CLASS
Continuum and Line Analysis Single-dish Software

A GILDAS working group software

February 6, 2002

Continuum and Line Analysis Single-dish Software
or

Sensible
Simple

Continuum and Line Analysis Sophisticated System (*)
Sympathetic
Super

Version 4.0

The GILDAS working group is a collaborative project of the Observatoire de Grenoble (1,3) and IRAM (2), and comprises: G. Buisson¹, L. Desbats¹, G. Duvert¹, T. Forveille¹, R. Gras³, S. Guilloteau¹,², R. Lucas¹,², and P. Valiron¹.

(1) Laboratoire d’Astrophysique
    Observatoire de Grenoble
    BP 53 X
    414 Rue de la Piscine
    F-38402 Saint Martin d’Hères CEDEX

(2) Institut de Radio Astronomie Millimétrique
    300 Rue de la Piscine
    F-38406 Saint Martin d’Hères

(3) CEPHAG
    Observatoire de Grenoble
    F-38402 Saint Martin d’Hères CEDEX

Contributions from and invaluable discussions with J. Cernicharo, P. Begou³, S. Delahaye³, A. Dutrey¹,², C. Kahane¹, P. Monger, J.L. Monin¹, R. Neri² and all GILDAS users are gratefully acknowledged.
# Contents

1 Introduction ........................................... 7

2 Cookbook .............................................. 8
   2.1 Getting on the air .................................... 8
   2.2 Adding Spectra ....................................... 8
   2.3 Visualization ......................................... 9
   2.4 Calibration .......................................... 9
   2.5 Analysis ............................................. 9
   2.6 Writing .............................................. 9
   2.7 End of session ....................................... 9

3 CLASS Manual .......................................... 10
   3.1 More about CLASS ................................... 10
      3.1.1 Files ............................................ 10
      3.1.2 R and T Memories ................................... 10
      3.1.3 Observation and Version Numbers .................. 10
      3.1.4 Adding Spectra .................................... 10
      3.1.5 Selection Criteria ................................... 11
      3.1.6 The Plot ........................................... 12
      3.1.7 Baselines ......................................... 13
      3.1.8 Frequency Switching ................................ 13
      3.1.9 Profile Analysis .................................... 13
      3.1.10 Miscellaneous ..................................... 15
   3.2 Continuum Processing ............................... 15
      3.2.1 Introduction ...................................... 15
      3.2.2 Differences between Continuum and Spectroscopy Mode 16
   3.3 Skydip Processing .................................... 16
   3.4 On-the-fly Data Processing .......................... 16
   3.5 Map Processing: GILDAS .............................. 17
      3.5.1 Maps of Spectra ................................... 17
      3.5.2 GILDAS Interface .................................. 17
      3.5.3 Contour maps: Building a data cube ............... 17
      3.5.4 Velocity-Position Plots ............................ 17
      3.5.5 Continuum Maps ................................... 17
   3.6 The PRINT Command .................................. 18
      3.6.1 GREG Tables ...................................... 18
   3.7 Publication Quality Plots ............................ 19
      3.7.1 Hardcopy ......................................... 19
      3.7.2 Annotations ....................................... 19
      3.7.3 Plot Quality ....................................... 20
   3.8 Variables ............................................ 20
      3.8.1 The Rope to Hang Yourself ........................ 20
      3.8.2 Index Variables ................................... 20
      3.8.3 Default Header Variables .......................... 21
      3.8.4 Advanced Processing .............................. 22

4 From CLASS to FITS ..................................... 23

5 LAS Language Internal Help ............................ 27

6 ANALYSE Language Internal Help ..................... 28
## CONTENTS

### 7 FITS Language Internal Help
- 7.1 DIRECTORY ......................................................... 29
- 7.2 DISMOUNT .......................................................... 29
- 7.3 INITIALIZE .......................................................... 29
- 7.4 LIST ................................................................. 29
- 7.5 MOUNT ............................................................... 29
- 7.6 READ ................................................................. 29
- 7.7 REWIND ............................................................. 30
- 7.8 SELECT ............................................................. 30
- 7.9 SKIP ................................................................. 30
- 7.10 WRITE .............................................................. 30
- 7.11 END ................................................................. 30

### 8 GTVL Language Internal Help .................................................. 32

### 9 Error Messages and Recovery Procedures ................................. 33
- 9.1 Notifying the Programmer or System Manager .......................... 33
- 9.2 Recovering Corrupted Data Files .................................... 33
- 9.3 LAS and ANALYSE Messages and Recovery Procedures .......... 34
  - 9.3.1 A and B ......................................................... 34
  - 9.3.2 C ................................................................. 36
  - 9.3.3 D and E ......................................................... 38
  - 9.3.4 F ................................................................. 39
  - 9.3.5 G through I ...................................................... 41
  - 9.3.6 K and L ......................................................... 42
  - 9.3.7 M ................................................................. 43
  - 9.3.8 N-Q ............................................................. 45
  - 9.3.9 R ................................................................. 47
  - 9.3.10 S ............................................................... 48
  - 9.3.11 T through Z .................................................. 50
1 Introduction

CLASS is a software package for reducing spectroscopic data, and also continuum drifts obtained on a single-dish telescope. CLASS supersedes the older LAS program which was restricted to spectroscopic data.

The originality of CLASS with respect to similar systems already in use is in the way an observation may be identified. In addition to the traditional scan number which can be used to uniquely refer to an observation, the system also enables one to use Selection Criteria as in a data base management system. This faculty, added to a powerful command monitor, SIC, allows easy manipulation of large volumes of data; the list of observation numbers to be added to get the mean spectrum at one position need no longer be typed in, but may be found by CLASS itself.

CLASS is divided in different parts, called “Languages”, which have somewhat independent functions (*). Language LAS contains all the general utility functions to handle the data structure, plot the spectra or drifts and calibrate them. Language ANALYSE contains functions to analyze calibrated spectra in more detail. Language GTVL includes commands for basic graphic actions performed by the graphic library used by CLASS; the same library is used by GreG.

On a standard installation, CLASS is entered by just typing CLASS. In addition to this manual, the reader should consult the SIC manual, and for further processing, the GILDAS and GreG manuals.
2 Cookbook

This part is a list of recipes enabling the beginner or the occasional user to get on the air very rapidly, without losing his time searching the system’s on-line HELP facility. A CLASSic session is the following:

```
DEVICE XLAND WHITE       ! 1
FILE IN BRU.BUR          ! 2
SET LINE 13CO(1-0)        ! 3
SET SOURCE IC348         ! 4
SET TELESCOPE IRAM-30M-B30 ! 5
SET OBSERVED 15-AUG-1984  ! 6
SET ANGLE SECONDS        ! 7
SET COORDINATES EQUATORIAL ! 8
FIND/OFFSET 0 25         ! 9
SUM                      ! 10
!
SET UNIT VELOCITY         ! 11
SET MODE X -1 14         ! 12
SET MODE Y -0.5 7.5      ! 13
SET PLOT HISTOGRAM       ! 14
PLOT                      ! 15
HARDCOPY/PLOT            ! 16
!
SET WINDOW 3 6 8 10       ! 17
BASE 4/PLOT               ! 18
PLOT                      ! 19
!
LINES 0                   ! 20
GAUSS                     ! 21
FIT                       ! 22
RESIDUAL                  ! 23
PLOT                      ! 24
!
FILE OUT REDU1T.3OM NEW   ! 25
SWAP                      ! 26
WRITE                     ! 27
!
SAVE IC348                ! 28
EXIT                      ! 29
```

2.1 Getting on the air

CLASS needs a data file, as well as (usually) a graphic output device. The first thing is then to define the device to be used, (command 1), here an X-window terminal, on which an landscape style window with white background is created. The input file containing spectra is then opened (command 2).

2.2 Adding Spectra

Usually one begins by adding up the individual spectra obtained at a single position on the sky. To do this, define the Selection Criteria of the spectra, including the Line Name (3), the Source Name (4), the Telescope and backend configuration used (5), and the date of observations (6). In this example we consider all the observations of the 13CO(1-0) transition performed using the IRAM 30-m telescope in the direction of IC348 on 1984 August 15th. Commands (7) and (8) further define the angular coordinate
system to be used, and the corresponding units. Executing the FIND command (9) builds up an Index of these observations, where the selection criteria defined by the previous SET commands are further extended by the OFFSET option which limits the search to the position (0.25.) (in equatorial coordinates and arc seconds, as defined by (7) and (8)). The SUM command then adds up all the spectra of the current index.

2.3 Visualization

To see the result, command (11) defines the X axis units (here Velocity, but channel number or Frequency may be used). Then commands (12,13) define the plotting limits (in Velocity and Antenna Temperature respectively). Command (14) specifies that the spectrum will be plotted as a histogram, instead of the simpler, but less realistic, segmented line. The PL0T command (15) draws the axes, labels, the spectrum and its title. Each of these operations may be performed separately if needed.

A hardcopy of this plot can be obtained as in command line (16). This creates a file which is automatically sent to a plot spooler. The plot appears on the local system plotter device.

2.4 Calibration

Generally the spectrum baseline is not perfect. One may thus want to subtract a baseline, calculated by a least-squares fit, excluding the intervals defined in command line (17) 3-6 and 8-11 (in velocity units according to (11)) in which the observed line is supposed to be present. This is done by command line (18), which computes a 4th degree polynomial (argument of the BASE command), plots the result superimposed on the preceding plot, then subtracts this polynomial from the spectrum. The original version of the spectrum is copied in the second "memory" of the system (T memory). The result of the baseline subtraction, in memory R, is displayed by the PL0T command (19). Note that the limits defined in (12,13) stay in use.

2.5 Analysis

If the result is satisfactory, the spectrum is further analyzed by more sophisticated means, such as model-fitting by Gaussian curves. One defines the number of curves to be used (20); 0 means the automatic minimization with a single Gaussian, without initial values for the parameters. The GAUSS command (21) activates the minimization algorithm, which finally converges (yes it does!). One plots the fitted Gaussian by the FIT command (22). The residuals may also be determined by the RESIDUAL command (23) which first copies the R "memory" into the T "memory", then subtracts the Gaussian. These residuals are then plotted (in the current limits) by the PL0T command (24).

2.6 Writing

If everything seems OK, one usually saves the results of the fit. One must first open an output file (25); in this example the file is created by means of the last argument "NEW"; to open an existing file, omit it. Then go back to the true spectrum, not the residuals, by exchanging "memories" R and T by the SWAP command (26), and copy the contents of the R "memory" onto the output file by the WRITE command (27). The limits of the plot, the baseline, the Gaussian parameters as well as the list of observations used to obtain this final result are automatically saved in the same operation.

2.7 End of session

You can simply get out by EXIT (29), but if you want to keep the current "environment" of CLASS, e.g. all the parameters defined by SET commands, use the SAVE command (28), which writes the corresponding SET commands on a command file (here IC348.CLASS). Later you will be able to re-execute all these commands and rebuild the previous environment, by entering : @ IC348, thus avoiding a lot of tedious typing.

That's it, you now know (nearly) everything. Good luck...
3 CL\emph{ASS} Manual

3.1 More about CL\emph{ASS}

3.1.1 Files

CL\emph{ASS} uses two files of data; one for input and one for output, which may be the same actual file. The input file is only used to read. An observation contains several independent sections. These file are defined by the command \texttt{FILE (IN,OUT,BOTH) Filename}, possibly followed by \texttt{NEW} if a new file is to be initialized. The default extension of files is site dependent: it is .\texttt{30m} at Pico Veleta, .\texttt{BUR} at the Groupe d\textquotesingle Astrophysique in Grenoble and on Plateau de Bure, and so on.

Commands \texttt{FIND} and \texttt{GET} operate only on the input file. Command \texttt{WRITE} operates only on the output file. When both are identical, \texttt{WRITE} creates physically a new version of the spectrum in the file, keeping the previous one. In some cases one may override the old version by the new one by the \texttt{UPDATE} command. To do this, the file must be used for both input and output, and each of the sections of the new version must fit in the space used by the old one. Other commands are subject to similar restrictions, in particular \texttt{KEEP} to save the gaussian fit results. If you want to play safe, always use \texttt{WRITE}.

CL\emph{ASS} also keeps a log file (named \texttt{SY$\$$LOGIN:CL\textbackslash emph{ASS\.LOG}, created by the SIC monitor} and a message file \texttt{SY$\$$LOGIN:CL\textbackslash emph{ASS\.MES}, which are systematically purged at the end of each session. These files may be used to keep track of a batch or interactive work. Finally the session environment may be saved as a command file containing a set of \texttt{SET} commands.

3.1.2 R and T Memories

CL\emph{ASS} keeps 2 observations in memory, one in an area called the R memory, the other in the T memory. The R memory is the only one that may be accessed directly; the T memory is only used for operations on spectra (additions,...). This works like a HP calculator. The command \texttt{SWAP} exchanges both memories.

3.1.3 Observation and Version Numbers

Within CL\emph{ASS}, an observation should represent a single observing configuration, e.g. for spectra a single direction observed at a single central frequency with a single spectral resolution and in one polarization only. All versions of a given observation represent different stages of the data reduction. Each time the observation is modified (using \texttt{UPDATE} or \texttt{WRITE}) the version number increases. In principle, in CL\emph{ASS}, only the last version of a given observation is relevant. Previous versions are never deleted, with the exception of \texttt{UPDATE} command which overwrites the current version, so you can go back to previous stages of reduction in case of big mistakes.

Provided you respect this use of the version number, data reduction can be largely automated. Failing to do so, i.e. using the same observation number for very different things, implies that you have to remember yourself which version corresponds to which configuration. This was unfortunately the case for previous releases of the OBS\emph{erving program on the 30-m (See CAL manual), but should be solved in the new version for which Scan numbers and Observations numbers are different. Note that, for bookkeeping purpose, CL\emph{ASS} keeps also track of a Scan number, which can be used as a selection criterion.

3.1.4 Adding Spectra

Four parameters define the way spectra are added. These are the align mode, the combination mode, the integration weighting, and the behaviour with respect to bad channels.

Four alignment modes are available, by the means of the command

\texttt{SET ALIGN Mode :}

- \texttt{CHANNEL} in which spectra are added channel by channel. This is only useful when the spectra have been obtained in strictly identical conditions. Warning messages are given when this is not the case.

- \texttt{VELOCITY} in which the velocity scale is used to align the spectra. This enables you to add spectra of different origin. An interpolation is performed if needed. If individual spectra have differing spectral resolutions, the lowest spectral resolution is used for the result.
• FREQUENCY in which the rest frequency is used to align the spectra.
• POSITION, in which continuum drifts are aligned regarding to the position along the drift.

CHANNEL and POSITION only are relevant for Continuum observations, while CHANNEL VELOCITY and FREQUENCY are relevant for Line observations.

Two combination modes are possible with the command

**SET ALIGN MODE Combination:**

• INTERSECT where only the intersection of individual spectra is kept
• COMPOSITE where the reunion of the individual spectra is kept (as in a spectral scan for example)

Three weighting types may be used, with the command

**SET WEIGHT Type:**

• TIME for weights proportional to the observing time, divided by the square of the system noise;
• SIGMA for weighting by the inverse square of the rms noise of each individual spectrum.
• NONE or EQUAL for equal weighting. Caution: equal weighting behaves differently in SUM and ACCUMULATE commands. SUM produces the average of spectra, while ACCUMULATE gives the sum of the two spectra. After division by the number of added spectra, ACCUMULATE will thus give the same result as SUM.

Bad channels are dealt with in two possible ways, defined by the command

**SET BAD Mode:**

• OR where resulting channels are declared bad if they were declared as such in at least one of the individual spectra;
• AND where resulting channels are declared bad if they were bad in all individual spectra.

Default values are ALIGN CHANNEL INTERSECT, WEIGHT TIME, and BAD OR.

Two other parameters control whether summing spectra is allowed or not. Positions are checked according to SET MATCH Tolerance or SET NOMATCH. If (absolute) positions differ by more than the tolerance parameter, an error message is generated. The tolerance is specified in current angle units. The homogeneity of the calibration is checked according to the SET CALIBRATION Beam_Tolerance Gain_Tolerance or SET CALIBRATION OFF commands. Beam_Tolerance is the maximum difference allowed in the beam efficiencies to add spectra (default 0.02) and Gain_Tolerance the maximum difference between the gains in the image band (default 0, which means not checked).

There are two ways of adding spectra: the commands SUM and ACCUMULATE. SUM operates globally on all the spectra in the index, while ACCUMULATE adds the R and T memories into R. SUM is generally better for systematic methods, ACCUMULATE for special cases. The drawback of ACCUMULATE is in the need for initialization; one needs a spectrum in T and a spectrum in R to begin with ...

### 3.1.5 Selection Criteria

Default selection criteria are defined by the SET command. For most selection criteria, an option to the FIND command exists, with the same name, which may be used to impose temporary values to the FIND command; the default values are unchanged by the FIND options.

• SET TYPE Name is more than a selection criterium, since it specifies on which type of observations the CLASS program works. Name can be “Continuum”, “Line” or “Spectroscopy”, or “Skydip”.

• SET LINE Name for the line name to be used. A line name of the form ABC indicates that all lines beginning by ABC are to be selected. The default is *, i.e. any line name.

• SET NUMBER n1 n2 for the range of observation numbers. Default is * *, i.e. any observation number ; * n2 specifies all observation numbers smaller than n2.
• **SET OBSERVED** d1 d2 for the range of observing dates. A date is specified in the format dd-mmm-yyyy, e.g. 19-jan-1985. Default is * *, i.e. any date ; 19-JAN-1985 * means any date later than January 19th, 1985.

• **SET OFFSET** o1 o2 for offsets of the position to be used (in the system and units specified by SET COORDINATE and SET ANGLE). Default is * *.

• **SET RANGE** w e s n is a less restrictive way to specify position offsets. A rectangular area of sky is defined by its west, east, south and north limits (in current angle units).

• **SET REDUCED** d1 d2 for a range in reduction dates ; same specifications and defaults as for SET OBSERVED.

• **SET SOURCE** Name for the source name ; same specifications as SET LINE.

• **SET SCAN** s1 s2 for a range of original scan numbers. Scan numbers should not be confused with Observation numbers (the numbers by which an observation is uniquely identified). They are essentially “historical” numbers defined by the acquisition system, but usually with different “observations” (in the CLASS meaning) for a single scan. The scan number is kept only for bookkeeping purpose.

• **SET TELESCOPE** Name for the Telescope name. For the IRAM 30-m telescope, the telescope name contains coded into the last 3 letters the backend used for the observations. Similar conventions are used for the POM-2 and IRAM 15-m telescopes.

The tolerance parameter defined by **SET MATCH** also influences on the position searches, since this parameter (in the current angle unit) is used to check agreement with the specified limits. Another option to the **FIND** command is **/ALL** which enables one to find all the versions of of observations satisfying the selection criteria (otherwise only the most recent version is selected). Note that the system is intended to work only with the last version of observations, so that the use of the **/ALL** option should remain exceptional.

### 3.1.6 The Plot

Plotting data is done by using **GreG**. All **GreG** commands are available in **CLASS**.

Plotting spectra is controlled by several parameters

• **SET UNIT** type defines the unit of the X axis, which may be C (for Channel number), V (for Velocity), F (for Frequency) or I (for Image).

• **LAS\SET PLOT** **Type** defines the plotting type **PLOT** (Normal or Histogram) ; Normal gives straight lines connecting the data points (this is the default since it is faster). Histogram gives a more realistic representation of spectroscopic data.

• **SET MODE** X (or Y) **Type** defines the plotting limits in X or Y, where type stands for **TOTAL** (all channels plotted in X, complete scale in Y), **AUTO** (take the plotting limits in use when the spectrum was last written), or two numbers for fixed limits ; X or Y specify the axis on which the type is to apply. For X axis, the limits are in the current units (C, V or F). For F, specify the offset from the rest frequency in MHz (note: the caption and the numbers on the axis will refer to absolute rest frequencies).

Several commands result in plotting. These are :

• **BOX**, which plots the frame. The Y axes are labelled in temperature units ; the X axes may be in the following units : Velocity, Frequency, Image frequency, or Channel number. The upper X axis may be labelled in a different unit than that of the lower axis. Units for both axes are entered by the command **SET UNIT L U**, where L and U stand for the units of lower and upper axes and may be any of V, F, I, or C. The second parameter U is optional ; if not entered, it defaults to L.

**BOX** accepts the option **/UNIT** which specifies a unit temporarily different from the current one (given by the **SET UNIT** command). The parameter **UPPER** will modify only the unit for the upper axis of
the frame. For instance: BOX /UNIT F UPPER will give velocities on the lower axis (if this is the current unit specified by SET UNIT V) and rest frequencies on the upper axis.

- **SPECTRUM**, which plots the spectrum, in the current mode, clipped into the current box. An offset may be given as argument to plot two spectra above each other for comparison.
- **TITLE**, which writes a header above the frame. The title format is controlled by the SET FORMAT command.
- **PLOT**, which performs all of CLEAR; BOX; SPECTRUM; TITLE in a single operation.

### 3.1.7 Baselines

The `BASE` command subtracts polynomial baselines of degree < 30. The fitting algorithm uses Chebyshev polynomials, and does not allow any extrapolation outside the fitting range. It is thus important to fit the baseline out to the maximum extension of the wanted spectrum. If extrapolation is needed, a constant value will be used outside the fitting range, equal to the polynomial value at the boundary. The algorithm warns if the polynomial degree is too high.

One first defines line windows by the command `SET WINDOW`. Values may be entered numerically as arguments:

```
SET WINDOW w11 wu1 [w12 wu2 [...] ]
```

or graphically with the crosshair cursor if SET CURSOR ON has been selected. In this case enter the values in the same order as above by typing "N" or " " (space bar); "C" cancels the last value entered; "H" types a help message and "E" terminates the operation.

Up to 20 windows may be defined. `BASE` then fits a polynomial to the rest of the spectrum. However, only the “visible” parts of the spectrum are used and bad channels are taken out. The degree of the polynomial is defined by SET BASE n, or temporarily by the BASE command itself with its argument. The `/PLOT` option plots the fitted baseline in the current box. The area in the windows as well as the rms noise, are computed.

A baseline can be computed for one spectrum, and then subtracted from a different one using `BASE LAST`. This may be helpful for example at Pico-Velleta where you may remove from the 100 kHz backend the baseline determined from the 1 MHz one. Be sure that you do not change the X-unit between the time you computed the baseline and the time you remove it...

Sinusoidal baselines may also be subtracted, using the command

`BASE SINUS Amplitude Period Phase`

where Amplitude Period and Phase are initial guesses for a minimization routine. A linear baseline is added to the sinusoid in any case.

### 3.1.8 Frequency Switching

Spectra obtained by Frequency Switching need to be folded before further processing. It is usually a good idea to remove a baseline before the spectra are folded in order to use as much baseline as possible. The folding is done by command `FOLD` which reads from the corresponding section all the necessary parameters. `FOLD` only operates in the R memory. The number of channels is decreased to keep only the relevant part of the resulting spectrum.

### 3.1.9 Profile Analysis

The `CLASS` user may analyse spectra by fitting profiles. The minimization method is taken from the `MINUIT` system of CERN, modified and optimised for this purpose. Reliability proves to be good. Five types of profiles are presently available, and can be selected by the `METHOD` command:

- **METHOD GAUSS**
  
  This is the default type of profile. One may use up to five gaussians, which might depend on each other as specified by a system of control codes associated with each variable. For each of these gaussians, the primary parameters are 1) Area, 2) Position, and 3) Width (FWHM). The current X
unit (for the lower axis) is used. Code 0 means that the parameter is adjustable; 1 that it is fixed; 2 that the parameter (head of group) is adjustable and that another parameter, coded 3, is fixed with respect to it; 4 that the parameter is a fixed head of group.

- **METHOD SHELL**
  Profiles are like those encountered in envelopes of stars. The primary parameters are Area, Position, Width and Horn to Center ratio. The aspect of the profile varies from parabola (as obtain in optically thick lines) for Horn/Center = -1 to flat-topped lines (unresolved optically thin lines) for Horn/Center = 0 and double peaked profiles (resolved optically thin lines) for Horn/Center > 0. The profile is symmetric. Presently only code 0 and 1 can be used, and up to 5 independent lines can be fitted in a single spectrum. The X unit must be frequency.

- **METHOD NH3(1,1) or NH3(2,2) or NH3(3,3)**
  Profiles taking into account hyperfine structure of ammonia with a gaussian distribution of velocity are fitted. Primary variables are 1) The product (Main Group Opacity) times (Excitation Temperature minus Background Temperature) 2) Velocity 3) Line Width (FWHM) and 4) Main Group Opacity. Up to 3 independent lines can be fitted, and only codes 0 and 1 are allowed. The X unit must be Velocity.

- **METHOD HFS File Name**
  This method is similar to the previous one, but the HyperFine Structure parameters are read from a file instead of being known by **CLASS**. The first line of this file must contain the number of hyperfine components (< 40). The other lines must contain, for each component, the velocity offset and the relative intensity. The parameters are the same as for NH3 method.

- **METHOD CONTINUUM**
  This method is used for continuum drifts. It fits a gaussian and a linear baseline in the drift. If beam-switching was used and the reference beam is along the drift direction, two dependent gaussian are used to optimize signal to noise. The method does not require any user input.

The commands of this profile analysis system are:

- LINES, GAUSS, FIT, DISPLAY, ITERATE, RESIDUAL, KEEP.

- **LINES N** defines the number of components and prompts for the initial values of the parameters for each component. This command has no effect for method CONTINUUM. Parameters are read in list directed format in the following order:

  Code, Intensity, Code, Position, Code, Width, [Code, Parameter 4]

The code is an integer number between 0 and 4. Note that, though the program works on the area (or other quantities as for NH3 methods), you have to give the intensity, since this quantity is more intuitive than area. The use of the list directed format makes things easier when only one parameter has to be modified (cf Fortran norms). The number of lines N may be zero; in this case the program finds out reasonable starting values by itself.

Values may be also entered graphically if SET CURSOR ON was selected. After entering LINES N, first point the cursor to one side of the line, strike one key, point the cursor the other side, strike another key. The program computes the moment of the spectrum between these boundaries and use it to set up starting values. Proceed like this for all components. One drawback of this way of entering values is that you cannot change the control codes. It should be used only for entirely independent and free lines.

- **GAUSS**
  activates minimization, then prints out the results after convergence. A Simplex method is first used to ensure convergence, then a Gradient method to refine the results, and compute the errors.

- **ITERATE**
  is similar to GAUSS, but starts from the previous minimization results. Only the Gradient method is used. Consequently, this command is only useful close to the minimum.
• **FIT N**
  plots the Nth component obtained by fitting; if N is not given, the sum of all components is plotted.

• **RESIDUAL N**
  subtracts the Nth component from the current spectrum, or the sum of all components is N is not given). In this process, the R spectrum is first copied into T, then the difference is done in R.

• **DISPLAY**
  Prints the results of fitting from the current spectrum, without recomputing it ...

• **KEEP**
  Saves the fit results in the input file, which must be opened also for output. **KEEP** is in fact a reduced version of **UPDATE**, and to be used with the same care as **UPDATE**.

• **SET MASK** ...
  Defines masks in the spectrum for the fit. This commands has the same syntax and behaviour as **SET WINDOW**. Masked regions will not be used for the fit.

Fit results are always saved by a **WRITE** command.

### 3.1.10 Miscellaneous

• **DIVIDE** makes the ratio of the R and T spectra. The two spectra must have the same velocity scale.

• **FFT** plots the power spectrum of the current observation. It might help identify spurious ripples. Editing of the fourier transform is possible, so that these ripples may be suppressed.

• **NOISE** generates a gaussian noise as intense as in the current spectrum using the rms value determined by the **BASE** command, or using a rms value given as an argument. **NOISE Value NEW** will create a noisy spectrum of given noise level into R, after copying R in T.

• **RESAMPLE** resamples the R spectrum on the specified grid. If the final sampling is coarser than the original one, a smoothing occurs to the final sampling.

• **SMOOTH** operates a Hanning smoothing by default and divide the number of channels by two. Other arguments can be specified to use other methods. **SMOOTH AUTO** uses a sophisticated variable-resolution algorithm, but it requires the channels to be really independent and this is apparently seldom the case in radio astronomy. **SMOOTH GAUSS width** convolves the spectrum by gaussian of given Width in current units; it does not take care of bad channels. **SMOOTH BOX N** make the average of N adjacent channels and divides the number of channels by N.

### 3.2 Continuum Processing

#### 3.2.1 Introduction

So far, we have handled only Spectroscopic data, but Continuum data can be processed by **CLASS**. Currently, only continuum drifts can be reduced. The basic idea is to treat continuum drifts as spectra would be. Accordingly, very few commands behave differently in Continuum and Spectroscopy modes.

Continuum mode is accessed by typing command

**SET TYPE CONTINUUM**

the prompt changes to **CAS>** (Continuum Analysis System). You can return to Spectroscopy mode later on typing command

**SET TYPE SPECTROSCOPY** or **LINE**

and the prompt changes to **LAS>** (Line Analysis System).
3.2.2 Differences between Continuum and Spectroscopy Mode

Some commands have slightly different behaviour in Spectroscopy and Continuum modes.

- **SET UNIT** has no effect in Continuum mode.
- **SET ANGLE** also controls the plotting units in Continuum mode.
- **METHOD** : only **GAUSS** and **CONTINUUM** methods are allowed in Continuum mode.
- **LINES** has no effect with **CONTINUUM** method.
- **HEADER** uses a different format for Continuum and Spectroscopy modes.
- **STRIP** produces a map from a set of parallel drifts. The index must define such an ensemble of drifts.
- **FITS** format support is experimental for continuum data.
- With **CONTINUUM** method, **PRINT FIT** command only outputs a single component, and the component number is not written.
- **SET ALIGN CHANNELS** and **SET ALIGN POSITION** are the only available alignment modes in Continuum mode.

Except for these restrictions, the behaviour of other commands is similar. Note that command **FIND** only selects data of the current type.

3.3 Skydip Processing

**CLASS** is able to reduce skydip data. Skydip mode must be selected using command

**SET MODE SKYDIP** which also changes the prompt to **SAS** (Skydip Analysis System). Commands **FIND**, **HEADER**, **GET**, **PLOT** and **WRITE** may be used as for Continuum and Line modes, but the only other valid command is **REDUCE** which fits the sky emission using atmospheric information available in the data, and displays the results.

3.4 On-the-fly Data Processing

The format of **CLASS** data files has been recently extended to allow the reduction of data obtained in the On-the-fly (OTF) mode by the 30-m telescope. In that mode the data is taken while the telescope beam is moving across the source to be mapped. The raw data is pre-processed (amplitude calibration) by the CAL program available at the telescope, and written in a **CLASS** file. Data from a scan across the source is written as a single observation, thus containing many spectra at different positions. Each spectrum is included in the observation as a ‘record’ or ‘dump’, labeled by time, and telescope position. This new kind of data is handled by **CLASS** in a way similar to the normal data, but a few commands behave differently:

- **GET** has an option **/RECORD n** to load an individual record of an OTF scan in the R buffer. After that this data may be plotted in the usual way.

- **PLOT** has an option **/SCAN** which displays the entire scan as a 2-dimensional image, with velocity/frequency in the X axis, record number in the Y axis, intensity being rendered as grey/color scale.

- **BASE, FFT, RESAMPLE** work on all the records of an OTF scan at once.

- **GRID** may be used to tabulate, grid and plot the OTF data into **GILDAS** data cubes that may be further processed with **GILDAS** tasks.
3.5 Map Processing: GILDAS

3.5.1 Maps of Spectra

Using the MAP command, it is possible to produce a plot of spectra in the current index, arranged in a map. Use the option /CELL Size_x Size_y to specify the size of a spectrum, in current angle units. Without this option a default is taken (the actual separation of the spectra). Option /GRID will produce frames around the spectra. The argument MATCH can be given to fix the aspect ratio of the boxes to the cell sizes.

The map size can be controlled using commands SET PAGE and SET BOX LOCATION. Labels can be suppressed by option /NOLABEL (and ticks will not be drawn if of size 0,0). Option /NUMBER will add the observation number with each spectrum.

On multi-window displays (e.g. X-Window terminals), after the MAP command has been used, the POPUP command may be used to display in another window a spectrum selected either from its observation number or from its offsets. POPUP can also be used after the STAMP command. The STAMP command allows to display many observations at once, without requesting the X and Y axis scales to be fixed.

3.5.2 GILDAS Interface

For mapping purpose, CLASS is able to produce images at the GILDAS format. GILDAS (the Grenoble Image and Line Data Analysis System) is an ensemble of routines which is able to process images of up to four dimensions. It contains a variety of routines to transpose, resample, reproject, smooth, add, merge, divide, etc... images, and a superset of GreG which allows to produce contour plots from these images. Other programs can also do false color displays of the images on image processors.

3.5.3 Contour maps: Building a data cube

The command CUBE builds a 3-D image from the current index. It is assumed that the current index defines such an image (i.e. corresponds to a single line, observed with a single resolution, and towards a single source). The first axis of the cube corresponds to the current X unit, the second to RA (or L) and the third to DEC (or B). Projection information is written so that the coordinate system is automatically recognized when processed by GILDAS.

Please refer to the GILDAS documentation for later processing. The recommended first steps are i) a transposition (task TRANSPOSE to put the axes in RA,DEC,Velocity ordering and ii) the tasks FILL_CUBE (for undersampled data) or MAKE_CUBE (for oversampled data) to resample the data cube on a finer grid for nice contouring.

An alternative processing technique is to use command GRID, which creates a table suited for map making using the task GRID_CUBE. This route is recommended for oversampled data, while the use of MAP and FILL_CUBE is probably better for undersampled data.

Finally GRID /IMAGE will directly produce data cubes from oversampled or undersampled data, using a Gaussian convolution kernel for interpolation, and reasonable defaults for the cube size and pixel size in both axes.

CUBE and GRID can also produce images or tables of several mathematical functions of the input spectra rather than simple channel maps. Please refer to the internal help for details.

3.5.4 Velocity-Position Plots

Although by adequate use of transpositions, you can produce velocity-position plots from data cubes, it is also possible to create a 2-D image for processing by GILDAS to produce Velocity-Position plots by using command STRIP. This command works on the current index, which must correspond to a real strip. Relevant information is written to a 2-D image which can later be processed by GILDAS. The first axis corresponds to the current X unit (Velocity or Frequency or Channels) and the second axis to the offset.

3.5.5 Continuum Maps

Command STRIP can also be used to produce a continuum map from an ensemble of parallel continuum drifts. The index must define such an ensemble.
3.6 The PRINT Command

In addition to the previous commands, the command PRINT offers a way to list a number of additional informations. It has several functions:

- PRINT FIT, which prints the results of profile fits. For each spectrum, N lines are written (N being the number of fitted components), and each line contains in the following order (1) the component number, (2) then observation number, (3,4) the two cartography offsets, (5,6) area of gaussian and corresponding error, (7,8) same for position, (9,10) same for width, (11) intensity, (12,13) rms on the baseline and on the line. Offset are in the current coordinate system and units. The current method is used.

   For Continuum method, only a single gaussian is written. The written information is oriented towards pointing measurements : (1) the observation number (2,3) Azimuth and elevation (4,5) area of gaussian and error, (6,7) position, (8,9) width, (10) intensity, (11,12) rms on baseline and signal, (13,14) collimations. All angular values are in the current angle unit. The values are followed by the source name.

- PRINT AREA, prints the area of the line computed by the BASE command. Each line contains (1,2) offsets, (3) area, (4) rms noise.

- PRINT AREA V1 V2 V3 V4 ..., prints areas within velocity slices (if such is the current X unit, but one could use channels or frequencies). Ranges are V1-V2, V2-V3, V3-V4, ... etc. Each line will contain (1,2) the offsets, followed by the areas in order.

- PRINT CHANNEL List, prints values of channels in the list. The list is specified in the FOR n1 TO n2 BY n3 format. Total number of channels is however limited to 15.

- PRINT MOMENT V1 V2 V3 V4 ..., prints moments (area, position, width), of the data within the velocity (or channels or frequencies, depending on the current units) V1-V2, V3-V4, etc... Each line contains (1,2) the offsets, (3,4,5) the moments for V1-V2, (6,7,8) for V3-V4 etc...

- PRINT POINTING, prints results of CONTINUUM method fits printed in a format adapted to pointing constants measurements. The output is suited for further processing and determination of pointing constants using the POINT program.

- PRINT FLUX, prints results of CONTINUUM method fits printed in a format adapted to flux determination. The output is suited for further processing using the FLUX language in the ASTRO program.

As all commands using an ensemble of spectra, PRINT works on the whole current index. Output is by default printed on the screen, but may be directed onto a file by the /OUTPUT filename option.

Alternatively, the same information may be written to a “Table” (a special kind of GILDAS image). The Table format is much faster and suppresses some of the limitations of the formatted output on the number of columns written. Table format is obtained using option /TABLE Tablename.

3.6.1 GreG Tables

The command GREG is intended to produce a direct interface with GreG for plots of spectra. It creates a GILDAS Table which can be read by GreG. The table contains the following columns for Spectra:

1. Intensity
2. Channel number
3. Velocity
4. Offset frequency
5. Rest frequency
6. Image frequency
7. Fitted profiles if any - fit(i), i=0,nline - in column 7+i, for the current method.
The output table can be put later in a formatted way using **GILDAS** task **LIST** if needed. For continuum data, the table contains

1. Intensity
2. Channel number
3. Angular offset (radian)
4. Fitted profile if any.

The table may be used as input to **GReG** to produce fancy plots, or by the **GILDAS** software for other applications. In particular, the **SIC** monitor (command **LET**) is able to subtract any of the fits from the spectrum to produce residuals if needed. It is possible to merge different tables, add columns to a table, etc... For example, from two spectra at the same velocity resolution it is possible to merge the two tables and compute the ratio of the spectra, as well as the errors on this ratio.

### 3.7 Publication Quality Plots

With the hardcopy facility provided by the graphic library used by **CLASS**, and the **DRaw** command which allows most of the wanted annotations, it is perfectly possible to produce quality publication plots of spectra directly. Note that all **GReG** commands are available in **CLASS** to fully annotate plots, superpose spectra with related data, stack various plots ...

#### 3.7.1 Hardcopy

The command **HARDCOPY** is the same as the **GReG** **HARDCOPY** command. Please refer to **GReG** documentation or the internal help for a description.

Note that you do not need a graphic terminal to prepare a plot. It is only much more convenient, but the plot and the way it is displayed are two completely independent things.

#### 3.7.2 Annotations

Command **DRAW** can be used to annotate a plot. The basic operations performed by command **DRAW** are:

- **DRAW TEXT Xpos Ypos "Text" Centering**
  to draw a text at position (Xpos,Ypos) (in current units) with the specified centering code. This command works more or less like the **GReG** command of same name. Please refer to the **GReG** manual for details. In particular, you can include Greek letters and Symbols in the text using the escape character \. A strange thing may appear on the screen, but it is O.K. on the plot. If you have doubts, use **ZOOM** **REFRESH** before command **HARDCOPY** to see exactly what your plot looks like.

- **DRAW UPPPER Xpos "Text"**
  to draw a vertically oriented text at position Xpos, with a vertical line connecting the beginning of the text to the current spectrum. This text and line are written at position Xpos, in units of the upper axis. Typically, this command is used to mark spectral line identifications.

- **DRAW LOWER Xpos "Text"**
  same as above, but with Xpos in units of the lower axis.

- **DRAW WINDOW [Level]**
  shows the current windows by marks on the graphic plot. Level is an optional arguments indicating at what Y value the marker should be put (Default 0).

- **DRAW MASK [Level]**
  same as above but for the current masks.

- **DRAW KILL [Channel]**
  kills the specified channel (current one if using the cursor) by attributing it the “blanking” or “undefined” value.
• DRAW FILL [Channel]
  Fills the specified channel (current one if using the cursor) by interpolation between the nearest non-blanked channels. The channel must have been killed before.

Any other character will not draw anything, but simply returns the cursor position, with corresponding values of the velocity, frequency, image frequency, channel number.

3.7.3 Plot Quality

CLASS is usually used for interactive look at spectra, hence its default values are all oriented towards fast plotting on screen. These defaults can be changed by command SET. If the value of a parameter is not controlled by CLASS, the command will be passed on to GreG for processing. The following GreG presentation parameters are useful:

• SET BOX LOCATION
  It can be set to LANDSCAPE, PORTRAIT, SQUARE or 4 numbers indicating the position of the box in the plot page (in centimeters).

• SET CHARACTER Size
  Control the size of characters in centimeters

• SET FONT Quality
  Select the character quality to be used, SIMPLEX or DUPLEX. The fonts are identical to the ones used by GreG, and the character handling is the same (in command DRAW TEXT).

• GREG\SET PLOT ...
  Define the plot page size. This command is identical to command SET PLOT_PAGE in GreG (but in CLASS, SET PLOT is already used to indicate whether spectra are broken lines of histograms). Refer to the GreG manual for details.

• SET TICK Size
  Define the tick size in centimeters

3.8 Variables

3.8.1 The Rope to Hang Yourself

CLASS makes use of SIC variables to allow more flexibility in the processing, in particular in procedures. SIC variables are extremely powerful, with the side effect that if you want, you can corrupt your data by overwriting some information. CLASS attempts to prevent the most disastrous errors by defining some of the most critical variables as READONLY. They cannot be overwritten by the user, but their values can be used in expressions, either arithmetic or logical. However, an unprotected mode is available for specific processing using the command SETVARIABLE.

3.8.2 Index Variables

The variable FOUND refers to the number of observations in the index. It is declared Read-Only of course. Its main use is as a test variable for actions which should be performed only if something exists in the index. As an example, the following procedure reduces a mapping observing run by averaging all spectra at each position, and requires only two user input : the two steps in Lambda and Beta.

SAY "Procedure MAP,CLASS : reduce mapping observations by averaging all" -
"spectra for each point. Input required from user : XSTEP and YSTEP" -
"the map spacings in both directions"
!
LAS\FIND
SIC\IF (FOUND.EQ.0) THEN
SIC\RETURN
SIC\ENDIF
SIC\DEFINE DOUBLE XMIN XMAY YMIN YMAX XSTEP YSTEP ! Local variables
!
! Determine offset extrema.
LAS\GET FIRST
SIC\LET XMIN = OFF_LAMBDA
SIC\LET XMAX = OFF_LAMBDA
SIC\LET YMIN = OFF_BETA
SIC\LET YMAX = OFF_BETA
SIC\FOR I 2 TO FOUND
    LAS\GET NEXT
    SIC\LET XMIN = MIN(XMIN,OFF_LAMBDA)
    SIC\LET XMAX = MAX(XMAX,OFF_LAMBDA)
    SIC\LET YMIN = MIN(YMIN,OFF_BETA)
    SIC\LET YMAX = MAX(YMAX,OFF_BETA)
SIC\NEXT
!
! Now make map. XSTEP and YSTEP must be specified by user
SAY "Enter step in Lambda"
LET XSTEP =
SAY "Enter step in Beta"
LET YSTEP =
SIC\FOR XOFF XMAY TO XMAX BY XSTEP
    SIC\FOR YOFF YMIN TO YMAX BY YSTEP
        LAS\FIND/OFFSET XOFF YOFF
        SIC\IF (FOUND.GT.0) THEN
            LAS\SUM
            LAS\WRITE
        SIC\ENDIF
SIC\NEXT
SIC\RETURN

The FIND command does not return an error, but set FOUND = 0, if it finds nothing. A second variable related to the index is the INDEX array, of dimension FOUND, which contains the observation numbers of all observations in the index.

3.8.3 Default Header Variables

The most important header parameters are defined by default as SIC variables in a protected mode; the others, of less frequent use, can be accessed if required by the user (see "Advanced Processing"). The default variables are (RW means Read-Write variable, RO, Read-Only).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TELESCOPE</td>
<td>Character*12, RW,</td>
<td>Telescope name</td>
</tr>
<tr>
<td>NUMBER</td>
<td>Integer,</td>
<td>RW, Observation number</td>
</tr>
<tr>
<td>VERSION</td>
<td>Integer,</td>
<td>RO, Version number</td>
</tr>
<tr>
<td>DATATYPE</td>
<td>Integer,</td>
<td>RO, Type of observation</td>
</tr>
<tr>
<td>QUALITY</td>
<td>Integer,</td>
<td>RO, Quality of data</td>
</tr>
<tr>
<td>SCAN</td>
<td>Integer,</td>
<td>RO, Original scan number</td>
</tr>
<tr>
<td>UTOBS</td>
<td>Double,</td>
<td>RO, UT of observation (Radians)</td>
</tr>
<tr>
<td>LSTOBS</td>
<td>Double,</td>
<td>RO, LST of observation (Radians)</td>
</tr>
<tr>
<td>AZIMUTH</td>
<td>Real,</td>
<td>RW, Azimuth of observation (Radians)</td>
</tr>
</tbody>
</table>
ELEVATION        Real,        RW, Elevation of observation (Radians)
TSYS             Real,        RW, System temperature
TIME             Real,        RW, Integration time (Seconds)
SOURCE           Character*12, RW, Source name
LAMBD A          Double,      RW, Longitude of source (Radians)
BETA             Double,      RW, Latitude of source (Radians)
OFF_LAMBD A      Double,      RW, Offset in longitude (Radians)
OFF_BETA         Double,      RW, Offset in latitude (Radians)
EPOCH            Real,        RW, Epoch of coordinates (Years)
LINE             Character*12 RW, Line name
CHANNELS         Integer,     RO, Number of channels
REFERENCE        Real,        RW, Reference channel
FREQ_STEP        Real,        RW, Frequency step by channel (MHz)
VELO_STEP        Real,        RW, Velocity step by channel (km/s)
VELOCITY         Real,        RW, Velocity of reference channel
FREQUENCY        Double,      RW, Rest frequency at reference channel
IMAGE            Double,      RW, Image frequency " " " "

BEAM_EFF         Real,        RW, Telescope beam efficiency
FORWARD_EFF      Real,        RW, Telescope forward efficiency
GAIN_IMAGE       Real,        RW, Image to signal band ratio
WATER            Real,        RO, Water vapor content (mm)
PRESSURE         Real,        RO, External pressure (hPa)
AMBIENT_T        Real,        RO, External temperature (K)
CHOPPER_T        Real,        RO, Chopper temperature (K)
COLD_T           Real,        RO, Cold load temperature (K)
TAU_SIGNAL       Real,        RO, Opacity in signal band
TAU_IMAGE        Real,        RO, Opacity in image band
ATM_SIGNAL       Real,        RO, Atmospheric temperature in signal band
ATM_IMAGE        Real,        RO, Atmospheric temperature in image band

RX                Real[8192]  RO, X values of data points
RY                Real[8192]  RW, Y values of data points

8192 is currently the maximum size of the spectra, but the variables RX and RY are redimensioned to the effective number of channels for each spectrum.

3.8.4 Advanced Processing

All header parameters can be defined as SIC variables for specific processing of the data, either as Read-Only or as Read-Write, using the command SET VARIABLE. Read-Write mode is to be used with caution, since even critical variables (e.g. the number of channels) can be modified. Refer to command SET_VARIABLE for more details.

By using the appropriate variables and the SIC mathematical and logical facilities, customized data processing becomes possible, as well as complete data editing.
4 From CLASS to FITS

To bring back CLASS spectra to your home institution, where the CLASS format may not be supported, CLASS offers the possibility of writing standard FITS format tapes. A program named CFITS is used to perform CLASS to FITS conversion (and vice-versa). For a description of the FITS format see the original paper by Wells et al. (Astron. and Astrophys. Suppl.). CFITS contains two languages, LAS\ and FITS\.

The LAS\ language is a subset of the language used in CLASS program, and contains only the following commands:

<table>
<thead>
<tr>
<th>CLEAR</th>
<th>DEVICE</th>
<th>DROP</th>
<th>DUMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>FIND</td>
<td>GET</td>
<td>HARDCOPY</td>
</tr>
<tr>
<td>HEADER</td>
<td>IGNORE</td>
<td>LIST</td>
<td>PLOT</td>
</tr>
<tr>
<td>SET</td>
<td>SHOW</td>
<td>SWAP</td>
<td>TAG</td>
</tr>
<tr>
<td>UPDATE</td>
<td>WRITE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The FITS\ language contains the following commands:

- DISMOUNT : logically dismounts the tape.
- INITIALIZE : initialize the tape
- HEADER : read the FITS header of the current file on tape. The tape is left after the end of file.
- LIST : lists all the FITS header from the current file to the end of tape.
- MOUNT : logically mount the tape. This command must be used before anything else can be done with the tape.
- READ : read the current FITS file into the R memory. The scan read in can then be written to the CLASS output file by command LAS\ WRITE.
- Rewind : Rewind the tape
- SKIP N : Skip N files on the tape. N can be positive or negative. If N = *, move to end of tape.
- WRITE [HERE] : Write the scan from R memory to the tape. The scan is written at end of tape, unless you specify argument HERE. An End of Tape mark is written after the file.

FITS headers written by CFITS depend on the informations present in the corresponding CLASS headers. Any missing information will also be omitted in FITS (and vice versa). A typical FITS header written by CFITS looks like this:

```
SIMPLE = T       / (1)
BITPIX = 16      / (2)
NAXIS = 4        / (3)
NAXIS1 = 253     / (3)
NAXIS2 = 1       / (3)
NAXIS3 = 1       / (3)
NAXIS4 = 1       / (3)
BSSCALE = 0.1038147092913E-03 / (4)
BZERO  = -0.24138058264353E+01 / (5)
DATAMIN = -0.5815605640411E+01 / (5)
DATAMAX = 0.9877878247506E+00 / (5)
BUNIT  = 'K'     / (5)
CTYPE1 = 'FREQ'  / (5)
CRVAL1 = 0.00000000000000E+00 / Offset frequency
CDELT1 = 0.10000000014901E+06 / Frequency resolution
CRPIX1 = 0.13450000000000E+03 / (5)
```
CTYPE2 = 'RA' 
CRVAL2 = 0.8338750229169E+02 
CDELT2 = -0.5555556975722E-02 
CRPIX2 = 0.0000000000000E+00 
CTYPE3 = 'DEC' 
CRVAL3 = -0.177777752148E+01 
CDELT3 = 0.0000000000000E+00 
CRPIX3 = 0.0000000000000E+00 
CTYPE4 = 'STUKES' 
CRVAL4 = 1.0000000000000E+00 
CDELT4 = 0.0000000000000E+00 
CRPIX4 = 0.0000000000000E+00 
TELESCOP = 'IRAM-30M-B20' 
OBJECT = 'ORI-I-2' 
GLAT = 0.0000000000000E+00 / Galactic latitude 
GLON = 0.0000000000000E+00 / Galactic longitude 
EPICH = 0.1950000000000E+04 
BLANK = 0.9878914356232E+00 / Blanking value 
LINE = '*' / Line name 
RESTFREQ = 0.1152712040000E+12 / Rest frequency 
VLSR = 0.1300000000000E+05 / Velocity of ref. channel 
DELTAV = -0.2600757479668E+03 / Velocity resolution 
IMFREQ = 0.1074062118530E+12 / Image frequency 
TSYS = 0.4787839660645E+03 / System temperature 
OBSTIME = 0.7500000000000E+02 / Integration time 
SCAN-NUM = 0.4386000000000E+04 / Scan number 
TAU-ATM = 0.8740132451057E+00 / Atmospheric opacity 
NPHASE = 2 / Number of frequency phases 
DELTAF1 = -0.6000000000000E+07 / Frequency offset Phase 1 
PTIME1 = 0.3750000000000E+02 / Duration of Phase 1 
WEIGHT1 = 0.1000000000000E+01 / Weight of Phase 1 
DELTAF2 = 0.5000000000000E+07 / Frequency offset Phase 2 
PTIME2 = 0.3750000000000E+02 / Duration of Phase 2 
WEIGHT2 = -0.1000000000000E+01 / Weight of Phase 2 
BEAMEFF = 0.56 / Beam efficiency 
FORWEFF = 0.88 / Forward efficiency 
GAINMSG = 1.0000000000000E+00 / Image sideband gain ratio 
ORIGIN = 'LAS-Grenoble-VAX' 
DATE = '7/9/85' / Date written 
DATE-OBSE = '29/5/85' / Date observed 
DATE-RED = '7/9/85' / Date reduced 
ELEVATE = 0.5064780612975E+02 / Telescope elevation 
AZIMUTH = 0.191966046612E+03 / Telescope azimuth 
UT = '12:50:47.384' / Universal time at start 
LST = '06:09:00.479' / Sidereal time at start of observation 
HISTORY REL 0.5064780612975E+02 / Telescope elevation 
HISTORY RAZ 0.191966046612E+03 / Telescope azimuth 
HISTORY RUT 12:50:47.384 Universal time at start of observation 
HISTORY RST 6:09:00.479 Sidereal time at start of observation 
HISTORY SCAN 4383-4386

1. Although only one axis is really necessary, it is very convenient to define four, use the first one for the channels, and the three last ones to code the positions and stokes parameters.
2. The first axis is used to define effectively the spectrum. Thus $NAXIS1$ is the number of channels.

3. $NAXIS2$, $NAXIS3$, and $NAXIS4$ are all one for a single spectrum. Note however that it is possible to store a raster map with a similar header as this one.

4. Could be Janskys

5. First axis defined in terms of frequency (in the signal sideband in case of double sideband operations). The frequency of a specific channel is given by

$$F(i) = RESTFREQ + CRVAL1 + (i - CRPIX1) \times CDELT1$$

in which the Rest frequency $RESTFREQ$ is defined later in the header.

6. Second axis, Right Ascension RA (as in this case) or Galactic Longitude GLON. The information as presented here is slightly incomplete, since it would be in general necessary to have an information about the kind of projection used. On most radio telescopes, it is simply assumed that the angular offset in RA is divided by the cosine of Declination to represent "true" angular offsets (valid only for a small field). Small telescopes may need more elaborate projection systems. In the current example, the position really observed is

$$\text{Dec} = \text{CRVAL3} + (1 - \text{CRPIX3}) \times \text{CDELT3}$$

$$\text{Ra} = \text{CRVAL2} + (1 - \text{CRPIX2}) \times \text{CDELT2} / \cos(\text{Dec})$$

That is, $CDELT2$ and $CDELT3$ represents angular offsets from the reference position $(\text{CRVAL2}, \text{CRVAL3})$ in a Global Sinusoidal projection ($\text{RADIIo}$ projection).

7. Stokes parameters as defined in the basic paper of Wells et al.

8. Galactic latitude and longitude of the reference position, i.e. of the position $(\text{CRVAL2}, \text{CRVAL3})$. If one was using galactic coordinates instead of equatorial ones, the RA and DEC would appear here instead.

9. Epoch of these coordinates

10. Molecular line name, for bookkeeping

11. Rest frequency

12. LSR Velocity of the reference channel. Heliocentric velocities can be used also.

13. Velocity spacings of the channels. This information is duplicate with the rest frequency and frequency spacing of channels, but convenient. The velocity of a given channel is thus given by

$$V(i) = VLSR + (i - CRPIX1) \times \text{DELTAV}$$

14. Image frequency, for double sideband operation.

15. System temperature, necessary for some weighting when adding a number of spectra.

16. Integration time, used for the same reason as above.

17. Scan number, for bookkeeping.

18. Atmospheric opacity in the signal sideband.

19. For multi-phased spectra (i.e. frequency switching) number of phases.

20. For each phase, the frequency offset, the phase length and weight.

21. The telescope beam efficiency

22. The telescope forward efficiency

23. The ratio of gains in the image and signal sidebands (in case of double sideband operation)
24. Some “History” comments. Whether this information should be given with specific keywords or in an History record is still an open question. This information is not really needed for further data reduction, but it helps bookkeeping.

25. The list of scan numbers of the spectra added to produce this one.

The FITS interface for Continuum data is still experimental. Try it, and send your comments...

A minimal number of keywords has been defined as part of the FITS standard, but additional ones can be (and have been) added by various groups. Thus, several “flavors” of FITS coexist. Unknown keywords are normally ignored, but CFITS supports keyword redefinition. If you receive a tape with scan number coded as NUMBER (instead of SCAN-NUM), all you need to do is to define a SIC symbol named NUMBER with translation SCAN-NUM. This is done by typing SIC\$SYMBOL NUMBER SCAN-NUM

The IAU FITS committee is currently revising some definitions of the FITS format, in particular to allow projection information to be written, and sub-keywords in a single line. Hence, it is more than likely that the FITS format produced by CLASS will be modified next year. Nonetheless, older tapes written by CFITS (or any compatible program) will forever be acceptable as input. We also plan to keep (as an option) the capability to write tapes with the present format for a long time (a few years).
5 LAS Language Internal Help
6  ANALYSE Language Internal Help
7 FITS Language Internal Help

7.1 DIRECTORY

FITS\DIRECTORY DirOne

Lists specified directory. Only meaningful for disks and CDroms...

7.2 DISMOUNT

FITS\DISMOUNT

Dismounts and unloads currently mounted removable medium (Tape, CDrom...).

7.3 INITIALIZE

FITS\INITIALIZE

Initialize the tape. It must be logically mounted to do so...

7.4 LIST

FITS\LIST [keyword]

List the headers of all files from current tape position, or list only the specified FITS keyword.

7.5 MOUNT

FITS\MOUNT Device

Logically mount the removable medium on the specified device. (Logical name TAPE_DEVICE is used by default). This command must be used before anything else is done on device (even INIT for tapes). If it was already mounted by the user, the tape is rewound.

7.6 READ

FITS\READ [FileName] [/CHECK]

This command is used to Read data from FITS format. Not all FITS tapes may be read by CLASS, and some header information might be incomplete when doing so.

On tape, the new scan is read from current tape position, and the tape is left at the beginning of next file. On disks and CDroms, the file name argument is compulsory.

Option /CHECK can be used to list the header while reading the file.
7.7 REWIND

FITS\REWIND

Rewinds the tape (but does not dismount it). Although CFITS normally keeps track of file numbers on tape, it used to occasionally get lost. In the unlikely event that this happens again, rewinding is the only way to recover correct numbers.

7.8 SELECT

SELECT Parametre Value

Defines optional parametres to be used when writing a tape.
- SELECT FORMAT Form
  Determines the format to be used on the output tape. Currently supported formats are I*2 and I*4. They correspond to BITPIX=16 and BITPIX=32 in FITS header.

7.9 SKIP

FITS\SKIP N

Skip N (positive or negative) files on tape. If N = *, move to end of tape.

7.10 WRITE

FITS\WRITE [argument] [/CHECK]

This command is used to write data in FITS format. The argument interpretation depends on the current device. In disk mode (before any MOUNT was issued), it is interpreted as the name of the FITS file. On tape, there are three possibilities:
- FITS\WRITE *
  writes all the scans currently selected in the CLASS index at end of tape. This command is the most efficient way to write a FITS tape.
- FITS\WRITE
  writes the current scan at the end of the tape
- FITS\WRITE HERE
  writes the current scan at the current tape position.
Option /CHECK may be used to list the FITS header while writing the tape files.

7.11 END
8 GTVL Language Internal Help
9 Error Messages and Recovery Procedures

Commands produce messages, either to signal their actions, or whenever they fail to execute properly to indicate the reason of the failure. The syntax of the message if the following

C-Facility, Error message text

where C is a one letter severity code, Facility is the name of the command or of a subroutine called by the command, and “Error message text” is a short explanation.

Messages can be divided into four categories according to the severity code

- S (Success) or I (Information) messages indicate normal successful operations.
- W (Warning) messages indicate that the operation completed, but that the result may not be significant.
- E (Error) messages indicate a failure to complete the command. The result of the command is undefined, but these messages are usually signaled when checking the input parameters before executing the command. In particular, the Input and Output files are never affected by these errors.
- F (Fatal) messages indicate severe errors which happened while a command is executing. They may have left the input or output files in an incoherent state. These errors are typically hardware errors.

9.1 Notifying the Programmer or System Manager

Some recovery procedures ask you to submit an SPR (Software Performance Report). This means to signal the error to either the CLASS authors, the local contact person, or the system manager. Signalling the error does not mean simply just telling “Gee, I typed ABRAcADAbRA and the program crashed”, it implies at least giving a copy of the CLASS.LOG and CLASS.MES files, and if requested, of the data files used when the error occurred. The utilities are too complex to allow any action if these files are not given.

Please remember that if you do not signal errors, they will (at best) never be corrected. At worst, hardware errors which would have been recoverable when signalled in time may become completely unrecoverable.

9.2 Recovering Corrupted Data Files

First, let us express our most sincere hopes that you will never need to read this section... Due to hardware (or even software but this seldom happens) problems data files may become corrupted. Although this is exceptional, it can be a dramatic circumstance. The structure of the data files is fairly complex, and missing information in the index may prevent reading otherwise valid spectra in the data file. If you suspect a file may have been corrupted, first verify it by the following commands:

LAS> FILE IN Dubious_file_name
LAS> SET DEFAULT
LAS> FIND/ALL
LAS> FOR I 1 TO FOUND+1
LAS> GET NEXT
LAS> NEXT

If this series of command ends with the message
'E-GET, End of current index'
everything is correct.

If not the file is corrupted and you may try to recover it partly. If the FIND command executes correctly, drop (by command DROP) from the index all the scans which cannot be read correctly (these have been lost forever). Then executes the following commands

LAS> FILE OUT Recovered_File_Name NEW
LAS> GET FIRST
LAS> WRITE
9 ERROR MESSAGES AND RECOVERY PROCEDURES

LAS> FOR I 1 TO FOUND
LAS> GET NEXT
LAS> WRITE
LAS> NEXT

If the FIND command does not execute correctly, try a LIST IN command to get the scan and version numbers. If it works or if you already have such a list, you may try to get each scan individually by specifying the scan and version numbers, and write them to another output file. In this way you might be able to recover some spectra, but it is not guaranteed.

If nothing like this works and if your file contained vital information you cannot reconstruct in any way, you may consider sending it to the CLASS authors which may be able to do something more. To our knowledge, such a desperate case actually happened only once in the early times of CLASS (at that time named LAS), to the authors who spent about 3 hours to find a way to repair the file, but managed to do it. Do not forget to supply a complete log of the errors (CLASS, LOG and CLASS.MES files).

9.3 LAS and ANALYSE Messages and Recovery Procedures

The messages not present in this list should come from the SIC monitor itself. Please refer to the SIC documentation for these.

9.3.1 A and B

E-ACCUMULATE, Coordinate systems are not compatible
LAS, SUM or ACCUMULATE commands. Two observations had different coordinate systems and could not be added.
User action: Convert them to a common coordinate system, or use the SET NOMATCH command to turn off position checking.

E-ACCUMULATE, Different calibrations
LAS, SUM or ACCUMULATE commands. Two observations were done with different major calibration parameters (Beam Efficiency and possibly Forward Efficiency) and could not be added.
User action: May be they were not on the same temperature scale. The weighting (and values) is incorrect in such a case. Modify the corresponding calibration parameter and retry. You may bypass this calibration checking by using command SET CAL OFF.

W-ACCUMULATE, Different reference channels R: <Real> T: <Real>
LAS, SUM or ACCUMULATE commands. Two spectra were added with the ALIGN mode CHANNELS, but had different reference channels, and hence possibly different velocity or frequency scales.
User action: Check if these spectra were to be added. If you want to add them and preserve a common velocity (or frequency) scale, use the ALIGN mode VELOCITY (resp. FREQUENCY).

W-ACCUMULATE, Different resolution R: <Real> T: <Real>
LAS, SUM or ACCUMULATE commands. Two spectra were added with the ALIGN mode CHANNELS, but had different resolutions and hence possibly different velocity or frequency scales.
User action: Check if these spectra were to be added. If you want to add them and preserve a common velocity (or frequency) scale, use the ALIGN mode VELOCITY (resp. FREQUENCY).

W-ACCUMULATE, Different rest frequencies R: <Real> T: <Real>
LAS, SUM or ACCUMULATE commands. Two spectra were added with the ALIGN mode CHANNELS or VELOCITY, but had different rest frequencies and hence possibly different velocity or frequency scales.
User action: Check if these spectra correspond to the same molecular line, use the MODIFY FREQUENCY to reset a common velocity scale and the ALIGN mode VELOCITY before adding them.

W-ACCUMULATE, Different velocities R: <Real> T: <Real>
LAS, SUM or ACCUMULATE commands. Two spectra were added with the ALIGN mode CHANNELS or
FREQUENCY, but had different velocities and hence possibly different velocity or frequency scales.
*User action*: Check if these spectra were to be added. If these spectra correspond to the same object, use the MODIFY VELOCITY to reset a common rest frequency scale and the ALIGN mode FREQUENCY before adding them.

W-ACCUMULATE, Drift not aligned in position
LAS, SUM or ACCUMULATE commands. Two continuum drifts were added but are not coincident in position.
*User action*: Select POSITION alignment using command SET, and retry.

W-ACCUMULATE, Origin table overflow
LAS, SUM or ACCUMULATE commands. Too many spectra were added and the list of original scan numbers is full. Scan numbers of any other added spectrum will be lost.
*User action*: none. The spectrum still contains the sum the all desired spectra, only the list is wrong.

E-ACCUMULATE, Positions are not compatible
LAS, SUM or ACCUMULATE commands. Two spectra have different positions and have not been added.
*User action*: If you want to add them, use the SET NOMATCH command before.

E-ACCUMULATE, <String> ranges do not intersect
LAS, SUM and ACCUMULATE commands. Two spectra were to be added with the ALIGN mode specified by <String> and INTERSECT mode, but do not overlap.
*User action*: what do you want to add ?...

W-ACCUMULATE, Spectra not aligned in sky frequency R: <Real> T: <Real>
LAS, SUM and ACCUMULATE commands. Two spectra were added with ALIGN mode CHANNELS, but have a different sky frequency scale.
*User action*: use the SET ALIGN FREQUENCY if you want to preserve a common frequency scale.

E-ACCUMULATE, Too many channels requested
LAS, SUM and ACCUMULATE commands. Two spectra were added with ALIGN mode COMPOSITE, but the final spectrum requires more channels than reserved in the program.
*User action*: most likely you are trying to make a composite spectrum with two unrelated frequency bands, such as CO and HCN... Note also that making composite spectra is not necessary to plot a few GHz from a spectral survey.

E-ADD, Cannot add Continuum with Line data.
LAS, ACCUMULATE command: The two observations are not of the same kind. SUM command: The index contains observations of different kinds.
*User action*: It is time to get some rest and a cup of coffee...

W-ADD, Gives sum of spectra for EQUAL weight
LAS, ACCUMULATE command: This command gives the sum of the spectra, not the average.
*User action*: This behaviour is consistent with that of SUM which gives the average over the index. Using several ACCUMULATE commands, and then dividing the final result by the number of added observations yields the same result as SUM.

W-ADD, Still experimental for Continuum
LAS, ACCUMULATE and SUM commands.
*User action*: Poor guinea pig... Do not worry too much, it usually works fine, but we would appreciate receiving detailed comments if you find something bizarre.

E-BASE, Insufficient memory for work space
LAS, BASE command. The work space to compute the best fit polynomial could not be dynamically allocated because the program exhausted its virtual memory quota.
*User action*: i) clear any plot and restart. ii) run CLASS in a single process (not from a sub-process). iii) Ask your system manager to extend the virtual memory limit.
E-BASE, No line window
LAS, BASE command. No line window has been defined.
User action : define one or more windows using the SET WINDOW command.

E-BASE, No valid baseline
LAS, BASE command : the windows covers the full spectrum and leave no channels to compute the baseline
User action : define windows which do not cover the spectrum, or (if possible) compute the baseline from another backend and use the command BASE LAST to apply it to this spectrum.

E-BASE, Sinus not converged
LAS, BASE command with SINUS method. The minimization routine failed to converge to a valid solution.
User action : The solution may still be sensible, look at it. If not, retry with different guesses.

9.3.2 C

E-CBUF, Write error block <Integer>
LAS, Input/Output routines : an error occurred during a write operation to the output file. This may be due to a large number of reasons (for example the disk may be off line, write locked or damaged) but is more or less fatal.
User action : See “Recovering Corrupted Data Files”

E-CHECK, Flag > 4
ANALYSE, LINES command : the input values contained an invalid flag.
User action : reenter valid values.

W-CHECK, Line <Integer> alone in a <String> Group
ANALYSE, LINES command. Input values were entered for dependent lines, but no lines depend on the one specified by <Integer>.
User action : although this is just a warning, it is likely that you made a typing mistake. Verify your input values and codes.

E-CHECK, No Independent <String>
ANALYSE, LINES command : input values were entered for dependent lines, but there is no independent line in VELOCITY, AREA or WIDTH (as specified by <String>).
User action : correct typing error.

E-CHECK, Several groups in <String>
ANALYSE, LINES command : input values were entered for dependent lines, but all lines must be dependent of a single one in this case. This may be due to a typing error, but also to a problem which cannot be solved by the current minimization routine.
User action : Correct typing error if any.

E-CUBS, Observation not open for write nor modify
LAS, Input/Output routine : an observation is being closed after or write or modify operation, but has never been opened for such operation.
User action : Submit an SPR. The output file may have been corrupted, see “Recovering Corrupted Data Files”

W-CONVERT, Earth velocity not converted
LAS, GET command or other observation reading routine. The velocity of the source was relative to Earth, and has not been converted to the current velocity system.
User action : None, but remember the velocity is wrong...

W-CONVERT, Velocity type was unknown, Set to <String>
LAS, GET command or other observation reading routine. The velocity type was undefined, and has been (arbitrarily) declared as <String> (LSR, Helio, etc...), but no velocity conversion was done.
9  ERROR MESSAGES AND RECOVERY PROCEDURES

W-COVAR, label too long, truncated
  LAS, plotting routines. A label was too long to be drawn.
  User action: None. Submit an SPR if you wish, with the label of course.

F-COVAR, Output file not opened
  LAS, Input/Output routine: an operation was requested on the output file, but no such file has been opened.
  User action: open the output file, and resubmit the command.

F-COVAR, Write error block <Integer>
  LAS, Input/Output routines: an error occurred during a write operation to the output file. This may be due to a large number of reasons (for example the disk may be off line, write locked or damaged) but is more or less fatal.
  User action: Notify your system manager. The output file may have been corrupted, see “Recovering Corrupted Data Files”

E-CUBE, Cannot create output image
  ANALYSE, CUBE command. The output image could not be created. This message is followed by a message 'E-CUBE: CUBEIMAG', ....' which gives the reason of the failure. The most likely reason is an exceeded quota (either disk or virtual memory)
  User action: if this is due to a disk quota, delete unwanted files and retry. If this is due to a virtual memory quota, clear any plot, run the program without parallel sub-processes, and if none of these solution works, ask your system manager.

E-CUBE, Cannot write output image header
  ANALYSE, CUBE command. The output image header could not be written. This error is due to an hardware error on the disk.
  User action: Notify the system manager, and retry again later.

I-CUBE, Check tolerance (SET MATCH) please
  ANALYSE, CUBE command. The output cube is too large, possibly because of roundoff errors in the offsets.
  User action: increase the SET MATCH to overcome roundoff errors.

E-CUBE, Continuum data not supported
  ANALYSE, CUBE Command. Spectral line data is required to make a data cube. Use command STRIP to build a continuum map from a set of scans.

E-CUBE, Cube too large, <Integer> by <Integer>
  ANALYSE, CUBE command. The output cube is too large, most likely because of roundoff errors in the offsets.
  User action: see “I-CUBE, Check tolerance (SET MATCH) please”

W-CUBE, Incoherent axis scale for scan <Integer>
  ANALYSE, CUBE command. One of the spectra in the index has an axis scale which does not match the cube definition. The spectrum has been written to the cube nevertheless, and the cube command proceeds.
  User action: May be this spectrum should not be placed in the cube, so drop it from the current index and rebuild the cube. If spectra have been obtained under slightly different conditions (for example with different sidebands) you must bring them on a common axis scale (use RESAMPLE) before combining them in a cube.

E-CUBE, Incoherent number of channels for scan <Integer>
  ANALYSE, CUBE command. One of the spectra has a number of channels different from the cube definition. The spectrum is not written to the cube, and the cube command aborts.
  User action: drop this spectrum from the current index and restart command.
9 \textbf{ERROR MESSAGES AND RECOVERY PROCEDURES}

E-CUBE, Incoherent offset type for scan \texttt{<Integer>}

\textbf{ANALYSE, CUBE command.} The coordinate system of the scan differs from the one of the cube. The scan is not written, and command CUBE aborts.

\textit{User action:} drop the scan from the index, and restart command.

E-CUBE, Index is empty

\textbf{ANALYSE, CUBE command.} A cube is being build from nothing...

\textit{User action:} define your current index before by using command FIND.

E-CUBE, Missing file name

\textbf{ANALYSE, CUBE command.} The output cube name has not been specified

\textit{User action:} specify file name

E-CUBE, No input file connected

\textbf{ANALYSE, CUBE command.} No input file is opened.

\textit{User action:} open input file and define the current index before using command CUBE.

9.3.3 D and E

W-DEVICE, No device active

\textbf{LAS, DEVICE command.} No graphic device is currently connected.

\textit{User action:} it is nevertheless possible to produce plots and hardcopies without a graphic device. A graphic device can be connected later at any time to visualize the plots.

W-DISPLAY, No Fit for scan \texttt{<Integer>}

\textbf{ANALYSE, DISPLAY command.} There is no fit (of the current method) for this scan.

E-DIVIDE, Spectra do not have the same velocity scale

\textbf{ANALYSE, DIVIDE command.} The R and T spectra must have the same number of channels and a common velocity scale to divide T by R.

\textit{User action:} Use command RESAMPLE to rescale both spectra on the same grid.

E-DIVIDE, Spectra have different number of channels

\textbf{ANALYSE, DIVIDE command.} The R and T spectra must have the same number of channels and a common velocity scale to divide T by R.

\textit{User action:} Use command RESAMPLE to rescale both spectra on the same grid.

E-DRAW, Invalid Centering

\textbf{ANALYSE, DRAW command with argument TEXT.} The centering code for the character string is invalid.

\textit{User action:} valid code range is 1-9.

E-DRAW, No cursor available

\textbf{ANALYSE, DRAW command.} The DRAW command is used without a completely explicit form, but no cursor is available to supply the missing parameters, either because no graphic device is connected, or because the current graphic device has no cursor.

\textit{User action:} supply all missing parameters to the command, or connect a graphic device with cursor.

E-DRAW FILL, Channel is not blanked

\textbf{ANALYSE, DRAW command with argument FILL.} The channel to be interpolated already contains a valid value. This may be because you typed F by mistake in the cursor mode.

\textit{User action:} If you really want to interpolate this channel, kill it before by using command DRAW KILL.

E-DRAW FILL, Channel outside spectrum

\textbf{ANALYSE, DRAW command with argument FILL.} The channel to be interpolated is outside the spectrum boundaries. This may be because you typed F by mistake in the cursor mode, or because the cursor lies outside the spectrum boundaries.

\textit{User action:} Move the cursor to the channel to be interpolated before typing the code F.
E-RAW KILL, Channel outside spectrum
   ANALYSE, DRAW command with argument KILL. The channel does not exist in this spectrum, probably
   because the graphic cursor does not point where you would like.
   User action: move graphic cursor towards the channel to kill.

E-DROP, Current index is empty
   LAS, DROP command. The current index is empty, so that no scan could be dropped from it...

I-DROP, <Integer>;<Integer> not in current index
   LAS, DROP command. The scan to be dropped is not in the current index. This is most likely due
to a typing error.

E-DROP, End of current index encountered
   LAS, DROP command with argument NEXT. The end of the index has been reached.

Error encoding <Real>
   Sexagesimal label routines. The value to be encoded in sexagesimal form is out of valid range.
   User action: Submit an SPR.

I-EIX, New data present
   LAS, NEW_DATA or FIND command with argument NEW. New data has been found in the input file,
exection resumes.

W-EIX, No input file opened
   LAS, NEW_DATA or FIND command with argument NEW. No new data can be found in nothing...

F-EIX, Read error file <String>
   LAS, NEW_DATA or FIND command with argument NEW. A read error occured while reading the
header block of the input file when looking for new data.
   User action : Retry, this is usually due to an interlock problem. If the error persists, check that no
more than 2 processes are accessing the specified file.

I-EIX, Switching to <String> Mode
   LAS, NEW_DATA command. The new data found is of a different type than the current one. Switching
to the new type occurred.
   User action : Remember not all actions are possible on every type of observations.

I-EIX, Waiting loop aborted by <"C>
   LAS, NEW_DATA or FIND command with argument NEW. The wait has been interrupted by user action.
   No new data was found.

9.3.4  F

E-FILE, IN OUT or BOTH please
   LAS, FILE command. The first argument is incorrect (typing error).

E-FILE, No default is provided for the file name
   LAS, FILE command. The second argument (file name) is missing.

I-FIND, <Integer> observations found
   LAS, FIND command, information message.

E-FIND, Nothing found
   LAS, FIND command, information message. Variable FOUND is reset to 0.
   User action : if you believe something should have been found, check the various selection criteria.
   One is probably different from what you had in mind.

I-FIND, <Integer> observations in index
   LAS, FIND command with argument APPEND, information message. The index has been appended
   and compressed to avoid duplicate entries to a given scan.
W-FITCONT, Solution not converged
W-FITGAUSS, Solution not converged
W-FITNH3, Solution not converged
W-FITSHELL, Solution not converged

**ANALYSE**, GAUSS command. The minimization failed to converge to a valid solution.

**User action**: Look at the results using command FIT. If they seem good enough, use command ITERATE until convergence is reached. If the warning persists, may be the problem is badly defined and the solution unreliable.

**E-FITNH3**, Unit X must be Velocity

**LAS**, GAUSS command for method NH3. The X unit must be velocity to fit NH3 hyperfine structure.

**User action**: use command SET UNIT X V, reset new guesses using command LINES and retry.

**E-FITSHELL**, No baseline removed

**ANALYSE**, GAUSS command, SHELL method. A base line must have been removed before fitting.

**User action**: use the BASE command to remove a baseline, and retry.

**E-FITSHELL**, Unit X must be Frequency

**ANALYSE**, GAUSS command, SHELL method. “Circumstellar shells” like profile fitting is only possible in Frequency units.

**User action**: use command SET UNIT X F, check the initial guesses and retry.

**F-FIX**, Current index is full

**LAS**, FIND command, usually with APPEND argument. Too many spectra match the current selection criteria, and the index is too large. Some spectra have been omitted from the index.

**User action**: Restrict the search by adding more severe selection criteria. Note in any case that you will have troubles processing the connected input file because it contains too many spectra. It would be wise to split this file in smaller ones, according to selected criteria.

**E-FIX**, More than <Integer> observations found

**LAS**, GET or other commands involving index finding operations, such as DROP. There are too many scans matching the requested criteria. Most likely, this error is due to an incoherent input file which contains twice the same scan (number AND version).

**User action**: It is possible to get rid of the duplicate scans by the following recipe. i) define a new output file to receive the valid spectra. ii) use a FIND/ALL without any selection criterium. iii) use the GET NEXT command to retrieve scans. iv) use the WRITE command for all scans but the duplicate scans that you should either skip or write under a new scan number (WRITE Number command). The new output file should be correct. If not see “Recovering Corrupted Data Files”.

**E-FIX**, Read error

**LAS**, FIND or other commands involving index finding operations. This may be due to a lack of input file (not all commands are explicitly protected against this specific mistake), but also to hardware problems.

**User action**: after checking for the input file, retry the operation. If the error persists, submit an SPR, and see “Recovering Corrupted Data Files”

**W-FOLD**, Cannot fold a single phased spectrum

**LAS**, FOLD command. The current spectrum was not observed in frequency switching mode, so that it cannot be folded.

**E-FOLD**, No channels left in spectrum

**LAS**, FOLD command. The align mode is INTERSECT, and the frequency switch is so large that no channel overlap from one phase to the others.

**User action**: Set the align mode to COMPOSITE, and redo the command.

**E-FSHELL**, Wrong Arguments : <Real>

**ANALYSE**, GAUSS command with method SHELL. The minimization routines went into a forbidden
region for the parameters. The minimization aborts.
User action: retry the fit with a different choice of input parameters.

9.3.5 G through I
E-GAUSS, No baseline removed
   ANALYSE, GAUSS command with method GAUSS. Minimization requires that a baseline be removed before.
   User action: remove a baseline with command BASE and retry the fit.
E-GET, End of Current Index Encountered
   LAS, GET command with argument NEXT. There are no more scans in the index.
I-GET, Entry <Integer> Found <Integer>;<Integer>
   LAS, GET command, information message.
E-GET, Index is empty
   LAS, GET command with argument FIRST. There is no first observation.
E-GET, No input file connected
   LAS, GET command. No input file is opened, so that the requested scan cannot be read...
E-GET, Observation <Integer>;<Integer> not Found
   LAS, GET command. The specified scan was not found in the input file. The version number (second <Integer>) is output only if it was specified in the command.
   User action: check for typing errors.
I-GREG, Creating Formatted <String>
   LAS, GREG command. This information message indicates that the output file <String> is formatted.
I-GREG, Creating Table <String>
   LAS, GREG command. This information message indicates that the output file <String> is a GILDAS Table. Note that this is the default.
E-GREG, Cannot open <String>
   LAS, GREG command. The output file (formatted or table) could not be created. This message is followed by more precise information about the failure. Most likely failures are: invalid name, exceeded quota, privilege violation.
   User action: it depends on the precise reason. Purge and delete any unwanted file if a quota problem is the basic cause and retry.
E-HARD, Conflicting options
   LAS, HARDCOPY command. Two conflicting options (e.g. /PORTRAIT and /LANDSCAPE) were given
   User action: remove the unwanted option
E-HARD, Invalid plot mode <String>
   LAS, HARDCOPY command. The scaling mode is invalid. Refer to the HELP HARDCOPY for valid scaling modes.
E-HARD, Non standard plot page, Specify option please
   LAS, HARDCOPY command. The plot page size does not match the default values, so that the orientation for hardcopy must be specified (/LANDSCAPE or /PORTRAIT).
S-HARD, <String>/<String> created
   LAS, HARDCOPY command with argument, information message. The first string indicate the vector file name, and the second the orientation.
S-HARD, Plot request successfully enqueued on GAG_PLOTTER
   LAS, HARDCOPY command with option /PLOT.
I-INIT, <String> initialized
LAS, FILE OUT command with third argument NEW specified, information message. A new empty output file has been initialized.

E-INIT, Open error file <String>
LAS, FILE OUT command with third argument NEW specified. The output file could not be created. This message is followed by more explanations about the precise reason (exceeded quota, privilege violation, invalid file name, etc...)
User action: respond according to the precise reason.

E-INIT, Write error file <String>
LAS, FILE OUT command with third argument NEW specified, information message. The output file was created, but could not be initialized properly. This is most likely due to a disk hardware error. User action: Notify the system manager and retry.

I-INPUT, <String> successfully opened
LAS, FILE IN command. Information message: the specified input file has been opened.

I-INPUT, <String> is reopened
LAS, FILE IN command. This message appears whenever a new input file could not be opened, to indicate which file is still currently opened.

I-INPUT, No input file opened
LAS, FILE IN command. This message appears whenever a new input file could not be opened, and no input file was previously opened.

F-INPUT, Non standard file <String>
LAS, FILE IN command. The specified file could be opened, but is not a valid CLASS file. User action: Check the file name.

F-INPUT, Open error file <String>
LAS, FILE IN command. The specified input file could not be opened. The message is followed by more details about the reason of the open failure.
User action: verify the file name you typed, and retry.

F-INPUT, Read error file <String>
LAS, FILE IN command. The specified file could be opened, but the first block could not be read. This is most likely because the file is not a standard CLASS file, but anything else. If you are sure that the file is a valid CLASS file, this indicates a hardware problem.
User action: Check the file name or notify the system manager accordingly.

E-ITERATE, Bad input parameters
ANALYSE, ITERATE command. The input parameters are non valid. This may happen because you have changed the X unit or the R spectrum till the last fit.
User action: verify X unit and change the input parameters, or use the GAUSS command.

9.3.6 K and L

E-KEEP, Input file must equal output file
ANALYSE, KEEP command. The KEEP command does an update of the spectrum, and the input and output files must be identical for this.
User action: open the input file as the output file and retry, or use the WRITE command.

I-KEEP, Observation <Integer>;<Integer> successfully updated
ANALYSE, KEEP command, information message.
9  ERROR MESSAGES AND RECOVERY PROCEDURES

E-KSHELL, Wrong Arguments: <Real>
ANALYSE, GAUSS or FIT commands with method SHELL. The minimization algorithm produced an
invalid solution.
User action: This also means that the fit is not converged. Restart it with different input parameters.

E-LAS, <String> Not yet implemented
LAS or ANALYSE languages, You are trying to use an undocumented, unimplemented command.
User action: Don’t

W-LAS, Save file could not be opened
LAS, SAVE command. The output file could not be opened, may be because of an exceeded disk
quota, or invalid file name.
User action: check the disk quota (and space) and the file name and retry.

E-LINES, Error opening guess file <String>
ANALYSE, LINES command. The input file for guesses could not be opened. This message is followed
by more details about the precise reason of the open failure. React accordingly.

E-LINES, Error reading guesses from <String>
ANALYSE, LINES command. The input file specified for guesses contains invalid data.
User action: check the input file format.

E-LINES, Invalid number of lines
ANALYSE, LINES command. The user specified either a negative number of lines or too many lines.

E-LINES, Not implemented for Continuum method.
ANALYSE, LINES command has no action for Continuum method.

W-LINES, Null area found, use manual mode
ANALYSE, LINES command. The area found in cursor mode is zero, may be because you forgot to
move cursor between the two edges...
User action: Try to give slightly different boundaries with the cursor, or use the manual mode
(/NOCURSOR or SET CURSOR OFF) as suggested.

W-LINES, Use of cursor with dependant or fixed lines not supported
ANALYSE, LINES command. The user is using the cursor to specify input values, but the flags
indicate that some parameters are fixed or dependent.
User action: It is not forbidden to use the cursor in such a case, but the user should realized that
the guesses will always be those found by the cursor routine, superseding any previous value (unless
a slash / is typed in the cursor routine).

I-LISTE, Input file is empty
I-LISTE, Input file contains
I-LISTE, Output file is empty
I-LISTE, Output file contains
I-LISTE, Current index is empty
I-LISTE, Current index :
LAS, LIST command, information messages

I-LISTE, List on file <String>
LAS, LIST command, information message specifying to which file the listing has been sent.

9.3.7 M

E-MAP, Continuum data not supported.
ANALYSE, MAP command. The current index contains continuum drifts, instead of spectra. MAP only
works on spectral line data.
E-MAP, Incoherent offset type for scan <Integer>

ANALYSE, MAP command. The current index contains offsets in different coordinate systems.

*User action*: drop the faulty spectrum from the index.

E-MAP, Index is empty

ANALYSE, MAP command. No map can be plotted.

*User action*: Define the current index using the FIND command and retry.

E-MAP, Map too large, <Integer> by <Integer>

I-MAP, Check tolerance (SET MATCH) please

ANALYSE, MAP command. The number of cells in the map defined by the index is too large, so that the spectra would be too small. This may be due to rounding errors in the offsets which lead to an anomalously small cell size.

*User action*: increase the tolerance in SET MATCH by a factor 2. If the error persists, give explicitly the cell size (/CELL option) if this does not help either, verify that the index does correspond to a single source.

E-MAP, Mode X and Y must be fixed

ANALYSE, MAP command. The X and/or Y axis has an automatic scaling enable, and must have a fixed scale.

*User action*: Specify a fixed scale for all spectra using the SET MODE command, and retry the MAP command.

E-MAP, No input file connected

ANALYSE, MAP command. No input file is opened.

*User action*: Open the input file, define the current index using the FIND command and retry the MAP command.

E-METHOD, Cannot open HFS description file.

E-METHOD, Error reading HFS description file.

ANALYSE, METHOD command with argument HFS. The Hyperfine structure of the line to be fitted could not be read from the specified file.

*User action*: Check the file name or content for proper format. See HELP METHOD.

E-METHOD, Too many HFS components

ANALYSE, METHOD command with argument HFS. The hyperfine structure specified has too many components.

*User action*: Relinking the program is required to increase this parameter. Contact the authors and supply the number of components required. You may solve adequately your problem by suppressing from the description the weakest components if they are negligible.

E-MOBS, Observation to be modified was not found

LAS, Input/Output routines, UPDATE or KEEP commands. The user attempted to modify an unexistant observation.

E-MOBS, Non-standard obs. block <Integer>

LAS, Input/Output routines, UPDATE or KEEP commands. The input/output file contains an invalid data block because it has been corrupted.

*User action*: see “Recovering Corrupted Data Files”, and notify your system manager because this is usually due to hardware errors.

I-MODIFY, Beam efficiency set to <Real>

ANALYSE, MODIFY command. The beam efficiency value was previously undefined and has been set to the specified value. No scaling occured.

E-MODIFY, Invalid beam efficiency <Real>

ANALYSE, MODIFY command. The beam efficiency value must be between 0 and the forward efficiency (less than 1).
E-MODIFY, MODIFY POSITION not supported for this type of projection.
ANALYSE, MODIFY command. Offsets relative to the new central position cannot be computed in the present coordinate or projection system.
User action: Submit an SPR, specifying the projection used. We will try to implement it properly in future releases.

I-MODIFY, Offsets set to <String>
ANALYSE, MODIFY OFFSETS command. Information message.

I-MODIFY, Reference channel set to <Real>
ANALYSE, MODIFY RECENTER command, information message.

I-MODIFY, Image frequency set to <Real>
ANALYSE, MODIFY IMAGE command, information message.

W-MODIFY, Unknown projection type, Radio projection assumed
ANALYSE, MODIFY command. Offsets relative to the new central position were in a unknown projection system.

F-MOX, Output file not opened
LAS, Input/Output routines. A WRITE operation was attempted before any output file was opened.
User action: open output file and retry the aborted command.

F-MOX, Read error block <Integer>
LAS, Input/Output routines. A WRITE operation failed because the output file index could not be read correctly.
User action: see “Recovering Corrupted Data Files” and notify your system manager.

F-MOX, Write error block <Integer>
LAS, Input/Output routines. A WRITE operation failed because the output file index could not be written correctly.
User action: see “Recovering Corrupted Data Files” and notify your system manager.

E-MOX, Wrong index address <Integer>
LAS, Input/Output routines. A WRITE operation failed because the index address is incorrect. This may be due to update a spectrum which does not exist in the output file.
User action: if the error was caused by an UPDATE or KEEP command, try a WRITE command instead; else, or if the error persists, submit an SPR.

9.3.8 N-Q

F-NOISE, No spectrum in memory
LAS, NOISE command. The command was specified without arguments, but there is no spectrum in the R memory to estimate a noise level.
User action: Specify a noise level, or get a spectrum in R.

I-OUTPUT, <String> is reopened
LAS, FILE OUT command, information message.

I-OUTPUT, <String> successfully opened
LAS, FILE OUT command, information message.

W-OUTPUT, No output file opened
LAS, FILE OUT command. The specified file could not be opened, or is not a valid CLASS file, but no previous output file has been reopened.

F-OUTPUT, Non standard file <String>
LAS, FILE OUT command. The specified file could be opened, but is not a CLASS data file.
User action: probably the file name has been mistyped.
E-OUTPUT, Old format not supported
  LAS, FILE OUT command. The specified file has an old data format, and cannot be used for output.
  *User action*: this file can be used for input. You may create a copy of it in new format by reading
  and writing to a new file all observations contained in it.

F-OUTPUT, Open error file <String>
  LAS, FILE OUT command. The specified output file could not be opened. The message if followed
  by more details about the specific reason for the open failure. The most likely reasons are i) a non
  existing file, ii) a privilege violation (the user has not write access to the output file), or iii) the file
  may also be locked by another user.
  *User action*: If you want to create a new output file, specify the 3rd argument NEW to the FILE OUT
  command.

F-OUTPUT, Read error file <String>
  LAS, FILE OUT command. The specified file could be opened, but most likely is not a CLASS data
  file.
  *User action*: verify the file name, which may have been mistyped. If it is correct, notify your system
  manager because the error is due to a hardware problem.

E-PEN, Number outside boundaries 0 - 15
  E-PEN, Default pen not changed
  LAS, PENCIL command. A bad pen number was specified, the pen number is left unchanged.

W-PEN, Colour outside boundaries 0 - <Integer>
  W-PEN, Default colour 0 used
  LAS, PENCIL command. A bad colour index has been specified, the colour index is reset to the
  default value 0.

W-PEN, Dashed pattern outside boundaries 1 - 7
  W-PEN, Default dashed pattern 1 used
  LAS, PENCIL command. A bad dashed pattern has been specified, the dash pattern is reset to the
  default value 1 (continuous line).

W-POLYNO, Baseline extrapolation is hazardous
  LAS, BASE command. Some of the line windows touch the spectrum edges, so that the baseline must
  be extrapolated in this range. Extrapolation is made by assuming a constant value in the range, and
  not by using the polynom because the solution is unstable.
  *User action*: If possible, try to avoid such situations by having a few channels free of lines at the
  spectrum edges. Note that the extrapolation is most likely incorrect.

W-POLYNO, Degree <Integer> would be even better
  LAS, BASE command. The specified degree would have given as good a fit to the baseline. It is
  hence possible that the current degree is too high and creates spurious oscillations.

E-POLYNO, NAG Error in <String>, ifail = <Integer>
  LAS, BASE command. The minimization failed or the minimum is ill-defined.
  *User action*: try to change the degree of the baseline, or the definition of the line windows to obtain
  a better solution.

W-PRINT, At entry <Integer> Scan <Integer>;<Integer>
  ANALYSE, PRINT command. The information to be listed is missing or incomplete for the specified
  scan. This message is usually preceded by "W-RSEC, Absent section <Integer>"

E-PRINT, Cannot print <String>
  ANALYSE, PRINT command. The specified keyword is not (yet) recognized.

E-PRINT, No input file connected
  ANALYSE, PRINT command. There is nothing to print...
E-PRINT, Option /TABLE invalid for FIT
ANALYSE, PRINT command. FIT results cannot be written to a GILDAS table, but only to a formatted file.

E-PRINT, No channel list
ANALYSE, PRINT CHANNEL command. The channel list is missing or incorrectly given.

E-PRINT, Syntax error in list: <String>
E-PRINT, Invalid list: <Integer> TO <Integer> BY <Integer>
E-PRINT, Incomplete list: <String>
E-PRINT, Empty list
ANALYSE, PRINT command (possibly with the CHANNEL argument). The list is incorrect in some way.
User action: Correct the faulty list and resubmit the command.

W-PRINT, Too many channels, list truncated
ANALYSE, PRINT CHANNEL command. The channel list is too long and has been truncated.
User action: check the output file to see which channels are missing and use another PRINT command to list the missing channels.

E-PRINT, Too many arguments in list
ANALYSE, PRINT command for velocity ranges. There are too many velocity ranges specified.

E-PRINT, Unable to open file <String>
ANALYSE, PRINT command with /OUTPUT option. The output file could not be created, possibly because of open file quota, disk quota, privilege violation to the directory, or simply invalid file name.

9.3.9 R
E-REBUF, Read error block <Integer>
LAS, Input/Output routines. The specified block could not be read. This is most likely due to a hardware error.
User action: see “Recovering Corrupted Data Files”. Notify your system manager.

W-RDATA, Too many channels, truncated
LAS, Input routines. The observation contains too many data points (more than 2048). Only 2048 channels were read.

W-REDUCE, Only valid for skydips
ANALYSE, REDUCE command. This command is only effective on skydips. Nothing happened.

E-RESCALE, Invalid unit type <String>
ANALYSE, RESAMPLE command. The resampling unit is unknown, it should be Velocity or Frequency.

E-RESCALE, New spectrum does not intersect the original one
ANALYSE, RESAMPLE command. The resampling formula creates a spectrum which does not intersect with the current one.
User action: use the HEADER command to see the current velocity or frequency scale for the current spectrum, and provide RESAMPLE arguments which are consistent with this scale.

E-RESCALE, Too many output channels, maximum is <Integer>
ANALYSE, RESAMPLE command. The parameters imply too many channels.
User action: decrease the resolution.

F-RIX, Input file not opened
LAS, Index routine. There is no input file connected.
F-RIX, Read error block <Integer>
LAS, Index routine. Unless it is preceded by the “E-RIX, Wrong index address” message, this error indicates a hardware problem.
User action: notify your system manager.

E-RIX, Wrong index address <Integer>
LAS, Index routine. The specified address does not exist. This error is usually due to an attempt to use a command such as GET or GET NEXT after an input file change.
User action: For these commands, use a FIND command before, or specify an observation number.

E-ROBS, Non-existing index entry <Integer>
LAS, Input/Output routine. The specified entry does not exist in the input file. This error is usually due to an attempt to use a command such as GET or GET NEXT after an input file change.
User action: For these commands, use a FIND command before, or specify an observation number.

E-ROBS, Non-standard obs. block <Integer>
LAS, Input/Output routine. The specified block does not have the valid CLASS data format.
User action: Submit an SPR, and see “Recovering Corrupted Data Files”

E-ROBS, Read error in index block
LAS, Input/Output routine. The main index block could not be read.
User action: Retry later. Submit an SPR and see “Recovering Corrupted Data Files” if the error persists.

F-ROX, Output file not opened
LAS, Index routine. There is no output file connected.

E-ROX, Wrong index address <Integer>
LAS, Index routine. The specified address does not exist in the output file.
User action: Submit an SPR.

F-ROX, Read error block <Integer>
LAS, Index routine. The specified block could not be read in the output file.
User action: notify the system manager. The file may have been corrupted, see “Recovering Corrupted Data Files”.

W-RSEC, Absent section <Integer>
LAS, Input/Output routines. The specified section does not exist in the current scan.

E-RUNLAS, No spectrum in memory
LAS, main program. The current command requires a scan, but no one has been read yet.

9.3.10  S
E-SET BOXLOCATION, Four arguments required
LAS, SET BOX command, information message, see help.

E-SET MODE, Invalid axis type <String>
LAS, SET MODE command. Only two axis are recognised by this command: X and Y.

E-SET PLOT_PAGE, Two arguments required
LAS, SET PLOT command, information message, see help.

I-SET, Clearing the plot and resetting BOXLOCATION
LAS, SET DEFAULT command, information message.

W-SET, Invalid <String> <String>
LAS, SET command, information message, see help.
I-SET, Default file extension set to <String>
LAS, SET command, information message, see help.

I-SET, Message display level set to <Integer>
LAS, SET MESSAGE, information message. Note that the message level has effect upon the number of
messages output by CLASS. The default level is 4. Higher values may suppress important messages.
Lower values can be used.

W-SKYDIP, Solution not converged
ANALYSE, REDUCE command. The minimization failed to converge.
User action: None, but the results may be unreliable.

E-SKYDIP, Stupid calibration No oxygen in atmosphere
E-SKYDIP, Stupid calibration No water in atmosphere
E-SKYDIP, Stupid calibration Zero atmospheric opacity
ANALYSE, REDUCE command. The minimization converged to a stupid solution.
User action: The calibration parameters of the skydip are probably incorrect. Discard the result.

E-SMOOTH, Box width out of range
ANALYSE, SMOOTH BOX command. The Box width is less than 2 or too large.

W-SMOOTH, Cross-validation not converged
ANALYSE, SMOOTH AUTO command. The cross-validation did not converge, no smoothing has been
done.
User action: None. There is no way to force the cross validation to converge.

W-SMOOTH, Error in AUTO smoothing
ANALYSE, SMOOTH AUTO command. An error in the cross-validation algorithm happened. No smooth-
ing has been done.
User action: Submit an SPR, supplying the data on which AUTO smoothing was tried.

E-SMOOTH, Number of points out of range
ANALYSE, SMOOTH command with argument NOISE.
User action: Decrease the number of channels

E-SMOOTH, Too few channels
ANALYSE, SMOOTH command. There are too few channels to make the smoothing.

E-SPECTRUM, No spectrum in memory.
LAS, SPECTRUM or PLOT commands. Not spectrum has been read yet.

E-STAMP, Mode X and Y must be fixed
ANALYSE, STAMP command. All spectra must have the same scale. Use the SET MODE command
to do so.

W-STAMP, More spectra in index than plotted.
ANALYSE, STAMP command. All spectra in index could not be plotted.
User action: Change the number of spectra along each direction, or drop observations from the
index.

E-STAMP, No input file connected
ANALYSE, STAMP command. No input file is currently opened.
User action: Open the desired input file, define an index using command FIND and retry the com-
mmand.

E-STRIP, Cannot create output image
E-STRIP, Cannot write output image header
ANALYSE, STRIP command. The output file could not be created.
User action: this is usually a problem of file name, privilege or disk quota. Check these parameters.
E-STRIP, Continuum data not supported
   ANALYSE, STRIP command with LINE type. The index contains data of continuum type.
   User action: drop the invalid observation from the index.

E-STRIP, Index does not define a strip
   ANALYSE, STRIP command. The index contains spectra which cannot be aligned along one of the
two main directions.
   User action: Check the index by command LIST, and build one which only contains offsets changing
in one direction. If you want to do a strip in a diagonal direction, the only possibility is to modify
the offset so that they describe a linear strip. Do not forget the changes if you do so...

E-STRIP, Index does not define a continuum map
   ANALYSE, STRIP command in CONTINUUM type. The index contains continuum drift which cannot
be aligned.
   User action: Check the index by command LIST, and build one which only contains offsets changing
in one direction.

E-STRIP, Too few points in position
   ANALYSE, STRIP command. The index contains less than two spectra.
   User action: Most likely your index does not contain what you believe. Check the index by command
LIST, and build the one you want.

E-STRIP, Too few points continuum drifts
   ANALYSE, STRIP command in CONTINUUM type. The index contains less than two drifts.
   User action: Most likely your index does not contain what you believe. Check the index by command
LIST, and build the one you want.

W-SUM, Added folded to unfolded spectra.
   LAS, SUM command. The index contains observations made in frequency switching mode which have
not been folded, and some which have been folded (or were made in other switching mode).
   User action: Check your index, drop the incoherent spectra.

I-SUM, Entry <Integer> found <Integer> ; <Integer>
   LAS, SUM command, information message.

E-SUM, Index is empty
   LAS, SUM command.
   User action: Build the index using command SUM, and try again.

E-SUM, No input file connected
   LAS, SUM command. There is no input file.
   User action: define an input file using command FILE, build the index using command FIND, and
try again.

E-SUM, Sum interrupted by "^C"
   LAS, SUM command. The sum was interrupted by the user by pressing "^C". In general, the R
spectrum contains nothing sensible in this case.
   User action: None. Do not write the spectrum.

9.3.11 T through Z

W-TAG, Observation <Integer> not found
   LAS, TAG command. The observation to be tagged does not exist, or is not of the current type.

E-TAG, Quality out of range
   LAS, TAG command. Quality must be in the range 0-9 (0 Unknown, 1 to 8 from Excellent to Very
bad, 9 reserved for deleted observations).
E-UPDATE, Can only update last versions
LAS, UPDATE command. Only last versions can be updated, the command had no effect.
User action: Use command WRITE instead of command UPDATE, but note that the new spectrum will become the last version in this case.

E-UPDATE, Input file must equal Output file
LAS, UPDATE command. The command had no effect.
User action: open the input as the output file too, and retry the command.

E-WBUF, Read error block <Integer>
LAS, Input/Output routines. A READ operation failed. The output file may be corrupted.
User action: notify your system manager. If the output file is corrupted, try to recover as many spectra as possible (cf “Recovering Corrupted Data Files”).

E-WBUF, Write error block <Integer>
LAS, Input/Output routines. A WRITE operation failed. This usually corresponds to an exceeded disk quota. This error may also be due to a write-lock error on the output device. In these cases, the WRITE operation can be done later. The problem can also be more serious and indicate a corrupted output file.
User action: check disk quota. Notify your system manager. If the output file is corrupted, try to recover as many spectra as possible (see “Recovering Corrupted Data Files”).

E-WDATA, Observation not open for write nor modify
LAS, Input/Output routines. An UPDATE or WRITE operation failed because the observation was not opened for write.
User action: This indicates an internal logic error in CLASS and should never occur. Submit an SPR.

E-WDATA, Insufficient space available for data section
LAS, Input/Output routines. An UPDATE or WRITE operation failed because the observation length is declared too short.
User action: This indicates an internal logic error in CLASS and should never occur. Submit an SPR.

F-WOBS, No output file opened
LAS, Input/Output routines. A WRITE operation failed because there is no output file.
User action: Open an output file.

F-WOX, Output file index is full
LAS, Index output routines. A WRITE operation failed because there are too many spectra in the output file. Nothing has been written.
User action: There are two possibilities i) open a new output file and write the spectrum. ii) you may consider compressing the output file by keeping only the last versions of the spectra, and then open the compressed output file and write the spectrum.

F-WOX, Read error block <Integer>
LAS, Index routines. A WRITE operation failed because the index could not be read correctly. This probably indicates a hardware problem.
User action: notify your system manager.

F-WOX, Write error block <Integer>
LAS, Index routines. A WRITE operation failed because the index block could not be written correctly. This probably indicates a hardware problem.
User action: notify your system manager.

I-WRITE, Observation <Integer>;<Integer> successfully <String>
LAS, WRITE or UPDATE commands, information message. <String> indicates whether the observations was updated or written.
E-WSEC, Observation not open for write or modify
LAS, Input/Output routines. A WRITE operation failed because the observation is defined as readonly.
User action: This indicates an internal logic error in CLASS. Submit an SPR.

F-WSEC, Too many sections
LAS, Input/Output routines. A WRITE operation failed because there are too many sections in the current spectrum to be written in the current file. This should not happen in the basic CLASS, but may occur to people using extended CLASS capabilities with user-defined sections.
User action: Submit an SPR, specifying the number of user-defined sections.

F-WSEC, Section <Integer> already written
LAS, Input/Output routines. A WRITE operation failed because the specified section already exists.
This indicates an internal logic error.
User action: Submit an SPR.

F-WSEC, Insufficient room available for section <String>
F-WSEC, Absent section <String>
LAS, Input/Output routines. An UPDATE operation failed because the specified section is longer in the R spectrum than in the output file.
User action: use the WRITE command instead (immediately to avoid leaving the observation half modified in the output file).

E-ZOOM, No cursor available
LAS, ZOOM command without arguments. No cursor is available to define the zooming window.
User action: supply the zooming window explicitly.
Index

ACCUMULATE, 11
BASE, 9, 13, 15, 16
/PLLOT, 13
LAST, 13
SINUS, 13
BOX, 12
/UNIT, 12
CUBE, 17
DIRECTORY, 29
DISMOUNT, 29
DISPLAY, 15
DIVIDE, 15
DRAW, 19
FILL, 20
KILL, 19
LOWER, 19
MASK, 19
TEXT, 19
UPPER, 19
WINDOW, 19
END, 30
EXIT, 9
FFT, 15, 16
FILE, 10
FIND, 9–11, 21
/ALL, 12
FIT, 9, 14
FOLD, 13
GAUSS, 9, 14
GET, 10, 16
GREG, 18
GRID, 16, 17
HARDCOPY, 9, 19
/PLLOT, 9
INITIALIZE, 29
ITERATE, 14
KEEP, 10, 15
LINES, 14
LIST, 29
MAP, 17
/CCELL, 17
/GRID, 17
/NLABEL, 17
/NNUMBER, 17
METHOD, 13
CONTINUUM, 14
GAUSS, 13
HFS, 14
NH3, 14
SHELL, 14
MOUNT, 29
NOISE, 15
PLOT, 9, 13, 16
POPUP, 17
PRINT, 18
/OUTPUT, 18
/TABLE, 18
AREA, 18
CHANNEL, 18
FIT, 18
MOMENT, 18
READ, 29
REDUCE, 16
RESAMPLE, 15, 16
RESIDUAL, 9, 15
REWIND, 30
SAVE, 9
SELECT, 30
SET, 9–11, 20, 22
ALIGN, 10
BAD, 11
BASE, 13
CALIBRATION, 11
CURSOR, 13, 14
FORMAT, 13
LINE, 11
MASK, 15
MATCH, 11, 12
MODE, 12
NOMATCH, 11
NUMBER, 11
OBSERVED, 11
OFFSETS, 9, 12
PLOT, 12
RANGE, 12
REDUCED, 12
SCAN, 12
SOURCE, 12
TELESCOPE, 12
TYPE, 11, 15
INDEX

UNIT, 12
WEIGHT, 11
WINDOW, 13
SKIP, 30
SMOOTH, 15
AUTO, 15
SPECTRUM, 13
STAMP, 17
STRIP, 17
SUM, 9, 11
SWAP, 9, 10

TITLE, 13

UPDATE, 10

WRITE, 9, 10, 30