as using an appropriate value for TRC_NPOLY.

Initial suggested value: POLY.

TRC_INTERACT = _LOGICAL

Determines whether order fitting/clipping is interactive.

Set for interactive order fitting/clipping.

If not set, order fitting/clipping proceeds automatically. When automatic fitting is engaged there are two parameters which may be adjusted to optimise the performance of the program. These are: TUNE_CLPMXDEV, which defines the clipping point, and TUNE_CLPBY, which is the number of points to clip in each iteration.

The default value is NO.

TRC_NPOLY = _INTEGER

The default number of coefficients to be used when fitting functions to the order trace.

The type of function to be used is set using the parameter TRCFIT. (Default value is POLY for polynomial.)

When TRCFIT=POLY then the number of coefficients should be set to degree of polynomial + 1.

When TRCFIT=SPLINE the number of coefficients should be set to 2x(number of knots + 7).

For example, setting TRC_NPOLY to 4 would permit the fitting of polynomials of the form:

`constant + ax + bx squared + cx cubed` Initial suggested value: 4.

TRC_VETO = _LOGICAL

Set to YES if the order consistency is to be checked interactively. Automatic checking is performed otherwise.

When automatic checking is engaged there are two parameters which may be adjusted to optimise the performance of the program.

These are TUNE_CLPMXDEV which defines the clipping point, and TUNE_CLPBY which is the number of points to clip in each iteration.

When TRC_VETO is set, the option to reject a re-fitted order at each iteration is available.

It is strongly recommended that this option is not used until you have a clear understanding of the algorithm used by the consistency checker, as it is possible to 'drive' the fit off by injudicious rejection of the re-fits proposed by the program.

TUNE_AAACODE = _INTEGER

Used only when the program is making use of the Astronomy & Astrophysics Abstracts system of information categorisation (TUNE_USEAAA=YES). This system consists of a set of numbers corresponding to topics in Astronomy. TUNE_AAACODE is used to supply an AAA standard information category for the data being reduced. These parameters are only used in conjunction with the archiving facility, when TUNE_ARCHIVE=YES.

Initial suggested value: 1.

TUNE_AIRTOVAC = _LOGICAL

Determines whether the wavelength scales in output datafiles are converted from air to to vacuum wavelengths.

If set to YES, then the vacuum wavelengths are converted using the procedure:

$$
\begin{aligned}
wavm &= (\mathrm{old}_{\mathrm{wavelength}} \times 10^{-4})^{-2} \\
sum &= 64.328 + 29498.1/(146.0 - wavm) + 255.4/(41.0 - wavm) \\
\mathrm{new}_{\mathrm{wavelength}} &= \mathrm{old}_{\mathrm{wavelength}} \times (1.0 + 10^{-4} \times sum)
\end{aligned}
$$

Only the wavelength scales in the output file are effected; the reduction database wavelength scales remain unchanged.

Initial suggested value: NO.

TUNE_ARCHIVE = _LOGICAL

Set to YES if automatic archiving of results files is to be performed. The archive is maintained by a central site and will support query facilities. A 2-year data protection scheme is applied, so your spectra may not be accessed by other parties until 2 years after its inclusion in the archive. See HELP ECHOMOP ARCHIVING for a more detailed discussion.

The default value is NO.

TUNE_AUTLOC = _LOGICAL

Set this parameter if the positions of the orders are to be determined automatically.

When not set, the program allows interactive selection of the order centres on a graphic display. This is useful if you wish to select a subset of the available orders in the frame, or to review the order centres found by the program.

The default value is NO.

TUNE_AUTOMATE = _CHAR

This hidden parameter allows multiple options to be automatically invoked upon startup of the monolithic task.

Any options required must be supplied on the command line or the you will be prompted as usual. For example:

```
TUNE_AUTOMATE=1,2,3,EXIT
```

would run the first 3 processing options and then exit.

If EXIT is not given as an option, ECHOMOP will start the menu task ECHMENU when the list of options specified by TUNE_AUTOMATE has been exhausted.

TUNE_BATCH = _LOGICAL

Set to YES for batch-mode operation.

TUNE_BLZRSET = _LOGICAL

This parameter controls whether or not the blaze function is to be reset to unity after being applied. The default is to do so, thus protecting against accidental re-application of the blaze function to an extracted spectrum. In some cases it may be required to keep the blaze function and apply it to a number of extracted object spectra. In this case the TUNE_BLZRSET=NO setting should be used.

The default value is YES.

Initial suggested value: YES.

TUNE_CLONE = _CHAR

The reduction database from which to clone information.

The reduction database should have been created by ECHOMOP previously. Objects which would normally be calculated by ECHOMOP will instead be copied from the named file. This allows the cloning of sections of a reduction amongst multi-frame datasets.

TUNE_CLPBY = _INTEGER

This parameter sets the number of points automatically clipped per order-trace fit/clip iteration.

The TUNE_CLPBY most-deviant points are clipped before a re-fit is attempted, this process is then repeated until either: all points have deviations of less than TUNE_CLPMXDEV from the polynomial; or, so many points have been clipped that a meaningful fit can no longer be attempted. In the latter case the order will be automatically disabled. You may manually re-fit the trace if the order is still to be processed.

The default value is 1.

TUNE_CLPMXDEV = _REAL

The maximum deviation used when automatically clipping traced points to fit order tracing polynomials.

It is expressed in units of pixels and will usually be a fraction of a pixel.

Any traced point which is further than this limit from the fitted polynomial (for that X-coordinate) is liable to be clipped. Automatic clipping only stops when either all points have deviations of less than TUNE_CLPMXDEV from the polynomial; or when so many points have been clipped that a meaningful fit can no longer be attempted. In this latter case, the order will be automatically disabled. You may re-fit manually if the order is still to be processed.

The default value is 0.5.

TUNE_CNSDEV = _REAL

Specifies the threshold for deciding whether the values generated by the order specific polynomial and the all-order polynomial, are consistent at a particular increment. TUNE_CNSDEV is the maximum difference between the two (in pixels) which is accepted as consistent.

The default value is 0.5 pixels.

TUNE_CRCLEAN = _LOGICAL

When set to YES then the optimal extraction will attempt to identify cosmic rays by examining the local variance. This option will only function when full optimal extraction has been selected using EXTRACT_MODE=O. It will not normally be used as more flexible tasks for cosmic-ray removal are provided as separate programs. See HELP ECH_DECOSMIC.

TUNE_CRINTER = _LOGICAL

Determines how cosmic-ray-contaminated pixels are to be treated. (Only used when cosmic-ray checking is enabled by TUNE_CRTRC=YES.)

When set, cosmic-ray pixels are replaced in the output image by interpolated values.

When not set, cosmic-ray pixels are marked with a flag value.

The default is NO.

TUNE_CRMAX = _INTEGER
The number of pixels expected to be contaminated by cosmic rays. (Only used when cosmic-ray checking is enabled by TUNE_CRTRC=YES.)

You may use an estimate, or else a value can be generated by an automatic modeller.

If no estimate is available, then zero should be used. The program will then display a histogram of the ratio of original divided-by median image and allow manual selection of a clipping point. The number of pixels to be clipped (flagged as cosmic rays) will then be shown.

TUNE_CRTRC = _LOGICAL
Set if the trace frame is to be checked for possible cosmic-ray contamination.

This will not usually be necessary and is provided for the case where only a faint, contaminated object frame is available for order tracing.

Initial suggested value: NO.

TUNE_CRXBOX = _INTEGER
X-dimension of the area used to calculate median values.

TUNE_CRYBOX = _INTEGER
Y-dimension of the area used to calculate median values.

TUNE_DB_SCOPE = _INTEGER
This parameter sets the number of nearest-neighbours to take account of when constructing/using a reference feature database (FDB). It will not normally require adjusting unless a very odd line list is being used.

TUNE_DEKABV = _INTEGER
This parameter sets the upper extraction limit for the dekker in the spatial direction.

A non-zero value for this parameter will over-ride the automatic dekker extent calculated by ech_spatial.

TUNE_DEKABV can be used when the dekker extent and position relative to the order traces are well known. Typically, it will be used when the first of a set of similar frames has been manually extracted and automation of the extraction of the remaining frames is required.

Initial suggested value: 0.

TUNE_DEKBLW = _INTEGER
This parameter sets the lower extraction limit for the dekker in the spatial direction.

A non-zero value for this parameter will over-ride the automatic dekker extent calculated by ech_spatial.

TUNE_DEKBLW can be used when the dekker extent and position relative to the order traces are well known. Typically, it will be used when the first of a set of similar frames has been manually extracted and automation of the extraction of the remaining frames is required.

Initial suggested value: 0.

TUNE_DEKTHR = _REAL
The threshold level for automatic determination of the dekker extent.

This is the percentage (of maximum intensity) below which the spatial profile is to fall before being classified as outside the dekker. In general an Arc frame will be used to establish the dekker limits, and a value of 80% will suffice.

The default value is `0.8` (*i.e.* 80%).

TUNE_DIAGNOSE = _LOGICAL

This parameter should be set only for a short period when a problem has occurred (and can be reproduced). It enables the production of a log file containing detailed information of use to the maintainer of ECHOMOP. This log is stored in the file `echomop_diagnostics.log`.

In the event of a problem mail this file, with details of the problem, to the Starlink Software Librarian at: `starlink@jiscmail.ac.uk`.

TUNE_DPRBTHR = _REAL

The maximum difference between observed and theoretical (based on observed mean, sigma) CDF's is calculated. The Kolmogorov-Smirnov test is used to evaluate the probability that the measured deviation is a chance occurrence (*i.e.* that the two distributions are actually the same). In general any significant probability indicates that the two distributions compare very well. When the K-S probability exceeds the `TUNE_DPRBTHR` threshold, clipping stops and the fit is accepted.

The default value is `0.9` (*i.e.* 90%).

TUNE_DSGMTHR = _REAL

The sigma clipping threshold for deciding which pixels to clip at each iteration of CDF matching, is set by this parameter. Normally clipping should be set to 2 to 3 sigma. At each iteration the deviation of each pixel from its predicted value is evaluated in terms of the mean and sigma for all pixels (per-order basis).

TUNE_FCHECK = _LOGICAL

Set this parameter if the processed frames are to be checked for 'bad' row and/or columns.

The trace frame is checked and other frames are then assumed to have identical 'bad' features.

TUNE_FFINTER = _LOGICAL

Set this parameter if the degree of polynomial used to model the flat field is to be under interactive control.

TUNE_FFLMED = _LOGICAL

Whether the median is to be used when calculating the behaviour of the flat field using only local pixels.

Set if the median value is to be used.

This applies only when `TUNE_FFNXPLY=1` (no polynomial model), in which case the local median or mean is taken over the nearest `TUNE_FFLSMP` pixels. This is useful for cases where there is significant small-scale structure in the flat field.

Initial suggested value: `NO`.

TUNE_FFLSMP = _INTEGER

The number of samples per point to use when the flat-field model is based on the local behaviour only.

This parameter is only used when the X-polynomial degree, TUNE_FFNXPLY=1, which implies a constant flat field.

The parameter TUNE_FFLMED controls whether a local mean or median value is used.

Initial suggested value: 10.

TUNE_FFNXPLY = _INTEGER

Specifies the number of polynomial coefficients in X for the flat-field model, and the re-fitted model.

A model of the flat field is made by fitting a polynomial at each X- and Y-increment. These polynomials are then used to calculate the balance factors at the 'Object' pixel positions.

Note: If the parameter is set to zero then no modelling in the X-direction will be performed. If the parameter is set to 1, then no polynomials are used (they would be constant), but the balance factors are calculated using the local mean value based on a TUNE_FFLSMP pixel sample. This option should be used when under-sampling *etc.* is preventing the fitting of polynomials at all.

The default value is 1.

TUNE_FFNXREJ = _INTEGER

Specifies the number of clip/reject/remodel cycles in X (dispersion direction) flat-field modelling.

The model of the flat field is made by fitting a polynomial at each X- and Y-increment. These polynomials are then used to calculate the balance factors at the 'Object' pixel positions.

TUNE_FFNYPLY = _INTEGER

Specifies the number of polynomial coefficients in Y for the flat-field model, and the re-fitted model.

The model of the flat field is made by fitting a polynomial at each X- and Y-increment. These polynomials are then used to calculate the balance factors at the 'Object' pixel positions.

The default value is 0.

TUNE_FFNYREJ = _INTEGER

Specifies the number of clip/reject/remodel cycles in Y (dispersion direction) flat-field modelling.

The model of the flat field is made by fitting a polynomial at each X- and Y-increment. These polynomials are then used to calculate the balance factors at the 'Object' pixel positions.

TUNE_FFSUBSMP = _LOGICAL

Whether subsampling is to be used when doing X-polynomial fits.

Set if subsampling is to be done.

This parameter is only used when polynomial fitting in the X-direction is enabled (by setting TUNE_FFNXPLY>1) and causes the program to subsample (10 samples per-pixel) the spatial profile during fitting. This is useful when the balance factors 'outside' the dekker, or at the very edge of the dekker (where the intensity changes quickly from pixel-to-pixel) are needed.

Initial suggested value: NO.

TUNE_FFTHRESH = _REAL

Specifies the rejection threshold for the flat field.

Any pixels which deviate by more than TUNE_FFTHRESH sigma from the fitted polynomial model will be clipped and the model re-fitted.

The polynomial model of the flat field is made by fitting a polynomial at each X- and Y-increment. These polynomials are then used to calculate the balance factors at 'Object' pixel positions.

The default value is 10.0 sigma.

TUNE_FIBRES = _LOGICAL

Set to YES if a multi-fibre object frame.

TUNE_FINCPLY = _INTEGER

The number of polynomial coefficients to be used for fitting each individual increment. Increments consist of all pixels at a specific distance above/ below the order trace. Polynomials are fitted through each increment in turn, object or sky.

TUNE_FLUX = _LOGICAL

YES if the program is to conserve flux. If not set, it will maintain the mean height of the data.

TUNE_HELIO = _INTEGER

Parameter is 'hidden'

TUNE_IDINMN = _INTEGER

Specifies the minimum number of features to be used when generating ratios for initial identification. In general, a good solution can be found using only the strongest 8-16 features. The program slowly increases the number of features it uses until an adequate solution is found. In cases where the orders are distorted it may be advantageous to increase the minimum number of features to cover the majority of located features.

Initial suggested value: 8.

TUNE_IDINMX = _INTEGER

Specifies the maximum number of features to be used when generating ratios for initial identification. In general, a good solution can be found using only the strongest 8-16 features. The program slowly increases the number of features it uses until an adequate solution is found. However, there may be large numbers of weak features present which are not in the reference database. This parameter allows the setting of an absolute maximum on the number of features (per order) which are to be considered.

Initial suggested value: 30.

TUNE_IDMDLT = _INTEGER

Specifies the maximum number of neighbouring features (on EACH side) to examine when generating ratios for matching. Increasing this will lead to exponential increases in CPU time, so it should be used with caution when all else fails.

The default value is 6.

TUNE_IDMXDIF = _REAL

Specifies the maximum difference between the ratios derived from observed features, and those in the database with which a match is attempted. The difference is evaluated by calculating

giving a fractional measure of the difference. For example: difference=0.01 indicates that the two quantities differ by approximately 1% of their average magnitude. Values much larger than 0.1 are likely to generate a lot of coincidence matches; values less than 0.01 may well miss 'good' matches in less-than-ideal data.

The default value is `0.03`.

TUNE_IDSDLT = _INTEGER

Specifies the starting number of neighbouring features (on **each** side) to examine when generating ratios for matching. Increasing this will lead to exponential increases in CPU time, so it should be used with caution when all else fails.

Higher values are tried automatically by the program if no solution can be found. The number of neighbours considered is increased monotonically until it reaches the maximum of `TUNE_IDMDLT`, when the program gives up.

The default value is `3`.

TUNE_IDSTRNG = _REAL

Specifies the minimum strength of features to be used for initial identification. It is specified by relating the strength of a feature to that of the strongest feature present in that order. Any features of strength max/`TUNE_IDSTRNG` or greater, will be included in the initial set.

The default value is `10.0`, thus all features which are at least one tenth as strong of the strongest feature will be eligible for identification.

TUNE_INTR = _LOGICAL

Selects linear interpolation when re-binning.

TUNE_IUE = _INTEGER

Non-zero if IUE type data frame.

TUNE_LOG = _LOGICAL

Used only when `TUNE_SCRMODE=1`, and indicates that the output wavelength scale has bins which vary on a LOG scale.

TUNE_MAX2DPLY = _INTEGER

Maximum number of coefficients for 2-D fits.

TUNE_MAX2DPNTS = _INTEGER

Maximum number of data points to be used in 2-D fits.

TUNE_MAXLINES = _INTEGER

Maximum number of arc-line slots.

TUNE_MAXPOLY = _INTEGER

The maximum number of polynomial coefficients to be allowed for the duration of this reduction.

For POLY trace fits, the order of the polynomials used for each fit may be varied up to this limit, but may not be set to higher values.

For SPLINE trace fits this is the maximum number of fit parameters which can be used. A high vale may be needed. The number of parameters required is 2x(knots+7).

The default value is 50, which should be acceptable in most cases.

TUNE_MAXRFLN = _INTEGER
Set to the maximum number of reference lines per order which are allowed. This parameter is used to dimension objects in the reduction database and should not normally need to be altered. You may feel tempted to lower it to avoid the seeming profusion of 'false' line candidates produced by ech_linloc. This is not a good idea if the automatic line-identification mode is being used as it requires the presence of faint feature positions for reliable operation.

TUNE_MERGE = _LOGICAL
Set to YES if the scrunched object spectrum is to be merged into the 1-D spectrum. The default is to zero the 1-D spectrum before output. Setting TUNE_MERGE=YES allows the spectra from multiple frames to be co-added into the same output spectrum using a weighted merge.

The algorithm is the same as used in the FIGARO ECHMERGE program.

Initial suggested value: NO

TUNE_MINCR = _REAL
Specifies the threshold level for deciding whether pixels located as 'possibly cosmic-ray-contaminated' should be flagged as such.

Only those pixels with intensities greater than TUNE_MINCR will be flagged. This provides a limit for the case where a cosmic-ray identifier becomes over enthusiastic due to the nature of the frame.

Normally this parameter should be set to zero, and therefore, have no effect.

Initial suggested value: 0.0.

TUNE_MRGMAXX = _INTEGER
Number of rhs pixels to ignore.

TUNE_MRGMINX = _INTEGER
Number of lhs pixels to ignore.

TUNE_MRGWGHT = _CHAR
Type of weighting to use for merge.

TUNE_MXBADSMP = _INTEGER
The maximum number of consecutive bad samples permitted before abandoning an order trace.

The default value is 10.

TUNE_MXSKYPIX = _INTEGER
The maximum number of pixels allowed in the spatial direction for each order.

This parameter is used to dimension objects in the reduction database and should only need to be altered when dealing with objects where the spatial extent is large.

Note that increasing this parameter leads to a substantial increase in the size of a reduction database, so it should be set to as small a value as sensible for your data.

TUNE_MXSMP = _INTEGER

The maximum number of independent X-samples to use when tracing an order.

If the X-dimension is less then 500 every X-increment will be sampled.

The default value is `500`.

TUNE_NOARC = _LOGICAL

Set to `YES` if there is no arc frame to use for wavelength calibration. This may also be specified by replying `NONE` when prompted for the name of the arc frame.

Initial suggested value: `NO`.

TUNE_NOFLAT = _LOGICAL

Set this parameter if there is no flat-field frame to use for balance-factor calculation.

This may also be setting the parameter `FFIELD` to `NONE` when prompted for the name of the flat-field frame.

In the absence of a flat-field frame, all balance factors are set to unity.

Initial suggested value: `NO`.

TUNE_OBJABV = _INTEGER

This parameter sets the upper extraction limit for the object in the spatial direction.

A non-zero value for this parameter will over-ride the automatic object extent calculated by `ech_spatial`.

`TUNE_OBJABV` can be used when the object extent and position relative to the order traces are well known. Typically, it will be used when the first of a set of similar frames has been manually extracted and automation of the extraction of the remaining frames is required.

Initial suggested value: `0`.

TUNE_OBJBLW = _INTEGER

This parameter sets the lower extraction limit for the object in the spatial direction.

A non-zero value for this parameter will over-ride the automatic object extent calculated by `ech_spatial`.

`TUNE_OBJBLW` can be used when the object extent and position relative to the order traces is well known. Typically, it will be used when the first of a set of similar frames has been manually extracted and automation of the extraction of the remaining frames is required.

Initial suggested value: `0`.

TUNE_OBJPOLY = _INTEGER

Specifies the degree of polynomial to use when trying to fit to the object intensity at any single X/lambda increment. The normal mode of operation for extraction is to form a model of the profile by averaging subsamples from all orders. The polynomial model (per increment) should only be used when the profile varies markedly **along** an order due to atmospheric or instrumental peculiarities.

The default value is `0`.

TUNE_OBJREJ = _INTEGER

Specifies the number of reject cycles for which object profile samples will be clipped, and the model re-fitted.

The polynomial model of the object's spatial profile is made by fitting a polynomial at each X-sample. These polynomials are then used to estimate the profile at each spatial sample of the object.

TUNE_OBJRTHR = _REAL

Specifies the reject-threshold for object pixels.

Any pixels which deviate by more than TUNE_OBJRTHR sigma from the fitted polynomial model, will be clipped, and the model re-fitted. The polynomial model of the object profile is made by fitting a polynomial at each X-increment. These polynomials are then used to estimate the signal at each spatial sample of the object.

Initial suggested value: 5.0.

TUNE_PAGE = _INTEGER

This parameter specifies the number of lines on the terminal screen.

When set to a non-zero value, after each TUNE_PAGE lines of non-menu output the program will pause and wait for a key to be hit before continuing. A variety of options are available at this point:

D,d,uparrow,KP8,p,P,b,B	Scroll back one line.
U,u,downarrow,KP2,f,F,n,N	Scroll forward a line.
space	Resume output.
return	Output next line only.
pageup	Previous full screen.
pagedown	Next full screen.

The program maintains a buffer of lines output to the screen; you may move around using the keys mentioned above. Only the space or return keys will resume regular program output. If the parameter value is not positive then no paging is done.

Initial suggested value: 0.

TUNE_PARTORD = _LOGICAL

Set this parameter if the program is to make use of order slope and inter-order distance data to look for partial orders at the top and bottom of the trace frame.

The default value is NO.

TUNE_PFLSSAMP = _INTEGER

The maximum number of subsamples used when determining the subsampling rate to be used for the generation of the averaged profile.

This parameter should only need to be changed when processing frames with a very large number of pixels sampling the object in the spatial direction.

The default value is 301 subsamples, which will normally be more than is needed.

TUNE_PFSAMP = _INTEGER

The maximum number of profile subsamples.

The default value is `101`.

TUNE_PREBAL = _LOGICAL

Set this parameter if the flat-field frame supplied has already been pre-processed to contain balance factors for the pixel-to-pixel variations.

If set, the modelling will not take place and the balance factors will simply be copied from the supplied frame. This should be used for cases where the polynomial model cannot generate appropriate values.

Note: The removal of the blaze function is a separate operation and will require the use of the original raw flat-field frame (or equivalent).

TUNE_QUAD = _LOGICAL

Selects quadratic interpolation when re-binning. This is usually more accurate than linear interpolation.

The default value is `YES`.

TUNE_QUICK = _LOGICAL

Set if data are to be processed in quick-look mode.

This enables fast reduction by automatically tailoring parameters. In particular: all fits are clipped automatically; no flat fielding is performed; and a simple summed extraction is used.

Prior to quick-look extractions the object extent should be specified by the parameters `TUNE_OBJBLW` and `TUNE_OBJABV`.

Initial suggested value: `NO`.

TUNE_REPORT = _CHAR

This parameter is used to control the variety of reporting output by the program.

It will usually be left blank to indicate default reporting to the terminal/batch log file. Options include:

F Specifies full reporting of task/module entry.

L Specifies duplicate log of reports to disk file

(`ech_report.log` in the current directory).

B Brief reporting, task starts only.

S Silent, no terminal reports at all.

E Error reports to override other setting (BS).

P Print log file on completion (only with L).

Specifiers may be concatenated in any order, thus

```
BELP
```

would specify minimal reporting, except for task startup and error reports. Reports to be copied to a log file and printed upon completion.

Initial suggested value: `' '`.

TUNE_REVCHK = _LOGICAL

Normally the arc line identification task assumes that wavelength increases from left to right (increasing X-coordinate). If you are in the unfortunate position of not being sure that this is the case then setting this parameter will cause the program to check both possible orientations. `TUNE_REVCHK=YES` will take twice as much time as `TUNE_REVCHK=NO`.

The default value is `NO`.

TUNE_RFLNTHR = _REAL

Specifies the degree to which arc line features are required to exceed the continuum intensity before they are considered as valid features. The default value of 1.25 is deliberately set low to allow the automatic line identifier access to very faint features. This parameter should not need adjusting unless the arc spectrum has a difficult continuum or other special problems associated with it. The line identification task provides interactive control over the minimum intensity for 'fitted' features via its (**T**)hreshold option.

The default value is `1.25`.

TUNE_SATRTN = _REAL

Specifies the threshold level for deciding whether any pixels should be flagged as saturated.

Flagged pixels do not contribute to the extraction in any way. Only pixels with intensities less than the `TUNE_SATRTN` level will be used by subsequent tasks.

Initial suggested value: `1.0E20`.

TUNE_SCFRACT = _REAL

Specifies the mixing fraction for two wavelength ranges (fitted). This is only required when there is a 'before' and an 'after' arc frame, and there has been significant movement of the image in between the two. The fraction is used as follows:

$$\lambda = \lambda_1 + (\lambda_2 - \lambda_1) \times \text{Fraction.}$$

No facility is provided for calculating the appropriate fraction but it will usually be estimated as the the ratio of elapsed times between the object frame, and the before/after arc frames (assumes a constant 'drift' of the instrumental setup).

TUNE_SCRADD = _INTEGER

Used only when `TUNE_SCRMODE=0`, and indicates the number of input bins to be added into each output bin. This is used mainly to increase the S/N of a weak spectrum, at the expense of resolution.

TUNE_SCRMODE = _INTEGER

If equal to zero, selects mode 0, in which input bins can be grouped and shifted to form the output bins. (See also `TUNE_SCRADD`).

If not equal to zero, selects mode 1, in which the input bins are transferred to the output bins in accordance with the wavelength scale(s).

TUNE_SKEW = _REAL

The number of bins the input array is to shifted (in mode 0 only). A positive value shifts data to higher pixel numbers - this is a FIGARO convention, and is the **opposite** of that used by `Lolita`.

TUNE_SKVRCORR = _LOGICAL

When this parameter is set to YES then weights are modified to include uncertainty in sky fit optimal variances equivalent to maximising:

$$\frac{\sum_i W_i^2 V_i + \sum_{i,j} W_i W_j V_{ij}}{(\sum_i W_i F_i)^2}$$

where:

W_i is the weight on pixel i,
V_i is the variance on the value of pixel i,
V_{ij} is the covariance of the skyfit between pixels i and j,
F_i is the normalised profile at pixel i.

This reduces to solving:

$$A_{ij} W_j = F_i$$

where:

$A_{ij} = V_{ij}$ for $i \neq j$, and
$A_{ii} = V_i + V_{ij}$.

TUNE_SKYHILIM = _REAL

Specifies the upper threshold used when automatically determining the location of 'sky' pixels in the object spatial profile.

The algorithm used is:

— Median filter profile and multiply by 1.05
— Calculate the threshold point: Min + (Max-Min) x TUNE_SKYHILIM
— Take the minimum of the two quantities

This minimum is used as the maximum sky intensity allowed along the spatial profile.

The default value is 0.5 (*i.e.* 50%).

TUNE_SKYINTER = _LOGICAL

Set if the modelling of sky backgrounds is to be under interactive control.

TUNE_SKYLINW = _INTEGER

Specifies the maximum expected width of sky lines in pixels.

This is only used when a wavelength-dependent model is being used to predict sky intensities using whole-order polynomials. In this case, any pixel whose intensity differs from the fit by more than sigma TUNE_SKYLTHR will be treated as a sky-line candidate, and its actual intensity entered into the model in place of the fit prediction. The same procedure is also applied to the pixels' TUNE_SKYLINW nearest neighbours to ensure that the entire sky line is entered into the model even when its wings are within the sigma constraint described above. This parameter only has effect when TUNE_SKYXPLY>0.

The default value is 5 pixels.

TUNE_SKYLTHR = _REAL

Specifies the threshold level for the treatment of sky-line candidates in the wavelength-dependent sky model produced when TUNE_SKYXPLY is greater than zero.

In this case, any pixel whose intensity differs from the fit by more than TUNE_SKYLTHR x fit-sigma will be treated as a sky-line candidate, and its actual intensity entered into the model in place of the prediction. The same procedure is also applied to the pixels TUNE_SKYLINW nearest neighbours to ensure that the entire sky line is entered into the model even when its wings are within the sigma constraint described above.

Initial suggested value: 3.0.

TUNE_SKYPOLY = _INTEGER

When TUNE_SKYPOLY>0 it specifies the degree of polynomial to use when trying to fit to the sky intensity at any single constant-X / lambda increment. These polynomials are used to predict the contributory sky intensity in the object pixels.

The default value is TUNE_SKYPOLY=0 to obtain sky averaging (*i.e.*, sky model is constant at each order/column).

Set TUNE_SKYPOLY=-1 when there is no sky to be modelled (*i.e.*, sky model is zero everywhere).

Initial suggested value: 0.

TUNE_SKYREJ = _INTEGER

Specifies the number of reject cycles for which sky will be clipped, and the model re-fitted. The polynomial model of the sky background is made by fitting a polynomial at each X-increment. These polynomials are then used to estimate the sky level at the positions of the 'Object' pixels.

TUNE_SKYRTHR = _REAL

Specifies the reject-threshold for sky pixels. Any sky pixels which deviate by more than TUNE_SKYRTHR sigma from the fitted polynomial model, will be clipped, and the model re-fitted. The polynomial model of the sky background is made by fitting a polynomial at each X-increment. These polynomials are then used to estimate the sky level at the positions of the 'Object' pixels.

Initial suggested value: 5.0.

TUNE_SKYSIM = _LOGICAL

Set to YES if simulations are to be used to estimate the variance of the sky using polynomial fits in the wavelength direction.

These simulations are CPU intensive.

The default value is NO.

TUNE_SKYXPLY = _INTEGER

Specifies the degree of polynomial to use when trying to fit to the sky intensity over the whole order at a spatial increment.

This facility will be most useful when the sky is very uniform, or possibly when trying to model the scattered light in the inter-order region. The intensities predicted by the wavelength-dependent model are used as input to the spatial model only when they fall

within a user-defined distance from the observed value, and when they are not possible sky-line features.

(See also TUNE_SKYLINW and TUNE_SKYLTHR).

The default value is 0.

TUNE_TRCNS = _REAL

Specifies the degree to which traces may be inconsistent. Expressed as a percentage of X-samples. For example, if a trace may be a maximum of 10% inconsistent, measured against the other traces, then TUNE_TRCNS should be set to 0.1.

The default value is 0.05.

TUNE_TWTHR = _REAL

The 'trace-width-threshold' is provided for cases when the S/N of the frame being examined to locate the orders is very low.

The threshold is the degree to which off-trace pixels must fall below the pixel at trace centre (in the spatial direction) in order to be considered as 'off-order'. *I.e.*, the trace width is estimated by stepping pixel-by-pixel above and below the trace until the intensity falls below (1.0 - TUNE_TWTHR) x intensity-at-trace-centre.

The default value is 0.95 (95%).

TUNE_UHRF = _LOGICAL

Set if the data is from the UHRF (Ultra-high Resolution Facility of the UCLES).

TUNE_USEAAA = _LOGICAL

Set to YES if the program is to make use of the 'Astronomy & Astrophysics Abstracts' system of information categorisation. This consists of a set of numbers corresponding to topics in Astronomy. If TUNE_USEAAA is active then you will be shown the currently recognised categories and asked to select the one most relevant to the data being reduced. This parameter is only used in conjunction with the archiving facility, when TUNE_ARCHIVE=YES. The prompting process may be overridden by supplying a valid AAA code using the parameter TUNE_AAACODE.

Initial suggested value: YES.

TUNE_USE_NXF = _REAL

Specifies the fraction of the X-dimension of the frame to be used for determining the profiles.

This is set low (0.2=20%) by default for speed.

Very poor data may benefit from an increase in this parameter, which provides a more accurate sampling of the object profile, and thus more accurate extraction weights. The central fraction from each order is sampled and the average profile over all orders is calculated.

A special mode is selected by setting TUNE_USE_NXF>=1, which makes the sampling independent for each order. In this case the fraction of the X-dimension of the frame used is TUNE_USE_NXF-1, for example, TUNE_USE_NXF=1.2 would select individual-order profiling using 20% of each order.

The default value is 0.2 (*i.e.* 20%).

TUNE_XBOX = _INTEGER
This parameter allows the sampling size (in X-pixels) used when performing order location to be altered.

This should be necessary when only one frame is available for tracing the orders, and this frame is of poor quality.

Some care must be taken to ensure that the sampling box is kept sufficiently small that all samples come from the same order for each sampling-box position.

TUNE_XZONE = _INTEGER
Number of horizontal image zones.

TUNE_YBLAZE = _LOGICAL
Set to YES if automatic Y-blaze correction is to be used. This correction works by examining the scrunched orders and re-scaling them so that the extremes of adjacent orders match up. This should only be used when no flux calibration is available and there is a real need for it. It requires that the blaze fit be scrunched before being applied and therefore necessitates a blaze-fit followed by the scrunch option. The blaze correction is only applied to scrunched (and merged) orders in this case. When disabled then blaze fitting is applied to the extracted orders directly and these may be scrunched later if required.

Initial suggested value: NO.

TUNE_YZONE = _INTEGER
Number of vertical image zones.

USE_MEDIAN = _LOGICAL
YES if median is to be used.

WAVFIT = _CHAR
Specifies the wavelength fitting function to use.

W_NPOLY = _INTEGER
Number of coefficients of wavelength fitting function.

W2_NX_POLY = _INTEGER
Maximum order of 2-D X-axis polynomial.

W2_NY_POLY = _INTEGER
Maximum order of 2-D Y-axis polynomial.

2D_INTERACT = _LOGICAL
Set to YES if you want to interactively control the 2-D polynomial fitting of the distortion corrections. Polynomials are fitted in two perpendicular directions (rows and columns) and generate the point-to-point predicted deviation from wavelength fit for each pixel centre. Interactive operation allows you to view an exaggerated version of the fitted points, and to selectively clip points on the basis of their deviations from the fit.

5 AUTOMATIC REDUCTIONS

The main reduction processes may optionally be performed automatically. Invoking ECHMENU with the TUNE_BATCH=Y parameter will cause the following to be performed automatically by setting the appropriate parameters. During automatic reduction the all the steps in a standard reduction are run in the proper order. The following parameters are used to control the tasks:

- Bad row/column removal (TUNE_FCHECK=NO)
- Cosmic-ray removal (TUNE_CRTRC=NO)
- Order location (TUNE_AUTLOC=YES)
- Order trace fitting (clipping)
- Sky/Object/Dekker locations
- Sky modelling
- Flat-field balance factor calculation
- Object profile modelling
- Extraction (object and arc)
- Arc line candidate location
- Arc line identification (TUNE_AUTOID=YES)
- Wavelength polynomial fitting
- 1-d extraction to fitted wavelength scale
- Scrunching spectrum to new wavelength scale
- Writing the resulting spectra to a file

The main method of performing automatic reductions is use of the TUNE_AUTOMATE parameter. This parameter specifies the sequence of operations to perform, where the operations are numbered as they would be in an ECHMENU run. Unless the final operation specified is EXIT, then interactive operation is restored after all requested operations have been performed. *e.g.*:

```
echmenu TUNE_AUTOMATE="'5,6,7,8,EXIT'"
```

would do: flat-field, sky, object profile modelling, extract the orders, and finally exit.

Alternatively you can create a script containing the sequence of ECHOMOP commands required (*e.g.* ech_trace, ech_trplt *etc.*) and then invoke the script from the shell.

On systems where ICL is available, ICL scripts can be used to perform the same operations.

In all cases you should take care to ensure that all the required parameter values are provided if unattended operation is intended.

6 USING PRIOR REDUCTIONS—CLONING

ECHOMOP provides easy access to facilities to enable previous reductions to be used as baselines for a current reduction. For example, the order traces may be in the same positions, the wavelength calibrations may be the same *etc.*

To include data from previously-created reduction databases the number of the option (in ECHMENU main menu) should be followed by a '<' character indicating that the information which would otherwise be generated should be copied from a previous reduction file. *e.g.*:

```
2<source-file
```

would copy the order-trace data (created by step 2 of a reduction) from the reduction database `source-file`, instead of calculating the relevant values.

To invoke 'cloning' when using single-function tasks (*e.g.* `ech_trace`) set `TUNE_CLONE` to the name of the source reduction file. *e.g.*:

```
ech_trace tune_clone=source-file
```

this would accomplish the same as the ECHMENU example above.

This option is most useful for cloning arc line identifications, as Option 10 (wavelength calibration) will automatically detect any global shift and re-identify the lines using the previous identifications as a template. To use the facility for this, the following sequence of options should be used:

- `9` (Locate candidates.)
- `10<old-reduction-file` (Clone identifications.)
- `9` (Locate candidates, they have been overwritten by the cloning.)
- `10` (Identify lines.)

7 ECHOMOP TIPS

7.1 Multi-Object Spectra

Although ECHOMOP was written with the UCLES Echelle Spectrograph in mind. It is in no way limited to processing data from this particular instrument, or even, to data from échelle spectrographs. The package can equally well be used on multi-object spectra obtained, for example, using popular fibre based instruments. If you are processing this type of data the following point needs to be remembered:

- Each spectrum will need to be independently wavelength calibrated as the expected échelle inter-order relationship checked for by completely automatic calibration will not be valid. Automatic identification of lines will still be possible on an object by object basis however, and if all the spectra have the same range (in wavelength), then the wavelength

limits obtained on successfully calibrating the first object can then be used as constraints for the others. In this instance the `ech_idwave` task/ECHMENU Option 9 would be run interactively for the first object, and then via a command such as the example below (for object 2), for the remaining objects:

```
% ech_idwave idx_num_orders=2 auto_id=yes arc_type=$ARCDIRS/THAR.ARC \
           ech_ftrdb=$ARCDIRS/THAR \
           /other parameter settings/ \
           min_wave=3250.0 max_wave=3295.0
```

7.2 Longslit Spectra

ECHOMOP is capable of reducing longslit spectra taken with a variety of instruments. In order to tune ECHOMOP for this task it will usually be necessary to increase the two parameters:

- `TUNE_MXSKYPIX`

  the maximum number of pixels in the spatial direction.

- `TUNE_PFLSSAMP`

  the number of sub-samples in spatial direction (*e.g.*, 10 x `TUNE_MXSKYPIX`).

7.3 Extended Objects

Although explicit support for extraction of extended objects is not provided in this release, it is possible to use ECHOMOP to extract an increment at a time. To do this the normal extraction should first be performed to obtain a reliable sky model. Then the `ech_spatial` task/ECHMENU Option 4 should be used to mask out all object increments except one. The extraction should then be re-done and the resulting spectrum copied into a section of a user-created data array using KAPPA facilities. This procedure needs repeating for each increment required and could be automated using a command file.

7.4 Detector Types

ECHOMOP has special facilities for particular detector dependent problems. In particular, for CCDs and similar detectors, the data may become obscured by cosmic-ray events over a long exposure. The package provides 2 different methods of identifying and flagging such pixels. CCDs also necessitate the provision of a number of extra pieces of information to the reduction process. Care must be taken to ensure these values are accurate as they can have a severe impact upon the resultant spectrum.

For IPCS detectors, frames can be highly distorted in both X- and Y-directions. ECHOMOP provides for a detailed 2-D polynomial fit to be made to the distortions present in each order.

7.5 Cosmic Rays

ECHOMOP provides three major methods of cosmic-ray identification:

- The first approach is to apply two median filters to the data frame. One in the X-direction (along rows) and the other in the Y-direction (along columns). Both of these median images are then divided into the original and the resulting image histogrammed. You are then required to select a threshold point on the displayed histogram. All pixels generating samples above the clip threshold are flagged as cosmic rays. This method does not rely on the échelle nature of the image and may be used on data frames of non-spectral type with some success.

- The second cosmic-ray removal routine, is more sophisticated in its approach and requires that the order traces are already known. It works by examining the variations from expected intensity (calculated from the average profile) and then calculating the probability that such variations are due to chance. Significant pixels are then rejected and the whole order re-processed.

  Iterations continue until the overall significance of all outliers in the order is below a user-supplied threshold. Each order is processed independently in two parts, first the sky pixels, and then the object pixels.

  The significance of a fit is calculated by assuming a Gaussian distribution of deviations from expected intensities and using the Kolmogorov-Smirnov statistic (Ref. Numerical Recipes. Cambridge Press Sec. 13.5)

A post processor is used to counteract the tendency of cosmic-ray identifiers to be fooled by sky line pixels. This task examines the geometry of all 'connected' cosmic-ray pixels and restores any which it adjudges are actually due to bright sky emission lines.

- If you have many 'identical' object frames then these may be processed all together to provide a more reliable cosmic-ray rejection. The command:

  ```
  % $ECHOMOP_EXEC/decos_many
  ```

  provides this facility and should be the preferred method when multiple frames are available. The procedure calculates a median image and then rejects pixels which deviate by more than n sigma from the median value (note that this means that all the frames must be of equal exposure time). Bad pixels are flagged by placing values in the quality array of the frame concerned.

7.6 Readout Noise

CCD detectors are subject to an error due to the final amplification immediately prior to the data leaving the chip. This is called readout noise and is due to uncertainty in the operation of the on-chip amplifier.

Recent developments have led to very significant improvements in this area (giving readout noise of less than 1 count per-pixel), but many older chips are still in use.

ECHOMOP looks for a value for readout noise specified in the data frame itself (for CCD frames). If no value can be found then you will be prompted for a value by the `READOUT_NOISE` parameter.

Some of the more modern CCDs use multiple on-chip amplifiers in order to provide a rapid readout facility. Each of these will have its own independent readout noise value. Currently ECHOMOP assumes that a single readout noise value is applicable for the whole frame so each sub-section of a multiple readout CCD image must be processed separately.

7.7 Flux Calibration

This release of ECHOMOP provides no special support for flux calibration of the extracted spectra. FIGARO provides extensive support for this procedure, including many standard star data tables.

You should note that due to the extremely high resolution of the spectra produced by UCLES, that there may not be sufficiently high resolution data available for the flux standards.

A common situation is that the FIGARO table for the standard contains (at most) one sample (flux) point per order, and that such points are often 40 Angstrom averages. In such cases it is difficult to ascribe much confidence to the resulting flux values.

A program of high resolution observations (of commonly used flux standards) to combat this problem is being planned.

A limited set of high resolution tables are provided. Type:

```
% ls $ECHOMOP_DATA/*.tab
```

to see which ones. These tables may be used with the FIGARO flux calibration programs to flux calibrate spectra output from ECHOMOP.

7.8 Blaze Function

If flux calibration is not being performed it is sometimes desirable to remove the 'blaze' function from the extracted spectrum to assist in fitting line profiles *etc.* during data analysis.

A task is provided for this purpose which operates by fitting polynomials to the flat-field orders. The fits may be automatically or interactively clipped and the resulting blaze spectrum is normalised such that its median intensity is unity. The normalised blaze is then divided into the extracted spectrum.

After a blaze function has been applied to the extracted order all its values may be reset to unity to ensure that the order(s) cannot be re-flattened in error. If the blaze is to be re-applied then the correct procedure is to first re-extract the order(s) concerned and then re-fit the blaze.

7.9 Scattered Light

In some cases you may want to subtract the contribution of scattered light from the object *instead* of using the sky pixels (which implicitly also contain scattered light information). The single function task `ech_mdlbck` performs this scattered light background fitting and places the resulting fitted values in the 'sky fit' arrays used by the extraction routines. The background fitter uses a similar set of parameters to the the normal sky modeller but applies its fits to the sets of inter-order pixels at each X-position in the frame. Note that this involves a lot of I/O and can be quite a slow operation for large frames.

7.10 Arc Frames

The wavelength calibration process may be performed using either a single ARC frame, or by using two frames (before and after the object exposure) and having the wavelength scale interpolated.

The advantage of using a single frame (possibly generated by co-adding the before and after arc frames) is that this allows the resolution of the final spectrum to be easily investigated. This is because the arc spectrum is always extracted in an identical manner to that of the object; using the first (and in this case only) supplied arc frame.

Using two frames and having to interpolate the wavelength scale will be most useful when there is a significant shift during the exposure such that co-adding the two arcs would produce a frame with 'doubled' lines. You should note that in this situation the generated wavelength scale is only a guess based on the assumption that the difference between the two arcs was due only to a stable linear shift between the exposures. You should ensure that the fractional weights assigned to the two arcs (parameter `TUNE_SCFRACT`) are appropriate to the specific situation by investigating the times of the exposures.

7.11 Viewing 2-D Correction

The effect of the 2-D distortion correction can be examined by using the `ech_scrn2d` task to create a 'corrected' image of the orders. This should be displayed using the regular imaging commands. It is also possible to use software intended for processing single spectra to process such corrected images. You should take care however to note that the pixels in such images are no longer independent as they represent the scrunching of two or more pixels in most cases.

7.12 Viewing Spectra

It is often useful when reducing 'difficult' data, to be able to examine various intermediate 'spectra' generated during the reduction process. ECHOMOP provides a very flexible method of examining any intermediate arrays produced by it or other tasks. The `ech_plot` task/ECHMENU Option 27 provides zooming, re-binning, re-scaling *etc.* on any array in the reduction database. Arrays may be specified by name or by using a set of reference names by which ECHOMOP tasks access them. To obtain a directory of known reference names use the D(irectory) option in `ech_plot`.

7.13 Heliocentric Correction

ECHOMOP can apply heliocentric corrections to the output wavelength scales during 'Write results' and 'Multi-merge' operations. In order for this to be possible it is necessary for to provide the package with the date and time of the observations involved. This is done by entering the values in the following manner (SETOBJ is a FIGARO command):

```
% setobj object=rdctn-database.OBS_DATE value="dd/mm/yy"
% setobj object=rdctn-database.OBS_UTTIME value="hh:mm:ss"
```

If these values are not provided a warning message will be printed, otherwise the applied correction will be shown.

8 INPUT DATA FILES

ECHOMOP performs most of its data access via Starlink subroutine libraries. All data frames should be held in files recognised by the CONVERT utility. For efficiency, store your data in the Starlink Extensible N-dimesional Data Format (NDF).

A few ASCII format files are also read by ECHOMOP:

Structure definition files (`.DEF` files, stored in the `ECHOMOP_EXEC` directory) are used by the data-access subroutines to create and access files used to store data.

A list of identified features and their properties (`ECHARC.LIS`) is used for data exchange with the ECHARC program.

8.1 Reduction Database

The reduction database is a file used to contain various information produced during the reduction; it is a database of reference information for the reduction. For example, it might contain a list of order locations, trace polynomials, sky and object channels, wavelength polynomial fits, *etc.*

The file is created at the start of a reduction cycle with a supplied name, and grows in size until the reduction is complete. At this point it contains a fairly complete record of any intermediate results, thus allowing particular sections of a reduction to be tuned and repeated as required.

In general you will not need to access the file except via ECHOMOP. The package provides facilities for examining and plotting data from the file. You can, however, use `hdstrace` to look at any data in the file you wish. See SUN/102 for more details.

8.2 Data Frame Format

The input data frames should be in NDF format. The data themselves being stored in a 2-dimensional array accessible via the path:

```
/filename/.DATA.DATA_ARRARY
```

This capability may be useful when a foreign data format has been translated into an NDF, but with non-standard object names.

8.3 Saving Disk Space

ECHOMOP makes substantial use of disk space throughout the reduction process. This is due to the intermediate storage of information in the reduction file. Much of the information stored is on a per-order basis. For data frames with many orders the disk usage may be substantially reduced by the following:

```
% cp $ECHOMOP_SOURCE/ECH_COMPACT.DEF ECH_RDCTN.DEF
```

and:

```
% source $ECHOMOP_EXEC/echomop.csh      (UNIX)

ICL> LOAD ECHOMOP_EXEC:ECHOMOP          (ICL)
```

This must be done before any reductions are embarked upon as it will not save any space retroactively on reductions already in progress.

When this procedure is used, it is essential to ensure that the per-order mode of operation is used and that the orders are fully processed one-by-one. *i.e.*, flat-field, model sky, extract, and save result for order 1, then for order 2 *etc.* This is because the compact version only stores reduction arrays for the last processed order. Thus flat-field modelling order 3 and then order 4 would cause the calculated model for order 3 to be overwritten by the one for order 4.

8.4 Arc Line Lists

Line list files are stored in ASCII format for ease of maintenance. ECHOMOP reads the line list(s) and creates its database file. These feature databases are created by ECHOMOP itself using the ASCII format files as input. Line list filenames are expected to have a .ARC extension and will be looked for in the directory pointed to by ARCDIRS.

The format of arc line lists is as follows:

```
!  comment lines starting with a !
   nnnn.mmm   iii.jj    text
   nnnn.mmm   iii.jj    text
   nnnn.mmm   iii.jj    text
```

where:

`nnnn.mmm` is a wavelength in Angstroms.

`iii.jj` is an (optional) intensity estimate.

`text` is a description of the line *e.g.*, Th II.

In addition a special comment line is recognised which allows the units of wavelength to be specified:

```
!WAVELENGTH_UNITS=some-unit-specifier
```

The default units are Angstroms. The units string is only used for output messages. Future ECHOMOP releases may support greater use of the wavelength units in constraining fits *etc.*

User-created lists should be placed in the directory pointed to by the `ADAM_USER` environment-variable.

8.5 Structure Definition File

Definition files are used by ECHOMOP to specify the structure of the files for storing objects produced during the reduction. Each of these objects is referred to internally by the package using a reference name. The definition file contains translations of these reference names into actual object paths within datafiles.

For instance; the object referred to as OBS_LINES internally is actually stored in the reduction file in the following object:

```
MORE.ECHELLE.WAVE_FIT_1D.OBS_LINES[tune_mxrfln,num_orders]
```

All such objects can be renamed in the reduction file by altering their specified paths.

The default definitions are likely to be adequate for most circumstances. For more information see the on-line HELP sub-topic *Redefining Object paths*.

8.6 Multiple Arc Frames

Multiple arc lamp images are usually taken to monitor any possible shift in the object spectrum which may have occurred during the exposure.

When no shift is evident then the simplest course is to add the two arc frames together and provide the result to ECHOMOP whenever it prompts for the arc frame name.

When a shift can be discerned, you must make a judgement as to whether meaningful wavelength calibration can still be applied. If so then there are two courses of action available:

- Add the two frames together as usual, providing a single image for calibration, albeit with degraded resolution. This permits an estimate of the resolution in the final spectrum because the arc frame is extracted in an identical manner to the object, and the resolution observed directly.
- Calibrate each arc frame separately and interpolate between the wavelengths (possibly taking into account time elapsed between arc1 - object - arc2).

  This technique allows the calibration to proceed even when a gross shift has taken place, but you should be cautious about using the resulting wavelength scale unless independent confirmation can be obtained from features in the object spectrum itself.

8.7 Multiple Object Frames

ECHOMOP is primarily orientated to processing single frame observations. Facilities exist in FIGARO for the combining of multiple spectra obtained from the reduction of a set of data frames. ECHOMOP does however provide assistance for the situations where the data frames are well registered, removing the necessity for steps such as Order location, tracing, and profiling, to be performed for each frame. The 'clone' option allows the selective copying of sections of a previous reduction file, which then becomes the baseline for the reduction of another similar frame. Optionally the definition file itself can be altered to permit the extension of the reduction database to include multiple spectra from a single observed source. For more information see the on-line HELP sub-topic *Redefining Object paths*.

8.8 Flux Calibration Frames

If flux calibration is to be done then the calibration frame must be presented in a Starlink datafile (NDF) in the same form as the data frame. ECHOMOP presently insists that **all** frames are of the same X- and Y-dimensions (a pretty safe bet). Take care to ensure the correct extraction of the calibration frame. For example, a different CCD bias voltage may have been used to that used for the object data frame *etc.*

Flux calibration is not usually performed on the very high resolution spectra produced by UCLES. This is due in the main to the lack of suitable reference tables. As more UCLES data becomes available this situation should improve.

A detailed description of the steps required is available in the FIGARO documentation.

9 OUTPUT DATA FILES

The various output files created during a reduction are described.

9.1 Output Spectra

The reduced spectra may be output as:

- a graphics plot file for printing.
- a file containing a single orders' spectrum.
- a file containing one spectrum per order in a 2-D array.
- a file containing a single merged spectrum.
- an ASCII file listing data pairs of wavelength v intensity.

9.2 Reduction Database

The reduction database is a file used to contain various information produced during the reduction; it is a database of reference information for the reduction. For example, it might contain a list of order locations, trace polynomials, sky and object channels, wavelength polynomial fits, *etc.*

The file is created at the start of a reduction cycle with a supplied name, and grows in size until the reduction is complete. At this point it contains a fairly complete record of any intermediate results, thus allowing particular sections of a reduction to be tuned and repeated as required.

In general you will not need to access the file except via ECHOMOP. The package provides facilities for examining and plotting data from the file. You can, however, use `hdstrace` to look at any data in the file you wish. See SUN/102 for more details.

10 FIGARO ECHARC IMPORT/EXPORT

In order to provide backward compatibility with the extensive line fitting ECHARC software, ECHOMOP provides the ability to import/export sets of line positions/identifications between itself and ECHARC. These facilities are provided from the main menu of the `ech_idwave` task/ECHMENU Option 10.

The exchange of information takes place on two levels. ECHARC requires an arc spectrum array organised on an NX by number-of-orders basis, plus a set of order numbers (1...n) in another array. These are setup in a Starlink (NDF) file. This file should be given to ECHARC as its input image. ECHARC reads from and writes to the other file; an ASCII file listing the order number, position, and wavelengths of all identified features.

This file has the name `ECHARC.LIS` and is placed in the current directory. ECHARC also uses the arc line reference lists (*i.e.*, `/arctype/.ARC`) and the same file should of course be used with both programs.

Note that the ECHMENU ECHARC export option must be used first in order to create the file required to use ECHARC, this **must** be done even if no lines are identified before ECHARC is run.

A NOTES FOR OBSERVERS

The reduction of spectra collected using the modern CCD and IPCS detectors often require complex procedures in order to ensure that the best possible results are obtained. In many cases, the dataset which the observer brings back from the Observatory is incomplete in the sense that some of the data required for an ideal reduction is missing. It is therefore vital for the observer to ensure that equal effort is paid to obtaining a comprehensive set of calibration frames, to that which was expended on the 'Object' frames. Different types of object will produce different requirements in terms of calibration and the following are merely guidelines.

A.1 Orientation of Spectra

The detectors are frequently aligned on the Spectrograph such that the central order falls (mainly) parallel to the detector rows or columns. The orders themselves however, are often not parallel to each other. The algorithms used for tracing orders assume the orders run approximately in the horizontal direction. It is therefore necessary to rotate any column-parallel frames before reduction (through 90 degrees). ECHOMOP is very flexible as to the exact order orientation and highly sloped and severely distorted orders can usually be extracted without difficulty **provided that a trace frame of sufficient quality is provided**. This will generally be in the form of a bright star exposure, or a template trace frame such as an exposure of a continuum lamp (with a narrow dekker preferably). If the target object under study is sufficiently bright, then the object frame itself can of course be used as the trace template frame. ECHOMOP is capable of tracing very poor object frames, **but** with the result that the modelled order positions will be less well determined, impacting on all subsequent phases of a reduction. You are encouraged to provide the best order-tracing template possible. The Figure below shows an example of the order trace paths plot.

Images not conforming to this format will need to be rotated through 90 degrees.

A.2 CCD Bias Frames

A bias current is routinely applied to CCD detectors to ensure that, as near as possible they are operating in a linear manner. This current has the effect that a non-zero count is recorded in all pixels. The observer should ensure that a number of 'bias frames' are obtained to adequately estimate the effect. The average of the bias frames should be taken, and the result subtracted from all other frames before using them in a reduction. Failure to account for the bias may have a detrimental effect on the signal-to-noise-ratio (S/N) in the final spectrum.

ECHOMOP does not provide any special facilities for the subtraction of the bias current from CCD frames. However, it does require that such subtraction be performed **before** processing takes place. The bias subtraction may often be done automatically at the observatory by the data recording software but it is up to you to check if this is the case. KAPPA or FIGARO can be used to subtract the bias if necessary.

For example:

```
% csub /inframe/ /nnn/ /outframe/    (KAPPA)

% icsub /inframe/ /nnn/ /outframe/  (FIGARO)
```

will subtract bias `/nnn/` from data `/inframe/` and put the resulting data in `/outframe/`.

For cases where the bias subtraction represents a significant source of error it may be desirable to determine the error due to bias subtraction on a pixel-by-pixel basis using many frames. The resulting error estimates must then be copied into the object data frames error arrays. ECHOMOP will incorporate these errors when calculating the variances on the extracted spectra.

A.3 CCD Overscan Region

CCD frames typically include a number of rows/columns not exposed to the light. The pixels in these rows often contain high counts and can affect some of the automatic reduction procedures. It is prudent to clip off these rows/columns before reduction proceeds. They will be the same for all frames and can thus be removed using a simple looping command procedure.

A.4 CCD Dark Count

Dark frames are essential for some applications. They are advisable when using on-chip binning. For CCDs, the dark current may be high if it has been exposed to too much light, or if power has been interrupted to a cold chip. A number of long exposure (30 to 60 minutes) dark frames should be combined using a median filter, and the result scaled according to the object frame exposure time. This frame may then be subtracted from the object frame before reduction begins. If you wish to ensure that the error properties are accounted for then the dark frame should be copied into the object frame variance array.

Again for CCDs only, a further advantage of collecting long exposure dark frames is the option of deriving an independent estimate of the rate of cosmic-ray events as a function of their intensity. This can provide a useful comparison with long exposure object frames and a check on how effective cosmic-ray removal has been.

A.5 Flat Fielding

A bright quartz lamp should be used with a dekker size a few steps larger than that used for the object observations, but order overlapping should of course be avoided. It is important to ensure that all orders are as bright as possible. It may be necessary to use different flat-field exposures to obtain good frames when observing in a spectral region where the quartz lamp intensity or detector efficiency changes rapidly. Each flat-field frame should be constructed by calculating the median of a large number of runs where possible. If you wish to include the errors on the flat-field then they should be placed in the flat-field variance array. If no variance estimates are provided then root-N statistics will be assumed. ECHOMOP can utilise different flat-field frames for different orders if required.

A.6 CCD Correction From Observed ADU to Photon Counts

Many of the algorithms used assume that photon counting statistics apply. However, it is unnecessary to correct raw frames since the conversion factor (ADUs/photon) is prompted for when required.

A.7 Arc Frames

In general arc frames should be obtained to bracket each object exposure in time, allowing us to check that no shift has occurred during the exposure. UCLES is extremely stable over relatively long periods and in practice shifts are not detectable. The advantage of using a single frame (possibly generated by co-adding the before and after arc frames) is that this allows the resolution of the final spectrum to be easily investigated. This is because the arc spectrum is always extracted in an identical manner to that of the object; using the first (and in this case only) supplied arc frame. As with the flat-field frame it may also be advantageous to take multiple arc frames, each exposed correctly to ensure high S/N in particular orders. ECHOMOP can use a different arc frame for each order if required.

A.8 Preparing for a Reduction

All raw data should be pre-processed in the following ways before using ECHOMOP:

- The frames necessary for the reduction should be first converted into Starlink standard data files. ADAM provides a set of tape reading and format translation utilities to help you do this.
- Rotate the data frame 90 deg. if necessary to achieve an orientation where the long axis of each order is in the X-pixel (horizontal) direction.
- Subtract the BIAS current value from all raw frames using the KAPPA CSUB command.
- DO NOT multiply (CCD) frames by the prevailing ADU to electron conversion factor prior to reduction. ECHOMOP tasks prompt for this value and use it to generate resulting spectra with counts in photons.

If a cosmic-ray-contaminated frame is to be used for tracing the paths of the orders across the frame then:

- Use `ech_decos1` to clean cosmic rays from the trace frame. This task flags cosmic rays in a quality array and does not actually alter any of the data frame values. This is the recommended approach in this situation. The `ech_decos1` task may be used stand alone, or it may be invoked from the main task by specifying the parameters `TUNE_CRTRC=YES` and `TUNE_FCHECK=YES` at the start of a reduction.
- If you have many 'identical' object frames then these may be processed all together to provide a more reliable cosmic-ray rejection. The command procedure:

  ```
  % $ECHOMOP_SOURCE/decos_many
  ```

  provides this facility and should be the preferred method when multiple frames are available. The procedure calculates a median image and then rejects pixels which deviate by more than n sigma from the median value (note that this means that all the frames must be of equal exposure time). Bad pixels are flagged by placing values in the quality array of the frame concerned.

B PROBLEM HINTS

B.1 Needs Rotating

A data frame has been specified whose dimensions are orthogonal to those of an already processed frame for this reduction. The most likely cause is that one of the frames has been rotated and the other not. The required frame geometry is to have the orders long axis aligned with the horizontal (X-axis). The solution is to ensure that all frames are consistent and then re-start the reduction. In the case that the first frame specified needs to be rotated then the reduction process **must** be re-started from the beginning using a new reduction file (delete the current reduction file). In the case that that the first frame specified was OK, then simply correcting the other frame and continuing with the reduction will suffice.

B.2 Bad X Dimension

A data frame has been specified whose X-dimension is different from that of the first frame specified during the reduction. All frames **must** have the same X- and Y-dimensions. The most likely cause is that the frames are from different datasets. If you really want to use this combination of data frames then you must use the facilities of FIGARO to 'grow' one of the data frames up to the dimensions of the other. Take care to ensure that any image registration problems are removed before reduction is attempted. In general it will be necessary to re-start the reduction with a new reduction file after any such adjustments have been made.

B.3 Bad Y Dimension

A data frame has been specified whose Y-dimension is different from that of the first frame specified during the reduction. All frames **must** have the same X- and Y-dimensions. The most likely cause is that the frames are from different datasets. If you really want to use this combination of data frames then you must use the facilities of FIGARO to 'grow' one of the data frames up to the dimensions of the other. Take care to ensure that any image registration problems are removed before reduction is attempted. In general it will be necessary to re-start the reduction with a new reduction file after any such adjustments have been made.

B.4 Vmem Exhausted

The program has run out of virtual memory space. The system command 'vmstat' produces a brief display of system memory status. In particular, note the value which appears under the 'free' column. A low value here indicates that the whole system is running low, otherwise it is likely that the problem is specific to your own limits. You may check these with the 'limit' command.

B.5 Vmem Release

The program has been unable to release allocated memory correctly. The most likely cause of this is attempting to continue processing after having run out of virtual memory (Vmem_exhausted). The only solution is to abort the task and re-start.

B.6 Dimension Conflict

An attempt has been made to access a reduction object using dimensions different from those operative at the time of its creation. This will usually be due to the changing of a tuning (TUNE_???) parameter which is used to specify the dimension, or due to a change in wavelength scale during scrunching (which changes dimensioning objects NX_REBIN and NO_OF_BINS). The name of the parameter is shown along with its 'old' and current value.

If a tuning parameter is involved then there are two courses of action available. The first option is to reset the parameter to its previous value and continue with the reduction. The second option is to delete from the reduction file the object which has been created using the 'old' value. Such objects must then be re-created using the new parameter value by invoking the relevant processing options. The program will prompt as to which course of action is required. It should be noted that in some cases deleting an object may necessitate the re-running of an earlier option in order to re-create it, and re-calculate its contents. In this case you will be informed as to which step needs repeating.

B.7 No Orders

The task has been unable to find any orders in the frame supplied. The most likely cause is that the frame has been incorrectly specified and contains no orders. Another possibility is that the frame has the wrong orientation (orders long axis horizontal) although at least one order would usually be 'found' in such a case. If the frame is correctly specified and orientated then possible causes are as follows:

Cosmic-ray noise, use the ech_decos1 task/ECHMENU Option 1.2 to clean the cosmic rays before re-trying order location.

Very poor S/N data, set the parameter TUNE_AUTLOC=N to specify the positions of the orders manually using a graphics cursor.

B.8 Bad Diagonals

The program is unable to determine the slope of the orders across the frame. In this situation the slope is assumed to be 0. The most likely cause is problems caused by cosmic-ray contamination which can be avoided by using the ech_decos1 task/ECHMENU Option 1.2 to clean the frame prior to slope determination.

The slope value is only used when tracing very poor data and it will not usually cause problems if its value is left at zero.

In extreme cases you can set the slope manually using the FIGARO command SETOBJ:

```
% setobj object=-reduction-file-.MORE.ECHELLE.ORDER_SLOPE value=nn.nn
```

where nn.nn is the increase in y coordinate per X-pixel increment.

B.9 Wave-fit Start/end

The automated polynomial fit in wavelength has generated a wavelength range which is inconsistent with the expected scope of wavelengths (either as specified, or as calculated from previously identified orders). You should investigate the identification of this order manually to verify the situation. In most cases allowing a larger search scope for the order and/or manually rejecting the incorrect identification using the AI option in `ech_idwave` will lead to a more favorable identification.

B.10 Wave-fit Deviations

The fitted polynomial has been rejected due to an excessive RMS deviation measure. Usually the rejection will be valid and the program will proceed to locate a better set of identifications. However, if looking at the other orders shows that these identifications are likely to be correct, then `ech_idwave` should be used in its manual mode of operation. Specifying an appropriate search range for the order and using the AI option will let the program relocate the set of identifications it rejected when running automatically. You can then accept the set and interactively improve it by adding/removing individual line identifications.

B.11 Cannot Create DB

The program is unable to create an arc line database file. This is usually due to either the destination directory being write-protected, or a disk quota being exceeded.

B.12 Feature List Too Small

ECHOMOP uses a special arc line information database to optimise its search algorithm. This algorithm requires sets of approx. 12 lines with which to work. The database consists of lines and their 10 nearest neighbours on each side (l/r). The arc-line list supplied (`.ARC` file) does not contain sufficient lines for the database to be useful. Either more lines must be added to the list, or another program (such as ECHARC) must be used for the line identification phase. It is possible to add dummy entries to the line list just to overcome this limitation, however in this case it is unlikely that automatic identification will work and the program will probably need to be run interactively.

B.13 No Trace Width

The program failed to determine the width of the order traces. The algorithm for this process requires that the intensity falls to less than the fraction specified by `TUNE_TWTHR` of the intensity at the peak. You may either adjust the parameter (probably increasing it), or directly specify a value for the trace width using FIGARO SETOBJ:

```
% setobj object=-rdctn-file-.MORE.ECHELLE.TRACE_WIDTH value=nn
```

where `nn` is the approximate width in pixels.

B.14 Corrupt DB

A 'read' operation on the arc line database has failed. The most likely cause is that an inadvertent database creation has been performed which has resulted in an incomplete copy of the database. Check the database files using

```
% ls -arc-list-type-.sdf
```

and delete/rename any partial files.

B.15 Bad Sky Below

The program was unable to obtain a satisfactory estimate of the position of the 'sky' in the region below the object. The algorithm for determining the sky extent uses the median intensity across the profile, and the relative peak intensity. The relative peak intensity is multiplied by the value of the parameter TUNE_SKYHILIM and compared with 1.05 times the median intensity. The lower of the two values is used as a threshold. When the intensity falls below this threshold, then the program denotes the pixels from the current pixel to the edge of the dekker as 'sky'. You may vary TUNE_SKYHILIM to adjust the threshold used. It is also possible to manually adjust the status of each pixel in the profile by setting PFL_INTERACT=Y.

B.16 Bad Sky Above

The program was unable to obtain a satisfactory estimate of the position of the 'sky' in the region above the object. The algorithm for determining the sky extent uses the median intensity across the profile, and the relative peak intensity. The relative peak intensity is multiplied by the value of the parameter TUNE_SKYHILIM and compared with 1.05 times the median intensity. The lower of the two values is used as a threshold. When the intensity falls below this threshold, then the program denotes the pixels from the current pixel to the edge of the dekker as 'sky'. You may vary TUNE_SKYHILIM to adjust the threshold used. It is also possible to manually adjust the status of each pixel in the profile by setting PFL_INTERACT=Y.

B.17 Bad Lower Dekker

The program was unable to obtain a satisfactory estimate of the position of the dekker in the region below the trace. The algorithm for determining the dekker position steps out from the peak intensity until the intensity falls below a tunable threshold. If the predicted edge is reached before this occurs then the lower dekker is set to this predicted value (simply calculated as half-way between the orders).

You may vary TUNE_DEKTHR to adjust the threshold used. It is also possible to manually adjust the status of each pixel in the profile by setting PFL_INTERACT=Y.

B.18 Bad Upper Dekker

The program was unable to obtain a satisfactory estimate of the position of the dekker in the region above the trace. The algorithm for determining the dekker position steps out from the

peak intensity until the intensity falls below a tunable threshold. If the predicted edge is reached before this occurs then the lower dekker is set to this predicted value (simply calculated as half-way between the orders).

You may vary `TUNE_DEKTHR` to adjust the threshold used. It is also possible to manually adjust the status of each pixel in the profile by setting `PFL_INTERACT=Y`.

B.19 Lost Left Trace

The program has failed to trace the order outwards from the centre of the frame towards the left-hand side (decreasing X). If tracing is being attempted using a cosmic-ray-contaminated frame then the `ech_decos1` task/ECHMENU Option 1.2 should be used to clean the frame prior to tracing.

The parameter `TUNE_MXBADSMP` determines the number of failed increments before the trace is 'lost'. The parameter `TUNE_MXSMP` determines the number of sampled increments across the frame and thus their separation. In general increasing `TUNE_MXSMP` will assist the program in tracing difficult orders. You should also experiment with the various modes of tracing (controlled by `TRACE_MODE` parameter) which should allow most cases to be traced. As a last resort the program also provides a manual trace mode in which points specified using a graphics cursor (ICUR program) can be used as the basis for the tracing process.

B.20 Lost Right Trace

The program has failed to trace the order outwards from the centre of the frame towards the right-hand side (increasing X). If tracing is being attempted using a cosmic-ray-contaminated frame then the `ech_decos1` task/ECHMENU Option 1.2 should be used to clean the frame prior to tracing.

The parameter `TUNE_MXBADSMP` determines the number of failed increments before the trace is 'lost'. The parameter `TUNE_MXSMP` determines the number of sampled increments across the frame and thus their separation. In general increasing `TUNE_MXSMP` will assist the program in tracing difficult orders. You should also experiment with the various modes of tracing (controlled by `TRACE_MODE` parameter) which should allow most cases to be traced. As a last resort the program also provides a manual trace mode in which points specified using a graphics cursor (ICUR program) can be used as the basis for the tracing process.

B.21 Untraceable

The program could not trace the order at all. Check that:

- the correct frame is being traced.
- the frame is oriented so that the orders run with the dispersion direction parallel to the X-axis of the frame.

If tracing has to be attempted using a cosmic-ray-contaminated frame, try using the `ech_decos1` task/ECHMENU Option 1.2 to clean the frame prior to tracing.

The parameter TUNE_MXSMP determines the number of sampled increments across the frame and thus their separation. In general, increasing TUNE_MXSMP will assist the program in tracing difficult orders. It is worth experimenting with the various modes of tracing (set by the TRACE_MODE parameter). Usually, at least one of the available options will handle difficult or unusual cases.

As a last resort a manual-trace mode, in which points specified using a graphics cursor (ICUR program) are used as the basis for the tracing process, is available.

B.22 Cannot Create ECH_RDCTN

The reduction database can not be created at this stage. You must either use ECHMENU Option 1, or task ech_locate, as the first operation. These ensure that the reduction database used by all other task/operations is correctly created and initialised with the data frame dimension information.

B.23 Cannot Create ECH_FTRDB

The specified file cannot be found and its automatic creation is illegal using the selected option.

Please use the special task ech_ftrdb to perform arc line database creation. This process should normally be left to the node manager who should place the database files in a commonly accessible directory (usually the same place as the .ARC files). If you wish to create a private database for your personally specified line list then use task ech_ftrdb to do so. If you are not trying to create a private database then the most likely problem is that the environment-variable ARCDIRS has not been used to prefix the arc database name (*e.g.*, ARCDIRS:THAR), or that the named arc database does not exist (ask the system manager).

If you wish to include a set of personally specified line list databases in the default search paths used by ECHOMOP, then their location should be added to the list held in environment-variable ARCDIRS. The list of locations is searched in order of their appearance in the environment-variable. The default search path is:

. — the current directory.

ADAM_USER — ADAM user directory.

ECHOMOP_EXEC — ECHOMOP base directory.

B.24 Not a Reduction Format

The specified file exists **but** is not a reduction database as defined by ECHOMOP tasks. The most probable cause is that a data frame file name has been provided by mistake.

The reduction database **must** be created before the reduction can start. The task ech_locate/ ECHMENU Option 1 is currently the only one authorised to invoke reduction file creation so you should use this if you need a new reduction database.

B.25 No Update Access

The program is unable to obtain update access to an array. This may be either a DATA, QUALITY or ERROR array; and the most most probable cause is that the file containing the array is protected against write access in some way. If the file is not owned by you, and in your directory, then it is usually easiest to copy it into a directory to which you have full access rights.

B.26 Cannot Create Here

The program is unable to create a reduction file as specified. The most probable cause is due to the directory where the file creation is to be performed being protected against write access, or does not exist at all. Type:

```
% ls -l /name/of/directory
```

to verify that the file exists, and to see what access rights the current directory has.

If you own the directory in which file creation was attempted and have the required access rights then the most probable cause is the exhaustion of system resources required, most notably disk space. Type:

```
% quota
```

to see if you have run out of space.

B.27 No Cloneable Object

The program is unable to copy reduction database objects from the specified reduction database. The possible causes are that the object in question does not exist in the reduction file specified (either using `TUNE_CLONE` or the < indirection operator in ECHMENU), or that the object is dimensioned differently from the corresponding object in the primary reduction database. Objects can only be copied when their dimensionalities are identical (*i.e.* they have the same number of X-pixels, number of orders *etc.*).

To check the dimensions of the object you are trying to clone from, use the FIGARO EXAM command and the object path as reported by the program. If the only dimensions which differ are controlled by `TUNE_` parameters, then you may adjust the relevant parameter to allow the object to be successfully copied (Note that this in turn may necessitate re-running parts of the reduction to re-generate other objects also dimensioned using the same `TUNE_` parameter).

B.28 Read-Only RDCTN File

The program is unable to obtain UPDATE access to an object in the reduction database. This is probably due to the file being protected against write access. Type:

```
% ls -l /name/of/directory
```

to check the file protection mask.

The program will continue **but** please note that it will be unable to write the results of the current step back into the reduction database. You are therefore strongly recommended to copy the file to a location where you have full access to it before proceeding.

C RELEASE NOTES

C.1 VERSION 3.3-0

These notes formed a NEWS item at the time of release of version 3.3-0, they describe the many changes from the previous, v3.2-0, release.

Changes & New Features

- The parameter `ARC_TYPE` is now ignored by the wavelength-calibration (Option 10) and wavelength-scale consistency checking (Option 20) tasks. The parameter should be removed from any scripts which use these tasks before the next release of ECHOMOP, at which point the parameter will be withdrawn.
- The central order number and central wavelength of an echellogram can be used to constrain the automatic wavelength calibration process. See parameter information for `CENTRAL_ONUM` and `CENTRAL_WAVE` for more details.
- The output option (Option 14) can now produce DIPSO stacks of echelle data. For example, to output a stack called `ECHOMOP_STK.sdf` containing the extracted object orders:

  ```
  % ech_result result_format=stack result_type=extobj stack=ECHOMOP
  ```

  See the on-line HELP for more details.
- When writing an ASCII file in Option 14 the name of the file can now be changed by setting the parameter `ASCII_FILE`.
- The Thorium-Argon arc-line list and database have been expanded.
- A line list and database for Thorium-Neon arcs has been added. The relevant files are `THNE.ARC` and `THNE.sdf`.
- An extra line has been added to the Copper-Argon line list and database.
- Non-standard PGPLOT calls have been removed from the source code. This will lead to a minor performance improvement.
- In the plotter, the ASCII-dump option now produces a files which DIPSO SP2RD can read. The output filename is no longer fixed.
- Deferred PGPLOT errors (which would be reported on exit from ECHOMOP) have been removed.
- The value `none` is now acceptable to indicate that no flat field (`FFIELD` parameter) or no arc (`ARC` parameter) is to be used. Previously only the upper-case `NONE` could be used. Note that only these two values indicate no available frame; a value of `None` or `NoNe` will be assumed to be the name of a frame.

- In Option 1 (Locate orders) the criterion for detecting an order has been slightly changed. Previously, a candidate order would have to have at least 1% of the intensity of the brightest order found minus the median value of the central column in the image. This does not allow for automatic detection of all orders if the brightness of orders and the inter-order background level vary significantly across the frame. The criterion has been changed to simply: 'intensity at least 1% of the brightest order'.
- HELP is now available within the plotter task.

Documentation notes

- The database files used to store information about a specific reduction have been retitled 'reduction databases'. Previously these were known as 'reduction structure files'. This change is reflected in all documentation and source code comments.

Fixes

- When changing between arc-line databases there is no longer a prompt to choose between old and new values of TUNE_WCAL_INDEX. This was a bug which would cause the program to crash if the user guessed 'new' as the response to the prompt.
- A memory management problem causing a crash if the abort '!' option was used in Options 3, 6, or 11 had been removed.
- The parameter CENTRAL_ONUM can now be used, previously the value was forced to zero within the program.
- In Option 4, the slit setup can now be edited. Previously, each invocation of Option 4 would recalculate the settings. The settings can now be edited without this 'reset' by setting PFL_MODE to the value E before starting the option.
- The program now reports an error when unable to successfully access an input file. The program reprompts the user. Previously the program would crash when trying to use 'unaccessed' data.
- The 'Path syntax' in the plotter is now correctly displayed.
- In browse mode in the plotter SPLINE traces are now supported. Previously POLY was assumed with unpredictable side-effects.
- Limits in plotter browse mode are now correctly enforced, avoiding common crashes.
- All advertised style options in the plotter now work.
- A bug in Option 3 (Trace clip), which would decrease the number of knots for a SPLINE fit by 2 on each invocation, has been removed.
- In Option 14 (Output results) the OSPECT output product can now be used without causing a crash.
- The conversion from air to vacuum wavelengths enabled by TUNE_AIRTOVAC is applied to the output data file when requested. Previously, the correction was applied to the reduction database wavelengths, not to the output file.

- A possible divide-by-zero error in Option 5 (Flat field) has been removed.
- A missing range check in Option 9.2, which might give rise to segmentation faults, has been removed.

C.2 VERSION 3.2-0

These notes formed a NEWS item at the time of release of version 3.2-0, they describe the many changes from the previous, v3.1-0, release.

General note: Several modifications have been made to enhance the speed of the program. Generally a reduction of about 35% in execution time can be expected.

New features

- A new task, `ech_genflat`, has been added. This task outputs flat-field balance factors as generated by `ech_ffield` to an image. The image can then be inspected using, e.g., KAPPA DISPLAY.
- The design of the interface in parameter editors has been altered to make them a little easier to use.
- The parameter `TUNE_USE_NXF` has been 'enhanced'. Previously, this parameter set the fraction of an order (in the dispersion direction) to be used when profiling; setting the parameter to a value of 1.0 selected a 'special' mode where each order was separately profiled. This behaviour remains the same; however, it is now possible to select individual-order profiling and set the fraction of each order to be used. For example, a value for `TUNE_USE_NXF` of 1.2 selects the central 20% of each order and individual-order profiling.
- 'QUIT' and 'Q' have been added as aliases for EXIT in the `echmenu` top-level menu.
- 'Q' has been added as an alias for 'E' (EXIT) in the `ech_plot` top-level menu.
- The task `ech_fcheck` now checks both input and trace frames for bad-pixel values in their data arrays. The trace frame is now checked for saturated pixels.
- In the plotting task the full option menu is displayed only once. The menu can be redisplayed by pressing 'M' as in other tasks.
- In the plotting task when displaying reduction data the default prompt now automatically sets its self to point at the next order for the last data plotted. For example a plot of 'OBJ[1,1]' — the first order of the object—will set the default value to 'OBJ[1,2]' —the next order. This means it is much faster to check through all the orders.

Documentation

- ECHMENU option 1.5 has been documented in SUN/152.
- Some errors in SUN/152 have been fixed, the text clarified in a few places, and several missing parameter details added.
- Some errors in the on-line and hypertext HELP texts have been corrected.
- The out-of-date and uninformative 'Inputs-Outputs' and 'Method' entries in the on-line and hypertext HELP text have been removed.

Parameter default changes

- The default value of `TUNE_MXSMP` has been changed from 200 to 500. This reflects the increase in size of CCDs in recent years.
- The advertised default value of `TUNE_TWTHR` (0.9) was incorrect and has been corrected to 0.95.
- The advertised default value of `TUNE_AUTLOC` (`YES`) was incorrect and has been changed to `NO`.
- The advertised default value of `TRC_NPOLY` (7) was incorrect and has been changed to 4.
- The default value of `TUNE_ARCHIVE` has been changed to `NO`.
- The default value of `MIN_DISPERSION` has been changed to 0.01 which better reflects the dispersions used in échelle instruments.

Other changes

- ECHOMOP no longer uses the NAG library.
- The module definition database has been removed. Modules are defined as the program starts up.
- Only one call to PSX_UNAME is made per invocation of the monolith. Previously, several routines called PSX_UNAME.
- Calls to PGPOINT have been changed to call PGPT.
- The subroutine calculating median values has been enhanced.
- There is only one routine for mean/median/most-common-value calculation.
- Several table look-up routines have been modified to exit-on-match, rather than searching the full table for an entry.
- The trace-clipping task menu is only displayed once which speeds up the process of trace clipping. The full menu can be displayed using option 'M' in the same style as the profiling tasks.
- In the spatial-profiling task the display of 'below'-trace pixel-distances has been changed to reflect the syntax used. For example the dekker edge is now said to be '20 pixels below trace' rather than '-20 pixels below' which suggests that it is above the trace.
- Messages of the form 'Parameter XXX is set to non-default value: nn' have been shortened.
- Some of the main menu text has been changed.
- Several (140) calls to ECH_SET_CONTEXT which had no effect have been removed from subroutines. This may lead to a small speed-up for some operations.
- When `DISPLAY=YES` overlaid trace plots are labelled with the relevant order number. The display scaling is calculated per-order, which gives more satisfactory results.

- The routine ECH_FIND_CENTRE has been modified, speeding up centre-of-gravity mode tracing and echmenu option 9 (locate arc line candidates).
- The file ech_dynamix_index_index.f (which was not used) has been removed.
- The file ech_makefits.f (which was not used and is not needed) has been removed.
- The file ech_plot_id_lines.f (which was not used) has been removed.
- In the dekker/object-profiling task (echmenu option 4) graph titles now include the number of the order displayed.
- Many FORMAT statements have been changed by the addition of '1P' scale factors. This makes output numbers easier to read.
- The files ech_loop_nvariable.f and ech_loop_variable.f have been removed due to the process of speeding up the main ECHOMOP routine and introduction of extra functions.
- The IDX_ parameters were being typed as _REAL by the program whilst they were (correctly) declared as _INTEGER in interface files. This has been corrected.
- In the order-blaze fitter the plots of function-verses-fit now use the same colour scheme as in the order trace fitter.
- The displays and menus in the interactive flat-field modeller have all been tidied up.
- In the trace-clipping task the '.' option now deletes the nearest point to the cursor as advertised. Previously this deleted the nearest point in X. This change also affects the blaze-fitter and sky-modeller.
- ECH_FATAL_ERROR and ECH_CALC_TRACE have been modified to speed up the program.

Fixes

- In the order-fitting/clipping process it is now possible to select an order of fit less than 4.
- In the `ech_spatial` task it is no longer possible to set the value of DEK_BELOW greater than DEK_ABOVE and vice versa. Previously this would cause various problems.
- Internal arrays in profiling tasks have been enlarged and range checks are now performed prior to array usage. This has removed some intermittent crashes.
- Most FORMAT statements using Hollerith characters have been updated.
- An error in the data access layer relating to default object dimensions has been fixed.
- Previously, in plots of order-trace versus fitted-curve the points of the order trace were shifted down. This gave the impression that the fit was offset by a constant. This bug has been removed. This bug also affected the order trace overlaid on image plots.
- A limitation in ECH_MEAN_MEDIAN whereby only the first 5000 values of a dataset would be used in median determination has been removed.

- An error causing a crash for `USE_MEDIAN=YES` in the order-location task has been removed. `USE_MEDIAN=YES` now works OK.
- An error in the call sequence causing unaligned memory access and crashing `ECH_DECOS2` has been removed.
- Spurious 'Unknown fitting function' messages appearing when using `ECH_DECOS2` have been removed.
- Centroid-mode tracing now includes a filter for bad values in images. Previously, these bad values would break the centroiding code causing a crash with a floating-point exception.
- The spatial-profiling task now includes various filters for bad values which would break it.
- A commonly-occurring divide-by-zero error in the optimal extraction algorithm implementation has been removed.
- The simple extraction algorithm will now handle bad-pixel values as well as, or instead of, quality arrays in object images.
- The simple extraction algorithm will now handle bad-pixel values in arc images.
- The profile-weighted extraction algorithm will now handle bad-pixel values as well as, or instead of, quality arrays in object images.
- The profile-weighted extraction algorithm will now handle bad-pixel values in arc images.
- The HELP facility is now accessible from the echmenu program (bug in v3.1-0).
- The quick-look extraction algorithm will now handle bad-pixel values as well as, or instead of, quality arrays in object images.
- The quick-look extraction algorithm will now handle bad-pixel values in arc images.
- After running the quick-look extraction the colour of plots is now reset to black rather than being left blue.
- The 2-D simple-extraction algorithm will now handle bad-pixel values in arc images.
- The 2-D simple-extraction algorithm will now handle bad-pixel values as well as, or instead of, quality arrays in object images.
- The arc-line width estimator routine can now handle bad values in the image. The routine has been improved to scale data according to the number of good values obtained, previously no scaling was applied.
- After running the trace-consistency checker the colour of plots is now reset to black rather than being left red.
- The blaze-fitting option now checks for the case `TUNE_NOFLAT=YES`, previously it would simply crash in this case.
- After running the order-trace plotting task the colour of plots is now reset to black rather than being left red.

- Image displays autoscale to suit the data rather than using fixed (incorrect) values dependant on the image dimensions.
- Order tracing with DISPLAY=YES now works, previously this caused a crash due to a type mismatch in a subroutine call.
- Previously, order tracing with fitted traces displayed overlaid on the traced image only worked when some other plot had been made in the session. This can now be the first plot in a session.
- A crash when running the trace-consistency checker with DISPLAY=YES has been removed.
- Evaluation of SPLINE-fitted order traces in the order-tracing task was not done correctly, causing the program to crash in some cases. This bug has been removed.
- Reporting of messages relating to NULL (!) and ABORT (!!) responses to parameter prompts is no longer deferred.
- The faulty arc-line database file $ARCDIRS/THAR.sdf has been replaced.
- Incomplete database entries in $ARCDIRS/ THAR.sdf and CUAR.sdf have been filled. The first 10 and last 11 entries, which should be partially present, were completely omitted. The database builder has been corrected to add the partial entries.
- When using ech_idwave, in the case of one identified feature in an order, the program would crash when trying to determine the wavelength range for the order. This bug has been removed.
- Option 'M' in the ech_idwave task, order-processing menu, no longer crashes the program.
- In the ech_idwave task, in the event of a wavelength polynomial being of such a low-order that all points are fitted exactly (giving zero RMS error) the program no longer crashes on a divide-by-zero.
- The interactive mode of ech_idwave, options < and > no-longer cause a crash when the display is not zoomed.
- In the interactive mode of ech_idwave, option 'I' would work only once per option 'P' this has been fixed.
- In the interactive mode of ech_idwave, option 'I', the list of nearby features is centred on the selected feature, rather than the next feature up in wavelength.
- Arc line database files no longer have to have write access enabled. This means that files in $ARCDIRS do not have to be copied to an ECHOMOP user's working directory.
- Several missing parameters have been added to the interface files.
- An infinite loop on detection of bad columns or rows in ech_fcheck has been removed.
- A divide-by-zero error in ech_ffield for local mean or median calculation has been removed.

- In `ech_ffield` the calculation of local medians now uses the correct start point in the X-direction. Previously, the median for the start point X-`TUNE_FFLSMP` rather than X-(`TUNE_PFLSMP`/2), was being used.
- In `ech_fitblz`, a floating point overflow could occur when automatic fitting became unstable. The overflows are now clipped.
- The type of the parameter `TUNE_SCFRACT` was incorrectly set as _INTEGER. It is now _REAL.
- Several internal changes to the scrunching routines have been made to prevent scrunched spectra being reflected in the wavelength axis, *i.e.*, all the fluxes being negative.
- Errors in the sky modeller which would lead to attempted processing of points outside the bounds of an image have been removed.
- The sky modeller now filters out bad-pixel values.
- The object profile modeller now filters out bad-pixel values.
- In several tasks, notably the extractions, weighting of the contribution from 'boundary' pixels has been corrected to a smooth function. The function previously had two discontinuities at the top and bottom edges of the extraction channel—introducing a 'jump' in some data.
- Errors in the pixel-weighting scheme implementation and pixel selection in the blaze fitter have been removed.
- ECHMENU Option 1.2 (`ech_decos1`) is now allowed to be an automated step. Previously the value of `TUNE_CRTRC` was always taken as `NO` in this case. In the same way, Options 11.3 and 11.4 now also work as elements in a `TUNE_AUTOMATE` request.
- NDF_SQMF has been used to switch off automatic QUALITY component checking as ECHOMOP does this itself.
- The copy-last-plot-to-hardcopy device option in `ech_plotter` now works.
- Several divide-by-zero opportunities in the 2-D sky modeller have been removed.
- The 2-D sky modeller now handles bad-pixel values in input data.
- The routine which identifies order numbers now works, this helps automatic wavelength calibration proceed faster.

Index