CORONAL DIAGNOSTIC SPECTROMETER

**SoHO**

**CDS SOFTWARE NOTE No. 39**

Version 1.3 14th December 1995

# Automatic Line Fitting to CDS Spectra

B J Kellett
Rutherford Appleton Laboratories
Chilton, Didcot
Oxon OX11 0QX

bjke@astro1.bnsc.rl.ac.uk

# 1 Introduction

Given the expected large quantity of CDS spectral data likely to be taken, it is desirable to have a quick and easy method of reducing a CDS spectrum to a set of identified lines and estimated intensities. This note describes one such method and how it can be used to directly fit CDS spectra. Later in the note a direct application of some of these fitting routines is discussed to illustrate how the software can be applied.

# 2 Automatic Fitting Routines

## 2.1 PLOT_SPEC

At present, the only way into the automatic line fitting routines is via the PLOT_SPEC procedure. This takes the full CDS FITS data structure and allows the user to select by WINDOW or BAND, EXPOSURE, and DET_Y (or SOLAR_Y if the SOLAR keyword is supplied), and then display the resulting spectrum. (The user also gets the option to display in WAVELENGTH (default) or PIXELS, and also to make a hardcopy of the plot. However, clearly, a spectrum in pixels would not be very useful for performing spectral line identifications!). The procedure currently has five optional keywords:-

**INFO** The procedure will print the actual GT_SPECTRUM command used to extract the displayed spectrum.

**HARDCOPY** Instructs the procedure to automatically produce a hardcopy of the extracted spectrum (the default action is to ask the user if a hardcopy is required).

**OUTPUT** Allows for the extraction of the displayed spectrum into a simple ASCII file.

**SOLAR** Instructs the procedure to use SOLAR_Y positions rather than the default DET_Y coordinates.

**SPEC** Instructs the procedure to return a structure containing the extracted wavelengths and fluxes. It is this structure that the fitting routines use.

**NB:** The selection by DET_Y or SOLAR_Y *will* accept ranges or multiple entries. So, for example, entering "67:75,90:99" would **average** the 9 spectra from 67 to 75 *and* 10 spectra from 90 to 99 (you don't enter the quotes!). Alternatively, you might enter "67,75,100" to average the three spectra indicated, or "67:75 99" would produce the average of the ten spectra 67–75 + 99. The ":" denotes a range and a "," or " " denote separators.

So, the actual IDL commands required to extract a spectrum ready for doing an automatic fit would be:-

```
% a = readcdsfits ( filename )          !read in a CDS data structure
% plot_spec,a,spec=sp
                                         !A summary of the input data
                                         !is displayed and various
                                         !options have to be selected.
```

In the above example, PLOT_SPEC would return a structure (sp) that currently contains two fields – sp.w is the wavelength and sp.flux is the spectrum in whatever units are in the original data structure. It is this data structure that is required for input to the fitting routines.

Automatic software like that described below cannot be expected to make judgements on exactly which element or atomic transistion is *actually* responsible for a given "observed" line. However, it should be able to *indicate* which line or lines are closest to a given observed feature. Since a great deal of effort went into the preparation of the CDS "Blue Book" line list, the fitting software defaults to selecting these lines for comparison with the "observed" lines. It should be noted, however, that the complete tabulation of Kelly is also available. So, in the descriptions given below, be aware of which line list is actually being used. It will be either the "Kelly" or "Blue Book" list (normally indicated by a "/k" or "/b" switch).

## 2.2 CDS_AUTOFIT

CDS_AUTOFIT is designed to be a first "quick look" at a CDS spectrum. It takes as input a data structure such as returned by PLOT_SPEC (for instance), and then, with NO input from the user, proceeds to try and find any significant (emission) lines in the data and fit them with simple gaussians on a quadratic background. It will optionally return a structure containing the fitted parameters (needed if the data is to be re-plotted, for example).

```
% cds_autofit, sp [, result = gfit ]
```

In doing the fitting and "selecting" the lines, CDS_AUTOFIT makes various assumptions on line-width and other parameters. For example, it rejects any lines it finds which are either too narrow or too broad (in its opinion!).

The fitted data is displayed by CDS_PLOT_FIT. If the user would like to redisplay the plot, the actual call that CDS_AUTOFIT makes to CDS_PLOT_FIT is:-

```
% cds_plot_fit, sp, results = results, /fill, /bluebook, /log
```

## 2.3 CDS_PLOT_FIT

This is a general routine for plotting a CDS spectral structure as defined by the PLOT_SPEC routine. But it is also a lot more than that! It can plot on a log or linear y (flux) scale; it can be restricted to only plot a given wavelength interval; and it can overlay the CDS Blue Book or Kelly set of lines. In addition, if the fitted parameters are supplied, it can also plot the gaussians and continuum for each of the lines that CDS_AUTOFIT found in the range displayed. CDS_PLOT_FIT takes six optional keywords:-

**FILL** Controls whether the fitted profiles are displayed or not (assuming the RESULTS keyword is also supplied). If FILL is not supplied, but the fitted results are given, only the positions of fitted lines are indicated (by arrows).

**BLUEBOOK** Specifies that the CDS "Blue Book" line list should be used/plotted.

**Kelly** Specifies that the Kelly line list should be used/plotted.

**LOGPLOT** If set, the spectrum is plotted on a log scale. (Only applies to the y-axis).

**RESULTS** Supplies the fitted parameters from CDS_AUTOFIT. (See FILL [above] for more details).

**WDIFF** Only applies if the KELLY or BLUEBOOK keyword is set. Sets the wavelength difference to search for line identifications. The default value is +/- 1.75 pixels [IN ANGSTROM UNITS] of the input spectrum.

**WRANGE** If set, can supply low and high wavelength pairs for the wavelength range. Multiple pairs are allowed.

**ALL** Used with the KELLY and BLUEBOOK keyword to display all the lines in the list which lie in the wavelength range displayed.

[NB: By default, the results structure returned by the various line fitting procedures will contain the Blue Book lines. So CDS_PLOT_FIT can use these for immediate plotting (if the /bluebook switch is specified *WITH* the results=r switch). But CDS_PLOT_FIT can also read in either set of lines if necessary - which is always the case with the Kelly line list].

So, some examples might be:-

```
% cds_autofit, sp , results = r     !do fitting and return results.
% cds_plot_fit, sp                  !just plot the spectrum.
% cds_plot_fit, sp, results = r     !plot spectrum, and indicate lines found
% cds_plot_fit, sp, r = r, /fill    !plot spectrum, and fill fitted lines.
                                    !fitted line intensities will also be
                                    !plotted.
% cds_plot_fit, sp, /kelly, /all    !plot spectrum and show all Kelly lines.
                                    !/all is needed because the default action
                                    !is to only identify lines close to fitted
                                    !lines, and there are none in this case!
% cds_plot_fit, sp, r = r, /f, /b   !plot spectrum, fill fit lines, and
                                    !display closest Blue Book lines.
% cds_plot_fit, sp, r = r, /f, /k   !plot spectrum, fill fit lines, and
                                    !display closest Kelly lines.
% cds_plot_fit, sp, r = r, /f, /l   !plot log spectrum, fill fit lines.
% cds_plot_fit, sp, wr = [wl1,wh1,wl2,wh2] !plot spectrum in two ranges;
                                    !from wl1 to wh1 and wl2 to wh2. (Get two
                                    !separate plots on same page).
% cds_plot_fit, sp, r = r, /f, /b, wd=0.25  !plot spectrum, fill fit lines,
                                    !and display CDS Blue Book lines out to
                                    !+/- 0.25 A.
```

(NB: Keywords can be shortened to only one or two letters, provided the shortened form is still unique! WRANGE will be automatically set if the results file is from CDS_FIT_WAVE).

### 2.4 CDS_FIT_WAVE

This is a somewhat more specific form of CDS_AUTOFIT in that only a single line or set of specified lines is/are fit, and the user specifies the precise region(s) to be fitted. The procedure allows the user to control the actual range of data to be fit by giving a central wavelength and the option to set the "width" of the fitted region (data window). It also allows the user the ability to adjust the "sensitivity" or "threshold" of the fit. (This, basically, sets the height above the continuum that a feature must exceed for the fitting procedure to consider that there is a line present). However, after that the fitting procedure is still essentially automatic. The width of the data window should not be too large or too small. Values in the range 12–50 pixels are probably ok, but since the automatic fitting procedure can only fit a maximum of 3 lines in any given range, larger widths may cause problems. CDS_FIT_WAVE will then attempt to fit the data with from one to three gaussians to the region(s) of the spectrum specified, display the results and also supply some details of the fits. The actual "work engine" is the same as used by CDS_AUTOFIT (a routine called CDS_MULTIG), but "ordinary" users shouldn't normally need to know that!

```
% result = cds_fit_wave ( spec, wcentre [, width = npixels ] )
% result = cds_fit_wave ( spec, wcentre [, threshold = thres ] )
% result = cds_fit_wave ( spec, [w1,w2,w3,...] [, width = npixels ] )
                                  ^^^^
                    i.e. fit multiple lines
```

For each region specified, the fitting procedure will display the data and fit (or fits). Details will also be given in the IDL input window. The plot displays TOTAL INTEGRATED COUNTS for each fitted line (with estimated errors from the IDL "CURVEFIT" routine). The intensities *above* the line profiles are the **fitted** values. Below each fitted profile, the **"observed"** intensity is also given. This is the sum of the observed channels (indicated below the fit with either triangle, square or diamond symbols), MINUS the (sum of the) fitted background/continuum. These results are also placed in the results structure.

The results structure returned contains all the details of the fit. The structure is very similar to that returned by CDS_AUTOFIT. (In fact, the two are identical except that the CDS_FIT_WAVE structure also contains details of the fitted wavelength range(s) used so that CDS_PLOT_FIT can automatically display the results correctly). CDS_PLOT_FIT can once again be used to (re)display the results, and can handle up to 12 separate "regions" – i.e. the user could display 12 separate line fits on one page. Obviously, the more data windows to plot, the smaller the resulting plots! At present, the default plotting mode for multiple regions is to scale ALL the plots to the SAME scale – i.e. the scale is determined by the maximum line intensity. This is to allow weak and strong lines to be compared on the same plot. If the spectrum is a full NIS band (unwindowed), the plot will automatically be divided into 6 ranges and displayed on two pages.

### 2.5 Test Data Files

In order for people to try out some of the above routines they will need some test data of some kind (until the real data becomes available anyway!). At present there is one NIS and one GIS test data file. The NIS file contains three windowed regions of the two bands (two in Band 1, 317.67–324.84

ÅÅ and 350.12–357.29ÅÅ, and one in Band 2, 560.12–571.70 ÅÅ). The GIS test file contains 6 windows and a 5×5 data array.

**NIS:** /cs/data/nis/test_nis_1.fits — NIS test data file.

**NIS:** /cs/data/gis/test_gis_1.fits — GIS test data file.

# 3 An Example Application of the Automatic Fitting Routines

In the tests performed so far, on "simulated" data at least, the automatic fitting software seems to be able to get some quite respectable fits to the data – certainly for the stronger lines. But, if a user wanted to fits lots of lines, the whole procedure might get somewhat laborious. For anyone willing to dig into the details of the various routines discussed above, it is quite possible to construct powerful applications of their own.

An example is given below. Basically, it provides a procedure for performing velocity slices through data cubes. In essence, it is a straight marriage of PLOT_SPEC and CDS_FIT_WAVE with a few lines of code inbetween to select which line to fit and then use the full Kelly line list to make the line identification (while also indicating which, if any, of the Kelly lines are also in the Blue Book line list!).

## 3.1 CDS_VEL_SLICE

CDS_VEL_SLICE is a function to return fitted wavelengths and line intensities and derived velocities for a CDS data cube. It uses essentially the same interface as PLOT_SPEC to assist the user to select the particlular data window and exposure (or exposures) to use. Having made some basic choices, the "first" spectrum will be displayed, and the user prompted to enter a chosen wavelength. This can be fairly approximate to start with. The procedure will then uses some of the features of CDS_PLOT_FIT to display the particular line of interest and the possible line identifications for the line to assist the user to select the correct rest wavelength for the feature of interest (to convert from fitted wavelength into velocity, a rest wavelength is required). If none of the lines are correct, the "0" option is always to enter the required details "by hand". The next option is whether to recenter the data window. The choices are to recenter on the actual rest wavelength (the "K" option), to re-enter the central wavelength from the keyboard or to continue with the current value. This can be important, since the fitting procedure tends to give more reliable results if the feature of interest is fairly centrally located.

Next, the width of the data window can be set. Care should be taken not to include to many other lines in the region to be fitted, but, at the other extreme, to supply enough data to fit both the line and continuum well. The first fit will then be shown (as an example of the likely quality of the results CDS_VEL_SLICE might be able to get) and the option is then to either proceed or try another central wavelength.

Once the user is happy with all the choices made, the procedure will then go away and assembles the required spectra, do the automatic line fitting and (hopefully!) returns with a three-panel display of the results. It will tabulate the "current" Solar X and Y coordinates, so that the user can see how well the procedure is progressing. Since the IDL CURVEFIT routine is at the very heart of the fitting engine, and, by its very nature, curve fitting can be a fairly hit or miss affair, it is quite possible that the fitting process will get "stuck" somewhere. The reasons for this are not always entirely obvious! A little more detail can be gleaned by adding a "debug" switch when calling CDS_VEL_SLICE. This will display the details of each fit in a little detail and might give a clue as to the problem. One can then try again, perhaps with a different central wavelength or data window width.

Assuming the fitting process went OK, the results are then displayed by a second routine – CDS_VEL_PLOT (which currently possesses no options!). The top panel should be a plot of the fitted wavelengths of the requested line either along the NIS slit direction (DETY pixels 0–119 but in SOLAR_Y arc-minutes), or in the time direction for a specified subset of DETY pixels (or SOLAR_Y if the /solar keyword is used). The middle panel takes the line identification details and converts the fitted line wavelengths into velocities (km/s), and the bottom panel shows how the fitted line intensity varies. If the spectra are in the time direction, the plot should also indicate the SOLAR_X values equivalent to the time axis (assuming that the raster was also moving in the X-direction).

This option to examine the data cube in the time dimension is switched on by an answer of "−1" to the Exposure question in the data selection/input part of the procedure (without the quotes!). The procedure will then prompt for DETY (or SOLAR_Y) values (either a single pixel m, or a range of pixels m:n, or any combination thereof).

```
  % result = cds_vel_slice ( a )            !a = cds data cube.
  % result = cds_vel_slice ( a, /solar_y )  !select on Solar Y rather than the
                                            !default DETY pixels.
  % result = cds_vel_slice ( a, /debug )    !plot details of each fit.

    In each of the above cases, multiple questions must be answered, mostly
with "sensible" defaults.

  % cds_vel_plot, result                    !redisplay fit.
```

(NB: CDS_VEL_PLOT currently as no switches. However, it is likely to gain more options in the near future, perhaps to specify which plot panel to display for example).

### 3.2 Batch Mode Operation

As described above, CDS_VEL_SLICE is very much an interactive procedure. However, it is quite likely that users will become aware of which input values work for a given line and wish to enter these values on the command line and bypass the interactive procedure. This mode of operation is allowed for, and is termed "batch operation". Some or all the necessary inputs can be composed on the command line and the procedure (in full batch mode) would provide some minimum level of basic output (but no display would be produced). This mode does require the user to have some basic understanding of the "correct" inputs necessary to produce useful results. If only a subset of the required input parameters are supplied, the user is prompted for the remained values.

The full list of optional keyword for CDS_VEL_SLICE is given below:-

**BATCH** Puts the procedure in batch mode and suppresses much of the user interaction and display parts of the procedure.

**DBAND** Currently not available as a user option, it would allow the selection of which full detector band to use. See the WINDOW keyword description below.

**DEBUG** Sometimes, the fitting procedure can "get stuck" somewhere in the CURVEFIT routine, from which it usually never returns! Specifying the DEBUG switch requests the fitting procedure to display some details of the fitting process in order that the cause of the problem may be assessed. Typically, the user needs to adjust the data window width or shift the central wavelength of the fitting region.

**DWIDTH** Set the width of the DATA WINDOW in pixels. This is the region of the full window that is actually used to for the fitting. The width should be small enough to avoid undue confusion with neighbouring lines, but large enough to include enough of the line profile and surrounding continuum to get a good overall fitted profile.

**EXPOSURE**$^\star$ Select which exposure to use. The input "$-1$" switches the CDS_VEL_SLICE mode from "velocity profile along the NIS slit" to "velocity with time and/or Solar_X".

**HARD** Instructs the procedure to automatically produce a hardcopy of the final results plot (the default action is to ask the user if a hardcopy is required, unless in batch mode).

**RECENTER** This is either "N" – no rencentering of the data window is requested, "Y" – recentering is required, user prompted for input (probably not feasible in BATCH mode!?), or "K" – recenter an the Kelly line identification. Default is currently N (input should be a string).

**SOLAR_Y** Instructs the procedure to use SOLAR_Y coordinates rather than the default DET_Y coordinates.

**WCENTER** Approximate central wavelength to use. The data window is chosen around this value (width specified by the DWIDTH keyword) and the exact value can be reset with the RECENTER keyword.

**WINDOW**$^\star$ Select which data window to use.

**YRANGE** A string input that specifies which Y data to use (if the EXPOSURE is set to $-1$ – "Time Slice"). This can be a single value, a range of values, or any combination. If the SOLAR_Y keyword is set, the coordinates should be specified in Solar_Y arc-seconds. The default is the DET_Y pixel values.

$^\star$A "feature" of IDL with regard to Key Words is that the input value "0" is NOT recognised. Since both the Window number and Exposure can both be zero, some alternative mechanism needs to be adopted to allow for this. For WINDOW **any** $-$ve input is taken to mean Window 0. For the EXPOSURE keyword, it is a little more complicated. An input of $-1$ switches to the "time slice" mode (see above), and **any other** negative number is taken to mean Exposure 0 is required.

Some example calls might be:-

```
1)     % r = cds_vel_slice ( a, win=1, exp=10, wc=345.6, dw=33 )

  Enters some values, but user will still need to choose the Kelly Line ID,
  answer the RECENTER question, confirm the first fit, and reply to the
  ``hard copy'' question after the results are displayed.
```

2) % r = cds_vel_slice ( a, win=1, exp=10, wc=345.6, dw=33, /b )

This will suppress most of the above interaction, except the procedure will still request an answer for the RECENTER question.

3) % r = cds_vel_slice ( a, win=1, exp=10, wc=345.6, dw=33, recen = 'k', /b )

Will proceed without any further input, and no results will be displayed.

4) % r = cds_vel_slice ( a, win=1, exp=-1, wc=345.6, dw=33 )

This will proceed like 1) except that the YRANGE will also be requested. And now the velocity fits will proceed in the TIME direction (exposure -1 option).

5) % r = cds_vel_slice ( a, win=-1, exp=1, wc=345.6, dw=33, recen = 'k', /b )

Selected Window *0* and Exposure 1 (the *second* exposure!).

6) % r = cds_vel_slice ( a, win=-1, exp=-2, wc=345.6, dw=33, recen = 'k', /b )

Selected Window *0* and Exposure *0* (the *first* exposure!).

7)
```
% r = cds_vel_slice ( a, win=-1, exp=-1, wc=345.6, dw=33, recen = 'k',   $
                      yrange = '24:32', /b )
```

Fully automatic in the Time direction of the data cube.