CLASS
Continuum and Line Analysis Single-dish Software

A GILDAS software

November 21st, 2006

Version 1.1

Questions? Comments? Bug reports? Mail to: `gildas@iram.fr`

The GILDAS team welcomes an acknowledgement in publications
using GILDAS software to reduce and/or analyze data.
Please use the following reference in your publications:
`http://www.iram.fr/IRAMFR/GILDAS`

Documentation

In charge: S.Bardeau[1], J. Pety[1,2].
Active developers: P. Hily-Blant[3], S. Guilloteau[4].
Main past contributors: B. Delforge, T. Forveille, R. Lucas.

Software

In charge: S.Bardeau[1], J. Pety[1,2].
Active developers: P. Hily-Blant[3], S. Guilloteau[4].
Main past contributors: B. Delforge, T. Forveille, R. Lucas.

1. IRAM
2. Observatoire de Paris
3. Observatoire de Grenoble
4. Observatoire de Bordeaux

Contents

1 Introduction **1**

2 Cookbook **3**
2.1 A CLASSic Session . . . 3
2.2 Reading a spectrum from a formatted input file . . . 4
2.3 Exporting a spectrum to a formatted file . . . 4
2.4 Building a Datacube from a CLASS File . . . 5
2.5 Fitting an HyperFine Structure . . . 5
2.5.1 Assumptions . . . 5
2.5.2 Parameters of the multiplet . . . 6
2.5.3 The fitting procedure . . . 6
2.5.4 Typical Analysis Sequence . . . 7
2.6 Subtracting Baseline on Large Datasets . . . 8
2.7 Reading/Writing FITS file . . . 10
2.7.1 Exporting CLASS Spectra through FITS . . . 10
2.7.2 Importing FITS spectra into CLASS . . . 10

3 User manuel **13**
3.1 Generalities . . . 13
3.1.1 Data and Log Files . . . 13
3.1.2 Observation and Version Number . . . 13
3.1.3 Scan and Subscan Number . . . 14
3.1.4 Entry Number and Index . . . 14
3.1.5 R and T Memories . . . 15
3.1.6 Large Set of Spectra . . . 15
3.1.7 Variables . . . 16
3.2 Spectra Line Processing . . . 17
3.2.1 Plotting Spectra . . . 17
3.2.2 Removing Baselines . . . 19
3.2.3 Folding Frequency Switched Spectra . . . 20
3.2.4 Adding Spectra . . . 20
3.2.5 Analyzing profiles . . . 21
3.2.6 Gridding Spectra on a 3-D Data Cube . . . 24
3.2.7 Miscellaneous . . . 24
3.3 Continuum and Skydip Processing . . . 25
3.3.1 Continuum . . . 25
3.3.2 Skydip Processing . . . 26

3.4 Communication with the outer world . . . 26
3.4.1 Listing Scientifically Valuable Results . . . 26
3.4.2 Making Publishable Quality Figures . . . 27
3.4.3 Importing and Exporting Spectra From and To FITS . . . 29

4 Developer Manual (04-apr-2025) 31
4.1 Internal CLASS Format . . . 31
4.1.1 Contents of one observation . . . 31
4.1.2 File organization . . . 31
4.1.3 File index . . . 32
4.2 The Header and Data Sections . . . 33
4.2.1 General Parameters . . . 33
4.2.2 Position information . . . 34
4.2.3 Spectroscopic information (for spectra) . . . 35
4.2.4 Baseline information (for spectra or drifts) . . . 37
4.2.5 Scan numbers of initial observations . . . 37
4.2.6 Default plotting limits . . . 37
4.2.7 Switching information (for spectra) . . . 37
4.2.8 Calibration parameters . . . 38
4.2.9 For Skydips observations. No associated data. . . . 38
4.2.10 Gauss fit results (for spectra or drifts) . . . 39
4.2.11 "Stellar shell" profile fit results (for spectra) . . . 39
4.2.12 Hyperfine structure profile fit results (for spectra) . . . 40
4.2.13 Hyperfine structure absorption profile fit results (for spectra) . . . 40
4.2.14 Continuum drift description (for drifts) . . . 41
4.2.15 Beam-switching parameters (for spectra or drifts) . . . 41
4.2.16 Double gaussian and baseline fit results (for drifts) . . . 42
4.2.17 Herschel Space Observatory (HIFI) . . . 42
4.2.18 Comment Section . . . 43
4.2.19 Data Section . . . 43
4.3 Old OTF data format . . . 43
4.3.1 Data Section Descriptor . . . 43
4.3.2 Multiple spectra Data Section . . . 44
4.4 CLASS FITS format . . . 44
4.4.1 Simple SPECTRUM Mode . . . 44
4.4.2 BINTABLE Mode . . . 47
4.4.3 Once FITS Always FITS . . . 55

5 Internal Helps 57
5.1 LAS Language Internal Help . . . 57
5.1.1 Language . . . 57
5.1.2 ACCUMULATE . . . 58
5.1.3 ASSOCIATE . . . 59
5.1.4 AVERAGE . . . 60
5.1.5 BASE . . . 62
5.1.6 BOX . . . 63
5.1.7 CATALOG . . . 63
5.1.8 CONSISTENCY . . . 64

5.1.9 COPY . . . 64
5.1.10 DROP . . . 64
5.1.11 DUMP . . . 65
5.1.12 EXTRACT . . . 65
5.1.13 FILE . . . 66
5.1.14 FIND . . . 67
5.1.15 FITS . . . 71
5.1.16 FOLD . . . 72
5.1.17 GET . . . 72
5.1.18 HEADER . . . 73
5.1.19 IGNORE . . . 74
5.1.20 LIST . . . 74
5.1.21 LOAD . . . 76
5.1.22 MERGE . . . 76
5.1.23 MODIFY . . . 76
5.1.24 MULTIPLY . . . 82
5.1.25 NEW_DATA . . . 82
5.1.26 PLOT . . . 82
5.1.27 SAVE . . . 83
5.1.28 SET . . . 83
5.1.29 SHOW . . . 102
5.1.30 SPECTRUM . . . 102
5.1.31 STITCH . . . 103
5.1.32 SWAP . . . 104
5.1.33 TAG . . . 104
5.1.34 TITLE . . . 105
5.1.35 UPDATE . . . 105
5.1.36 WRITE . . . 105
5.2 ANALYSE Language Internal Help . . . 106
5.2.1 Language . . . 106
5.2.2 COMMENT . . . 107
5.2.3 DIVIDE . . . 107
5.2.4 DRAW . . . 107
5.2.5 FFT . . . 109
5.2.6 FILL . . . 110
5.2.7 GREG . . . 110
5.2.8 LMV . . . 111
5.2.9 MAP . . . 113
5.2.10 MEMORIZE . . . 114
5.2.11 MODEL . . . 115
5.2.12 NOISE . . . 116
5.2.13 POPUP . . . 116
5.2.14 PRINT . . . 117
5.2.15 REDUCE . . . 120
5.2.16 RESAMPLE . . . 120
5.2.17 RETRIEVE . . . 122
5.2.18 SMOOTH . . . 122

5.2.19 STAMP . . . 123
5.2.20 STRIP . . . 123
5.2.21 TABLE . . . 124
5.3 FIT Language Internal Help . . . 126
5.3.1 Language . . . 126
5.3.2 DISPLAY . . . 126
5.3.3 ITERATE . . . 127
5.3.4 KEEP . . . 127
5.3.5 LINES . . . 127
5.3.6 METHOD . . . 129
5.3.7 MINIMIZE . . . 131
5.3.8 RESIDUAL . . . 134
5.3.9 RESULT . . . 134
5.3.10 VISUALIZE . . . 134
5.4 MAP Language Internal Help . . . 134
5.4.1 Language . . . 134
5.4.2 XY_MAP . . . 135
5.5 DSB2SSB Language Internal Help . . . 138
5.5.1 Language . . . 138
5.5.2 INITIALIZE . . . 139
5.5.3 DECONVOLVE . . . 139
5.6 EXPERIMENTAL Language Internal Help . . . 140
5.6.1 Language . . . 140
5.6.2 DIFF . . . 141
5.6.3 FILTER . . . 141
5.6.4 MEDIAN . . . 141
5.6.5 RMS . . . 142
5.6.6 SUBTRACT . . . 142
5.6.7 UNBLANK . . . 142
5.6.8 UV_ZERO . . . 143
5.6.9 VARIABLE . . . 143
5.6.10 WAVELET . . . 143

Chapter 1

Introduction

CLASS is a software package for reducing spectroscopic data obtained on a single-dish telescope. It also has basic functionalities to reduce continuum drifts like pointing or focus.

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.

On a standard installation, CLASS is entered by just typing `class`. 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 analyse calibrated spectra in more detail.
- Language FIT gathers the spectra fitting functionalities.

Those languages are described in this manual. In addition, CLASS imports many functionalities defined and documented in other GILDAS packages:

- The command line interpretor is imported through the SIC (basic), GUI (for widgets) and VECTOR (miscellaneous) languages.
- The plotting possibilities through the GTVL (basic graphic actions), GREG1 (curve plotting), GREG2 (image plotting) and GREG3 (data cube plotting) languages.
- And the ephemerids and atmospheric contributions through the ASTRO language.

In addition to this manual, the reader should thus consult the SIC manual, and for further processing, the GREG manual.

This version of the CLASS documentation reflects the full rewritting of CLASS in `FORTRAN90`. In this process, many things have been changed, hopefully improved. If you are an experienced CLASS user, you may first want to consult the IRAM memo which describes only the changes in this version of CLASS.

Chapter 2

Cookbook

This part is a list of recipes enabling the beginner or the occasional user the get on the air very quickly, without losing his time searching the system's on-line HELP facility.

2.1 A CLASSic Session

```
 1    device image white
 2    set angle seconds
 3    set coordinates equatorial
 4    file in raw.30m
 5    set line 13co(1-0)
 6    set source ic348
 7    set telescope iram-30m-b30
 8    set observed 15-aug-1984
 9    find /offset 0  25
10    set weight time
11    average
12    set unit v f
13    set mode x -1 14
14    set mode y -0.5 7.5
15    set plot histogram
16    plot
17    hardcopy /print
18    hardcopy spectrum.ps  /dev ps fast
19    hardcopy spectrum.eps /dev eps color
20    set window 3 6 8 10
21    base 4 /plot
22    plot
23    lines 0
24    gauss
25    fit
26    residual
27    plot
28    sic delete reduced.30m
29    file out reduced.30m new
```

```
30    swap
31    write
32    save ic348
33    exit
```

1 Define the output device.

2-3 Select the coordinate system and angle unit.

4 Open the input file.

5-9 Build the index according to various criteria.

10-11 Averaged all spectra with weights $w_i = \Delta t \Delta\nu / T_{\rm sys}^2$.

12-16 Plot the averaged spectrum in a given velocity interval, with velocity for the lower axis and rest frequency for the upper axis.

17-20 Make hardcopies (directly to the printer, in a ps or eps file).

21-23 Subtract a polynomial baseline. Plot the fitted baseline. Plot the baseline subtracted spectrum. The baseline does not take into account channels corresponding to x in the two specified windows, in current units (here velocity).

24-26 Perform single Gaussian fitting and plot the result.

27-28 Compute the residuals and plot them.

29-30 Open a new output file.

31 Recover the baseline subtracted spectrum.

32 Write it.

33 Write the input commands in file ic348.class.

2.2 Reading a spectrum from a formatted input file

```
dev xl w
greg1\column x 1 y 2 /file ''filename''
model y x
plot
```

2.3 Exporting a spectrum to a formatted file

```
file in toto
find
get f
sic output toto.dat
for i 1 to channels
  say 'rx[i]' 'ry[i]' /format g12.4 g12.4
next
sic output
```

2.4 Building a Datacube from a CLASS File

The building of a regular grid in CLASS is done in two steps, in a similar fashion as the production of a PdBI regular grid from UV tables. The first step involves the building of a table: the spectra are written as rows in the Gildas internal data format for efficiency (command `table`). When creating the TABLE, the spectra are *not* resampled on a grid. The second step is the resampling of the non-regularly spaced spectra on a regular grid (command `xy_map`).

A simple example is given below where the cube is produced from a single .30m file. Default parameters are used by the `xy_map` command to define both the convolution kernel (1/3 of the $HPBW$) and the output grid (1/2 of the $HPBW$).

```
file in map       ! Open the input file
find              ! Build the index
consistency       ! Check first that the index is consistent
let name thecube  ! Use global SIC variables to define output cube name
let type lmv      ! The output cube is named ‘‘thecube.lmv’’
table ’name’ new  ! Build the non-gridded table
xy_map ’name’     ! Grid the data from the table
go view           ! Check the quality of the data reduction with the VIEW tool
```

The following example illustrates the way to build a cube from several .30m files. This is typically the case when the map has been repeated several times and the data have been calibrated and reduced in different files.

```
file in map1      ! Open the input file
find              ! Build the index
consistency       ! Check first that the index is consistent
let name thecube  ! Use global SIC variables to define output cube name
let type lmv      !
table ’name’ new  ! Build the non-gridded table as a new table
file in map2      ! Open the input file
find              ! Build the index
table ’name’      ! Append the index to the current table
file in map3      ! Open the input file
find              ! Build the index
table ’name’      ! Append the index to the current table
...
xy_map ’name’     ! Grid the data from the global table
go view           ! Check the quality of the data reduction with the VIEW tool
```

2.5 Fitting an HyperFine Structure

2.5.1 Assumptions

A1: same excitation temperature for all the components of the multiplet

A2: Gaussian profiles for the opacity as a function of frequency

A3: the lines all have the same width

A4: the multiplet components do not overlap

A5: the main beam temperature is well suited for your source

2.5.2 Parameters of the multiplet

When selecting the HFS method, you must give the name of a file that contains the relative positions and intensities of the components of your multiplet:

```
LAS> method hfs hfs-n2hp.dat
```

Relative intensities may be normalized or not. Here follow two examples, for the N_2H^+ multiplet.

```
7               ! 1st line: Number of components
 6.9360  1.     ! 1st column: Velocity shift, for each component,
 5.9841  5.     !             (in km/s) with respect to the reference
 5.5452  3.     !             component you choose (here the 5th one).
 0.9560  5.     ! 2nd column: The relative strength of each component.
 0.0000  7.     !             Here, they are not normalized.
-0.6109  3.     !
-8.0064  3.     !

7                  ! As before.
14.9424 1./27.     ! The reference velocity is now that of the last
13.9906 5./27.     ! component.
13.5516 3./27.     ! The intensities are normalized (you could also
8.9624  5./27.     ! put numerical values rather that fractions).
8.0064  7./27.     !
7.3955  3./27.     !
0.      3./27.     !
```

Let us call v_i and r_i the positions and relative intensities of the N components of the multiplet. We define $S = \sum r_i$. In the first case, we thus have $N = 7$ and $S = 27$.

2.5.3 The fitting procedure

According to assumptions A3 and A4, the opacity of the ith component is written:

$$\tau_i(v) = \tau_i \cdot \exp\left[-4\ln 2\left(\frac{v - v_{0,i}}{p_3}\right)^2\right] \tag{2.1}$$

where p_3 is the common FWHM of all components. The central velocity of component i is $v_{0,i} = v_i + p_2$, where p_2 is the velocity of the reference component (*i.e.* the one with $v_i = 0$).

The opacity of the multiplet is the sum of the N opacities:

$$\tau(v) = p_4 \sum_{i=1}^{N} r_i \cdot \exp\left[-4\ln 2\left(\frac{v - v_i - p_2}{p_3}\right)^2\right] \tag{2.2}$$

Given the opacity $\tau(v)$, the antena temperature is given by

$$T_{\rm ant}(v) = \frac{p_1}{p_4}\left(1 - e^{-\tau(v)}\right) \tag{2.3}$$

From these equations, we deduce:

$$\begin{aligned} \tau(v_i + p_2) &= p_4 \cdot r_i \qquad \text{(assumption A4)} \\ \sum_i \tau_i &= p_4 \cdot S \\ T_{\rm ant}(v_i + p_2) &= \frac{p_1}{p_4}\left(1 - e^{-p_4 r_i}\right) \\ T_{\rm ant}(v_i + p_2) &\approx p_1 \cdot r_i \end{aligned}$$

where the last equality holds in the optically thin regime.

This implies that the physical meaning of p_4 depends on the value of S: If the relative intensities are normalized to unity, then $S = 1$ and p_4 equals the sum of all centerline opacities.

The results of the HFS fitting procedure are:

```
Line     T ant * Tau           V lsr          Delta V            Tau main

1    1.313 ( 0.018)    3.783 ( 0.001)    0.589 ( 0.003)    0.230 ( 0.006)
```

where `T ant * Tau`=p_1, `V lsr`=p_2, `Delta V`=p_3 and `Tau main`=p_4.

According to assumptions A1 and A5, the hyperfine structure fitting procedure allows you to deduce the excitation temperature, since (assuming the Rayleigh-Jeans regime is valid, which is not true at $\lambda < 3$mm...):

$$T_{\rm ant}(v) = T_A^*(v) = \frac{B_{\rm eff}}{F_{\rm eff}}[T_{\rm ex} - T_{\rm bg}](1 - e^{-\tau(v)}) \tag{2.4}$$

Finally, the excitation temperature is given by:

$$T_{\rm ex} = T_{\rm bg} + \frac{F_{\rm eff}}{B_{\rm eff}}\frac{p_1}{p_4} \tag{2.5}$$

Note: the main group opacity is limited to the range $0.1 - 30$ since outside these limits, the problem becomes degenerate because the opacity no longer appears in the equations (in the optically thin limit, line ratio no longer depend on the opacity, and in the thick limit, $\exp(-\tau) \ll 1$).

2.5.4 Typical Analysis Sequence

This routine produces the figure 2.1.

```
 1  set plot histo
 2  set format brief
 3  clear
 4  clear alpha
 5  file in prov.30m
 6  find
 7  get f
 8  modify source TOTO
 9  set unit v c
10  method hfs hfs-n2hp.lin
11  minimize
```

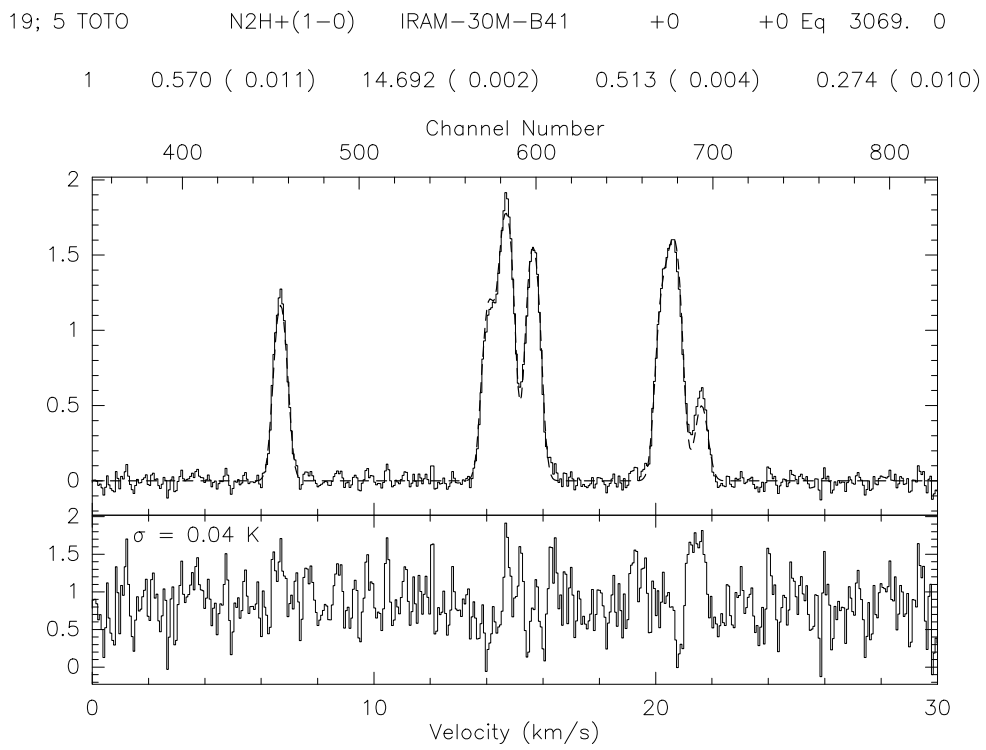


Figure 2.1: Result of the HFS fitting method (see 2.5.4).

```
12  set viewport 0.2 0.9 0.4 0.8
13  set mode x 0 30
14  box n o i p
15  spec
16  title
17  pen /c 0 /w 2 /das 2
18  visualize /pen
19  pen /def
20  display 1.5
21  set viewport 0.2 0.9 0.2 0.4
22  box p o i n /unit v upper
23  residual
24  spectr
25  define real rms
26  compute rms RMS ry
27  greg1\draw test 1 -0.5 "\gs = "'nint(rms*100)/100.'" K" 6 0 /box 7
```

2.6 Subtracting Baseline on Large Datasets

```
1  set verbose off
2  set level 10
```

```
 3  dev im w
 4  greg1\set plot portrait
 5  set format brief
 6  set angle sec
 7  file in co21.30m
 8  find /range -6 6 -6 6
 9  load
10  plot /index
11  set window /polygon 1
12  base 1 /index
13  plot /index
14  sic rename window-1.pol my-window-1.pol
15  sic delete co21-base.30m
16  file out co21-base.30m new
17  for i 1 to found
18    get n
19    base 1
20    write
21  next
22  file in co21.bas
23  find
24  load
25  plot /index
```

1. Makes CLASS quiet.
2. Turn off nearly all output messages.
3. Open an image window (necessary to plot 2D images).
4. Change the orientation to portrait.
5. Title in short format.
6. Set arcsecond units.
7. Open the file.
8. Build an index according to offsets.
9. Build a 2D array.
10. Display the whole index.
11. Define a polygon interactively to set spectral windows. By default the polygon is stored in a file window-1.pol.
12. Subtract a 1st order polynom to all records.
13. Display the result.
14. Change the polygon filename (to avoid overwritting it next time).

15. Delete output file if it exists.

16. Open the output file as new.

17-21 Make a loop over the index to subtract the baseline and write the result.

22-25 Display the result from the output file.

2.7 Reading/Writing FITS file

2.7.1 Exporting CLASS Spectra through FITS

```
1  set angle sec
2  file in co21.30m
3  find /range -2 2 -2 2
4  get f
5  set fits mode spectrum
6  fits write single.fits
7  find /range -2 2 -2 2
8  fits write index.fits /mode index
```

1. Angles are expressed in arcseconds.

2. Open the .30m file.

3. Build the index.

4. Store the first spectrum in `R` memory.

5. Default mode for writing `FITS` is set to single spectrum.

6. Write the `R` memory in `FITS` format file 'single.fits'.

7. Re-build the index.

8. Write the whole index in `FITS` format file 'index.fits'.

2.7.2 Importing FITS spectra into CLASS

```
1  set angle sec
2  fits read single
3  file out single.30m single /overwrite
4  write
5  file in single.30m
6  find
7  get f
8  plot
```

1. Angles are expressed in arcseconds.

2. Read a single `FITS` spectrum.

3-4 Open a new .30m file.

3. Write the `FITS` spectrum in the .30m file.

6-9 Check the result by plotting the spectrum from the .30m file.

Chapter 3

User manuel

3.1 Generalities

3.1.1 Data and Log Files

CLASS 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 subsections. These file are defined by the command `FILE (IN,OUT,BOTH) Filename`, possibly followed by `NEW` if a new file is to be initialized. The default extension of files is `.30m`. This can be changed with `SET EXTENSION .my_extension`.

CLASS also keeps two log files, named `class.log` and `class.mes`. They are created at CLASS start in the default directory `$HOME/.gag/logs`. This directory is defined in the SIC logical `gag_log:` which can be customized in the user customization file: `$HOME/.gag.dico`. The log files may be used to keep track of a batch or interactive work.

3.1.2 Observation and Version Number

Within CLASS, an *observation* should represent a single observing configuration, *e.g.* a single direction observed at a single central frequency with a single spectral resolution and in one polarization only (*i.e.* a single sky position, front-end and back-end combination). Each observation is given a number, named *observation number* at the time of creation of the CLASS data file. This number is then carried out in further manipulations.

Several version of a given observation may be stored in the same data file. Each version of a given observation thus represent different stages of the data reduction and all the versions of a given observation gives the history of the data processing. Each version of an observation is given a number (starting at 1), named *version number* at the time of creation of the CLASS data file. The version number increases automatically, each time the observation is modified (using `WRITE`). By default, only the last version of a given observation is relevant, *i.e.* `GET` reads the last version of an observation. It is possible (but not recommended) to store an observation without incrementing the version number with the `UPDATE` command so that 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 this, *i.e.* using the same observation number for very different things at time of creation of the CLASS data file, implies that you have to remember yourself which version corresponds to which configuration.

3.1.3 Scan and Subscan Number

For bookkeeping purpose, CLASS keeps track of a Scan and a Subscan number, which can be used as a selection criterium. The Scan number is attributed at observing time which is carried out in the CLASS data format. Moreover, two different observing mode are nowdays in common use in single-dish telescope:

- The pointed observing mode for which the telescope is pointed toward the source direction during all the integration to obtain only one spectrum. The scan is made of only one spectrum whose intensity is accumulated during the scan duration. Hence the Subscan number is always 1.
- The On–The–Fly (OTF) observing mode for which the telescope drift on source during the integration to make a small map. The scan is here composed of a collection of spectra dumped regularly (typically every 1 second) during a contiguous portion of time. Each dumped spectrum is also tagged by a subscan number whose value is incremented for each new OTF line (both to enable easy selection of a single line inside one OTF scan, and to ensure consistency with the 30m numbering). This subscan number is foreseen to always be greater than 1. There is one exception: when CLASS read data in old format, the subscan number is zero and a warning is issued.

3.1.4 Entry Number and Index

The `FIND` command offers the possibility to build an index of the observations matching several given selection criteria. The user can then easily process consistently only those observations. Each time a new index is formed using the `FIND` command, all the selected observations (which will belong to this index) are sequentially attributed a number, named *entry* number. This number goes from 1 to `found`, the number of observations in the current index. The entry number is never saved. It is just used as a number to process the current index in a loop.

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 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 "history" 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 spectra coming from Plateau de Bure Interferometer.
- `SET TYPE Name` specifies on which type of observations you deal with: "Continuum", "Line" or "Skydip".

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 to find all the versions of all 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.

Finally, the `SET SORT` force the `FIND` command to sort all the entries of an index in ascending order of a key parameter (*e.g.* lambda or beta offsets). `SET SORT number` implies the default order.

3.1.5 R and T Memories

CLASS keeps 2 observations in memory, in two different buffers, called `R` and `T`. The `R` memory is the only one that may be accessed directly; the `T` memory is only used for operations on spectra (additions,...). The `GET n` command places the spectrum corresponding to entry number `n` in the `R` buffer, while the previous `R` content is stored in the `T` buffer. The command `SWAP` exchanges both memories.

3.1.6 Large Set of Spectra

It is today possible with the IRAM–30 m to map a square degree field in CO (2-1). As an order of magnitude, this gives a final spectra cube of about 10^6 spectra with a slightly oversampling of $4''$. An observer who has just spent a few hours observing the same source in OTF mode may want to process all the dumped spectra at once even though they do not belong to the same scan.

The `LOAD` command gathers all the individual spectra currently in the index as a 2D array for future work, in particular visualization. This requires that all the spectra currently in the index are coherent, *i.e.* same source name, same line name and above all exactly the same frequency sampling. The latter can easily be achieved just by resampling. No checking is currently done about this, *i.e.* this is currently the responsibility of the observer to ensure a coherent frequency axis. The `/INDEX` option of commands like `PLOT` or `BASE` modifies there behavior to directly work on the 2D array defined with the `LOAD` command.

Nota Bene: The `STRIP` command which is producing a Velocity-Position plots, is *obsolescent*. Indeed the same functionality can be achived by the combination `LOAD; PLOT /INDEX` if the index is correctly choosen. And the `LOAD` command is much more powerful.

3.1.7 Variables

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 `SET VARIABLE`.

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 either to write a SIC loop that goes through the index or to test for actions which should be performed only if something exists in the index. 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.

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).

```
TELESCOPE       Character*12, RW, Telescope name
NUMBER          Integer,      RW, Observation number
VERSION         Integer,      RO, Version number
DATATYPE        Integer,      RO, Type of observation
                                  0 Line, 1 Continuum, 2 Skydip
QUALITY         Integer,      RO, Quality of data
SCAN            Integer,      RO, Original scan number
UTOBS           Double,       RO, UT of observation (Radians)
LSTOBS          Double,       RO, LST of observation (Radians)
AZIMUTH         Real,         RW, Azimuth of observation (Radians)
ELEVATION       Real,         RW, Elevation of observation (Radians)
TSYS            Real,         RW, System temperature
TIME            Real,         RW, Integration time (Seconds)

SOURCE          Character*12, RW, Source name
LAMBDA          Double,       RW, Longitude of source (Radians)
BETA            Double,       RW, Latitude of source (Radians)
OFF_LAMBDA      Double,       RW, Offset in longitude (Radians)
OFF_BETA        Double,       RW, Offset in latitude (Radians)
EQUINOX         Real,         RW, Equinox 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.

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.

3.2 Spectra Line Processing

3.2.1 Plotting Spectra

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).

- `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).

Single spectrum

Single spectrum are usually plotted using the following commands:

- `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.

Spectra map

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.

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.

Large set of spectra

An efficient way to look at a large set of coherent spectra (*e.g.* observed in OTF mode) is to plot them as a 2-D image where the intensity is colour coded. The image is formed by applying the

LOAD on the current index. Then the /INDEX option modify the single spectra plotting commands as follow:

- BOX /INDEX, which plots the frame, *i.e.* the entry number (Y axis) as a function of the velocity and/or frequency (X axis). The ranges of X and Y axes are controlled by the SET MODE command. SET MODE Y always control the intensity range.
- SPECTRUM /INDEX, which plots the image.
- TITLE /INDEX, which writes a header above the frame. Range of parameters (*e.g.* scan number, beta and lambda offsets, ...) are written.
- PLOT /INDEX, which performs all of CLEAR; BOX /INDEX; SPECTRUM /INDEX; TITLE /INDEX in a single operation.

3.2.2 Removing 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 polynom value at the boundary. The algorithm warns if the polynomial degree is too high. See section 2.6 for a typical baseline fitting session.

The user first defines line windows by the command SET WINDOW with the following syntax:

```
SET WINDOW [wl1 wu1 [wl2 wu2 [...]]] [/VAR array]
           [/POLYGON [N] [filename1...filenameN]]
           [/NOCURSOR]
```

The POLYGON option enable the definition of 2D polygons on images obtain with LOAD; PLOT /INDEX when working on all the index. Line window values may be entered numerically as arguments, red from variables and line polygons may be red from input files. If available, the cursor may be used to define the windows or polygons. In the window 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. In the case of polygons, each left clic defines a gon and a right clic terminates the operations. The polygons may leak out of the image. Several polygons may be defined in case the line appear at very different velocities.

Up to 100 windows or 5 polygons may be defined. BASE then fits a polynomial to the part of the spectrum outside the line windows. 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.

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.

When working on an individual spectrum (not the index), 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-Veleta 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...

When working on the whole index, the baseline are fitted spectrum per spectrum and the baseline–corrected spectra are stored in the 2D array ready for plotting with the next `PLOT /INDEX` command. However, baseline fitting results are lost and an explicit loop on the index entry must be use including a new baseline computation) to store the results with the `WRITE` command.

3.2.3 Folding Frequency Switched Spectra

Spectra obtained by Frequency Switching need to be folded to obtain the source spectra. 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 on the `R` buffer. The number of channels is decreased to keep only the relevant part of the resulting spectrum.

3.2.4 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 `SET ALIGN Mode`:

- `CHANNEL` in which spectra are added channel by channel. This is only useful when the spectra have been obtained in strictly identic conditions. Warning messages are given when this is not the case.
- `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, VELOCITY` and `FREQUENCY` are relevant for Line observations, while `POSITION` is relevant only for Continuum 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 `AVERAGE` and `ACCUMULATE` commands. `AVERAGE` 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 `AVERAGE`.

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 `AVERAGE` and `ACCUMULATE`. `AVERAGE` operates globally on all the spectra in the index, while `ACCUMULATE` adds the `R` and `T` buffers into `R`. `AVERAGE` 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...

3.2.5 Analyzing profiles

The CLASS user may analyse spectra by fitting profiles. The fitting commands are available from the `FIT` language. The minimization method is taken from the MINUIT system of CERN, modified and optimized for this purpose. Reliability proved 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 10 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` (see details below) 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 10 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 10 independent lines can be fitted, and only codes 0 and 1 are allowed. The X unit must be Velocity.

- `METHOD HFS FileName` 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.

`METHOD SHELL` in details. The fitted function is:

$$f(\nu) = \frac{\mathcal{A}}{\Delta\nu} \frac{1 + 4H\left[(\nu - \nu_0)/\Delta\nu\right]^2}{1 + H/3}$$

where the fitted parameters are:

1. $\mathcal{A}$: the area under the profile (in K MHz),
2. ν_0: the middle frequency (in MHz),
3. $\Delta\nu/2$: the half width at zero level (in MHz),
4. H: the Horn/Center parameter (dimensionless, see below)

The central value is $f(\nu_0) = \frac{\mathcal{A}}{\Delta\nu}\frac{1}{1+H/3}$ whilst the value at the edge is $f(\nu_0+\Delta\nu/2) = \frac{\mathcal{A}}{\Delta\nu}\frac{1+H}{1+H/3}$. The edge-to-center intensity ratio value is thus dictated by the Horn/Center parameter H according to

$$\frac{f(\Delta\nu/2)}{f(0)} = 1 + H$$

The center-to-edge frequency shift corresponds to an expanding velocity

$$v_{\rm exp} = c\,\frac{\Delta\nu/2}{\nu_0}$$

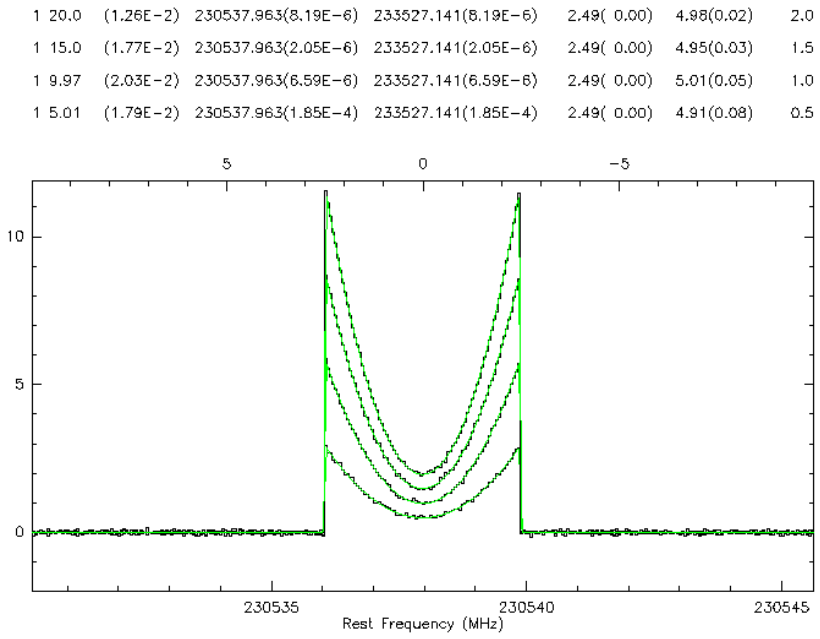
Figure 3.1 shows synthetic shell-like profiles, for which the area takes values $\mathcal{A} = 5$ to 20 by 5 K MHz. The full width at zero level is $\Delta\nu = 3.83$ MHz in all cases, which corresponds to $v_{\rm exp} = 2.49$ km.s^{-1}at 230.537 GHz. The Horn/Center parameter is $H = 5$ (top) or $H = -1$ (bottom, for which the intensity at the edge is zero).

The `FIT` commands are:

- `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



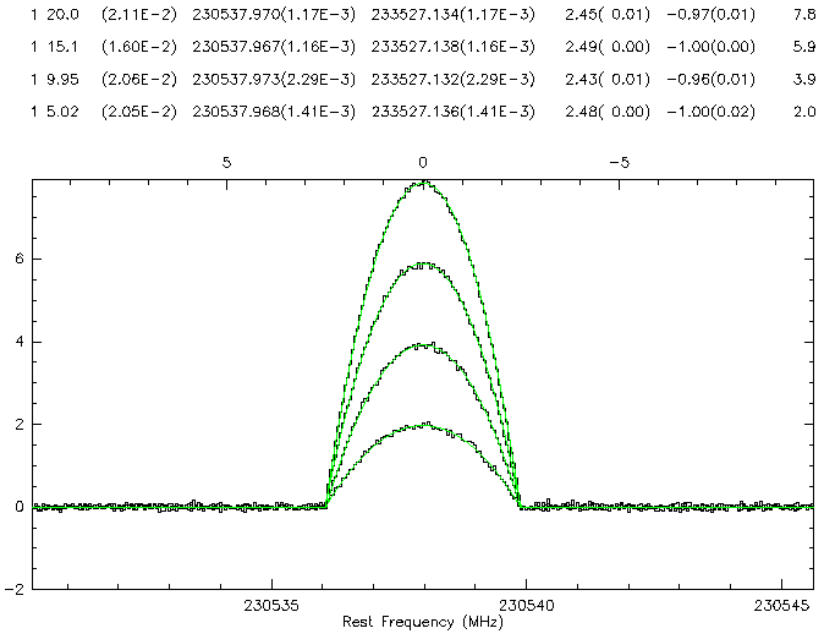


Figure 3.1: Four synthetic shell-like profiles, corresponding to increasing areas $\mathcal{A} = 5$ to 20 by 5 K MHz, a $\Delta\nu$ of 3.83 MHz or $v_{\rm exp} = 2.49$ km.s^{-1}at 230.537 GHz. The results of fitted shell profiles are indicated at the top. The upper scale is km.s^{-1}. *T*op: The Horn/Center parameter is $H = 5$. *B*ottom: Same as above for $H = -1$.

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 the cursor is available. 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.

- `MINIMIZE` 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 `MINIMIZE`, but starts from the previous minimization results. Only the Gradient method is used. Consequently, this command is only useful close to the minimum.

- `VISUALIZE [N] [/PEN]` 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 and made available through the corresponding variable section (see `SET VARIABLE` help).

3.2.6 Gridding Spectra on a 3-D Data Cube

See memo IRAM 2005-1 (CLASS evolution: I. Improved OFT support) available here
`http://iram-institute.org/medias/uploads/class-evol1.pdf`

3.2.7 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.

3.3 Continuum and Skydip Processing

3.3.1 Continuum

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 the `SET TYPE CONTINUUM` command. The prompt then changes to `CAS>` (Continuum Analysis System). You can return to Spectroscopy mode later on typing the `SET TYPE SPECTROSCOPY or LINE` command and the prompt changes to `LAS>` (Line Analysis System).

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 a set 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.

3.3.2 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 Communication with the outer world

3.4.1 Listing Scientifically Valuable Results

Command `PRINT` offers a way to list a number of valuable information on screen or in an ASCII/binary file:

- `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 program.

As all commands using a set 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`.

3.4.2 Making Publishable Quality Figures

CLASS has many functionalities to directly produce publishable quality Figures. All the GREG commands are imported in CLASS to fully annotate plots, superpose spectra with related data, stack various plots and then make hardcopy (like Post-Scripts files). A few guidelines are given here on essential GREG commands. For more details, users are advised to read the GREG manual.

Moreover, CLASS commands like `DRAW` and `GREG` implements fancy functionalities of common use when producing figures around spectra.

GREG functionalities

CLASS is mainly 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`).
- `SET PLOT_PAGE ...` Define the page size. Warning: You will get into trouble if you want to abreviate this command to `SET PLOT` as `SET PLOT` is a valid CLASS command used to indicate whether spectra are broken lines of histograms. A way out of this is to abbreviate the GREG command as `G\SET PLOT`.
- `SET TICK Size` Define the tick size in centimeters.

Note that you do not need to open a plotting window to produce a hardcopy through the `HARDCOPY` command. It is only much more convenient, but the plot (*e.g.* sequence of commands) and the way it is displayed (*e.g.* PS format, bitmap on a computer screen) are two completely independent things.

Bridge toward GREG functionalities

The spectra and the results of their analysis (like the fits) are in CLASS internal buffers not always easily accessible for plotting with the GREG commands. CLASS thus implements the `GREG` command which is intended to produce a direct interface with GREG for plots of spectra. It creates a GILDAS table which can be read using the standard GREG commands for further plotting. 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.

Annotations

CLASS has two `DRAW` commands to annotate plots: *i)* The standard GREG `DRAW` command which can be accessed by typing `G\DRAW` (please look at the GREG manual) and *ii* a special flavor of `DRAW` customized for annotating spectra. The CLASS flavor is the default one used when typing just `DRAW` The basic operations performed by this flavor of the `DRAW` command 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.

- `DRAW UPPER 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 line 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.

3.4.3 Importing and Exporting Spectra From and To FITS

No data reduction package has all the functionalities any user dream about. But a user may know that the functionality he needs is available in a very specific package. Here comes the need to exchange data between packages. The current standard answer to this problem is FITS. CLASS to FITS conversion (and vice-versa) is done by command `FITS`. In addition, all functionalities provided by the SIC command `DEFINE FITS` are of course available. For a description of the FITS format see the original paper by Wells et al. (Astron. and Astrophys. Suppl.).

The CLASS `FITS` command has the following syntax:

```
FITS READ Filename[.fits]
```

to read a FITS file and create CLASS data from it, or

```
FITS WRITE Filename[.fits] [/BITS Nbits] [/MODE SPECTRUM|INDEX]
```

to write a FITS file from CLASS data.

In addition, default values can be supplied by the `SET FITS` command.

```
SET FITS BITS Nbits
SET FITS MODE Spectrum|Index|None
```

From FITS to CLASS

```
FITS READ Filename[.fits]
```

will read a FITS file and create CLASS data from it. It is expected to work under the following conditions:

1. The `Filename.fits` file contains one spectrum, with (a subset of) the FITS keywords which are described in the previous section. FITS Keyword redefinition is possible.
2. **OR** The `Filename.fits` contains a BINTABLE, also with recognized FITS keywords.
3. **No more, no less**

This may look akwardly restrictive, but is already powerful if you have thought about your data destination when creating the FITS file.

FITS Keyword redefinition

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 to support their own needs. Thus, several "flavors" of FITS coexist. A detailed description of the CLASS FITS flavors is given in chapter 4. Unknown keywords are normally ignored, but CLASS supports FITS keyword redefinition. If you receive a file 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`.

From CLASS to FITS

To write a FITS file from CLASS data, use the following command:

```
FITS WRITE Filename[.fits] [/BITS Nbits] [/MODE SPECTRUM|INDEX]
```

The command will create a simple FITS file from the current Spectrum (in `SPECTRUM` mode), or a FITS BINTABLE from the current Index (in `INDEX` mode). The number of bits can be controlled. Default values for the mode and the number of bits can be supplied by the `SET FITS` command.

```
SET FITS BITS Nbits
SET FITS MODE Spectrum|Index|None
```

In `INDEX` mode, it is up to the user to make sure that the index is consistent (same number of channels, etc..., for all spectra in index).

Chapter 4

Developer Manual (04-apr-2025)

Please check directly the CLASS sources to get the most up-to-date information.

4.1 Internal CLASS Format

The CLASS Data Format is built upon the Classic Data Container. Please refer to the dedicated documentation for details.

4.1.1 Contents of one observation

Data, *i.e.* observational parameters as well as spectra, is organized in the following way:

- One observation is self-contained. All the information needed to reduce it is recorded on the same few disk blocks. It may be one spectrum or one continuum drift.
- Each observation is divided in several sections, containing header parameters or data:
  - General information (date, times, local coordinates, sequence number, ...)
  - Positional information (source, name, astronomical coordinates, equinox, offsets, ...)
  - Spectral information (number of spectra, line names ans frequencies, resolutions, number of channels, ...)
  - Data
  - ...

In the Classic nomenclature, one observation is saved into one *entry* in a Classic file. Each entry is composed of an entry descriptor + the observation itself (header+intensity array). Please refer to the section *The File Entries* in the Classic documentation for details about the way an entry is written in the file.

4.1.2 File organization

CLASS files strictly respect the Classic Data Container specifications (see the dedicated documentation for details). In particular, the file identification codes (first 4 bytes) are described in this documentation. CLASS files use the *kind* 1 in the File Descriptor.

4.1.3 File index

The Entry Indexes, gathered into Extension Indexes, are ruled by the Classic Data Container specifications. According to these, the index contents are let free to Class. They are described below for files version 1 and 2.

Table 4.1: Index version 1 (found in files version 1). Note that index length is 32 words (so that 4 of them fit in a 128 words record), even if 10 words are left unused. The subscan number was originally absent from the old version of the data format (see related memo for details); it is read as 0 from such files.

Position	Parameter	Fortran Kind	Unit	Purpose
1	`bloc`	Integer*4	Record	Record where observation is to be found
2	`num`	Integer*4		Observation number
3	`ver`	Integer*4		Observation version
4-6	`csour`	Character*12		Source name
7-9	`cline`	Character*12		Line name
10-12	`ctele`	Character*12		Telescope name
13	`dobs`	Integer*4	GAG date	Observation date
14	`dred`	Integer*4	GAG date	Reduction date
15	`off1`	Real*4	Radian	Lambda offset
16	`off2`	Real*4	Radian	Beta offset
17	`type`	Integer*4	Code	Coordinate system
18	`kind`	Integer*4	Code	Kind of data
19	`qual`	Integer*4	Code	Quality of data
20	`scan`	Integer*4		Scan number
21	`posa`	Real*4	Radian	Position angle (drifts)
22	`subscan`	Integer*4		Subscan number (absent from old data format)
23-32	(unused)			Padding to 32 words

Table 4.2: Index version 2 (found in files version 2)

Position	Parameter	Fortran Kind	Unit	Purpose
1-2	bloc	Integer*8	Record	Record where observation is to be found
3	word	Integer*4	Word	Word where the obs. starts in this record
4-5	num	Integer*8		Observation number
6	ver	Integer*4		Observation version
7-9	csour	Character*12		Source name
10-12	cline	Character*12		Line name
13-15	ctele	Character*12		Telescope name
16	dobs	Integer*4	GAG date	Observation date
17	dred	Integer*4	GAG date	Reduction date
18	off1	Real*4	Radian	Lambda offset
19	off2	Real*4	Radian	Beta offset
20	type	Integer*4	Code	Coordinate system
21	kind	Integer*4	Code	Kind of data
22	qual	Integer*4	Code	Quality of data
23	posa	Real*4	Radian	Position angle (drifts)
24-25	scan	Integer*8		Scan number
26	subscan	Integer*4		Subscan number

4.2 The Header and Data Sections

Here we describe the contents of the main header sections (except from the storage declarations in the CLASS program itself). Note that, in the data file, the actual length of some of the sections is variable, *e.g.* the length of the switching information section depends on the number of different phases in the switching procedure.

4.2.1 General Parameters

The General section in memory has the particularity to concatenate information written in the index on disk and written in the General section on disk.

Observation version 2 since 10-mar-2023 (added parallactic angle):

```
! GENERAL: General parameters, always present.
integer(kind=4), parameter :: class_sec_gen_id=-2
integer(kind=4), parameter :: class_sec_gen_len=11
type general
   sequence
   ! Written in the index on disk
   integer(kind=4) :: num     ! [         ] Observation number
   integer(kind=4) :: ver     ! [         ] Version number
   integer(kind=4) :: teles(3)! [         ] Telescope name
   integer(kind=4) :: dobs    ! [MJD-60549] Date of observation
   integer(kind=4) :: dred    ! [MJD-60549] Date of reduction
   integer(kind=4) :: kind    ! [     code] Type of data
```

```
   integer(kind=4) :: qual    ! [     code] Quality of data
   integer(kind=4) :: scan    ! [         ] Scan number
   integer(kind=4) :: subscan ! [         ] Subscan number
   ! Written in the section on disk
   real(kind=8)    :: ut      ! [  rad] UT of observation
   real(kind=8)    :: st      ! [  rad] LST of observation
   real(kind=4)    :: az      ! [  rad] Azimuth
   real(kind=4)    :: el      ! [  rad] Elevation
   real(kind=4)    :: tau     ! [neper] Opacity
   real(kind=4)    :: tsys    ! [    K] System temperature
   real(kind=4)    :: time    ! [    s] Integration time
   real(kind=8)    :: parang  ! [  rad] Parallactic angle
   integer(kind=4) :: xunit   ! [ code] X unit (if X coordinates section is present)
end type general
```

Observation version 1/2 (obsolete for writing):

```
! GENERAL: General parameters, always present.
integer(kind=4), parameter :: class_sec_gen_id=-2
integer(kind=4), parameter :: class_sec_gen_len=9
type general
   sequence
   ! Written in the index on disk
   integer(kind=4) :: num     ! [         ] Observation number
   integer(kind=4) :: ver     ! [         ] Version number
   integer(kind=4) :: teles(3)! [         ] Telescope name
   integer(kind=4) :: dobs    ! [MJD-60549] Date of observation
   integer(kind=4) :: dred    ! [MJD-60549] Date of reduction
   integer(kind=4) :: kind    ! [     code] Type of data
   integer(kind=4) :: qual    ! [     code] Quality of data
   integer(kind=4) :: scan    ! [         ] Scan number
   integer(kind=4) :: subscan ! [         ] Subscan number
   ! Written in the section on disk
   real(kind=8)    :: ut      ! [  rad] UT of observation
   real(kind=8)    :: st      ! [  rad] LST of observation
   real(kind=4)    :: az      ! [  rad] Azimuth
   real(kind=4)    :: el      ! [  rad] Elevation
   real(kind=4)    :: tau     ! [neper] Opacity
   real(kind=4)    :: tsys    ! [    K] System temperature
   real(kind=4)    :: time    ! [    s] Integration time
   integer(kind=4) :: xunit   ! [ code] X unit (if X coordinates section is present)
end type general
```

4.2.2 Position information

Observation version 2 since 26-mar-2015:

```
! POSITION: Position information.
integer(kind=4), parameter :: class_sec_pos_id=-3
```

```
integer(kind=4), parameter :: class_sec_pos_len=14
type position
   sequence
   integer(kind=4) :: sourc(3) ! [    ] Source name
   integer(kind=4) :: system   ! [code] Coordinate system
   real(kind=4)    :: equinox  ! [year] Equinox of coordinates
   integer(kind=4) :: proj     ! [code] Projection system
   real(kind=8)    :: lam      ! [ rad] Lambda of projection center
   real(kind=8)    :: bet      ! [ rad] Beta of projection center
   real(kind=8)    :: projang  ! [ rad] Projection angle
   real(kind=4)    :: lamof    ! [ rad] Offset in Lambda
   real(kind=4)    :: betof    ! [ rad] Offset in Beta
end type position
```

Observation version 1 (obsolete for writing):

```
! POSITION: Position information.
integer(kind=4), parameter :: class_sec_pos_id=-3
integer(kind=4), parameter :: class_sec_pos_len=17
type position
   sequence
   integer(kind=4) :: sourc(3) ! [    ] Source name
   real(kind=4)    :: equinox  ! [year] Equinox of coordinates
   real(kind=8)    :: lam      ! [ rad] Lambda
   real(kind=8)    :: bet      ! [ rad] Beta
   real(kind=4)    :: lamof    ! [ rad] Offset in Lambda
   real(kind=4)    :: betof    ! [ rad] Offset in Beta
   integer(kind=4) :: proj     ! [code] Projection system
   real(kind=8)    :: sl0p     ! [ rad] lambda of descriptive system
   real(kind=8)    :: sb0p     ! [ rad] beta   of descriptive system
   real(kind=8)    :: sk0p     ! [ rad] angle  of descriptive system
end type position
```

4.2.3 Spectroscopic information (for spectra)

Observation version 1.3 (Doppler correction added, SkyFr and Vteles removed):

```
! SPECTRO: Spectroscopic information (for spectra).
integer(kind=4), parameter :: class_sec_spe_id=-4
integer(kind=4), parameter :: class_sec_spe_len=17
type spectro
   sequence
   integer(kind=4) :: line(3)  ! [    ] Line name
   real(kind=8)    :: restf    ! [ MHz] Rest frequency
   integer(kind=4) :: nchan    ! [    ] Number of channels
   real(kind=4)    :: rchan    ! [    ] Reference channel
   real(kind=4)    :: fres     ! [ MHz] Frequency resolution
   real(kind=4)    :: foff     ! [ MHz] Frequency offset
   real(kind=4)    :: vres     ! [km/s] Velocity resolution
```

```
   real(kind=4)    :: voff     ! [km/s] Velocity at reference channel
   real(kind=4)    :: bad      ! [    ] Blanking value
   real(kind=8)    :: image    ! [ MHz] Image frequency
   integer(kind=4) :: vtype    ! [code] Type of velocity
   real(kind=8)    :: doppler  ! [km/s] Doppler correction -V/c (CLASS convention)
end type spectro
```

Observation version 1.2 (SkyFr and Vteles added), obsolete for writing:

```
! SPECTRO: Spectroscopic information (for spectra).
integer(kind=4), parameter :: class_sec_spe_id=-4
integer(kind=4), parameter :: class_sec_spe_len=18
type spectro
   sequence
   integer(kind=4) :: line(3) ! [    ] Line name
   real(kind=8)    :: restf   ! [ MHz] Rest frequency
   integer(kind=4) :: nchan   ! [    ] Number of channels
   real(kind=4)    :: rchan   ! [    ] Reference channels
   real(kind=4)    :: fres    ! [ MHz] Frequency resolution
   real(kind=4)    :: foff    ! [ MHz] Frequency offset
   real(kind=4)    :: vres    ! [km/s] Velocity resolution
   real(kind=4)    :: voff    ! [km/s] Velocity at reference channel
   real(kind=4)    :: bad     ! [    ] Blanking value
   real(kind=8)    :: image   ! [ MHz] Image frequency
   integer(kind=4) :: vtype   ! [code] Type of velocity
   real(kind=8)    :: skyfr   ! [ MHz]
   real(kind=4)    :: vteles  ! [km/s]
end type spectro
```

Observation version 1.1, obsolete for writing:

```
! SPECTRO: Spectroscopic information (for spectra).
integer(kind=4), parameter :: class_sec_spe_id=-4
integer(kind=4), parameter :: class_sec_spe_len=15
type spectro
   sequence
   integer(kind=4) :: line(3) ! [    ] Line name
   real(kind=8)    :: restf   ! [ MHz] Rest frequency
   integer(kind=4) :: nchan   ! [    ] Number of channels
   real(kind=4)    :: rchan   ! [    ] Reference channels
   real(kind=4)    :: fres    ! [ MHz] Frequency resolution
   real(kind=4)    :: foff    ! [ MHz] Frequency offset
   real(kind=4)    :: vres    ! [km/s] Velocity resolution
   real(kind=4)    :: voff    ! [km/s] Velocity at reference channel
   real(kind=4)    :: bad     ! [    ] Blanking value
   real(kind=8)    :: image   ! [ MHz] Image frequency
   integer(kind=4) :: vtype   ! [code] Type of velocity
end type spectro
```

4.2.4 Baseline information (for spectra or drifts)

```
! BASE: Baseline information (for spectra of drifts).
integer(kind=4), parameter :: class_sec_bas_id=-5
integer(kind=4), parameter :: mwind=100
integer(kind=4), parameter :: class_sec_bas_len=8+2*mwind
type base
   sequence
   integer(kind=4) :: deg         ! [         ] Degree of last baseline
   real(kind=4)    :: sigfi       ! [Int. unit] Sigma
   real(kind=4)    :: aire        ! [Int. unit] Area under windows
   integer(kind=4) :: nwind       ! [         ] Number of line windows
   real(kind=4)    :: w1(mwind)   ! [     km/s] Lower limits of windows
   real(kind=4)    :: w2(mwind)   ! [     km/s] Upper limits of windows
   real(kind=4)    :: sinus(3)    ! [         ] Sinus baseline results
end type base
```

4.2.5 Scan numbers of initial observations

```
! HISTORY: Scan numbers of initial observations.
integer(kind=4), parameter :: class_sec_his_id=-6
integer(kind=4), parameter :: mseq=100
integer(kind=4), parameter :: class_sec_his_len=2*mseq+1
type history
   sequence
   integer(kind=4) :: nseq        ! Number of sequences
   integer(kind=4) :: start(mseq) ! Start can number of seq.
   integer(kind=4) :: end(mseq)   ! End scan number of seq.
end type history
```

4.2.6 Default plotting limits

```
! PLOT: Default plotting limits.
integer(kind=4), parameter :: class_sec_plo_id=-7
integer(kind=4), parameter :: class_sec_plo_len=4
type plot
   sequence
   real(kind=4)    :: amin    ! [Int. unit] Min Y value plotted
   real(kind=4)    :: amax    ! [Int. unit] Max Y value plotted
   real(kind=4)    :: vmin    ! [     km/s] Min X value plotted
   real(kind=4)    :: vmax    ! [     km/s] Max X value plotted
end type plot
```

4.2.7 Switching information (for spectra)

```
! SWITCH: Switching information (for spectra).
integer(kind=4), parameter :: class_sec_swi_id=-8
integer(kind=4), parameter :: mxphas=8
integer(kind=4), parameter :: class_sec_swi_len=2+6*mxphas
```

```
type class_switch_t
   sequence
   integer(kind=4) :: nphas           ! [    ] Number of phases
   real(kind=8)    :: decal(mxphas)   ! [ MHz] Frequency offsets
   real(kind=4)    :: duree(mxphas)   ! [   s] Time per phase
   real(kind=4)    :: poids(mxphas)   ! [    ] Weight of each phase
   integer(kind=4) :: swmod           ! [code] Switching mode (frequency, position...)
   real(kind=4)    :: ldecal(mxphas)  ! [ rad] Lambda offsets
   real(kind=4)    :: bdecal(mxphas)  ! [ rad] Beta offsets of each phase
end type class_switch_t
```

4.2.8 Calibration parameters

```
! CALIBRATION: Calibration parameters.
integer(kind=4), parameter :: class_sec_cal_id=-14
integer(kind=4), parameter :: class_sec_cal_len=25
type calib
   sequence
   real(kind=4)    :: beeff    ! [     ] Beam efficiency
   real(kind=4)    :: foeff    ! [     ] Forward efficiency
   real(kind=4)    :: gaini    ! [     ] Image/Signal gain ratio
   real(kind=4)    :: h2omm    ! [   mm] Water vapor content
   real(kind=4)    :: pamb     ! [  hPa] Ambient pressure
   real(kind=4)    :: tamb     ! [    K] Ambient temperature
   real(kind=4)    :: tatms    ! [    K] Atmosphere temp. in signal band
   real(kind=4)    :: tchop    ! [    K] Chopper temperature
   real(kind=4)    :: tcold    ! [    K] Cold load temperature
   real(kind=4)    :: taus     ! [neper] Opacity in signal band
   real(kind=4)    :: taui     ! [neper] Opacity in image band
   real(kind=4)    :: tatmi    ! [    K] Atmosphere temp. in image band
   real(kind=4)    :: trec     ! [    K] Receiver temperature
   integer(kind=4) :: cmode    ! [ code] Calibration mode
   real(kind=4)    :: atfac    ! [     ] Applied calibration factor
   real(kind=4)    :: alti     ! [    m] Site elevation
   real(kind=4)    :: count(3) ! [count] Power of Atm., Chopp., Cold
   real(kind=4)    :: lcalof   ! [  rad] Longitude offset for sky measurement
   real(kind=4)    :: bcalof   ! [  rad] Latitude offset for sky measurement
   real(kind=8)    :: geolong  ! [  rad] Geographic longitude of observatory
   real(kind=8)    :: geolat   ! [  rad] Geographic latitude of observatory
end type calib
```

4.2.9 For Skydips observations. No associated data.

```
! SKYDIP: For Skydips observations. No associated data.
integer(kind=4), parameter :: class_sec_sky_id=-16
integer(kind=4), parameter :: msky=10
integer(kind=4), parameter :: class_sec_sky_len=10+4*msky
type skydip
```

```
   sequence
   integer(kind=4) :: line(3)     ! [   ] Line name
   real(kind=8)    :: restf       ! [MHz] Rest frequency
   real(kind=8)    :: image       ! [MHz] Image frequency
   integer(kind=4) :: nsky        ! [   ] Number of points on sky
   integer(kind=4) :: nchop       ! [   ]   -   -    -    - chopper
   integer(kind=4) :: ncold       ! [   ]   -   -    -    - cold load
   real(kind=4)    :: elev(msky)  ! [rad] Elevations
   real(kind=4)    :: emiss(msky) ! [  ?] Power on sky
   real(kind=4)    :: chopp(msky) ! [  ?] Power on chopper
   real(kind=4)    :: cold(msky)  ! [  ?] Power on cold load
end type skydip
```

4.2.10 Gauss fit results (for spectra or drifts)

Observation version 1 (dynamic size) since 22-feb-2021:
The section size is now dynamic and can offer space for less or more than 5 lines, always with `mgauslin` $\geq$ `nline`. `mgauslin`, and thus the array sizes (`mgausfit`), is implicit and can be guessed from the section length (by reverting the `class_sec_gau_len` formula below). Using `mgauslin` = 5 ensures backward compatibility with older CLASS versions. Other values are read incorrectly with such versions.

Observation version 1 (static size):

```
! GAUSS: Gauss fit results (for spectra or drifts).
integer(kind=4), parameter :: class_sec_gau_id=-9
integer(kind=4), parameter :: mgauslin=5
integer(kind=4), parameter :: ngauspar=3  ! Number of parameters per line
integer(kind=4), parameter :: mgausfit=ngauspar*mxgaus
integer(kind=4), parameter :: class_sec_gau_len=3+2*mgausfit
type gauss
   sequence
   integer(kind=4) :: nline           ! [            ] Number of components
   real(kind=4)    :: sigba           ! [   Int. unit] Sigma on base
   real(kind=4)    :: sigra           ! [   Int. unit] Sigma on line
   real(kind=4)    :: nfit(mgausfit)  ! [area,v0,fwhm] Fit results
   real(kind=4)    :: nerr(mgausfit)  ! [area,v0,fwhm] Errors
end type gauss
```

4.2.11 "Stellar shell" profile fit results (for spectra)

Observation version 1 (dynamic size) since 22-feb-2021:
The section size is now dynamic and can offer space for less or more than 5 lines, always with `mshelllin` $\geq$ `nline`. `mshelllin`, and thus the array sizes (`mshellfit`), is implicit and can be guessed from the section length (by reverting the `class_sec_she_len` formula below). Using `mshelllin` = 5 ensures backward compatibility with older CLASS versions. Other values are read incorrectly with such versions.

Observation version 1 (static size):

```
! SHELL: "Stellar shell" profile fit results (for spectra).
integer(kind=4), parameter :: class_sec_she_id=-12
integer(kind=4), parameter :: mshelllin=5
integer(kind=4), parameter :: nshellpar=4  ! Number of parameters per line
integer(kind=4), parameter :: mshellfit=nshellpar*mshelllin
integer(kind=4), parameter :: class_sec_she_len=3+2*mshellfit
type shell
   sequence
   integer(kind=4) :: nline            ! [] Number of components
   real(kind=4)    :: sigba            ! [] Sigma on base
   real(kind=4)    :: sigra            ! [] Sigma on line
   real(kind=4)    :: nfit(mshellfit)  ! [] Fit results
   real(kind=4)    :: nerr(mshellfit)  ! [] Errors
end type shell
```

4.2.12 Hyperfine structure profile fit results (for spectra)

Observation version 1 (dynamic size) since 22-feb-2021:
The section size is now dynamic and can offer space for less or more than 3 lines, always with `mhfslin` $\geq$ `nline`. `mhfslin`, and thus the array sizes (`mhfsfit`), is implicit and can be guessed from the section length (by reverting the `class_sec_hfs_len` formula below). Using `mhfslin` = 3 ensures backward compatibility with older CLASS versions. Other values are read incorrectly with such versions.

Observation version 1 (static size):

```
! HFS: "Hyperfine Structure" profile fit results (e.g. NH3, HCN, for spectra).
integer(kind=4), parameter :: class_sec_hfs_id=-13
integer(kind=4), parameter :: mhfslin=3
integer(kind=4), parameter :: nhfspar=4  ! Number of parameters per line
integer(kind=4), parameter :: mhfsfit=nhfspar*mhfslin
integer(kind=4), parameter :: class_sec_hfs_len=3+2*mhfsfit
type hfs
   sequence
   integer(kind=4) :: nline          ! Number of components
   real(kind=4)    :: sigba          ! Sigma on base
   real(kind=4)    :: sigra          ! Sigma on line
   real(kind=4)    :: nfit(mhfsfit)  ! Fit results
   real(kind=4)    :: nerr(mhfsfit)  ! Errors
end type hfs
```

4.2.13 Hyperfine structure absorption profile fit results (for spectra)

Observation version 1 (dynamic size) since 22-feb-2021:
The section size is now dynamic and can offer space for less or more than 5 lines, always with `mabslin` $\geq$ `nline`. `mabslin`, and thus the array sizes (`mabsfit`), is implicit and can be guessed from the section length (by reverting the `class_sec_abs_len` formula below). Using `mabslin` = 5 ensures backward compatibility with older CLASS versions. Other values are read incorrectly

with such versions.

Observation version 1 (static size):

```
! ABS: "Hyperfine Structure" absorption profile fit results (e.g. NH3, HCN, for spectra).
integer(kind=4), parameter :: class_sec_abs_id=-18   ! Absorption fit
integer(kind=4), parameter :: mabslin=5
integer(kind=4), parameter :: nabspar=3  ! Number of parameters per line
integer(kind=4), parameter :: mabsfit=1+nabspar*mabslin  ! First is continuum
integer(kind=4), parameter :: class_sec_abs_len=3+2*mabsfit
type absorption
   sequence
   integer(kind=4) :: nline          ! Number of components
   real(kind=4)    :: sigba          ! Sigma on base
   real(kind=4)    :: sigra          ! Sigma on line
   real(kind=4)    :: nfit(mabsfit)  ! Fit results
   real(kind=4)    :: nerr(mabsfit)  ! Errors
end type absorption
```

4.2.14 Continuum drift description (for drifts)

```
! DRIFT: Continuum drift description (for drifts).
integer(kind=4), parameter :: class_sec_dri_id=-10
integer(kind=4), parameter :: class_sec_dri_len=16
type drift
   sequence
   real(kind=8)    :: freq    ! [ MHz] Rest frequency
   real(kind=4)    :: width   ! [ MHz] Bandwidth
   integer(kind=4) :: npoin   ! [    ] Number of data points
   real(kind=4)    :: rpoin   ! [    ] Reference point
   real(kind=4)    :: tref    ! [   ?] Time at reference
   real(kind=4)    :: aref    ! [ rad] Angular offset at ref.
   real(kind=4)    :: apos    ! [ rad] Position angle of drift
   real(kind=4)    :: tres    ! [   ?] Time resolution
   real(kind=4)    :: ares    ! [ rad] Angular resolution
   real(kind=4)    :: bad     ! [    ] Blanking value
   integer(kind=4) :: ctype   ! [code] Type of offsets
   real(kind=8)    :: cimag   ! [ MHz] Image frequency
   real(kind=4)    :: colla   ! [   ?] Collimation error Az
   real(kind=4)    :: colle   ! [   ?] Collimation error El
end type drift
```

4.2.15 Beam-switching parameters (for spectra or drifts)

```
! BEAM: Beam-switching parameters (for spectra or drifts).
integer(kind=4), parameter :: class_sec_bea_id=-11
integer(kind=4), parameter :: class_sec_bea_len=5
type beam
   sequence
```

```
    real(kind=4)    :: cazim   ! [ rad] Azimuth of observation
    real(kind=4)    :: celev   ! [ rad] Elevation of observation
    real(kind=4)    :: space   ! [ rad] Beam spacing
    real(kind=4)    :: bpos    ! [ rad] Position angle of beams
    integer(kind=4) :: btype   ! [code] System for angle
end type beam
```

4.2.16 Double gaussian and baseline fit results (for drifts)

```
! CONTINUUM: Double gaussian and baseline fit results (for drifts).
integer(kind=4), parameter :: class_sec_poi_id=-15
integer(kind=4), parameter :: mpoifit=8
integer(kind=4), parameter :: class_sec_poi_len=3+2*mpoifit
type pointing
   sequence
   integer(kind=4) :: nline          ! Number of components
   real(kind=4)    :: sigba          ! Sigma on base
   real(kind=4)    :: sigra          ! Sigma on line
   real(kind=4)    :: nfit(mpoifit)  ! Fit results
   real(kind=4)    :: nerr(mpoifit)  ! Errors
end type pointing
```

4.2.17 Herschel Space Observatory (HIFI)

```
integer(kind=4), parameter :: class_sec_her_id=-20
integer(kind=4), parameter :: class_sec_her_len=96
type class_herschel_t
  integer(kind=8)   :: obsid       !  1- 2  [----] Observation id
  character(len=8)  :: instrument  !  3- 4  [str ] Instrument name
  character(len=24) :: proposal    !  5-10  [str ] Proposal name
  character(len=68) :: aor         ! 11-27  [str ] Astronomical Observation Request
  integer(kind=4)   :: operday     !    28  [day ] Operational day number
  character(len=28) :: dateobs     ! 29-35  [str ] Start date (ISO date)
  character(len=28) :: dateend     ! 36-42  [str ] End date (ISO date)
  character(len=40) :: obsmode     ! 43-52  [str ] Observing mode
  real(kind=4)      :: vinfo       !    53  [km/s] Informative source velocity in LSR frame
  real(kind=4)      :: zinfo       !    54  [----] Informative target redshift
  real(kind=8)      :: posangle    ! 55-56  [rad ] Spacecraft pointing position angle
  real(kind=8)      :: reflam      ! 57-58  [rad ] Sky reference (a.k.a. OFF), lambda coordi
  real(kind=8)      :: refbet      ! 59-60  [rad ] Sky reference (a.k.a. OFF), beta coordinat
  real(kind=8)      :: hifavelam   ! 61-62  [rad ] ON average H and V coordinate (HIFI), lamb
  real(kind=8)      :: hifavebet   ! 63-64  [rad ] ON average H and V coordinate (HIFI), beta
  real(kind=4)      :: etamb       !    65  [----] Main beam efficiency used when applying do
  real(kind=4)      :: etal        !    66  [----] Forward efficiency used when applying doAn
  real(kind=4)      :: etaa        !    67  [----] Telescope aperture efficiency
  real(kind=4)      :: hpbw        !    68  [rad ] Azimuthally-averaged half-power beam widtl
  character(len=8)  :: tempscal    ! 69-70  [str ] Temperature scale in use
  real(kind=8)      :: lodopave    ! 71-72  [MHz ] Average LO frequency Doppler corrected to
```

```
  real(kind=4)      :: gim0        !    73  [----] Sideband gain polynomial coeff 0 applied
  real(kind=4)      :: gim1        !    74  [----] Sideband gain polynomial coeff 1 applied
  real(kind=4)      :: gim2        !    75  [----] Sideband gain polynomial coeff 2 applied
  real(kind=4)      :: gim3        !    76  [----] Sideband gain polynomial coeff 3 applied
  real(kind=4)      :: mixercurh   !    77  [mA  ] Calibrated mixer junction current, horizon
  real(kind=4)      :: mixercurv   !    78  [mA  ] Calibrated mixer junction current, vertica
  character(len=28) :: datehcss    ! 79-85  [str ] Processing date (ISO date)
  character(len=24) :: hcssver     ! 86-91  [str ] HCSS version
  character(len=16) :: calver      ! 92-95  [str ] Calibration version
  real(kind=4)      :: level       !    96  [----] Pipeline level
end type class_herschel_t
```

4.2.18 Comment Section

```
integer(kind=4), parameter :: class_sec_com_id=-1
integer(kind=4), parameter :: class_sec_comm_len=256
type comment
   sequence
   integer(kind=4) :: ltext   ! Length of comment
   character ctext*1024       ! Comment string
end type comment
```

4.2.19 Data Section

The data section contains the spectrum intensity of each spectral channel. It thus consists of NDATA real values.

4.3 Old OTF data format

This section is kept for backward compatibility. Indeed CLASS is still able to read old OTF data format. Once CLASS, always CLASS...

4.3.1 Data Section Descriptor

```
! Data Section Descriptor.
! (Old OTF data format, kept for backward compatibility).
integer(kind=4), parameter :: class_sec_desc_id=-30
integer(kind=4), parameter :: class_sec_desc_len=4
type descriptor
   sequence
   integer(kind=4) :: ndump   ! Number of records
   integer(kind=4) :: ldpar   ! Length of data header (longwords)
   integer(kind=4) :: ldatl   ! length of line data      (")
   integer(kind=4) :: ldump   ! length of record         (")
   integer(kind=4) :: rec     ! Current Record Number
end type descriptor
```

4.3.2 Multiple spectra Data Section

A multiple spectra data section contains many spectra, obtained in a single observing scan: typically a drift scan across a source, a on-off scan with a multi-beam receiver, or a drift scan with a multi-beam receiver. The data section is considered as a multiple spectra data section is the special section 'Data Section Descriptor' is present. It consists of a `R_NDUMP` records (individual spectra); each spectrum has `R_LDATL` line channels, and is followed by a few data associated parameters (`R_LDPAR` words). The length of each record is thus `R_LDUMP=R_LDPAR+R_LDATL`; the total length of the data section should be: `NDATA=R_NDUMP*R_LDUMP`.

The data header presently contains:

offset	type	name	
0	Integer*4		Record number
1	Real*4	AZ	[rad] Telescope Azimuth
2	Real*4	EL	[rad] Telescope Elevation
3	Real*8	ST	[rad] Sideral Time
5	Real*4	LAMOF	[rad] Azimuth offset
6	Real*4	BETOF	[rad] Elevation offset
12	Real*4	UT	[sec] Universal Time in day DOBS

4.4 CLASS FITS format

Starting with dec06 release of GILDAS, the FITS support of CLASS is largely improved. In addition to the old possibility of writing one spectrum per file (see the "Simple SPECTRUM Mode" section), it is now possible to read from the same FITS file, one spectrum in the primary Header/Data Unit and collections of spectra in bintables stored as FITS extensions (see the "BINTABLE Mode"). Moreover, it is now possible to store in the same bintable, spectra of different number of channels. The distinction between the SPECTRUM and BINTABLE mode is mainly historical as the software automatically toggles from one mode to another.

4.4.1 Simple SPECTRUM Mode

FITS headers written by CLASS 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 CLASS looks like this :

```
SIMPLE  =                    T         /                                      
BITPIX  =                   32         /                                      
NAXIS   =                    4         /                                   (1)
NAXIS1  =                  921         /                                   (2)
NAXIS2  =                    1         /                                   (3)
NAXIS3  =                    1         /                                   (3)
NAXIS4  =                    1         /                                   (3)
BLANK   =           2147483647         / Blanking value                       
BSCALE  =  0.237091768440E-008         /                                      
BZERO   =  0.192957639694E+001         /                                      
DATAMIN = -0.316193056107E+001         /                                      
DATAMAX =  0.702108430862E+001         /                                      
```

```
BUNIT   = 'K (Ta*)'                / (4)
CTYPE1  = 'FREQ    '               / (5)
CRVAL1  =  0.000000000000E+000     / Offset frequency
CDELT1  =  0.390625000000E+005     / Frequency resolution
CRPIX1  =  0.578812988281E+003     /
CTYPE2  = 'RA---GLS'               / (6)
CRVAL2  =  0.315384333333E+003     /
CDELT2  =  0.835555750817E-002     /
CRPIX2  =  0.000000000000E+000     /
CTYPE3  = 'DEC--GLS'               / (6)
CRVAL3  =  0.681748750000E+002     /
CDELT3  =  0.505729609368E-002     /
CRPIX3  =  0.000000000000E+000     /
CTYPE4  = 'STOKES  '               / (7)
CRVAL4  =      1.0000000000000     /
CDELT4  =      1.0000000000000     /
CRPIX4  =      0.0000000000000     /
TELESCOP= '30M'                    /
INSTRUME= '30MEOHUI-V01'           /
OBJECT  = 'NGC7023'                /
RADESYS = 'FK5     '               /
EQUINOX =  0.200000000000E+004     / (8)
LINE    = '13co10'                 / Line name                    (9)
RESTFREQ=  0.110201354300E+012     / Rest frequency               (10)
VELO-LSR=  0.250000000000E+004     / [m/s] Velocity of referance ch (11)
SPECSYS = 'LSRK'                   / Reference frame
DELTAV  = -0.106265872717E+003     / Velocity spacing of channels  (12)
IMAGFREQ=  0.972814604954E+011     / Image frequency               (13)
TSYS    =  0.158461090088E+003     / System temperature            (14)
OBSTIME =  0.492044001818E+000     / Integration time              (15)
SCAN-NUM=  0.100000000000E+004     / Scan number                   (16)
TAU-ATM =  0.189633205533E+000     / Atmospheric opacity           (17)
NPHASE  =                    1     / Number of frequency phases    (18)
DELTAF1 =  0.000000000000E+000     / Frequency offset Phase 1      (19)
PTIME1  =  0.493851512671E+000     / Duration of Phase 1           (19)
WEIGHT1 =  0.000000000000E+000     / Weight of Phase 1             (19)
GAINIMAG=  0.501190014184E-001     / Image sideband gain ratio     (20)
BEAMEFF =  0.949999988079E+000     / Beam efficiency               (21)
FORWEFF =  0.949999988079E+000     / Image sideband gain ratio     (22)
ORIGIN  = 'CLASS-Grenoble  '       /
DATE    = '2023-11-07T00:00:00.000'  / Date written
TIMESYS = 'UTC             '       /
DATE-OBS= '2009-09-07T18:47:46.352'  / Date observed
DATE-RED= '2009-09-15T00:00:00.000'  / Date reduced
ELEVATIO=  0.486971109330E+002     /  Telescope elevation
AZIMUTH =  0.385419378637E+003     /  Telescope azimuth
UT      = ' 18:47:46.352'          /  Universal time at start
```

```
LST     = ' 17:42:05.449'              /  Sideral time at start
END
```

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 ) * 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
   `Dec = CRVAL3 + ( 1 - CRPIX3 ) * CDELT3`
   `Ra = CRVAL2 + ( 1 - CRPIX2 ) * CDELT2 / COS(Dec)`
   That is, `CDELT2` and `CDELT3` represents angular offsets from the reference position `(CRVAL2,CRVAL3)` in a Global Sinusoidal projection (`RADIO` projection).

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

8. Equinox of these coordinates.

9. Molecular line name, for bookkeeping.

10. Rest frequency.

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

12. 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 ) * DELTAV`

13. Image frequency, for double sideband operation.

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

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

16. Scan number, for bookkeeping.

17. Atmospheric opacity in the signal sideband.

18. For multi-phased spectra (i.e. frequency switching) number of phases.
19. For each phase, the frequency offset, the phase length and weight.
20. The ratio of gains in the image and signal sidebands (in case of double sideband operation).
21. The telescope beam efficiency.
22. The telescope forward efficiency.

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

4.4.2 BINTABLE Mode

In addition to the simple "one FITS file per Spectrum" mode, CLASS supports the Binary Table extension, in reading and writing. CLASS reads and processes all the Binary Table extension of the FITS file. The FITS keywords for the Binary Table format are similar to those of the simple Spectrum mode, but a complete set of spectra can be handled in a single binary table.

Keywords which would change from one spectrum to the next are handled as Binary columns of the table, with the `TTYPEi` FITS keyword holding the name of the corresponding simple FITS keyword. Reading such FITS files, CLASS will support any value for the `TTYPEi` keywords. If a column and a keyword have the same name, the column prevails. If several columns have the same name (try to avoid this), the last one prevails.

The spectra are handled in a `SPECTRUM` column.

Here is an example with a tiny primary Header Data Unit and a Binary Table extension header.

```
SIMPLE  =                    T
BITPIX  =                    8
NAXIS   =                    0
EXTEND  =                    T
BLOCKED =                    T
ORIGIN  = 'My observatory'
CREATOR = 'Me      '
END

...

XTENSION= 'BINTABLE'
BITPIX  =                    8         / Always 8.
NAXIS   =                    2         / Always 2: tables have 2 dimensions.
NAXIS1  =                 7275         / Number of bytes per row.
NAXIS2  =                    4         / Number of rows.
PCOUNT  =                    0         / Usually 0.
GCOUNT  =                    1         / Always 1.
TFIELDS =                   18         / Number of columns.
EXTNAME = 'MATRIX  '                   / Just a name, not important.
EXTVER  =                    1         / Always 1.
MAXIS   =                    4         / Number of dimensions in the data.
```

```
MAXIS1  =                 1793         / Dummy number of channels (see TTYPE1).
MAXIS2  =                    1         /
MAXIS3  =                    1         /
MAXIS4  =                    1         /
CTYPE1  = 'FREQ    '                  / Dim1: freq => MAXIS1 = Nb channels.
CRVAL1  =  0.0000000000000E+00         / Frequency offset, always 0.
CDELT1  =  0.1000000000000E+07         / Frequency resolution [Hz].
CRPIX1  =  0.2565000000000E+03         / Dummy reference channel (see TTYPE18).
CTYPE2  = 'RA      '
CRVAL2  =  0.0000000000000E+00
CDELT2  =  0.0000000000000E+00
CRPIX2  =  0.0000000000000E+00
CTYPE3  = 'DEC     '
CRVAL3  =  0.0000000000000E+00
CDELT3  =  0.0000000000000E+00
CRPIX3  =  0.0000000000000E+00
CTYPE4  = 'STOKES  '
CRVAL4  =  0.0000000000000E+00
CDELT4  =  0.0000000000000E+00
CRPIX4  =  0.0000000000000E+00
SUBSCAN =                    1         / Subscan number.  Often 1.
LINE    = 'CS(5-4)     '               / Name of your line, up to 12 chars.
OBJECT  = 'JUPITER     '               / Name of your source, up to 12 chars.
RESTFREQ=  0.2449356430000E+12         / Rest (signal) frequency at ref chan.
VELO-HEL=  0.0000000000000E+00         / Velocity at ref.  chan [m.s-1].
VELDEF  = 'RADI-HEL'                   / Type of velocity.
GAINIMAG=  0.9999999776483E-02         / Ratio Image/Signal.
BEAMEFF =  0.8600000143051E+00         / Beam efficiency.
FORWEFF =  0.8600000143051E+00         / Forward efficiency.
EQUINOX =  0.1950000000000E+04         / Epoch of coordinates.
DATE-RED= '15/07/97'                   / Date of reduction.
TTYPE1  = 'MAXIS1  '                   / Number of channels for this spectrum.
TFORM1  = '1J      '                   / [integer]
TTYPE2  = 'SCAN    '                   / Scan number.
TFORM2  = '1J      '                   / [integer]
TTYPE3  = 'TELESCOP'                   / Name of your backend.
TFORM3  = '12A     '                   / [Up to 12 characters]
TTYPE4  = 'TSYS    '                   / System temperature.
TFORM4  = '1E      '                   / [K]
TTYPE5  = 'IMAGFREQ'                   / Image freq at ref channel.
TFORM5  = '1E      '                   / [Hz]
TTYPE5  = 'DELTAV  '                   / Velocity resolution.
TFORM6  = '1E      '                   / [m.s-1]
TTYPE7  = 'TAU-ATM '                   / Atmospheric opacity.
TFORM7  = '1E      '                   / [neper]
TTYPE8  = 'MH2O    '                   / Water vapor content.
TFORM8  = '1E      '                   / [mm]
```

```
TTYPE9  = 'TOUTSIDE'                / Atm temperature near the antenna.
TFORM9  = '1E      '                / [K]
TTYPE10 = 'PRESSURE'                / Atm pressure on the antenna.
TFORM10 = '1E      '                / [hPa]
TTYPE11 = 'ELEVATIO'                / Elevation.
TFORM11 = '1E      '                / [deg]
TTYPE12 = 'AZIMUTH '                / Azimuth.
TFORM12 = '1E      '                / [deg]
TTYPE13 = 'DATE-OBS'                / Observing date.
TFORM13 = '23A     '                / 'dd/mm/yy' for ex.  See below.
TTYPE14 = 'UT      '                / Universal time at start.
TFORM14 = '1D      '                / [s]
TTYPE15 = 'LST     '                / Sideral time at start.
TFORM15 = '1D      '                / [s]
TTYPE16 = 'OBSTIME '                / Integration duration.
TFORM16 = '1E      '                / [s]
TTYPE17 = 'SPECTRUM'                / Your data.
TFORM17 = '1793E   '                / [K]
TTYPE18 = 'CRPIX1  '                / Reference channel for this spectrum.
TFORM18 = '1E      '                / [real]
END
```

Known keywords

`HISTORY` 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.

`MAXIS` Number of axes.

`MAXISi` Number of elements for axis `i`. `i` goes from 1 to `MAXIS`. The first axis is usually used to define effectively the spectrum. Thus `MAXIS1` corresponds to the number of channels. The second and third one can be used to define the coordinates of the spectra when storing a full spectra cube.

`CTYPEi` The type of physical coordinate on axis i: frequency, velocity, right ascension, time... The unit of the values of `CRVALi` and `CRPIXi` are conventionnally defined by the value of the `CTYPEi` keyword. The authorized values and associated units are

`FREQ or FREQUENCY` Frequency axis. Unit: [Hz]. Frequency for channel number n is defined by: `F(n) = RESTFREQ + CRVALi + ( n - CRPIXi ) * CDELTi`

`LAMBDA or WAVELENG` wavelength axis. Unit: [m^{-1}].

`VELO or FELO` Velocity axis. Unit: [$m.s^{-1}$]. `VELO` corresponds to the radio convention while `FELO` implies channels regularly gridded in frequency but expressed as velocity in the optical convention. CLASS will process `VELO` and `FELO` in the same way *** JP. The referential is defined by the second part of the keyword:

- `-LSR` or `CITY` or empty second part: Local Standart Rest velocity.
- `-HEL` Heliocentric (barycentric) velocity.

- `-OBS` or `-TOP` The frame of rest of the observer/telescope (topocentric).
- `-EAR` or `-GEO` Geocentric velocity

The full keyword may thus looks like: `FELO`, `VELOCITY`, `VELO-GEO`...

`RA-` or `RA` Right ascension axis. Unit: [degrees]. Right Ascension `RA` 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

```
Dec = CRVAL3 + ( 1 - CRPIX3 ) * CDELT3
Ra = CRVAL2 + ( 1 - CRPIX2 ) * CDELT2 / COS(Dec)
```

`DEC-` or `DEC` Declination axis. Unit: [degrees]. Same caveats as for `RA`.

`GLAT` Galactic latitude axis. Unit: [degrees]. Same caveat as for `RA`.

`GLON` Galactic longitude axis. Unit: [degrees]. Same caveat as for `RA`.

`TIME` or `UT` or `UTC` Time axis. Unit: [s]. The time since `DATE-OBS`.

`STOKES` Stokes axis. The Stokes parameter of the data as defined in the basic paper of Wells et al.

`CRPIXi` The array location of the reference pixel along axis i. `CRPIXi`'s value may be a fractional number of pixels and/or outside of the limits of the array. This descriptor is optional for degenerated axes (*i.e.* 0 or 1 element).

`CRVALi` The value of the physical coordinate on axis i at the reference pixel.

`CDELTi` The increment in physical coordinates along axis i. This descriptor is optional for degenerated axes. Warning: Although there is a difference between the frequency spacing between channels and the frequency resolution of each channel, CLASS currently handles only one value for both.

`BSCALE` A value of 1 is assumed when it is undefined. The following formula will be applied to the FITS data: `CLASS_spectrum = BZERO + BSCALE * FITS_spectrum`

`BZERO` A value of 0 is assumed when it is undefined. The following formula will be applied to the FITS data: `CLASS_spectrum = BZERO + BSCALE * FITS_spectrum`

`DATAMIN` Minimum value of your spectrum. Understood by CLASS but unused.

`DATAMAX` Maximum value of your spectrum. Understood by CLASS but unused.

`BLANK` Blanking value. If not specified, -1e38 assumed. `BSCALE` and `BZERO` are applied to the blanking value.

`PCOUNT` Usually 0. But if you provide your data as Variable Length arrays, then you will have to tweak this value which specifies the size of the variable length array heap, plus a gap (if any). See `THEAP`.

`GCOUNT` Always 1 in Binary tables.

`THEAP` Byte offset of heap area. The heap area is where the Variable Length Array data are stored. They are usually stored just after the binary table. The heap does not need to begin at the end of the main table data. The user may choose to provide a gap between the main table and the beginning of the heap. The size of this gap, in bytes, can be found using the value of the `THEAP` keyword. If there is no gap, the value of `THEAP` is `NAXIS1 * NAXIS2`. If there is, the size of the gap is the difference between the value of `THEAP` and `NAXIS1 * NAXIS2`. A value of `NAXIS1 * NAXIS2` is assumed when this keyword is not defined.

`BEAMEFF` The main-beam efficiency.

`FORWEFF or ETAFSS` The forward spillover and scattering efficiency.

`GAINIMAG` Ratio Image/Signal.

`MH2O` Water vapor content. Unit: [mm].

`PRESSURE` The ambiant atmospheric pressure. Unit: [hPa].

`TOUTSIDE` The ambient temperature. Unit [K].

`NPHASE` Value: from 1 to 8. For multi-phased spectra (i.e. frequency switching) number of phases.

`DELTAFj` Frequency offset for phase `j`. Unit: [Hz].

`PTIMEj` Duration of phase `j`. Unit: [s].

`WEIGHTj` Weight of phase `j`.

`TELESCOP` A string value giving the telescope name. CLASS truncates this string to the first 12 characters. There is a CLASS convention about `TELESCOP`, which is not mandatory: `TELESCOP` contains the name of the instrument, the name of the backend and the name of the subband (if any). For example: "HIFI-HRV-02" for subband 2 of the High Resolution Spectrometer (Vertical polarization) of HIFI (one of the Herschel's instrument).

`AZIMUTH` The azimuth at `TIME`. Unit: [degrees]. If the `TIME` axis is non-degenerated, then this is the azimuth at the `TIME` of the first pixel on the `TIME` axis.

`ELEVATIO` The elevation at `TIME`. Unit: [degrees]. Same caveat as for `AZIMUTH`.

`DATE-OBS` or `DATE_OBS` A string giving the observation date and optionally the time at the observation start using the new FITS y2k convention. Since CLASS does not handle the `TIMESYS` keyword yet to indicate the time system, UTC is assumed. Accepted formats are

- `'dd/mm/yy'` (1900 is added to yy)
- `'yyyy/mm/dd'`
- `'yyyy/mm/ddThh:mm:ss.s'`, where 'T' stands for the letter 'T' and has no meaning except as a separator between date and time. The trailing time is ignored, you have to provide it through `UT`, `UTC` or `LST`.

`DATE-RED` or `DATE_RED` or `DATE` Same as `DATE-OBS`, except it is the date at which the data have been reduced.

`SCAN or SCAN-NUM` or `SCAN_NUM` Scan number of the observation, according to the CLASS terminology.

`SUBSCAN` Subscan number of the observation, according to the CLASS terminology.

`UT or UTC` Universal time of the start of the observation provided as a string or a number. Then the format is

- 'hh:mm:ss.s', when this keyword contains a string.
- The number of seconds after midnight, when this keyword contains a number.

`LST` Sideral time of the start of the observation. See `UT` or `UTC` for information about the format.

`OBSTIME or EXPOSURE` The effective integration time. Unit: [s]. Necessary for some kind of weighting when adding several spectra.

`TSYS` The system temperature. Unit: [K]. Necessary for some kind of weighting when adding several spectra.

`TAU-ATM` or `TAU_ATM` The opacity at `RESTFREQ`. CLASS does not handle the opacity in the image band yet.

`EQUINOX` The equinox of coordinates. Possible values are 1950 or 2000.

`OBJECT` A string value giving an object name. CLASS truncates this string to the first 12 characters.

`LINE` Molecular line name, for bookkeeping. CLASS truncates this string to the first 12 characters.

`RESTFREQ` The observed frequency at the reference pixel of the frequency-like axis. Unit: [Hz].

`IMAGFREQ` The image sideband frequency corresponding to `RESTFREQ` for double sideband operation. Unit: [Hz].

`VELOCITY` The velocity at the reference channel. The referential frame must be defined by `VELDEF`.

`VELO-LSR or VLSR` The velocity at the reference channel, understood as a velocity in the Local Standard Rest frame. Thus, you don't need to use `VELDEF` to specify the frame. Unit: [$m.s^{-1}$].

`VELO-OBS or VELO-TOP` The velocity at the reference channel, understood as a velocity in the frame of rest of the observer/telescope. Thus, you don't need to use `VELDEF` to specify the frame. Unit: [$m.s^{-1}$].

`VELO-HEL` The velocity at the reference channel, understood as a velocity in the Heliocentric (barycentric) frame. Thus, you don't need to use `VELDEF` to specify the frame. Unit: [$m.s^{-1}$].

`VELO-EAR or VELO-GEO` The velocity at the reference channel, understood as a velocity in the Geocentric frame. Thus, you don't need to use `VELDEF` to specify the frame. Unit: [$m.s^{-1}$].

`DELTAV or DELTAVEL` 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 n is thus given by `V(n) = VLSR + (n - CRPIXi) * DELTAV`. If `DELTAV` is not provided, then the velocity resolution will be calculated by `DELTAV = -c / RESTFREQ * CDELT1`, where `CDELT1` is the frequency spacing and `c` the celerity of light.

`VELDEF` The velocity definition and frame (8 characters). The first 4 characters describe the velocity definition. Accepted definitions are

- `RADI` Radio convention.
- `OPTI` Optical convention.
- `RELA` Relativistic convention.

CLASS only handles `RADI` and does not handle `OPTI` or `RELA`. If nothing specified, `RADI` is assumed. This is the list of the values CLASS understands

- `LSR` or `RADI-LSR` Local Standart Rest frame.
- `HEL` or `RADI-HEL` Heliocentric (barycentric) frame.
- `OBS` or `RADI-OBS` Frame of rest of the observer/telescope. Note that `TOP` is not handled here.
- `EAR` or `RADI-EAR` Geocentric frame.

Known columns and data storage

Most of the keywords described above can be used as column names. The value contained in the column prevails on the value contained in the Keyword. This can be useful if you need to store spectra with different lengths in your table: You can use a column named `MAXIS1` containing the number of channels for each spectrum of the table (the trailing bytes requiered to fill the `DATA` column will be ignored then).

Here is the list of authorized column names which can also be keywords: `DATAMAX`, `DATAMIN`, `MAXISi`, `CRVALi`, `CRPIXi`, `CDELTi`, `BEAMEFF`, `FORWEFF` or `ETAFSS`, `GAINIMAG`, `MH2O`, `PRESSURE`, `TOUTSIDE`, `AZIMUTH`, `ELEVATIO`, `DATE-OBS` or `DATE_OBS`, `DATE-RED` or `DATE_RED` or `DATE`, `SCAN` or `SCAN-NUM` or `SCAN_NUM`, `SUBSCAN`, `LST`, `UT` or `UTC`, `TAU-ATM` or `TAU_ATM`, `TELESCOP`, `OBSTIME` or `EXPOSURE`, `TSYS`, `EPOCH` or `EQUINOX`, `OBJECT`, `IMAGFREQ`, `LINE`, `RESTFREQ`, `VELOCITY`, `VELDEF`, `VELO-LSR` or `VLSR`, `VELO-OBS` or `VELO-TOP`, `VELO-HEL`, `VELO-EAR` or `VELO-GEO`, `DELTAV` or `DELTAVEL`.

The following names are exclusively used for columns because their value (your data) can not be factorized

`MATRIX or SPECTRUM or SERIES or DATA` The spectra intensities. Unit: [K]. This is the only mandatory column. Indeed, why would we make a FITS file without any spectrum? This column must also be unique, *i.e.* only one data column per binary table. The number of elements of this column, specified in `TFORM`, depends on the way you want to store your data (Fixed or Variable Length Arrays). More explanations are available below

`WAVE` Frequency of each channel. Unit: [Hz]. This column is optional. If it is present, the spectra is considered as "irregularly sampled": each channel has its own frequency and the frequency axis description; In other words, the `RESTFREQ`, `CDELTi`, `CRVALi`) keyword are not used at all even when present. `MAXISi` is still used, since it gives the number of channels. If this column is absent, then the frequency axis will be described in the usual way.

Both the SPECTRUM and WAVE columns must have the same number of elements to obtain a meaningful result. Those columns may either be filled with

Fixed Length Arrays This may be the easiest way to store the data but it may waste a lot of space if you have spectra with different number of channels. The steps are:

1. Description of the data column.
   (a) First, chose a precision: E for single precision floating point number (4 bytes), D for double precision floating point number (8 bytes).
   (b) Second, browse your collection of spectra to find the one with the highest number of channels (for example, 1793).
   (c) Now you can give TFORMi the value 1793E:
   ```
   TTYPE17 = 'SPECTRUM'
   TFORM17 = '1793E   '
   ```
2. Description of the number-of-channel column. If your collection of spectra contains spectra with different sizes, then you must specify the number of channels of each spectrum. To do that, you have to use a column MAXISi. i corresponds to the number of your frequency/wavelength/velocity axis: usually 1.
   ```
   TTYPE1  = 'MAXIS1  '
   TFORM1  = '1J      '
   ```
   The values stored in this column will override the value given to the keyword MAXIS1 at the beginning of the bintable header. If all your spectra have the same number of channels, then you don't need a MAXISi column. But make sure the MAXISi keyword at the beginning of the bintable header has the correct value, which is the number of channels of your spectra.
3. Description of the reference channel. You may like to chose the middle of your spectra to be the reference for frequencies. If you have spectra with different sizes, then the reference channels may be different for each spectrum. To override the value of the CRPIX1 keyword, you must use a CRPIX1 column.
   ```
   TTYPE18 = 'CRPIX1  '
   TFORM18 = '1E      '
   ```
   If your reference channel is always the first one, you can forget about this column and give the value 1 to the CRPIX1 keyword.
4. Filling the olumns. When it comes to fill the column with the data, make sure you are always storing 1793 (for example) elements. If you are to store a spectrum with only 512 channels, then store those 512 elements and pad with 1281 zeroes (single or double precision zeroes, not bytes). You can see here the waste of space when you have a significant amount of spectra much smaller that the widest one.

Variable Length Arrays To save space and avoid filling half of your data columns with padding zeroes, you can use the Variable Length Array format described in the FITS standard (Definition of the Flexible Image Transport System, version 2.1b, section 8.3.5).

The idea is not to store the data directly in the bintable records, but in the heap area located after the bintable. In the SPECTRUM or WAVE column, you will find an Array Descriptor which basically consists in a pointer to a location in this heap area (an offset), and a number of

elements to read. By doing this, no padding is required any longer. The FITS standart defines two kinds of Array Descriptors: P (two 32bits unsigned integers values) and Q (two 64 bits unsigned integers values). CLASS only handles P.

You will not need a `MAXIS1` column in this case, since the number of channels is written in the `SPECTUM` column. You may still need a `CRPIX1` column to define your reference channels.

4.4.3 Once FITS Always FITS

This is the basic principle of FITS, but the IAU FITS committee keeps making revisions of the FITS format. So things are expected to change, with new versions of CLASS providing new FITS output, but still reading old FITS.

Chapter 5

Internal Helps

5.1 LAS Language Internal Help

5.1.1 Language

```
            LAS\ Command Language Summary

ACCUMULATE          Add R and T observation.
ASSOCIATE           Add an Associated Array to the R observation
AVERAGE             Average all the observations of the current index.
BASE [arg]          Subtract a baseline.
BOX                 Draw a frame for data.
CATALOG             Load a line catalog
CONSISTENCY         Check the consistency of the current index
COPY                Write the current index into output file.
DROP num [ver]      Take a scan out of the current index.
DUMP                List some informations on the R spectrum.
EXTRACT X1 X2 [U]   Extract a subsection of the R spectrum.
FILE type name      Define the input/output files.
FIND                Search the input file for observations.
FITS                Write or read a fits file
FOLD                Fold a Frequency Switched spectrum.
GET [N]             Read a scan in the input file.
HEADER              Display some header information on the R spectrum.
IGNORE List         Ignore scans from the Input file.
LIST [in|out]       List header information about an ensemble of scans.
LOAD                Build a 2D set of spectra from INDEX
MODIFY              Edit and change the scan header.
MULTIPLY fact       Multiply the R spectrum by fact.
NEW_DATA            Wait until new data present in input file.
PLOT                Plot the observation in R, with box and title.
SAVE [name]         Save the current parameters.
SET                 Enter a value for a parameter.
SHOW Arg            Display some parameter.
SPECTRUM            Plot the R observation or an associated array.
SWAP                Exchange the contents of the R and T buffers.
```

```
TAG Qual List      Change the quality of scans in the Output file.
TITLE              Write a header above the plotted frame.
UPDATE             Update R in the output file.
WRITE [Obs]        Write R in the output file.
```

5.1.2 ACCUMULATE

LAS\ACCUMULATE [/RESAMPLE [NX Xref Xval Xinc Unit]] [/NOCHECK Arg1 ... ArgN]] [/WEIGHT TIME|SIGMA|EQUAL|ASSOCIATED]

ACCUMULATE is used to add a pair of spectra together. The T spectrum is added to the R spectrum using the specified (/WEIGHT) or current (SET WEIGHT) weighting function (see HELP SET WEIGHT for details). To stack more than 2 spectra, you should build an index and AVERAGE it in a single step instead (this will prevent cumulative resamplings, respect the respective weights, and preserve at best the noise properties).

On return R contains the sum spectrum, and T the observation previously in R. If /WEIGHT or SET WEIGHT is ASSOCIATED (see HELP SET WEIGHT for details), the weight array of the sum spectrum is saved in the associated array named W.

The spectra are stacked channel by channel by default, if possible. Else the option /RESAMPLE must be invoked (see subtopic for details).

For EQUAL weights, ACCUMULATE computes the sum of the two spectra, allowing addition of an ensemble of spectra after division by the total number of spectra. For other weightings, ACCUMULATE computes the average of the two spectra.

ACCUMULATE also works on Continuum drift. The alignement may be Channel or Position in this case.

ACCUMULATE /NOCHECK

LAS\ACCUMULATE /NOCHECK [SOURCE|POSITION|LINE|SPECTROSCOPY|CALIBRATION|SWITCHING]

By default, ACCUMULATE checks the consistency between R and T. This is a security to avoid mixing inconsistent data (e.g. spectra from different sources). These checks can be individually (with one or more names) or all (no name) disabled using the option /NOCHECK. See HELP SET NOCHECK for details.

Note the key status of the SPECTROSCOPY check! ACCUMULATE uses this check to detect if the spectra X axes are aligned or not, to help its decision to enable resampling or not (see /RESAMPLE subtopic). Using /NOCHECK SPECTROSCOPY enforces the spectra alignment status:

ACCUMULATE /NOCHECK SPECTROSCOPY
will assume R and T are aligned and will stack them channel by channel (no resampling), while they are possibly not aligned.

ACCUMULATE /RESAMPLE /NOCHECK SPECTROSCOPY
will assume R and T are not aligned and will always resample them before stacking, even if not necessary (adds computation cost and possible round off errors).

ACCUMULATE /RESAMPLE

LAS\ACCUMULATE /RESAMPLE [NX Xref Xval Xinc Unit]

No /RESAMPLE
Without this option, a consistency test checks that R and T have both the same X axes, according to the current SET ALIGN mode. If yes, R and T are stacked channel by channel (no resampling). If not, an error is raised and /RESAMPLE is suggested.

/RESAMPLE (without arguments)
The /RESAMPLE option is a request to allow the resampling, but it is not enforced. A consistency test first checks if the spectra X axes are identical or not, according to the current SET ALIGN mode. If yes, R and T are stacked channel by channel (no resampling). If not, the X axes are resampled according to the SET ALIGN mode. The X axis of the output sum is automatically computed. In particular, the output spectrum has the coarsest resolution of the two input spectra. Those are then internally resampled and stacked on the output axis.

/RESAMPLE NX Xref Xval Xinc Unit
A custom output X axis can be given with the syntax /RESAMPLE NX Xref Xval Xinc Unit. In this case, the resampling is always performed. The input spectra are internally resampled and stacked on this axis. See HELP RESAMPLE for possible uses and caveats.

Remember that if it is enabled, the resampling is time consuming and has non trivial effects on the spectra noise.

5.1.3 ASSOCIATE

LAS\ASSOCIATE ReservedName Array
LAS\ASSOCIATE FreeName Array /BAD BadValue [/UNIT UnitString]
LAS\ASSOCIATE AnyName /DELETE

This commands adds or deletes an Associated Array to/from the R observation.

The first form allows to add a reserved Associated Array. With such arrays, the metadata like the bad value or the unit are restricted by the Associated Arrays specifications: they can not be customized.

The second form allows to add a free (non-reserved) Associated Array. In this case, the bad value is mandatory (no default). The unit is optional and defaults to a blank string. The Associated Array name is free, except that it must use symbols allowed in Sic variable names (i.e. alphanumeric characters).

The input array type and kind will be reused as the Associated Array type and kind. Its first dimension must be of size RY (i.e. same number of channels as R). A second dimension is allowed. The array argument can refer to a subset of a bigger array, as long as this subset respects the above rules.

The third form will delete the named array from the section (no error if no such array). If there are no more arrays in the section, the section is disabled.

5.1.4 AVERAGE

LAS\AVERAGE [/RESAMPLE [NX Xref Xval Xinc Unit]] [/NOCHECK [Arg1 ... ArgN] [/WEIGHT TIME|SIGMA|EQUAL|ASSOCIATED]

Average all the spectra of the current index using the specified (/WEIGHT) or current (SET WEIGHT) weighting function (see HELP SET WEIGHT for details). The spectra are stacked channel by channel by default, if possible. Else the option /RESAMPLE must be invoked (see subtopic for details).

On return the R spectrum contains the average spectrum. If /WEIGHT or SET WEIGHT is ASSOCIATED (see HELP SET WEIGHT for details), the weight array of the average spectrum is saved in the associated array named W.

Bad channels are handled according to SET BAD. A sum may be interrupted by <^C>, but the result is then undefined.

AVERAGE also works on Continuum drift. The alignement may be Channel or Position in this case.

AVERAGE /NOCHECK

LAS\AVERAGE /NOCHECK [SOURCE|POSITION|LINE|SPECTROSCOPY|CALIBRATION|SWITCHING]

By default, AVERAGE checks the consistency of all the input spectra. This is a security to avoid mixing inconsistent data (e.g. spectra from

different sources). These checks can be individually (with one or more names) or all (no name) disabled using the option /NOCHECK. See HELP SET NOCHECK for details.

Note the key status of the SPECTROSCOPY check! AVERAGE uses this check to detect if the spectra X axes are aligned or not, to help its decision to enable resampling or not (see /RESAMPLE subtopic). Using /NOCHECK SPECTROSCOPY enforces the spectra alignment status:

AVERAGE /NOCHECK SPECTROSCOPY
    will assume the spectra are aligned and will stack them channel by channel (no resampling), while they are possibly not aligned.

AVERAGE /RESAMPLE /NOCHECK SPECTROSCOPY
    will assume the spectra are not aligned and will always resample them before stacking, even if not necessary (adds computation cost and possible round off errors).

AVERAGE /RESAMPLE

    LAS\AVERAGE /RESAMPLE [NX Xref Xval Xinc Unit]

  No /RESAMPLE
    Without this option, a consistency test checks that the spectra all have the same X axes, according to the current SET ALIGN mode. If yes, the spectra are stacked channel by channel (no resampling). If not, an error is raised and /RESAMPLE is suggested.

  /RESAMPLE (without arguments)
    The /RESAMPLE option is a request to allow the resampling, but it is not enforced. A consistency test first checks if the spectra X axes are identical or not, according to the current SET ALIGN mode. If yes, the spectra are stacked channel by channel (no resampling). If not, the X axes are resampled according to the SET ALIGN mode. The X axis of the output sum is automatically computed. In particular, the output spectrum has the coarsest resolution of the input spectra. Those are then internally resampled and stacked on the output axis.

  /RESAMPLE NX Xref Xval Xinc Unit
    A custom output X axis can be given with the syntax /RESAMPLE NX Xref Xval Xinc Unit. In this case, the resampling is always performed. The input spectra are internally resampled and stacked on this axis. See HELP RESAMPLE for possible uses and caveats.

Remember that if it is enabled, the resampling is time consuming and has non trivial effects on the spectra noise.

5.1.5 BASE

```
LAS\BASE [Deg] [/PLOT [Ipen]] [/INDEX] [/OBS] [/CONTINUUM [Flux]]
LAS\BASE SINUS Amplitude Period Phase [/PLOT [Ipen]]
LAS\BASE LAST [/PLOT [Ipen]]
```

When working on single spectra, BASE copies R into T, then subtracts from R a polynomial baseline of degree Deg, or a sinusoidal baseline if the first argument is SINUS. Polynomial baseline degree must be in range 0-100. Areas defined by the SET WINDOW command are not used to fit the baseline. The T content may be recovered with SWAP.

BASE SINUS Amplitude Period Phase
(Phase is in the same unit as the period, usually km/s or MHz) Subtract a sinusoid. Minimization will be done including also a linear baseline in addition to the sinusoid.

BASE LAST
Use the last determined baseline instead of computing a new one. It can be useful to find a baseline from one backend and apply it to another. This also applies for sinusoidal baselines. 0th and 1st order polynomials, and sinus fit are extrapolated if the R spectrum extends beyond the LAST spectrum limits. 2nd order (and higher) polynomials are not extrapolated and the edge values are used beyond the LAST spectrum limits.

BASE /PLOT
Plot this baseline after the minimization, using the Ipen-th pen properties. The pencil 2 (red at startup) is used by default.

BASE /INDEX
Subtracts a baseline to all records that have been loaded. This option is not needed if SET ACTION INDEX. /PLOT and /INDEX options are incompatibles.

BASE /OBS
indicate that the baseline is to be fitted to the current R buffer observations. This option is not needed if the action level is OBS (default). /OBS and /INDEX are incompatible.

BASE /CONTINUUM [Flux]

1) Divide the spectrum by the baseline (rather than subtracting it), and then 2) multiply by the average flux. This is useful to keep the continuum level, when continuum is present and variations of atmospheric emission are properly subtracted out by fast enough switching. This is of course the case for correlation spectra (e.g. coming from an interferometer). The continuum level will be adjusted to the value of the argument 'Flux', if present; e.g. /CONTINUUM 1.0 will produce a spectrum of line-

to-continuum ratio. BASE /CONTINUUM does not work with BASE LAST or BASE SINUS.

5.1.6 BOX

LAS\BOX [Arg1] [Arg2] [Arg3] [Arg4] [/UNIT Type [LOWER|UPPER]] [/INDEX] [/OBS]

BOX draws a frame suited for the current R spectrum or index in memory. The units and limits are determined according to the options of the SET command (SET MODE ; SET UNIT), and the box size governed by SET BOX.

```
Arg1/Arg2 modify the labelling of the lower (left) axis
          = P for parallel   labels (default for X axis)
          = O for orthogonal labels (default for Y axis)
          = N for no labels
Arg3      indicates that the ticks are In or Out ofthe box
Arg4      controls the labelling of the upper X axis and
          may have the same values as Arg1 or 2 (P, O, N).
```

The default is BOX P O I.

/UNIT Type [LOWER|UPPER]
: If present, it modifies (with no permanent action) the unit of the X axis (lower or upper according to the subsequent argument, default is LOWER). Type may be any of the usual SET UNIT arguments. For Continuum data, the /UNIT command has no effect, but the unit is controlled by the SET ANGLE command. /INDEX produces a box suited for 2D color plot with a wedge (the Y-left axis gives the number of spectra).

/OBS
: Plot a box for a single spectrum.

/INDEX
: Plot a box suited for 2D display: Y axis is the temperature scale, and Z axis is the observation number. All three axis (X, Y and Z) may be defined by using the SET MODE command.

5.1.7 CATALOG

LAS\CATALOG [Filename] [/SORT NONE|FREQUENCY]

Read an ASCII file of lines (frequency in GHz and name) into the SIC structure named CLASS%LINE. When the filename is missing, CATALOG uses by default the ASTRO line catalog. Each row of the line file contains the description of a line as follows:

```
   115.271204      12CO-1-0
```

Comment lines start with the exclamation mark (!). By default the line catalog is sorted by increasing frequency. This can be changed by using the option /SORT.

The SIC structure containing the catalog has 3 members:

```
   CLASS%LINE%N          [     integer] number of lines read
   CLASS%LINE%FRES[N]    [      double] line frequency in MHz
   CLASS%LINE%NAME[N]    [character*12] line name
```

CATALOG /SORT

```
LAS\CATALOG [Filename] /SORT NONE|FREQUENCY
```

Control the sorting of the output catalog. If the option is not given the output catalog is sorted by ascending frequency. Currently only two sorting options are available, NONE, which means the SIC structure order is the same as that in the file, and FREQUENCY, which is the same as the default behaviour.

5.1.8 CONSISTENCY

```
LAS\CONSISTENCY [/NOCHECK [Keyword1] [Keyword2] ... ]
```

Check the consistency of the current index according to particular header parameters. The attributes to be checked are ruled by SET CHECK or NOCHECK (see HELP for details).

The consistency is implicitly tested with some commands: AVERAGE, ACCUMULATE, STITCH, LOAD, TABLE.

/NOCHECK
: Without keyword, this option disables all the checks. With a keyword, disable the named checks. Supported keywords are documented in HELP SET NOCHECK.

5.1.9 COPY

```
LAS\COPY [/SORTED]
```

Write all the observations in current index into output file. For efficiency reasons, COPY refuses to copy a sorted index (SET SORT is other than NONE), but user can enforce the copy with the /SORTED option.

5.1.10 DROP

```
LAS\DROP Obs [Ver]
```

Removes an Observation from the current index. The version number must be specified if it is not the last in the input file, even if it is the last version in the current index. This commands has an effect only on the current index, i.e. until a new index is built with a FIND.

5.1.11 DUMP

```
LAS\DUMP Argument [Buffer]
LAS\DUMP /SECTION [Name] [R|T|P]
```

With no argument, lists the content of all present header sections of the current R buffer. Arguments are:

```
ADDRESSES    Lists addresses
DATA [R|T|P] Lists in addition the data values of buffer R, T or P
FILE         Lists information about the input and output files
INDEX        Lists information about the current index
MEMORY       Lists the memory usage of the Class buffers
OTF          Lists OTF informations
PLOT         Lists current plot definition
```

/SECTION [Name] [R|T|P]
Lists parameters of a given section. Default is all for R buffer.

5.1.12 EXTRACT

```
LAS\EXTRACT [X1 X2 [Unit]] [/INDEX]
```

In its default mode, EXTRACT copies R into T, extracts a subsection of R, and save the result in R. If the option /INDEX is present, or if SET ACTION is INDEX, a subsection of each observation in the index is extracted and saved in the output file.

Without arguments, the cursor is invoked (if available) to define the subsection range. If the arguments X1 and X2 are present, the subsection is defined by this range, in the given Unit (if any) or by default in the current X axis unit (SET UNIT). X2 can be indifferently greater or lower than X1, and Unit is a single character in the four C)hannel, V)elocity, F)requency or I)mage. Note that if Unit is F or I, the number of extracted channels can vary for two similar observations (same setup) because of different Doppler factors.

If the range overlaps the input spectrum limits, a warning is raised and the missing data is filled with the blanking value. Requesting a range off the input spectrum is an error. The command FIND /FREQUENCY can help you to build an index with spectra overlapping one frequency or one range of frequencies.

```
On return, the observation header is consistently updated: the reference
channel is shifted and number of channels updated. The PLOT section
(which remembers the previous plotting limits) is disabled because it
probably does not makes sense for the extracted data array.

Only regularly sampled X axes are supported.
```

5.1.13 FILE

```
    LAS\FILE Mode Name [SINGLE|MULTIPLE] [/CONVERT] [/OVERWRITE]

Selects the input and output files.
  FILE IN Name        open an existing file as input
  FILE OUT Name       open an existing file as output
  FILE OUT Name Type  creates a new output file of the specified type
  FILE BOTH Name      select the same file for input and output.
  FILE UPDATE Name    open a file for Updates only.
See subtopics for details.

The default extension is .30m but can can be specified using command SET
EXTENSION.
```

FILE IN

```
    LAS\FILE IN Name

Open an input file for reading.

The input file can be of 2 different forms:
  - the usual Class Data Format (.30m or other produced by several tele-
    scopes, and by Class itself),
  - a VLM (Velocity-Position-Position) cube in the Gildas Data Format.
    LMV cubes must be transposed first to VLM before use.
The usual Class commands (FIND, GET, etc) work transparently on both
kind of files.
```

FILE OUT

```
    LAS\FILE OUT Name
    LAS\FILE OUT Name SINGLE|MULTIPLE [/OVERWRITE]

Open a Class file for writing.

The first form opens an old, already existing file, for appending new
observations into it.

The second form creates a new (empty) file. If a file with the same name
already exists, it is overwritten if the option /OVERWRITE is present.
An error is raised otherwise.
```

```
A new ouput file can be of 2 different kinds:
  - the type MULTIPLE allows to have several version of the  same  spec-
    trum (same ObservationNumber). This type is the "historical" one for
    CLASS. It can be opened in all modes.
  - with the type SINGLE, the ObservationNumber is a  unique  identifier
    of  the spectrum: only one version exists. This type has been intro-
    duced to simplify and speed up On-The-Fly processing, in  particular
    at the telescope. It can not be opened in BOTH mode.
```

FILE BOTH

```
    LAS\FILE BOTH Name

Open  a Class file both for reading old observations and writing new ob-
servations, or new versions of the old observations already in the file.

This mode can be used only with MULTIPLE files  (see  FILE  OUT),  where
several  versions  of the same observation can be saved in the file. See
HELP WRITE for details.
```

FILE UPDATE

```
    LAS\FILE UPDATE Name

Open a Class file for updating its observations.

In this mode, the spectra can be read, modified, and updated  in  place,
typically  via a TAG or UPDATE command after a MODIFY. The command WRITE
(i.e. appending a new observation into the file) is not allowed.
```

5.1.14 FIND

```
    LAS\FIND [APPEND|NEW_DATA|UPDATE] [Options]

FIND performs a search in the input file to build a new index, according
to  selection criteria defined by the SET command. These criteria may be
temporarily modified by the following options.

Observation numbers:
  /ALL                  all versions of each observations are searched for
                        (if not present: only the last version)
  /NUMBER n1 n2         search for the specified range of observation number
  /ENTRY  n1 n2         search for the specified range of entry numbers

General information:
  /SCAN s1 s2           search for the specified range of scan numbers
  /SUBSCAN s1 s2        search for the specified range of subscan numbers
  /SWITCHING mode       search for the switching mode used
  /TELESCOPE name       search for the telescope used
```

```
  /OBSERVED d1 d2       search for these observation dates (JJ-MMM-YYYY)
  /REDUCED d1 d2        search for these reduction dates (JJ-MMM-YYYY)
  /QUALITY q            search for the data of quality better than Q
  /SECTION name         search for observations which have this section defi

Spectroscopic information:
  /LINE name            search by line name
  /FREQUENCY f1 f2      search by signal or image frequency value or range

Position information:
  /SOURCE name          search by source name
  /OFFSET a1 a2 [unit]  search for this position (within SET MATCH tolerance
  /RANGE w e s n [unit] search for the specified range of offsets (plus SET
  /POSITION a0 d0 sys   search for observations by absolute position
  /MASK File            search for observations in the mask

Drift information:
  /DRIFT angle          search by compatible drift angle (degrees) i.e. modu
  /DIRECTION angle      search by identical drift angle (degrees) i.e. modul
```

FIND does not return an error if the index is empty, but the variable FOUND is set to 0. FOUND is always set to the number of observations in the index.

FIND APPEND

```
LAS\FIND APPEND [/OPT1 ...]
```

FIND by default resets the current index. Found observations may be appended to the current index by specifying the argument APPEND. A compression occurs to avoid duplication of observations in the index.

FIND NEW_DATA

```
LAS\FIND NEW_DATA [/OPT1 ...]
```

Argument NEW_DATA can be used to find the new (and only new) observations (if any) present in the input file. FIND options or equivalent SET tunings apply. If no new observations are present, the current index will be empty.

FIND NEW_DATA does not wait for new data, but you can use the command NEW_DATA to do so.

This possibility is intended for sites where data acquisition is done in CLASS format (Pico Veleta, Plateau de Bure) to use CLASS as an automated quick look facility.

FIND UPDATE

```
LAS\FIND UPDATE [/OPT1 ...]
```

The input index will first be appended with the new observations (if any) present in the input file. This saves time because a (much) smaller part of the input file is actually read. The current index is then reset and recomputed from the input index (i.e. with old and new observations), as usual.

FIND UPDATE does not wait for new data, but you can use the command NEW_DATA to do so.

This keyword can be omitted if the SET FIND mode is UPDATE.

FIND /MASK

```
LAS\FIND /MASK [FileName]
```

Select observations according the given mask.

The file name should refer to a GDF file providing a 2D (position-position) mask with <=0 or blank values (not selected), or >0 (selected) real values. The spectra from the input index are selected according to their position (lamof, betof) on the mask. Positions completely out of the mask are not selected. Note that this command does not check the consistency of the projection center, projection kind, and projection angle (only offsets are used).

This option is best suited for use with the weight file (.wei) produced by the command XY_MAP, in which zero weights are put where no data is available. Using this weight file with FIND /MASK will thus ignore pixels/spectra with no data at all (all blank values).

Another possibility is to produce a custom mask from an image, e.g.

```
G\IMAGE myimage.gdf  ! or mycube.lmv
G\LIMITS /RG
G\SET BOX MATCH
G\PLOT
G\BOX
G\POLYGON                          ! Define with cursor
G\MASK OUT                         ! Mask pixels OUT of the polygon
LET RG +1 /WHERE RG.NE.BLANKING[1] ! Select pixels IN the polygon
G\WRITE IMAGE mymask.gdf           ! Save into file
```

In this example the cursor is used to define the mask, but one can use his own recipe to define the in/out rules.

The default for FileName is the FILE IN VLM cube name if any, with the

extension replaced by ".wei". Else, there is no default.

FIND /POSITION

LAS\FIND /POSITION A0 D0 GALACTIC|EQUATORIAL [Equinox]

Select observations according to their absolute position. A0 and D0 are sexagesimal strings (degrees, or hour angle for A0 in case of equatorial coordinates). The coordinate system (and its equinox for equatorial system) must be provided as third (and fourth) argument to the option.

In practice, the option converts (if they are different) the given coordinates to the system used by the observation currently considered (e.g. galactic to equatorial 2000). Then it translates the absolute position to relative position. Finally, it compares the offset position of the observation to this computed reference. SET MATCH is used as tolerance (same as FIND /OFFSET). These operations are repeated for all observations in the file.

Beware this option is less efficient than its equivalent FIND /OFFSET (which uses only index elements). When considering absolute coordinates, the whole position section must be read for each observations. As the Class Data Format does not enforce consistency of the reference positions of all the observations, all the conversions described above must be repeated for all observations. But if you know your file is consistent, FIND /OFFSET will always be faster for the same result. For example, prefer FIND /OFFSET 0 0 instead of FIND /POSITION to select the spectra near the reference position.

FIND /SWITCHING

LAS\FIND /SWITCHING Mode

Select observations according to their switching mode:
- *: any defined or unknown switch
- POSITION: position switch
- FREQUENCY: unfolded frequency switch
- FOLDED: folded frequency switch
- WOBBLER: wobbler switch
- BEAM: beam switch
- MIXED: special code for mixed modes e.g. after averaging spectra
- UNKNOWN: if the switching information is missing or no switching performed (e.g. CALSKY).

Note that evaluating the switching mode is unefficient since this information is not saved in the CLASS index, and requires to read each observation header one by one. Identifying observations by e.g. their date and scan number is more efficient, either by proper bookkeeping at the

telescope, or with a once-for-all use of the /SWITCHING option.

5.1.15 FITS

```
    LAS\FITS READ  File[.fits]
    LAS\FITS WRITE File[.fits]   /MODE   SPECTRUM|INDEX   [/BITS  Nbits]
[/CHECK]
```

Read or write the File.fits FITS file to/from the CLASS Data Format.

FITS READ

```
    LAS\FITS READ File[.fits]
```

Read the File.fits FITS file to the CLASS Data Format. If the FITS file has been written (see HELP FITS WRITE) using:
- the SPECTRUM mode, the (unique) spectrum is loaded in the R buffer (memory only). If needed, you have to open an output file and WRITE it in order to save it on disk. BITPIX=-64 spectra are supported and converted to the internal 4 bytes real representation.
- the INDEX mode, the (several) spectra are converted and implicitly written to the current output file (it must have been opened before). The previous contents of the Class file is preserved (spectra are appended). REAL*8 SPECTRUM columns are not supported.

This command can read 2 flavors of FITS files:
- the CLASS-FITS format (i.e. the one written by FITS WRITE or by some 3rd party softwares). See the CLASS programmer documentation for details.
- the Herschel-HIFI FITS format. See the dedicated memo for details.

FITS cubes can be converted to the Gildas (not CLASS) Data Format by the command VECTOR\FITS, and then imported as a Class index by the command ANALYSE\LMV. See HELP V\FITS and HELP LMV for details.

FITS WRITE

```
    LAS\FITS   WRITE  File[.fits]  /MODE  SPECTRUM|INDEX  [/BITS  Nbits]
[/CHECK]
```

WRITE the File.fits FITS file from the CLASS Data Format. The resulting FITS file is a binary table (BINTABLE) with columns of parameters and channel intensities (dedicated CLASS-FITS format, see CLASS documentation for details). In order to produce a data cube, the commands TABLE and XY_MAP must be used first. Then the resulting GDF cube can be exported with the command V\FITS.

/MODE is a mandatory option which can take the following values as argument:

SPECTRUM a simple FITS file is written from the current R spectrum in memory is written. The number of bits per intensity value used when writing the FITS file may be controlled through the /BITS optional argument. It can be: 16 or I*2 (2 bytes integer), 32 or I*4 (4 bytes integer), -32 or R*4 (4 bytes real), or -64 or R*8 (8 bytes real). Default is 32.

INDEX a FITS BINTABLE is written from all the spectra in current index. The user MUST make sure that all index spectra are consistent (same source, same line, same frequency setup, ...). The intensity values (and irregularly sampled X axis values, if any) are always written as 4 bytes real. The -64 and R*8 values of /BITS are rejected in this mode.

The default mode and number of bits can also be specified through the commands:

```
SET FITS MODE SPECTRUM|INDEX
SET FITS BITS Nbits
```

The option /CHECK will print the FITS HEADER on the terminal.

5.1.16 FOLD

```
LAS\FOLD [/BOUNDARIES KEEP|DROP]
```

FOLD copies R into T, and folds a (not yet folded) frequency-switched spectrum.

The behavior regarding blank values (contaminating or not the resulting spectrum) is ruled by the SET BAD tuning (see HELP SET BAD for details). If blank channels are located at the folded spectrum boundaries, they can be kept (/BOUNDARIES KEEP) or implictly dropped (/BOUNDARIES DROP). Default is KEEP.

Because of the frequency throw, there are frequencies which are not covered by all phases (R%HEAD%FSW%NPHAS), typically 1 phase instead of 2. The resulting channels in the folded spectrum are located at the spectrum boundaries. If SET ALIGN second argument is Intersect (resp. Composite), those channels are rejected from (resp. included in) the resulting R spectrum. Default is Intersect.

5.1.17 GET

```
LAS\GET [N [version]|FIRST|LAST|NEXT|PREVIOUS|ZERO]
```

GET copies R into T, and loads the Observation number N in R. The current index initialized by FIND is first explored, then, if needed, the whole input file. If N is absent, the previous (last read) Observation is recovered.

```
Instead of an observation number, several keywords can also be used:
  - FIRST:    the first Observation of the current index is loaded,
  - LAST:     the last Observation of the current index is loaded,
  - NEXT:     the next Observation in the current index is loaded,
  - PREVIOUS: the previous Observation in the current index is loaded,
  - ZERO:     the internal Observation counter is reset to 0, so that
              next GET NEXT will get the first observation. In this
              unique case, R and T are not modified.
```

5.1.18 HEADER

```
    LAS\HEADER

Displays some header information on the R Observation. The FULL format
is used, with the informations selected in command SET FORMAT written.
The lines written contain (in output order, but not all lines are neces-
sarily present)
- General information line:
    Scan number and version
    Source name
    Line name
    Telescope name
    Date of observation
    Date of last reduction
- Position information line
    RA or l        right ascension or longitude (or azimuth)
    DEC or b       declination or latitude (or elevation)
    Equinox        if equatorial coordinates are used
    Offsets        (in current units)
    coordinate     Eq, Ga, Az
- Quality information line:
    Quality of calibration (as defined by a SET QUALITY)
    Tau      opacity at zenith
    Tsys     system temperature (outside atmosphere).
    Time     total integration time on source (minutes)
    El       elevation of source
- Spectral information lines (2):
    N        number of channel
    I0       reference channel (real)
    V0       velocity at reference channel
    Dv       velocity resolution (signed)
    F0       rest frequency at reference channel
    Df       frequency resolution (signed)
    Fi       image frequency at reference channel
- Calibration information line:
    B_ef    Beam efficiency of telescope
    F_ef    Forward efficiency of telescope
    G_im    Gain in image band
```

```
- Atmospheric information lines (2):
    H2O      millimeter of precipitable water vapor
    Pamb     ambient pressure
    Tamb     ambient (receiver cabin) temperature
    Tchop    Chopper temperature
    Tcold    Cold load temperature
    Tatm     Atmospheric temperature in the signal band
    Tau      Zenith opacity in the signal band
    Tatm_i  Atmospheric temperature in the image band
    Tau_i   Zenith opacity in the signal band
- Continuum Drift information lines (2) :
    N        Number of points
    I0       Reference point
    A0       Angle offset at reference point
    Da       Angular spacing between points (signed).
    F0       Observing frequency
    Df       Band pass
    Pos. Ang. : Position angle of the drift.
followed by the list of Scan numbers added in the observation.
```

5.1.19 IGNORE

```
    LAS\IGNORE List_of_Observations [/SCAN iscan]
```

This command can be used to declared the specified list of Observations (from the INPUT file) to be ignored in all FIND operations. They effectively become invisible to CLASS (except in a LIST IN command), until a FILE IN command is typed again. The input file is not physically modified however. The list of observation may have the same format as the index list of a FOR command.

/SCAN iscan
    Ignore all observations with scan number 'iscan'.

5.1.20 LIST

```
    LAS\LIST [IN|OUT] [/BRIEF] [/LONG]  [/SCAN]  [/OUTPUT  File]  [/TOC]
[/VARIABLE VarName]
```

List header information about an ensemble of Observations. LIST is used for a quick look to Observation headers, in a more or less detailed format. IN or OUT keyword specifies the file to be listed. If absent, the current index is listed. A medium-sized format is used by default.

```
LIST IN       List the whole input file (all versions, see FIND /ALL)
LIST OUT      List the whole output file (all versions)
/BRIEF        Brief format
/LONG         Long format
/SCAN         Scan based compact format
```

```
/SCAN /BRIEF  Scan numbers followed by the number of spectra in each scan
/TOC          Display the table of contents. See subtopic for details.
/OUTPUT File  Send the list to a file
```

Some options are particularly well suited for large ensemble of spectra: /SCAN displays the index on a scan number basis (defined by a unique SOURCE, LINE, TELESCOPE and SCAN number), /SCAN /BRIEF gives the list of scan numbers with their total number of spectra. /SCAN and /LONG options are incompatible.

LIST /TOC

```
LAS\LIST [IN|OUT] /TOC [Attr1 Attr2 ... AttrN] [/VARIABLE VarName]
```

LIST /TOC displays the Table Of Contents of the current index, or of the input (LIST IN) or output file (LIST OUT). A summary of the different values of each attribute is made, plus a final summary of all the combinations (setups). The attributes can be:

- SOURCE
- LINE
- TELESCOPE
- OFF1
- OFF2
- ENTRY
- NUMBER
- BLOCK
- VERSION
- KIND
- QUALITY
- OBSERVED
- SCAN
- SUBSCAN

The following attributes are more relevant in the context of continuum drifts:

- DRIFT (compatible continuum drift position angles i.e. same angle modulo 180 degrees within tolerance)
- DIRECTION (identical continuum drift position angles i.e. same angle modulo 360 degrees within tolerance)
- FESB (based on telescope name but only same frontend+sideband)
- FRONTEND (based on telescope name but only same frontend)

By default, SOURCE, LINE and TELESCOPE are used.

The table of contents is also saved in a SIC, user-defined structure thanks to the option /VARIABLE (default is TOC%). The content of the structure is updated after each call to the command LIST /TOC. One must take care that subsequent calls to FIND can lead to values in TOC% inconsistant with the current index.

5.1.21 LOAD

```
LAS\LOAD [/NOCHECK keyword1 [keyword2 [keyword3] ...]]
```

Pack the whole current index in a 2D array. By default, check the consistency of the index (see CONSISTENCY). The option /NOCHECK allows to ignore some inconsistency, according to the optional keywords.

5.1.22 MERGE

```
LAS\MERGE
```

Merge all the continuum drifts from the input index into a single one. The resulting drift is saved in R.

The drifts are merged by concatenating and sorting the resulting RX and RY values. The output observation is then flagged as "irregularly sampled".

5.1.23 MODIFY

```
LAS\MODIFY Item [Values...]
```

This is a general purpose command to edit and change the current observation header. Some actions are merely presentation "details", others do affect the information in the observation. Among these, RECENTER, WIDTH, IMAGE and SWITCH_MODE are typically used only when the software did not agree with the hardware when the spectra was taken (i.e. when the information in the observation header is wrong). They should be used with caution.

MODIFY actions are limited. You should use the SIC\ command LET to modify header variables which are not available within the MODIFY command (Caution: this is the rope to hang you).

MODIFY BLANKING

```
LAS\MODIFY BLANKING New_Blank
```

Change the blanking value (bad and NaN) in both header and data. This may be required before you can build a consistent strip or cube, specially if the spectra were initially loaded from a FITS tape.

MODIFY BANDS

```
LAS\MODIFY BANDS
```

Exchange signal and image bands. This allows line fitting on image band for example. Currently, the spectrum is not rescaled even if the band ratio is not 1.

MODIFY BEAM_EFF

```
LAS\MODIFY BEAM_EFF Value
LAS\MODIFY BEAM_EFF /RUZE [B0 Sigma]
```

Change the beam efficiency.

The first syntax allows to define the new beam efficiency from a single value for the whole spectrum. If the beam efficiency previously had a non zero value, the spectrum intensities and affected header parameters are rescaled according to the old and new values. Otherwise, only the beam efficiency is set to the new value.

The second syntax allows to use the Ruze's equation, which is more suitable for large band spectra:

```
Beeff(lambda) = B0*exp(-(4*pi*Sigma/lambda)**2)
```

B0 is the scaling factor with no units (between 0 and 1), and Sigma the width factor in microns. If the beam efficiency previously had a non zero value, a rescale factor is evaluated for each channel and is applied to the intensities, the new beam efficiency is set to the one at the center of the spectrum, and the affected header parameters are rescaled according to this new value. Otherwise, only the beam efficiency is set to this new value.
B0 and Sigma can be omitted for 30M observations after 01-APR-2009. Values measured at the 30M will be displayed and used in this case.

MODIFY DOPPLER

```
LAS\MODIFY DOPPLER [SIGN|*|Value]
```

Change or recompute the Doppler factor according to the CLASS convention (-V/c). With no argument, checks whether the header value is consistent with the CLASS value computed according to the date and observatory. The checking precision corresponds to 0.6 m/s in V. A logical SIC variable, DOPPLER_PB, is set to true if the values disagree. If an argument is present:

SIGN	change the sign of the Doppler factor but do not recompute it,
*	compute and update the Doppler factor according to the observing date-time and the observatory position (this position can be customized with SET OBSERVATORY). This is suited when the frequencies in the Class header are described in the observatory frame. See also undefined Doppler section below.
0	force the Doppler factor to 0, i.e. the frequencies in the Class header are already corrected for the Doppler factor (described in the LSR frame).
Value	set the Doppler factor to <Value>

```
Undefined Doppler factor:
    If the Doppler factor is -1 (no physical meaning), it is considered
  "undefined" by Class. In most of the cases, this happens when reading
  old Class data files where the Doppler factor was not available in the
  data format. Since then, telescopes are encouraged to use the newest
  data format and write directly a valid Doppler factor value at the
  time the Class files are created.

    At read time (GET or other commands reading a spectrum), Class con-
  siders the value -1 as a request to recompute automatically the
  Doppler factor from the telescope position on Earth and from observing
  date-time, assuming the frequencies in the spectrum header are de-
  scribed in the observatory frame. In other words, Class tries to apply
  implicitly the command MODIFY DOPPLER * on the spectrum.

    The above automatic patch works if the observatory can be recognized
  by Class from the telescope name (R%HEAD%GEN%TELES). If not, you will
  encounter this error:
    E-COMPUTE_DOPPLER, Telescope not understood from XXX
  In this case, use the command SET OBSERVATORY beforehand.

    Beware that the automatic patch makes sense if the frequencies are
  described in the observatory frame. If they are described in the LSR
  frame, the Doppler factor must be forced to 0.

    Finally, if you have an old Class file without the Doppler factor,
  it is advised to copy it to a newer Class Data Format. This will save
  the computed Doppler value directly in the new file instead of recom-
  puting again and again this factor each time the spectrum is read from
  the old file. If MODIDY DOPPLER * is what you want (frequencies in the
  observatory frame), this will do the job:
    FILE IN old.30m
    FIND
    FILE OUT new.30m SINGLE ! Uses newest data format
    COPY                    ! Implicit MODIFY DOPPLER * on each spectrum
  If you need a custom Doppler value (most probably 0), use:
    FILE IN old.30m
    FIND
    FILE OUT new.30m SINGLE ! Uses newest data format
    FOR I 1 TO FOUND
      GET NEXT
      MODIFY DOPPLER 0
      WRITE
    NEXT
```

MODIFY FREQUENCY

```
LAS\MODIFY FREQUENCY Rfreq
```

Compute the velocity scale corresponding to a new rest frequency Rfreq. The sky frequency scale is not modified. If present, the resolution section is also updated.

MODIFY ELEVATION_GAIN

```
LAS\MODIFY ELEVATION_GAIN
```

Correct the spectrum for elevation gain losses as a function of frequency and elevation. This is valid for point sources. For extended sources, the correction is less, depending on the size of the source.

The rest frequency and elevation are taken from the spectrum header, assuming that the gain stays constant over the entire bandpass of the spectrum.

This correction is only implemented for 30M data. The formulae for the gains are documented in Juan Penalver memo 16-Apr-2012. The gain is the ratio of aperture efficiency at the elevation of the observation, divided by the maximum aperture efficiency near 50 degrees.

MODIFY IMAGE

```
LAS\MODIFY IMAGE Rimage
```

Change the image frequency (in MHz).

MODIFY LINENAME

```
LAS\MODIFY LINENAME "New Name"
```

Modify the line name (for bookeeping).

MODIFY OFFSETS

```
LAS\MODIFY OFFSETS O1 O2 [Unit]
```

Force the offsets to new values. They should be expressed in the current observation system and projection. Values are expressed in the given unit if any (RAD, DEG, MIN, SEC, MAS), or current Class unit if absent (current SET ANGLE).

This may be required when you average spectra from different offsets, because the resulting "position" is meaningless (usually the position of the first observation added).

MODIFY PARALLACTIC_ANGLE

```
LAS\MODIFY PARALLACTIC_ANGLE
```

Recompute the parallactic angle from the latitude of the observatory and the azimuth and elevation found in the header.

The azimuth convention is assumed to be South at 180 degrees.

MODIFY POSITION

```
LAS\MODIFY POSITION Lambda Beta
```

Change the projection center to (Lambda,Beta). Units are degrees, except Lambda in hour angle if coordinate system is Equatorial. Sexagesimal strings or numeric real values can be used.

The offsets are also modified, as to be consistent with the center. This is useful when different parts of one source are mapped relative to several reference positions.

This command is a shortcut for:
MODIFY PROJECTION = Lambda Beta =
i.e. modify the projection center but preserve the projection kind and angle.

MODIFY PROJECTION

```
LAS\MODIFY PROJECTION Type [A0 D0 [Angle [DEG|RAD]]]
```

Change the projection system of the current R observation, and optionally the projection center and angle at the same time. Type "MODIFY PROJECTION ?" for a list of supported projections.

A0 and D0 should be sexagesimal strings in degrees, except A0 in hour angle if coordinate system is Equatorial. Angle is the desired projection angle, with its unit given just after (default degrees).

If an argument is absent or equal sign "=", it is left unchanged. The offsets are recomputed consistently according to the modified parameters.

MODIFY RECENTER

```
LAS\MODIFY RECENTER Rrchan
```

Force the reference channel to a new value in order to change the velocity, frequency and image frequency scales. The sky frequency scale is also altered in this command.

MODIFY SCALE

```
LAS\MODIFY SCALE Unit
```

THIS COMMAND IS EXPERIMENTAL AND MAY EVOLVE IN THE FUTURE. NOTE THAT THE DATA FORMAT DOES NOT REMEMBER YET WHAT IS THE CURRENT UNIT OF RY, IT IS YOUR RESPONSIBILITY TO REMEMBER WHICH RESCALING YOU PREVIOUSLY APPLIED.

Rescale RY from its former unit to the new unit. Recognized units are:

```
TA*
Tmb
Jy/beam
mJy/beam
Jy/sr
mJy/sr
```

MODIFY SOURCE

```
LAS\MODIFY SOURCE source_name
```

Modify the source name.

MODIFY SYSTEM

```
LAS\MODIFY SYSTEM System
```

Force the System in which the center of projection (central position of the source) is defined. Valid systems are GALACTIC, EQUATORIAL, HORIZONTAL and UNKNOWN. This command is typically used for "foreign" data when the SYSTEM information has not been decoded correctly by the conversion package. No correction is applied on spectrum position.

MODIFY TELESCOPE

```
LAS\MODIFY TELESCOPE Name
```

Change the telescope name, for instance from IRAM-30M to 30M-MRT.

MODIFY VELOCITY

```
LAS\MODIFY VELOCITY Rvoff
```

Compute the image and rest frequencies corresponding to a new velocity (in km/s). The sky frequency scale is unchanged by this command.

MODIFY WIDTH

```
LAS\MODIFY WIDTH Rfres
```

Change the frequency resolution (in MHz).

MODIFY SWITCH_MODE

```
LAS\MODIFY SWITCH_MODE FSW Nphase Foff1 Time1 Weight1 ...
LAS\MODIFY SWITCH_MODE PSW Nphase Loff1 Boff1 Time1 Weight1 ...
```

Change the description of the switching mode. This command is typically used for "foreign" data when this information has not been decoded correctly by the conversion package.

FSW stands for unfolded frequency switch. For each phase, one should specify:
- the offset for the phase (in MHz),
- the duration of the phase (in seconds),
- the weight given to phase (arbitrary units)

PSW stands for position switch. For each phase, one should specify:
- the lambda offset for the phase (in current angular),
- the beta offset for the phase (in current angular),
- the duration of the phase (in seconds),
- the weight given to phase (arbitrary units)

5.1.24 MULTIPLY

```
LAS\MULTIPLY Fact
```

Multiply the R Observation by Fact. MULTIPLY leaves the T Observation unaffected. Note that Tsys is also multiplied by the same (absolute) factor to keep proper weights. If only the data is to be multiplied, use the SIC\LET command on array RY:

```
SIC\LET RY Fact*RY
```

5.1.25 NEW_DATA

```
[LAS\]NEW_DATA [Interval]
```

Waits until new observations have been written to the input file, and then return. No modification of the input or current index is done. It is then expected to use FIND NEW_DATA|UPDATE to rebuild the indexes with these new observations (see corresponding HELP for details).

The waiting time between 2 check can be customized as first argument (in seconds, default 5).

5.1.26 PLOT

```
LAS\PLOT [ArrayName] [/INDEX] [/OBS]
```

PLOT /OBS

Display the Observation in R. No argument or argument "Y" will plot

```
    the  spectrum  Y intensities of the R buffer (RY), but an associated
    array can be plotted instead given its name. It is equivalent to the
    following sequence of commands:
        CLEAR DIRECTORY  ! i.e. clear only the current window
        BOX
        SPECTRUM [ArrayName]
        TITLE
    The  type of PLOT can be specified by SET PLOT. The limits for the X
    and Y axes are set by SET MODE X and SET MODE Y.

PLOT /INDEX
    Equivalent to the following sequence of commands:
        CLEAR DIRECTORY
        BOX /INDEX
        SPECTRUM /INDEX
        TITLE /INDEX
    Plot a 2-dimensional image (built with the LOAD command) with veloc-
    ity/frequency  for  the X axis, index number for the Y-axis, and in-
    tensity being rendered as grey/color scale. Use SET MODE Y to choose
    the  color lower and upper limits. Entering command POPUP after PLOT
    /INDEX enables displaying a single spectrum,  selected  by  clicking
    with  the  middle button of the mouse. Use SET MODE Z to control the
    second dimension of the plot (e.g. number of entries).

Default is /OBS. The default can be modified with SET ACTION.
```

5.1.27 SAVE

```
    LAS\SAVE Name

SAVE creates a procedure file of name "Name.CLASS", containing  all  the
current  parameters  of  the  program.  This file may be executed at any
time using the @ command: just type "@ name" after the LAS>  prompt,  or
pass   "@ name"   as   a   parameter  when  invoking  CLASS  (by  typing
"CLASS @ name").  This file is composed of standard CLASS commands,  and
may be edited with any text editor.
```

5.1.28 SET

```
    LAS\SET something [value1 [value2 [...]]] [/NOCURSOR]

This  command  is used to set a value for a CLASS parameter. If no argu-
ment is given, the default value for the parameter will be restored.
```

SET ACTION

```
    LAS\SET ACTION I[ndex] O[bservation]

Defines the default action level: current index or  current  observation
```

```
(R buffer). Default is OBServation.
```

SET ALIGN

```
    LAS\SET ALIGN Type Range

Defines  the  way the spectra to be averaged are aligned (Type) and com-
bined (Range). Possible combinations are:
  Type:  C[hannel]      for spectra and drift
         V[elocity]     for spectra only
         F[requency]    for spectra only
         P[osition]     for drift only
  Range: I[ntersect] or C[omposite]
Default is SET ALIGN C I.

Note: the default means that continuum drifts in opposite directions are
not properly stacked: ALIGN Position must be specified for that.
```

SET ANGLE

```
    LAS\SET ANGLE Unit

Specify  the angle unit for offsets. Unit is one of: RAD[ian], DEG[ree],
MIN[ute of arc], SEC[ond of arc], MAS[milli second of arc].  Default  is
SECond. The ANGLE unit is also used to display Continuum drifts.
```

SET BAD

```
    LAS\SET BAD AND|OR

Specify  the  way  the commands which have to mix channels behave in the
case of bad channels. There are two families of such commands:
  - resampling (part of) commands: RESAMPLE, and optionally  ACCUMULATE,
    AVERAGE, and STITCH if resampling is needed,
  - addition  (part  of) commands: ACCUMULATE, AVERAGE, and STITCH. FOLD
    is also affected.

Default is OR, i.e. the output channel is bad if any of the input  chan-
nels  is  bad. AND indicates to use all the non-bad channels (if any) to
compute the output channel.
```

SET BASELINE

```
    LAS\SET BASELINE arg

Specify the degree of polynomial baselines to be subtracted. Default  is
1.
```

SET BOX

```
LAS\SET BOX L[andscape] P[ortrait] D[efault]
```

Calls the GREG1\SET PLOT and GREG1\SET VIEWPORT commands with default values suited for CLASS plots. If no keyword, restore the default box: SET VIEW 0.1 0.95 0.15 0.70. The box size can be modified using the GREG1\ command directly as for instance

```
LAS> g\set view 0.2 0.3 0.2 0.3
```

SET CALIBRATION

```
LAS\SET CALIBRATION [Beam_Tol [Gain_Tol]] or OFF
```

Specify the tolerance on the Beam efficiency (Beam_Tol) and gain image ratio (Gain_Tol), on turn or calibration checking. A value of 0 means no check. The default values are 0.02 and 0 respectively.

These values are used by the command CONSISTENCY (and the other commands using it implicitely) to verify that the calibration is consistant. They are also used by command WRITE which writes the corresponding information in the output file only if Beam_Tol is non zero. Note that because the beam efficiency is less than 1, you could use Beam_Tol=1 to suppress the calibration checking but still write the information.

SET CHECK

```
LAS\SET CHECK [Arguments]
```

Enable checking all (no argument) or named attributes in the commands performing a consistency check. See HELP SET NOCHECK for details.

SET DEFAULT

```
LAS\SET DEFAULT
```

Restore all parameters to their default value. The default values for individual parameters are documented under the corresponding subtopic.

SET ENTRY

```
LAS\SET ENTRY e1 e2
```

FIND will build the current index from the other selectors, and then from this index it will reduce the number of entries found to the given range. This is useful for example to iterate the index by consecutive ranges. Default is * *.

SET EXTENSION

```
LAS\SET EXTENSION Ext
```

Defines the default extension for input and output files. Default is '.30m'.

SET FIND

```
LAS\SET FIND UPDATE|NOUPDATE
```

FIND will check or not for new entries in the input file (see FIND UPDATE). Default is no implicit update (NOUPDATE).

SET FORMAT

```
LAS\SET FORMAT Type
```

Set the format of the title of observations plotted. The same format is used by the LIST command. The format may be BRIEF (Observation number and version, Source name, Line name, Telescope name and position offsets), LONG or FULL which is similar to the long format, but also displays the list of original Scans. The default is LONG. FULL format is always used for command HEADER.

Other keywords can be specified to indicate which type of information should be written for the LONG and FULL formats. This is done by command
LAS\SET FORMAT Keyword [ON] [OFF] (ON to write the information corresponding to the specified keyword, OFF to ignore it). The keywords are

- POSITION for position of the source, offsets, type of coordinates and Equinox
- QUALITY for the Opacity, System temperature, Elevation, Integration time.
- SPECTRAL for the number of channel, reference channel, velocity and resolution, signal and image frequency and resolution.
- CALIBRATION for the beam and forward efficiencies and the gain image ratio.
- ATMOSPHERE for the water vapor content, pressure, temperatures and opacities in signal and image bands.
- CONTINUUM for continuum drift information
- ORIGIN for the list of added scans.

The default is Position - Quality - Spectral for spectra, Position - Quality - Continuum for drifts. A line of general information is always written in any case.

SET FREQUENCY

```
LAS\SET FREQUENCY Freq1|* [Freq2|*] [SIGNAL|IMAGE]
```

Add a selection by frequency for the command FIND. This criterion is suited only for spectroscopic data.

- SET FREQUENCY Freq1
  select all observations which frequency range intersects the given frequency,
- SET FREQUENCY Freq1 Freq2
  select all observations which frequency range intersects at least partially the given range,
- SET FREQUENCY * Freq2
  select all observations which frequency range is at least partially under Freq2,
- SET FREQUENCY Freq1 *
  select all observations which frequency range is at least partially above Freq1,
- SET FREQUENCY *
  select all observations i.e. disable the selection by frequency. This is the default.

An optional keyword can be added after the values to indicate whether the selection should be performed by SIGNAL or IMAGE frequencies. Default is SIGNAL.

Users should be aware that this selection criterion requires to read more informations in the observation header, and thus has a computation cost which can become important for large datasets.

SET LEVEL

```
LAS\SET LEVEL interactive_min_value logfile_min_value
```

SET LEVEL IS OBSOLETE AND HAS NO EFFECT ANYMORE.
It can be mimicked with the SIC MESSAGE command (see HELP for details). To alter the verbosity of CLASS commands to terminal and to message file, one should modify the message filters of the CLASS package.

For example:

```
LAS90> SIC MESSAGE CLASS S=FEW ! Change the on-screen CLASS filter
LAS90> SIC MESSAGE CLASS       ! Print the two CLASS filters
class    on-screen  active   filter: FEW------
class    to-mesfile active   filter: FEWRI--CU
```

This keeps only the most important messages (Fatal, Errors, and Warnings) to the terminal, whereas Results, Informations and Commands are also printed in the message file by default.

SET LINE

```
LAS\SET LINE Name
```

FIND will select all observations according to the specified line name. Default is *. Wildcards are accepted at any position in the line name (e.g. NAM*, N*ME or N*M*): they match 0 or more characters.

SET MASK

LAS\SET MASK [ml1 ml2 [...]] [/VAR array]

Defines a mask for the Gaussian fits by its boundaries. If available, the cursor will be used to specify missing arguments, unless option /NOCURSOR is present. When using the cursor, enter: N (or Left mouse button) to enter a value, C (or Middle button) to cancel the last entry, H for help, E (or Right button) for exit. The boundaries are in current unit. Default is no mask. The boundaries are in current unit. See DRAW MASK to display the mask(s).

SET MASK /VAR mask_array
Define the masks from rank 1 array mask_array[1+2*NMASK]. The array must be of the following form:

mask_array[1]=NMASK	number of masks
mask_array[2*i:2*i+1]	boundaries of mask num. 'i'

SET MAP

LAS\SET MAP [Clear|Normal]

Will Clear (or Not) the screen before processing a MAP command. However, a "MAP Where" command does not clear the screen even in a "LAS\SET MAP Clear" mode.

SET MATCH

LAS\SET MATCH [Tolerance [Unit]]

Turn on the checking of position-matching in CONSISTENCY (and the other commands using it). This tolerance is also used in FIND, MAP and a few other commands.

The Tolerance is the offset position tolerance expressed in the given unit if any (RAD, DEG, MIN, SEC, MAS), or current Class unit if absent (current SET ANGLE). If the Tolerance is absent, it resets to its default (2 arc seconds).

SET MODE

LAS\SET MODE X | Y | Z | ALL [mode]

Select the scale mode for axis X, Y or Z, or all. The mode can be:

```
  AUTO    : limits as found in the observation header (DEFAULT),
  TOTAL   : all data plotted,
  CURRENT : fix the current values in use (whatever if they were
            Automatic, Total, or already Fixed), so that the next
            spectra plotted will reuse the same
  [void]: i)  limits are entered explicitly by the user
                    LAS\set mode x x1 x2
          ii) limits are entered interactively when no mode is
              given (if the cursor is available)
```

For 2D data (built with the LOAD command), SET MODE Y defines the color range. The second dimension range (e.g. observation number) is defined with SET MODE Z.

SET NOCHECK

```
LAS\SET NOCHECK [Arg1] ... [ArgN]
```

Disable checking all (no argument) or named attributes in the commands performing a consistency check (see HELP CONSISTENCY). Default is to check everything.

Arguments can be:
- SOURCE
- POSITION
- OFFSET. Offsets are effectively checked if position checking, offset checking, and offset tolerance (SET MATCH) are enabled.
- LINE
- SPECTROSCOPY
- CALIBRATION. Calibration is effectively checked if calibration checking and calibration tolerance (SET CALIBRATION) are enabled.
- SWITCHING

SET NOMATCH

```
LAS\SET NOMATCH
```

Turn off checking of position-matching in CONSISTENCY (and the other commands using it). This does not change the tolerance for the other functions like FIND and MAP.

SET NUMBER

```
LAS\SET NUMBER n1 n2
```

FIND will select all Observations with numbers between n1 and n2. Default is * *.

SET OBSERVATORY

```
LAS\SET OBSERVATORY *
LAS\SET OBSERVATORY Name
LAS\SET OBSERVATORY Name Longitude Latitude Altitude
```

If OBSERVATORY is *, Class will guess the observatory from the telescope name (R%HEAD%GEN%TELES). This is the default.

If an observatory name known by GILDAS/Astro is provided as first argument, Class will use the known longitude, latitude and altitude values independently of the telescope name.

If an observatory name is provided with its explicit longitude, latitude (both sexagesimal strings) and altitude (in kilometers), Class will use those values independently of the telescope name.

The two last syntaxes are useful for R%HEAD%GEN%TELES strings encoding observatories unknown by Class.

SET OBSERVED

```
LAS\SET OBSERVED d1 d2
```

FIND will select all Observations with observing dates between d1 and d2. Dates are specified as DD-MMM-YYYY (i.e. 14-JUL-1789). Default is * *.

SET OFFSETS

```
LAS\SET OFFSETS o1 o2 [Unit]
```

FIND will select all observations with offsets at this position, within the specified tolerance (SET MATCH) around o1 and o2. Values are expressed in the given unit if any (RAD, DEG, MIN, SEC, MAS), or current Class unit if absent (current SET ANGLE).

Default is * * (select all offsets). See also SET RANGE.

SET PLOT

```
LAS\SET PLOT Type
```

Set the plotting type for commands PLOT and SPECTRUM. Choices are:
- N[ormal] (polyline connecting the points), or
- H[istogram], or
- P[oint] (disconnected points using the current Greg marker).

Normal is considered to be 50% faster plot than Histogram, but Histogram

gives a better reading of the channel positions and values. Default is Histogram.

SET POSANGLE

```
LAS\SET POSANGLE amin amax
```

Specify the position angle range (continnum drifts only) for subsequent searches.

SET PROCESSING

The following SET Parameter commands control data processing options:

```
LAS\SET ALIGN Type Range
LAS\SET BAD Check
LAS\SET BASE arg
LAS\SET CALIBRATION [Beam_Tol [Gain_Tol]] or OFF
LAS\SET MASK [ml1 ml2 [...]] [/VAR array]
LAS\SET MATCH Tol
LAS\SET NOMATCH
LAS\SET TYPE CONTINUUM or LINE or SKYDIP
LAS\SET VARIABLE Section_Name [Keyword]
LAS\SET VELOCITY L[sr]H[eliocentric]A[utomatic]
LAS\SET WEIGHT type
LAS\SET WINDOW [wl1 wu1 [wl2 wu2 [...]]] -
               [/POLYGON [N] [filename1...filenameN] [/VAR array]]
```

SET QUALITY

```
LAS\SET QUALITY q
```

FIND will select only observations of quality better than Q (i.e. less than Q). When originally written, unless the real time acquisition system detected a severe problem, all observations have quality 0, a priori the best. The quality flag of an observation can be changed using the TAG command. See HELP TAG for the recommended quality scale.

SET RANGE

```
LAS\SET RANGE West East South North [Unit]
```

FIND will select all observations with offsets in the specified range, plus the current SET MATCH tolerance (use SET MATCH 0 to find the observations exactly in the defined ranges). Values are expressed in the given unit if any (RAD, DEG, MIN, SEC, MAS), or current Class unit if absent (current SET ANGLE).

Default is * * * * (i.e. select all offsets).

SET REDUCED

```
LAS\SET REDUCED d1 d2
```

FIND will select all observations with reduction dates between d1 and d2. Dates are specified as DD-MMM-YYYY (i.e. 14-JUL-1789). Default is * *.

SET SCALE

```
LAS\SET SCALE Unit
```

THIS COMMAND IS EXPERIMENTAL AND MAY EVOLVE IN THE FUTURE. NOTE THAT THE DATA FORMAT DOES NOT REMEMBER YET WHAT IS THE CURRENT UNIT OF RY, IT IS YOUR RESPONSIBILITY TO REMEMBER WHICH RESCALING YOU PREVIOUSLY APPLIED.

Specify the RY scale unit to be used by the current R observation and next ones loaded in memory (GET, COPY, LMV, etc). Recognized units are:

```
*
TA*
TMB
JY/BEAM
mJY/BEAM
JY/SR
mJY/SR
```

If SET SCALE is * (default), the intrinsic RY unit is not modified. If SET SCALE is a valid unit, RY is rescaled from its former unit to this new unit.

SET SCAN

```
LAS\SET SCAN s1 s2
```

FIND will select all observations with Scan numbers between s1 and s2. The Scan number, attributed by the on-line acquisition system may be different from the observation number which is used by CLASS to refer to the observations. Default is * *.

SET SELECTION

The following SET Parameter commands affect data selection in FIND:

```
LAS\SET ANGLE Unit
LAS\SET LINE Name
LAS\SET FREQUENCY Freq
LAS\SET MATCH Tol
LAS\SET NOMATCH
LAS\SET NUMBER n1 n2
LAS\SET OBSERVED d1 d2
```

```
LAS\SET OFFSETS o1 o2
LAS\SET POSANGLE amin amax
LAS\SET QUALITY q
LAS\SET RANGE West East South North
LAS\SET REDUCED d1 d2
LAS\SET SCAN s1 s2
LAS\SET SOURCE name
LAS\SET SYSTEM Type [Equinox]
LAS\SET TELESCOPE Name
LAS\SET TYPE CONTINUUM or LINE or SKYDIP
```

SET SUBSCAN

```
LAS\SET SUBSCAN s1 s2
```

FIND will select all observations with subscan numbers between s1 and s2. The subscan number reflects the record number during an On-The-Fly acquisition. Default is * *.

SET SORT

```
LAS\SET SORT keyword
```

Defines the sorting when building the INDEX with FIND. The sorting is in ascending order. Default sorting is recovered with SET SORT NONE.

Keyword is one of:
BETA KIND LAMBDA LINE NUMBER NONE OBSERVED QUALITY SCAN SUBSCAN SOURCE TELESCOPE VERSION

SET SOURCE

```
LAS\SET SOURCE Name
```

FIND will select all scans according to the specified source name. Default is *. Wildcards are accepted at any position in the source name (e.g. NAM*, N*ME or N*M*): they match 0 or more characters.

SET SYSTEM

```
LAS\SET SYSTEM Type [Equinox]
```

Specify the coordinate system to be used by the current R observation and next ones loaded in memory. Type may be E[quatorial] followed by an equinox, G[alactic], or A[utomatic]. Default is Auto, i.e. uses the type found in the data file. If the equinox of Equatorial coordinates is not specified, the previous value is used. The initial default is 2000.00. Coordinates and offsets are automatically converted to the specified coordinate system and precessed to the specified equinox if needed.

SET TELESCOPE

```
LAS\SET TELESCOPE Name
```

FIND will select all scans according to the specified telescope name. Default is *. Wildcards are accepted at any position in the telescope name (e.g. NAM*, N*ME or N*M*): they match 0 or more characters.

SET TYPE

```
LAS\SET TYPE CONTINUUM or LINE or SKYDIP
```

Defines on which type of data the program is working.

- LINE (or SPECTROSCOPY) (the default when entering LAS) selects Spectral Line data only, and the program prompt is LAS> (Line Analysis System).
- CONTINUUM selects continuum drifts results only, and the program prompt changes to CAS> (Continuum Analysis System).
- SKYDIP selects only skydips, and the prompt changes to SAS>

In CONTINUUM mode, a continuum drift is treated as a spectrum would be in LINE mode. You can add drifts, subtract baseline, make gauss fits, change plotting scales etc... The current angular units is used in plots. SKYDIP mode has very limited capabiblities: only command REDUCE may be used to analyse the SKYDIP results.

SET UNIT

```
LAS\SET UNIT Lower [Upper]
```

Specify the unit to be used for the lower and upper X axis. Allowed units are C[hannel], F[requency], [Image] and V[elocity]. Default is Velocity. For Continuum drifts, this unit is ignored, but the angular unit specified by SET ANGLE is used.

SET VARIABLE

```
LAS\SET VARIABLE Section_Name [READ|WRITE|OFF]
```

Change the read-write status of the SIC variables corresponding to the section of the header of the R observation. The section names are:

SECTIONS (to enable edition of the R%HEAD%PRESEC array)
GENERAL
POSITION
SPECTRO
RESOLUTION
BASE
HISTORY
PLOT
SWITCH

```
  CALIBRATION
  SKYDIP
  GAUSS
  SHELL
  NH3 (or HFS)
  ABSORPTION
  DRIFT
  BEAM
  CONTINUUM
  ASSOCIATED
  HERSCHEL
  COMMENT
  USER
```

The corresponding variables are read-only by default, but can be changed to read-write with the keyword WRITE. They can always be found under the R% structure.

Some stand-alone aliases are defined by default (e.g. LINE is an alias of R%HEAD%SPE%LINE). These aliases can be deleted with the keyword OFF.

The SIC variables have the name of the corresponding FORTRAN variables. They are defined below:

```
              SECTIONS Information about present sections
LOGICAL       R%HEAD%PRESEC[:]  [   ] List of present sections
Section codes can be found in CLASSCODES%SEC%.

              GENERAL General parameters, always present:
INTEGER*8     R%HEAD%GEN%NUM    [     ] Observation number
INTEGER*4     R%HEAD%GEN%VER    [     ] Version number
CHARACTER*12  R%HEAD%GEN%TELES  [     ] Telescope name
CHARACTER*12  R%HEAD%GEN%CDOBS  [  day] Date of observation
CHARACTER*12  R%HEAD%GEN%CDRED  [  day] Date of reduction
INTEGER*4     R%HEAD%GEN%KIND   [ code] Type of data
INTEGER*4     R%HEAD%GEN%QUAL   [ code] Quality of data
INTEGER*8     R%HEAD%GEN%SCAN   [     ] Scan number
REAL*8        R%HEAD%GEN%UT     [  rad] UT of observation
REAL*8        R%HEAD%GEN%ST     [  rad] LST of observation
REAL*4        R%HEAD%GEN%AZ     [  rad] Azimuth
REAL*4        R%HEAD%GEN%EL     [  rad] Elevation
REAL*4        R%HEAD%GEN%TAU    [neper] Opacity
REAL*4        R%HEAD%GEN%TSYS   [    K] System temperature
REAL*4        R%HEAD%GEN%TIME   [    s] Integration time
REAL*8        R%HEAD%GEN%PARANG [  rad] Parallactic angle
INTEGER*4     R%HEAD%GEN%XUNIT  [ code] X unit, if X coordinate section is p
Codes  for  KIND,  QUAL and XUNIT can be found in SIC%CODE%KIND%, CLASS-
```

```
CODES%QUAL% and SIC%CODE%XUNIT% respectively.

                    POSITION Position information:
CHARACTER*12  R%HEAD%POS%SOURC    [    ] Source name
INTEGER*4     R%HEAD%POS%SYSTEM   [code] Coordinate system
REAL*4        R%HEAD%POS%EQUINOX  [year] Equinox of coordinates
INTEGER*4     R%HEAD%POS%PROJ     [code] Projection system
REAL*8        R%HEAD%POS%LAM      [ rad] Lambda of projection center
REAL*8        R%HEAD%POS%BET      [ rad] Beta of projection center
REAL*8        R%HEAD%POS%PROJANG  [ rad] Projection angle
REAL*4        R%HEAD%POS%LAMOF    [ rad] Offset in Lambda
REAL*4        R%HEAD%POS%BETOF    [ rad] Offset in Beta
Codes for SYSTEM and PROJ can be found in SIC%CODE%COORD% and
SIC%CODE%PROJ% respectively.

           SPECTRO Spectroscopic information (for spectra)
CHARACTER*12  R%HEAD%SPE%LINE     [    ] Line name
INTEGER*4     R%HEAD%SPE%NCHAN    [    ] Number of channels
REAL*8        R%HEAD%SPE%RESTF    [ MHz] Rest frequency
REAL*8        R%HEAD%SPE%IMAGE    [ MHz] Image frequency
REAL*8        R%HEAD%SPE%DOPPLER  [    ] Doppler correction -V/c (CLASS conv
REAL*8        R%HEAD%SPE%RCHAN    [    ] Reference channel
REAL*8        R%HEAD%SPE%FRES     [ MHz] Frequency resolution
REAL*8        R%HEAD%SPE%VRES     [km/s] Velocity resolution
REAL*8        R%HEAD%SPE%VOFF     [km/s] Velocity at reference channel
REAL*4        R%HEAD%SPE%BAD      [    ] Blanking value
INTEGER*4     R%HEAD%SPE%VTYPE    [code] Type of velocity
INTEGER*4     R%HEAD%SPE%VCONV    [code] Velocity convention
Codes for VTYPE and VCONV can be found in SIC%CODE%VELO% and
SIC%CODE%CONV% respectively.

                RESOLUTION Resolution information
REAL*4        R%HEAD%RES%MAJOR    [ rad] Major axis
REAL*4        R%HEAD%RES%MINOR    [ rad] Minor axis
REAL*4        R%HEAD%RES%POSANG   [ rad] Position angle

          BASE Baseline information (for spectra of drifts)
PARAMETER     MWIND=100
INTEGER*4     R%HEAD%BAS%DEG            [      ] Degree of last baseline
REAL*4        R%HEAD%BAS%SIGFI          [     K] Sigma
REAL*4        R%HEAD%BAS%AIRE           [K.km/s] Area under windows
INTEGER*4     R%HEAD%BAS%NWIND          [      ] Number of line windows
REAL*4        R%HEAD%BAS%W1(MWIND)      [  km/s] Lower limits of windows
REAL*4        R%HEAD%BAS%W2(MWIND)      [  km/s] Upper limits of windows
REAL*4        R%HEAD%BAS%SINUS(3)       [      ] Sinus baseline result
REAL*4        WINDOW(FOUND,1+2*RNWIND)  [  km/s] 2D plots: window for each r
```

```
              HISTORY Scan numbers of initial observations
PARAMETER     MSEQ=100
INTEGER*4     R%HEAD%HIS%NSEQ          [   ] Number of sequences
INTEGER*4     R%HEAD%HIS%START(MSEQ)   [   ] Start scan number of seq.
INTEGER*4     R%HEAD%HIS%END(MSEQ)     [   ] End   scan number of seq.

                    PLOT Default plotting limits.
REAL*4        R%HEAD%PLO%AMIN    [   ] Min Y value plotted
REAL*4        R%HEAD%PLO%AMAX    [   ] Max Y value plotted
REAL*4        R%HEAD%PLO%VMIN    [   ] Min X value plotted
REAL*4        R%HEAD%PLO%VMAX    [   ] Max X value plotted

          FSWITCH Frequency switching information (for spectra)
PARAMETER     MXPHAS=8
INTEGER*4     R%HEAD%FSW%NPHAS           [    ] Number of phases
INTEGER*4     R%HEAD%FSW%SWMOD           [code] Switching mode (frequency, p
REAL*8        R%HEAD%FSW%DECAL(MXPHAS)   [ MHz] Frequency offsets
REAL*4        R%HEAD%FSW%DUREE(MXPHAS)   [   s] Time per phase
REAL*4        R%HEAD%FSW%POIDS(MXPHAS)   [    ] Weight of each phase
REAL*4        R%HEAD%FSW%LDECAL(MXPHAS)  [ rad] Lambda offsets
REAL*4        R%HEAD%FSW%BDECAL(MXPHAS)  [ rad] Beta offsets of each phase
Codes for SWMOD can be found in SIC%CODE%SWITCH%.

                   CALIBRATION Calibration parameters
REAL*4        R%HEAD%CAL%BEEFF      [     ] Beam efficiency
REAL*4        R%HEAD%CAL%FOEFF      [     ] Forward efficiency
REAL*4        R%HEAD%CAL%GAINI      [     ] Image/Signal gain ratio
REAL*4        R%HEAD%CAL%H2OMM      [   mm] water vapor
REAL*4        R%HEAD%CAL%PAMB       [  hPa] Ambient pressure
REAL*4        R%HEAD%CAL%TAMB       [    K] Ambient temperature
REAL*4        R%HEAD%CAL%TATMS      [    K] Atmosphere temp. signal band
REAL*4        R%HEAD%CAL%TCHOP      [    K] Chopper temperature
REAL*4        R%HEAD%CAL%TCOLD      [    K] Cold load temperature
REAL*4        R%HEAD%CAL%TAUS       [neper] Opacity signal band
REAL*4        R%HEAD%CAL%TAUI       [neper] Opacity image band
REAL*4        R%HEAD%CAL%TATMI      [    K] Atmosphere temp. image band
REAL*4        R%HEAD%CAL%TREC       [    K] Receiver temperature
INTEGER*4     R%HEAD%CAL%CMODE      [ code] Calibration mode
REAL*4        R%HEAD%CAL%ATFAC      [    K] Applied calibration factor (Tcal)
REAL*4        R%HEAD%CAL%ALTI       [    m] Site elevation
REAL*4        R%HEAD%CAL%COUNT(3)   [count] Power of Atm., Chopp., Cold
REAL*4        R%HEAD%CAL%LCALOF     [  rad] Longitude offset for sky measurem
REAL*4        R%HEAD%CAL%BCALOF     [  rad] Latitude offset for sky measureme
REAL*8        R%HEAD%CAL%GEOLONG    [  rad] Geographic longitude of observato
REAL*8        R%HEAD%CAL%GEOLAT     [  rad] Geographic latitude of observator
Codes for CMODE can be found in CLASSCODES%CALIB%.
```

```
          SKYDIP For Skydips observations. No associated data.
PARAMETER     MSKY=10
REAL*8        R%HEAD%SKY%SREST         [MHz] Rest frequency
REAL*8        R%HEAD%SKY%SIMAG         [MHz] Image frequency
INTEGER*4     R%HEAD%SKY%NSKY          [   ] Number of points on sky
INTEGER*4     R%HEAD%SKY%NCHOP         [   ]    -   -    -    -  chopper
INTEGER*4     R%HEAD%SKY%NCOLD         [   ]    -   -    -    -  cold load
REAL*4        R%HEAD%SKY%ELEV(MSKY)    [rad] Elevations
REAL*4        R%HEAD%SKY%EMISS(MSKY)   [  ?] Power on sky
REAL*4        R%HEAD%SKY%CHOPP(MSKY)   [  ?] Power on chopper
REAL*4        R%HEAD%SKY%COLD(MSKY)    [  ?] Power on cold load

           GAUSS Gauss fit results (for spectra or drifts)
PARAMETER     MXGAUS=10                                    Maximum number of
INTEGER*4     R%HEAD%GAU%NLINE             [            ] Number of componen
REAL*4        R%HEAD%GAU%RMS_BASE          [   Int. unit] Sigma on base
REAL*4        R%HEAD%GAU%RMS_LINE          [   Int. unit] Sigma on line
REAL*4        R%HEAD%GAU%RESULT(3*MXGAUS)  [area,v0,fwhm] Fit results
REAL*4        R%HEAD%GAU%ERROR(3*MXGAUS)   [area,v0,fwhm] Errors

        SHELL "Stellar shell" profile fit results (for spectra)
PARAMETER     MXSHELL=10                            Maximum number of fitted c
INTEGER*4     R%HEAD%SHE%NLINE              [   ] Number of components
REAL*4        R%HEAD%SHE%RMS_BASE           [   ] Sigma on base
REAL*4        R%HEAD%SHE%RMS_LINE           [   ] Sigma on line
REAL*4        R%HEAD%SHE%RESULT(4*MXSHELL)  [   ] Fit results
REAL*4        R%HEAD%SHE%ERROR(4*MXSHELL)   [   ] Errors

       HFS "Ammoniac and Hyperfine Structure" profile fit results
                        (for spectra, ex: NH3, HCN)
PARAMETER     MXHFS=10                              Maximum number of fitted com
INTEGER*4     R%HEAD%HFS%NLINE            [   ] Number of components
REAL*4        R%HEAD%HFS%RMS_BASE         [   ] Sigma on base
REAL*4        R%HEAD%HFS%RMS_LINE         [   ] Sigma on line
REAL*4        R%HEAD%HFS%RESULT(4*MXHFS)  [   ] Fit results
REAL*4        R%HEAD%HFS%ERROR(4*MXHFS)   [   ] Errors

        ABS "Hyperfine Structure" ABSORPTION profile fit results
                        (for spectra, ex: NH3, HCN)
PARAMETER     MXABS=10                               Maximum number of fitted co
INTEGER*4     R%HEAD%ABS%NLINE              [   ] Number of components
REAL*4        R%HEAD%ABS%RMS_BASE           [   ] Sigma on base
REAL*4        R%HEAD%ABS%RMS_LINE           [   ] Sigma on line
REAL*4        R%HEAD%ABS%RESULT(1+3*MXABS)  [   ] Fit results
REAL*4        R%HEAD%ABS%ERROR(1+3*MXABS)   [   ] Errors

            DRIFT Continuum drift description (for drifts)
```

```
REAL*8          R%HEAD%DRI%FREQ    [ MHz] Rest frequency
REAL*4          R%HEAD%DRI%WIDTH   [ MHz] Bandwidth
INTEGER*4       R%HEAD%DRI%NPOIN   [    ] Number of data points
REAL*4          R%HEAD%DRI%RPOIN   [    ] Reference point
REAL*4          R%HEAD%DRI%TREF    [   ?] Time at reference
REAL*4          R%HEAD%DRI%AREF    [rad?] Angular offset at ref.
REAL*4          R%HEAD%DRI%APOS    [rad?] Position angle of drift
REAL*4          R%HEAD%DRI%TRES    [   ?] Time resolution
REAL*4          R%HEAD%DRI%ARES    [rad?] Angular resolution
REAL*4          R%HEAD%DRI%CBAD    [    ] Blanking value
INTEGER*4       R%HEAD%DRI%CTYPE   [code] Type of offsets
REAL*8          R%HEAD%DRI%CIMAG   [ MHz] Image frequency
REAL*4          R%HEAD%DRI%COLLA   [    ] Collimation error Az
REAL*4          R%HEAD%DRI%COLLE   [    ] Collimation error El
Codes for CTYPE can be found in SIC%CODE%COORD%.

         BEAM Beam-switching parameters (for spectra or drifts)
REAL*4          R%HEAD%BEA%CAZIM   [ rad] Azimuth of observation
REAL*4          R%HEAD%BEA%CELEV   [ rad] Elevation of observation
REAL*4          R%HEAD%BEA%SPACE   [ rad] Beam spacing
REAL*4          R%HEAD%BEA%BPOS    [ rad] Position angle of beams
INTEGER*4       R%HEAD%BEA%BTYPE   [code] System for angle
Codes for BTYPE can be found in SIC%CODE%COORD%.

    CONTINUUM Double gaussian and baseline fit results (for drifts)
PARAMETER       MPOIFIT=8
INTEGER*4       R%HEAD%POI%LCONT            [   ] Number of components
REAL*4          R%HEAD%POI%BCONT            [   ] Sigma on base
REAL*4          R%HEAD%POI%RCONT            [   ] Sigma on line
REAL*4          R%HEAD%POI%NCONT(MPOIFIT)   [   ] Results
REAL*4          R%HEAD%POI%ECONT(MPOIFIT)   [   ] Errors

                    HERSCHEL Herschel-HIFI section
INTEGER*8       R%HEAD%HER%OBSID          [    ] Observation id
CHARACTER*8     R%HEAD%HER%INSTRUMENT     [str ] Instrument name
CHARACTER*24    R%HEAD%HER%PROPOSAL       [str ] Proposal name
CHARACTER*68    R%HEAD%HER%AOR            [str ] Astronomical Observation Reques
INTEGER*4       R%HEAD%HER%OPERDAY        [day ] Operational day number
CHARACTER*28    R%HEAD%HER%DATEOBS        [str ] Start date (ISO date)
CHARACTER*28    R%HEAD%HER%DATEEND        [str ] End date (ISO date)
CHARACTER*40    R%HEAD%HER%OBSMODE        [str ] Observing mode
REAL*4          R%HEAD%HER%VINFO          [km/s] Informative source velocity in
REAL*4          R%HEAD%HER%ZINFO          [    ] Informative target redshift
REAL*8          R%HEAD%HER%POSANGLE       [rad ] Spacecraft pointing position an
REAL*8          R%HEAD%HER%REFLAM         [rad ] Sky reference (a.k.a. OFF), lam
REAL*8          R%HEAD%HER%REFBET         [rad ] Sky reference (a.k.a. OFF), bet
REAL*8          R%HEAD%HER%HIFAVELAM      [rad ] ON average H and V coordinate (
```

```
REAL*8         R%HEAD%HER%HIFAVEBET    [rad ] ON average H and V coordinate (
REAL*4         R%HEAD%HER%ETAMB        [    ] Main beam efficiency used when
REAL*4         R%HEAD%HER%ETAL         [    ] Forward efficiency used when ap
REAL*4         R%HEAD%HER%ETAA         [    ] Telescope aperture efficiency
REAL*4         R%HEAD%HER%HPBW         [rad ] Azimuthally-averaged half-power
CHARACTER*8    R%HEAD%HER%TEMPSCAL     [str ] Temperature scale in use
REAL*8         R%HEAD%HER%LODOPAVE     [MHz ] Average LO frequency Doppler co
REAL*4         R%HEAD%HER%GIM0         [    ] Sideband gain polynomial coeff
REAL*4         R%HEAD%HER%GIM1         [    ] Sideband gain polynomial coeff
REAL*4         R%HEAD%HER%GIM2         [    ] Sideband gain polynomial coeff
REAL*4         R%HEAD%HER%GIM3         [    ] Sideband gain polynomial coeff
REAL*4         R%HEAD%HER%MIXERCURH    [mA  ] Calibrated mixer junction curre
REAL*4         R%HEAD%HER%MIXERCURV    [mA  ] Calibrated mixer junction curre
CHARACTER*28   R%HEAD%HER%DATEHCSS     [str ] Processing date (ISO date)
CHARACTER*24   R%HEAD%HER%HCSSVER      [str ] HCSS version
CHARACTER*16   R%HEAD%HER%CALVER       [str ] Calibration version
REAL*4         R%HEAD%HER%LEVEL        [    ] Pipeline level
```

SET VELOCITY

```
    LAS\SET VELOCITY Mode

Specify the velocity kind to be used by the current  R  observation  and
next ones loaded in memory. Mode can be
  - L[sr]   Forces the velocity axis values to Lsr values
  - H[elio] Forces the velocity axis values to Heliocentric values
  - A[uto]  The velocity type is taken from the header of the spectrum
Default mode is Auto.
```

SET WEIGHT

```
    LAS\SET WEIGHT Type

Define  the weighting to be used in the averaging engine (AVERAGE, ACCU-
MULATE, STITCH), in the command TABLE,  and  in  the  command  UV_TABLE.
Choices are:
  T[ime],       for Time*abs(Fres)/(Tsys**2) (spectrum) or
                    Time*abs(Width)/(Tsys**2) (continuum),
  S[igma],      for 1/Sigma**2,
  E[qual],      for equal weights,
  A[ssociated], to use the Associated Array named "W" from each input
                spectrum. "W" provides one weight value per channel.

In  case  of  a  folded frequency switch spectrum, the spectrum is given
twice more weight (as if the integration time was twice longer).

The default is TIME. Sigma is not recommended unless  you  just  made  a
baseline  fit before. Equal weight behaves differently in AVERAGE/STITCH
```

```
and ACCUMULATE (which produces the sum).
```

SET WINDOW

```
    LAS\SET WINDOW /DEFAULT
    LAS\SET WINDOW [/NOCURSOR]
    LAS\SET WINDOW NONE
    LAS\SET WINDOW wl1 wu1 ... wlN wuN
    LAS\SET WINDOW /VAR Array
    LAS\SET WINDOW /POLYGON [N Filename1 ... FilenameN]
    LAS\SET WINDOW /ASSOCIATED
    LAS\SET WINDOW AUTO
```

```
Defines one or several line windows to be avoided by BASE when fitting a
baseline.  Window boundaries are expressed in current X unit (SET UNIT).
If current X unit is F (resp. I), the values are offset frequencies from
the  rest  frequency  (resp. image frequency). There are several ways to
use this command:

SET WINDOW /DEFAULT
  Undefine the current SET WINDOW method. Next calls to the command BASE
  will raise an error. This is the default at startup.

SET WINDOW [/NOCURSOR]
  With  this  form  without  argument,  the cursor is invoked to specify
  missing arguments, unless option /NOCURSOR is present. When using  the
  cursor, enter: N (or Left mouse button) to enter a value, C (or Middle
  button) to cancel the last entry, H for help, E (or Right button)  for
  exit. The boundaries are in current unit.

SET WINDOW NONE
  Set the number of windows to 0. This is valid e.g. if there is no sig-
  nal in the spectrum. BASE will exclude no channel during the  baselin-
  ing.

SET WINDOW wl1 wu1 ... wlN wuN
  Define  N  windows with their lower-upper boundaries, expressed in the
  current X unit.

SET WINDOW /VAR Array
  Define the window from rank 1 array  win_array[1+2*NWIND].  The  array
  must be of the following form:
    win_array[1]=NWIND      number of windows
    win_array[2*i:2*i+1]    boundaries of window number 'i'

SET WINDOW /POLYGON [N [Filename1 ... FilenameN]]
  Define  the  spectral window(s) through polygon(s). N is the number of
  polygons (default 1). If filenames are present, polygons are read from
```

these files. If not, they are defined interactively with the cursor, and saved in files "window-"'i'".pol", where 'i' is the polygon number. This option makes sense only if an index is loaded and plotted (PLOT /INDEX).
The resulting windows are visible in the array SET%WINDOW of size [Nspec,1+2*Mwind], where:
- Mwind is the largest number of windows per spectrum over all the index,
- SET%WINDOW[ispec,1] is the number of windows for i-th spectrum in the index,
- SET%WINDOW[ispec,2*jwind] and SET%WINDOW[ispec,2*jwind+1] are the min-max range of the j-th window.

SET WINDOW /ASSOCIATED
Find the spectral windows in the Associated Array named LINE (must be present in the R buffer). LINE is a flag array of 0 (channel out of a line) and 1 (channel in a line).

SET WINDOW AUTO
Reuse the spectral window(s) as found in the observation header (Baseline section must be present). There are two possibilities:
- if the Associated Array LINE was used, R%HEAD%BAS%NWIND is -1 and BASE will rely again on this array, or
- the spectral windows were saved in R%HEAD%BAS%NWIND, R%HEAD%BAS%W1, and R%HEAD%BAS%W2 from a previous call to BASE, these windows will be used again.

5.1.29 SHOW

LAS\SHOW Arg1 ... ArgN | ALL

Display some tuning(s) defined by SET. Arg can be "ALL" to display all parameters specified by SET. See SET command for details.

5.1.30 SPECTRUM

LAS\SPECTRUM [ArrayName] [Yoff] [/INDEX] [/OBS] [/PEN Ipen]

/OBS
Plot the spectrum in the R buffer in the units and scales defined by SET UNIT and SET MODE. No argument or argument "Y" will plot the spectrum Y intensities of the R buffer (RY), but an associated array can be plotted instead given its name.

Yoff (default 0) is a constant Y-offset added to the plotted values (useful to compare two spectra).

The option /PEN allows to use a i-th pen instead of the current one,

without modifying it in return.

/INDEX
Plots the 2D array from the last LOAD command.

Without option, the behavior depends on SET ACTION (see PLOT). Default is OBS.

5.1.31 STITCH

LAS\STITCH [/RESAMPLE NX Xref Xval Xinc Unit] [/NOCHECK [Arg1 ... ArgN]] [/IMAGE] [/WEIGHT TIME|SIGMA|EQUAL|ASSOCIATED]

Stitch (i.e. make the union of) all the spectra of the current index using the specified (/WEIGHT) or current (SET WEIGHT) weighting function (see HELP SET WEIGHT for details). The command behaves like AVERAGE /RESAMPLE except it ignores the SET ALIGN status and composes the spectra on the signal frequency axis. Use the option /IMAGE to compose them on the image frequency axis.

On return the R spectrum contains the average spectrum. If /WEIGHT or SET WEIGHT is ASSOCIATED (see HELP SET WEIGHT for details), the weight array of the average spectrum is saved in the associated array named W.

Bad channels are handled according to SET BAD. A sum may be interrupted by <^C>, but the result is then undefined.

STITCH /NOCHECK

LAS\STITCH /NOCHECK [SOURCE|POSITION|LINE|SPECTROSCOPY|CALIBRATION|SWITCHING]

By default, STITCH checks the consistency between all the input spectra. This is a security to avoid mixing inconsistent data (e.g. spectra from different sources). These checks can be individually (with one or more names) or all (no name) disabled using the option /NOCHECK. See HELP SET NOCHECK for details.

Note the key status of the SPECTROSCOPY check! STITCH uses this check to detect if the spectra frequency axes are aligned or not, to help its decision to enable resampling or not (see /RESAMPLE subtopic). Using /NOCHECK SPECTROSCOPY enforces the spectra alignment status:

STITCH /NOCHECK SPECTROSCOPY
will assume the spectra are aligned and will stack them channel by channel (no resampling), while they are possibly not aligned.

STITCH /RESAMPLE /NOCHECK SPECTROSCOPY

will assume the spectra are not aligned and will always resample them before stacking, even if not necessary (adds computation cost and possible round off errors).

STITCH /RESAMPLE

LAS\STITCH [/RESAMPLE NX Xref Xval Xinc Unit]

No /RESAMPLE
Without this option, STITCH behaves as AVERAGE /RESAMPLE, stitching all the input frequency axes. The resampling is possible but not enforced. A consistency test first checks if the spectra frequency axes are identical or not. If yes, the spectra are stacked channel by channel (no resampling). If not, the frequency axes are resampled. The frequency axis of the output sum is automatically computed. In particular, the output spectrum has the coarsest resolution of the input spectra. Those are then internally resampled and stacked on the output axis.

/RESAMPLE NX Xref Xval Xinc Unit
A custom output X axis can be given with the syntax /RESAMPLE NX Xref Xval Xinc Unit. In this case, the resampling is always performed. The input spectra are internally resampled and stacked on this axis. See HELP RESAMPLE for possible uses and caveats.

5.1.32 SWAP

LAS\SWAP

Exchange the contents of the R and T buffers.

5.1.33 TAG

LAS\TAG Quality_Code List_of_Observations

Attributes a quality to a given list of Observations. Quality_Code is an integer in the range 0-9, and the recommended quality scale is

0 Unknown
1 Excellent
2 Good
3 Fair
4 Average
5 Poor
6 Bad
7 Awful
8 Should never exist
9 Reserved for deleted Observations

The operation is immediate and occurs in the OUTPUT file for all versions of all Observations specified in the list. If no list is given,

the R memory is attributed the specified quality. A FIND operation will only select Observations of quality better than (i.e. less or equal to) the quality specified by the SET QUALITY command, or in the /QUALITY option.

5.1.34 TITLE

LAS\TITLE [/BRIEF] [/LONG] [/FULL] [/INDEX] [/OBS]

/OBS
Writes the header of the Observation currently in R above the plotted data. The format used is the one selected by SET FORMAT, which may be temporarily overridden by the options.

/INDEX
Writes the content of the loaded index in a compact form: Range of scan numbers, of spectrum, of observing dates and of offsets. It may be combined the other options (/LONG and /FULL give the same result).

Without option, the behaviour is fixed by SET ACTION level. Default is OBS.

5.1.35 UPDATE

LAS\UPDATE

Update the LAST version of the observation in the R buffer in the output file. You can update only the last version. The update OVERWRITES the observation. This command MUST BE USED WITH CAUTION; in case of complex manipulations of observations, you may get confused and erase precious data.

If more space is needed (e.g., the new version has more header information), an error is returned. You MUST then WRITE the observation to avoid loss of information.

The only typical case where it may be used is in the following sequence:
GET Number
MINIMIZE or MODIFY
UPDATE

5.1.36 WRITE

LAS\WRITE [ObsNum]

Writes the observation in the R buffer onto the output file. If a version of this observation is already in the file, a new version of the observation is created. A new observation number can be specified as ar-

```
gument.  If  the implicit (R%HEAD%GEN%NUM) or explicit (argument ObsNum)
value is 0, automatic numbering is used i.e. a new (unused  yet)  number
is chosen (in pratice highest number in use plus one).

The behaviour depends on the "File Type" (see FILE command):
- for  files  of type MULTIPLE, a new version of the spectrum is written
  if the ObsNum already exists in the file.
- for files of type SINGLE, the ObservationNumber is  automatically  set
  to  the  next  one available in the file. If an ObsNum is specified in
  the command, WRITE checks if it already exists. If so, it  returns  an
  error; if not, it sets the specified ObsNum to the written spectrum.
- for  both,  the  version value (R%HEAD%GEN%VER) is always incremented,
  assuming something has changed in R between the time it has been load-
  ed in memory (or last written) and the current write.
On  return  of the command, R%HEAD%GEN%NUM and R%HEAD%GEN%VER are set to
the values which have been used.

The information written depends on the operations done on  the  observa-
tion.  Existing  fit results are written. The calibration information is
written only if the calibration is checked  (see  SET  CALIBRATION  com-
mand).

In the case of (old) on-the-fly data, WRITE automatically splits the da-
ta into records with entry numbers starting at 1 unless the  observation
number of the first record is given explicitly.
```

5.2 ANALYSE Language Internal Help

5.2.1 Language

```
            ANALYSE\ Command Language Summary

 COMMENT             Manipulate the COMMENT section of spectra
 DIVIDE              Divide R spectrum by T spectrum
 DRAW                Call the cursor or plot comments
 FILL begin end      Fill from begin to end with noise, blank, etc.
 FFT                 Compute and edit Fourier Transform of R or P
 GREG [name]         Make a TABLE or formatted file from current scan
 LMV                 Convert a data cube into a set of spectra
 MAP [M|W]           Plot a map of spectra (or only their location)
 MEMORIZE Arg        Memorize the current observation
 MODEL Var           Generate a CLASS spectrum from a 1D variable
 NOISE [S [NEW]]     Generate gaussian noise
 PRINT [Arg]         Print values in a formatted file, or a table
 POPUP               Zoom a spectrum from a STAMP, MAP or PLOT /INDEX
 REDUCE              Reduce a SKYDIP
 RESAMPLE Arg        Resample a spectrum on a specified grid
 RETRIEVE Arg        Retrieve an observation from the memories
```

```
SMOOTH [Arg...]    Smooth the spectrum in R
STAMP              Display all spectra in index on one plot
STRIP File         Create an image for Position-Velocity plots
TABLE              Build a GILDAS table from the current index
```

5.2.2 COMMENT

```
ANALYSE\COMMENT [READ|WRITE|APPEND|EDIT|DELETE] ["comment"]
```

This command READs, WRITEs or DELETEs an existing comment section (max. length: 1020 characters) of the current R buffer.

```
READ      display the comment section
WRITE     write (or append) the input comment given between
          double quotes.
EDIT      write a comment given interactively
DELETE    delete and reset the comment section
APPEND    Obsolescent
```

5.2.3 DIVIDE

```
ANALYSE\DIVIDE Threshold
```

This command divides the spectrum in R by the spectrum in T, which is left unchanged. The two spectra must be on the same velocity scale (Use command RESAMPLE to do so). Threshold is a value to avoid zero divide. The output spectrum is blanked for channels where abs(T) < Threshold.

5.2.4 DRAW

```
ANALYSE\DRAW [Action [Parameters]] [/PEN Ipen]
```

This command executes the specified drawing action, or calls the graphic cursor and uses the returned character from the cursor to execute the corresponding action. The mnemonic for the cursor characters are :

```
F for "FILL" To interpolate a bad channel
K for "KILL" To flag a bad channel
L for "LOWER" To flag line identifications, ...
M for "MASK" To show active masks
T for "TEXT" To annotate your spectrum
U for "UPPER" Similar to LOWER
W for "WINDOW" To show current line windows
```

any other to display the channel number, frequency, velocity and antenna temperature at the cursor position. The explicit form does not call the cursor and can be used in procedures.

The pen used to draw the elements can be customized with the option /PEN.

DRAW FILL

```
ANALYSE\DRAW FILL Nchan
```

Interpolates the value of the specified bad channel using the nearest valid channels. Channel Nchan must be blanked before. In cursor mode, the channel number is retrieved from the cursor position.

DRAW KILL

```
ANALYSE\DRAW KILL Nchan
```

Deletes the specified channel, i.e. attributes the blanking value to it. In cursor mode, the channel number is retrieved from the cursor position.

DRAW LOWER

```
ANALYSE\DRAW LOWER value "text" [angle]
```

Draws a vertical line with label "text" (written vertically by default, or with orientation 'angle' in degree) at abscissa 'value' in lower axis unit. The line length is adjusted to approach closely the spectrum value at the corresponding abscissa.

DRAW MASK

```
ANALYSE\DRAW MASK [Level]
```

Plots the active masks, with bars at the mask positions and Y=Level (Default 0).

DRAW MOLECULE

```
ANALYSE\DRAW MOLECULE frequency name [angle]
```

Display a molecule name at the given frequency (irrespective of the units from the SET UNIT command), with a given orientation (angle, default 90). The name lies on the lower X-axis.

frequency	Frequency in MHz of the line
name	String giving the name to be written
angle	Orientation, in degree, with respect to X-axis

DRAW UPPER

```
ANALYSE\DRAW UPPER Value "text" [angle]
```

Draws a vertical line with label "text" (written vertically by default, or with orientation 'angle' in degree) at abscissa Value in upper axis

unit. The line length is adjusted to approach closely the spectrum value at the corresponding abscissa.

DRAW TEXT

```
ANALYSE\DRAW TEXT X Y "text" I
```

Writes (horizontally) the string "text" at plot position (X,Y) in paper units, with the centering code I for the string.

DRAW WINDOW

```
ANALYSE\DRAW WINDOW [Ylow [Yup]]
```

Plots the current line windows, with bars. Ylow (default at bottom of the plot) and Yup (default 10% of Y range) refer to the bottom and upper levels of the windows drawn.

5.2.5 FFT

```
ANALYSE\FFT [/REMOVE [wmin wmax]]
            [[s1 e1] [s2 e2] ... [s10 e10] /KILL]
            [/INDEX] [/OBS] [/NOCURS]
```

FFT computes a Fourier transform and plots it. Since Fourier transforms can not deal with sparse data, blank values in the input data (spectrum or index) are replaced by interpolated values.

The command acts either on a single spectrum (from the R buffer) or on the INDEX. The default behavior is defined with SET ACTION, but can always be ignored with /OBS or /INDEX. Identically, the option /NOCURS overrides the default cursor.

Three modes are available (see subtopic for details):
- no option: the FFT is computed and plotted. The RY values are not modified.
- /KILL allows to linearly interpolate pieces of the Fourier transform, either interactively or by supplying values.
- /REMOVE indicates that the Fourier transform is to be computed from a spectrum linearly interpolated in some intervals (specified or taken from the last fitted profiles).

If the action is applied on the R buffer, the result (including /KILL, if any) of the FFT is available in the variables R%FFT%X and R%FFT%Y.

There is no option to customize the FFT plot limits. A custom plot can be done as follows:

```
LAS> FFT  ! Fill the R%FFT%X and R%FFT%Y arrays
LAS> G\LIMITS /VARIABLES R%FFT%X R%FFT%Y  ! Auto X and Y limits
```

```
LAS> G\LIMITS 0.0 1.0 = =  ! Override X limits with custom values
LAS> CLEAR
LAS> G\BOX
LAS> G\HISTO R%FFT%X R%FFT%Y
LAS> G\LABEL "Inverse Frequency (MHz\\u-1\d)" /X
```

FFT /KILL

```
FFT [s1 e1] [s2 e2] ... [s10 e10] /KILL
```

Perform a linear interpolation (on real and imaginary parts) on the fourier transform. The intervals are defined interactively (if no value supplied and the cursor is available) or from the command line (if values in MHz-1 are supplied).

FFT /REMOVE

```
ANALYSE\FFT /REMOVE [wmin wmax]
```

If no value supplied, the sum of existing fits of the current method is subtracted before computing the Fourier transform. If an interval is supplied, linearly interpolate between wmin and wmax.

5.2.6 FILL

```
ANALYSE\FILL s1 e1 [[s2 e2] ...] [/INTER] [/NOISE [rms]] [/BLANK [bval]]
```

Performs the specified action in the interval(s) specified with starting and ending values (in current units).

/INTER
: Makes a linear interpolation. When combined with /NOISE [rms], a Gaussian noise is added to the linear interpoltion.

/NOISE [rms]
: Fill in the interval(s) with a Gaussian noise of a given rms. The rms defaults to the one computed in the BASE section. If this section is absent, it is computed from Tsys, resolution and integration time in the GENERAL section. If one of these values is null (e.g. spectrum was generated by the command MODEL), an error is raised. This option can be combined with /INTER.

/BLANK [bval]
: Fill in with blank value unless bval is specified.

5.2.7 GREG

```
ANALYSE\GREG Name [/FORMATTED [Format]]
ANALYSE\GREG Name /READ [X n] [Y m]
```

Writes in file Name the contents of R buffer as a GILDAS table, or as a

formatted file if the /FORMATTED option is present.

If the /READ option is present, performs the reverse operation: load the file content in the R buffer. Only available for formatted files; the equivalent capability for GILDAS tables is available through a combination of DEFINE IMAGE and MODEL.

For the table, the information stored is :

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

Line #0 means sum of all lines. Then all lines are written individually. The output table can be put later in a formatted way using GILDAS task LIST in you need.

The table may be used as input to GreG to produce fancy plots, or by the GILDAS software for other applications. In particular, SIC variables can be used to subtract any of the fits from the spectrum to produce residuals if needed. For plots, note that you can produce essentially the same within CLASS, except under very special circumstances, and moreover that only CLASS can handle the different X-unit simultaneously.

For the formatted list, less information is written (and it takes longer to write it of course) :

1. Current X unit in column 1
2. Intensity in column 2
3. Fitted profiles - (fit(i),i=0,nline) - in column 3+i

The /FORMATTED option can be passed a custom Fortran format, e.g. to extend the number of digits in each column. Default is "(1PG17.9,12(1PG12.4))". Do not forget to provide the enclosing parentheses and as many specifications as the number of columns (up to 13 depending on the number of line fits). For example "(F15.6,1X,F15.4)" is valid if no line fit is available.

For the /READ option, columns are X 1 Y 2 by default but can be changed. If X 0 is given as zero, then only Y is read and X is filled with channel numbers. Only very coarse header information is filled in. A lot of complementary header editing will often be needed.

5.2.8 LMV

ANALYSE\LMV DataCube [/SCAN LAMDBA|BETA] [/STEP DLAMBDA [DBETA]] [/LINE LineCube] [/FORCE What Value]

Read the spectra cube stored in input DataCube and convert it in a set of spectra automatically written in the currently opened output file. The spectra cube can be ordered in LMV, VLM, LVM, etc... FITS files are automatically converted.

By default, the command sorts the spectra to mimic the scanning path of a telescope along the lambda direction, shifted on beta for each scan. Each constant beta row has a unique SCAN number. The sorting can be rotated by 90 degree by using the /SCAN BETA option.

Fully blanked spectra in the input cube are ignored. If the cube spatial axes are largely oversampled, the /STEP option allows the user to select 1 spectrum every DLAMBDA (resp. DBETA) in the LAMBDA direction (resp. BETA). DLAMBDA and DBETA are expressed in pixel numbers. If DBETA is omitted, it defaults to DLAMBDA. The default value of DLAMBDA and DBETA (if the /STEP option is not invoked) is 1, implying a conversion of the whole cube.

User should take care that many parameters of the CLASS spectrum headers have no meaning here. They are left to their default values.

If /LINE provides a LineCube which matches exactly the DataCube (dimensions, coverage, source name, ...), each spectrum extracted from the DataCube is exported with a LINE Associated Array extracted from the LineCube. The LineCube should encode with 1 the parts of the cube where there is signal, and 0 where there is no signal. The LINE Associated Array can be reused later on with the commands SET WINDOW /ASSOCIATED and BASE.

If /FORCE is present, elements from the cube header can be overriden at import time. For now, only the scale unit can be customized with the keyword UNIT:

```
  /FORCE UNIT Value
```

where value is a keyword for supported scale units. This is useful for cubes imported from outer world where the scale unit syntax may vary. Type "LMV cubename /FORCE UNIT ?" for possible choices.

LMV FAXIS

```
    ANALYSE\LMV frequency/velocity axis
```

If your input file is a Gildas cube (3 dimensions) but it has no or incomplete frequency/velocity axis description (as found in the spectroscopic section), it can not be imported as spectra into Class.

However, you can fix this by setting by yourself this description in the file header. This can be achieved with Sic commands as follows:

```
  DEFINE HEADER H mycube.lmv WRITE
  LET H%F_AXIS   ??   ! [    ] Freq/Velo axis (1, 2 or 3)
  LET H%UNIT?    ??   ! [Keyword] Fill e.g. UNIT3 if F_AXIS is 3. Keyword mu
                                  be VELOCITY or FREQUENCY according to H%CO
  LET H%LINE     "SOMELINENAME"
  LET H%FREQRES  ??   ! [MHz ] Channel spacing
  LET H%IMAGFRE  ??   ! [MHz ] Image Frequency
  LET H%RESTFRE  ??   ! [MHz ] Rest frequency
  LET H%VELRES   ??   ! [km/s] Most likely -C*H%FREQRES/I%RESTFRE
  LET H%VELOFF   ??   ! [km/s] Velocity at reference channel H%CONVERT[2,H%F
  LET H%DOPPLER  ??   ! [    ] Doppler factor (Class convention)
  LET H%VTYPE    ??   ! [code] Velocity type (see codes in SIC%CODE%VELO%)
  LET H%SPEC     14   ! Section size => non-zero enables the section
  DELETE /VARIABLE H  ! Save the modifications in the file
```

If your input cube is a FITS cube, you can either:

- edit the FITS with you favorite FITS editor, and add the missing keywords, or
- convert the FITS to the Gildas Data Format:

  ```
  V\FITS mycube.fits TO mycube.lmv
  ```

  and follow the steps above.

5.2.9 MAP

ANALYSE\MAP [Match|Where|Keep] [/CELL Size_X [Size_Y]] [/GRID] [/NO-LABEL] [/NUMBER] [/BASE [Ipen]]

Makes a map of spectra or display the true locations of the spectra. Offset limits are automatically computed, and it adjusts the size of a spectrum depending in the current box size. If not in WHERE-mode, the only constraint of MAP in this mode is that the X and Y limits must be fixed so that the spectra share a common scale.

MATCH

To force matching the X to Y ratio, which is otherwise adjusted so as to make the largest possible drawing. The options control presentation details.

WHERE

Only displays the spectra locations with crosses (the marker can not be modified). Can be combined with MATCH. Global variables idx%loff and idx%boff contains offsets positions of all entries of the current index, and may also be used, with GREG1\ commands (g\limit, g\box, g\point), to display the spectra locations. In this case, the marker can be modified (g\set marker).

KEEP

Will not recompute the grid. Uses the last grid. Can be useful for

online display.

The actual graphic output is delayed until the end of the index has been reached. This process may be slow on some inplementations and graphic terminals when the number of spectra is large. The MAP command can be interrupted by typing Control-C.

MAP /GRID

ANALYSE\MAP /GRID

A grid is plotted to separate the spectra.

MAP /CELL

ANALYSE\MAP /CELL Size_X [Size_Y]

If the option /CELL is given, the cell size of the grid is taken to be Size_X by Size_Y (default Size_X) current angular units. Spectra always fill the cells.

MAP /NOLABEL

ANALYSE\MAP /NOLABEL

If the option /NOLABEL is present, the coordinates are not written but ticks are still present. Tick can be suppressed by specifying a tick size of zero in command SET TICK.

MAP /NUMBER

ANALYSE\MAP /NUMBER

If the option /NUMBER is present, the observation number is written for each spectrum in the upper left corner.

MAP /BASE

ANALYSE\MAP /BASE [Ipen]

If the option /BASE is present, a line at Y=0 will be drawn in addition to the spectrum, using pen number Ipen (default current pen).

5.2.10 MEMORIZE

ANALYSE\MEMORIZE [Name]
ANALYSE\MEMORIZE Name|* /DELETE

If a Name is specified, put a copy of the current observation in a "memory" called Name (limited to 12 characters). The observation can later

be retrieved using command RETRIEVE. If no Name is specified, list the currently defined memories. Up to 100 memories can be defined at a time.

MEMORIZE Name /DELETE can be used to delete one "memory" area. Use a wildcard (*) to delete all memories.

5.2.11 MODEL

ANALYSE\MODEL Y_Variable [X_Variable] [/BLANK Bval] [/REGULAR [Xref Xval RestFreq]] [/FREQUENCY LineName RestFreq] [/XAXIS Xref Xval Xinc Unit]

Generates a CLASS spectrum from a 1D SIC variable. Note that the 1D variable can be (and frequently is) a subset of a higher dimensionality variable. This command is typically used either to generate test data, or more frequently to transfer data from outside world through ASCII files.

A second variable, if present, is interpreted as an associated X axis scale for a presumably non linear or irregularly sampled frequency axis, as can occur for non resampled data from AOS backends (or optical spectra). Please note however that non linear axes are a largely untested feature in CLASS, you should avoid this solution when possible.

There are three (exclusive) ways to define the X axis: provide a X array, /XAXIS option, /REGULAR option (see below). The model header will have a minimal values set, you should update it with SET VARIABLE used, the command will copy RX and header from the previous R buffer in memory.

MODEL Y ! No X array
Velocity and frequency axes description is copied from the previous R buffer in memory.

MODEL Y X ! With an X array
If the X array is regularly sampled, it behaves as MODEL Y X /REGULAR. Otherwise the spectroscopic axis values are taken "as is" from the X variable and saved as an irregular axis.

MODEL Y X /REGULAR ! With an X array
Velocity axis is read from the X variable, and checked for regular spacing. The reference channel and velocity at reference channel (velocity offset) are taken at the middle of the X array.

MODEL Y X /REGULAR Xref Xval RestFreq ! With an X array
Velocity axis is read from the X variable, and checked for regu-

lar spacing. Velocity spacing is taken from the X array. The reference channel, the velocity at reference channel (velocity offset), and the frequency at reference channel (rest frequency) are passed as arguments to the option.

MODEL Y /XAXIS Xref Xval Xinc Unit ! No X array
Define the X axis with its reference channel (Xref), its value at reference channel (Xval), and the increment per channel (Xinc). Units are km/s or MHz if Unit is VELOCITY or FREQUENCY respectively. If Unit is FREQUENCY, Xval is expected to be the rest frequency.

Additional options:

/FREQUENCY LineName RestFreq
Indicate line name and the rest frequency. Default rest frequency is 300000 MHz.

/BLANK Bval
Allow the user to provide the blanking value. Default is -1000.

5.2.12 NOISE

ANALYSE\NOISE [S [NEW]]

Generates and draws a spectrum with only gaussian noise of rms S, to be compared with actual receiver noise.

S defaults to the rms computed in the BASE section of the current spectrum. If this section is absent, it is computed from Tsys, resolution and integration time in the GENERAL section. If one of these values is null (e.g. spectrum was generated by the command MODEL), an error is raised.

If the argument NEW is present, the spectrum is not plotted but the R buffer is copied in T, and the generated noise put into the R buffer. The baseline section is updated accordingly (in particuler the noise value in R%HEAD%BAS%SIGFI).

5.2.13 POPUP

ANALYSE\POPUP [ObsNum | OffX OffY]

Plot an observation in a popup window.

- POPUP without argument will invoke the cursor to select interactively the observation to be plotted. Point the cursor to one spectrum in the MAP, STAMP or PLOT /INDEX plots, and follow the usage explained in the

terminal window.

- POPUP Number will display a separate window containing a plot (same as PLOT) of an observation specified by its number (last version only).

- POPUP OffX OffY will display a separate window containing a complete plot (same as PLOT) of an observation with offsets matching OffX and OffY, which has been previously displayed by a command MAP. If more than one observation have the same offsets, only the first one will be displayed.

In the first and last cases, POPUP assumes that the index has not been modified since the last STAMP, MAP, or PLOT /INDEX commands.

5.2.14 PRINT

ANALYSE\PRINT [Arg] [/OUTPUT File] [/TABLE File]

PRINT lists some observation parameters in a Greg compatible format. It loops on all the observations in the current index (one line printed per observation). The kind of parameters to be printed is ruled by the optional argument Arg. Default is FIT.

FIT Results of profile fits in current method

The information written is (1) the number of lines fitted, (2) the observation number, (3,4) the offsets in current angle units, (5,6,...) fit results of the current method in the same order as in DISPLAY command. For the Gauss Method, this order is: (5) Line Area, (6) its incertainty (7) Position (8) incert. (9) Width (10) incert. (11) Intensity (12) Base RMS (13) Line RMS.

AREA Integrated area computed by BASE

The information written is (1) the observation number, (2,3) the offsets in current angle units, (4,5) Area and rms noise.

AREA List Area in velocity ranges

The list in the same format as the FOR-NEXT loop, for example v1 v2 v3, or v1 to v2 by 1. Area in velocity ranges v1 to v2 and v2 to v3, etc. Up to 19 ranges (20 values in list) may be listed at a time. The information written is (1) the observation number, (2,3) the offsets in current angle units, (4,5,...) the areas. If a velocity boundary do not fall exactly on a channel boundary, the proper channel fraction is added to the sum.

CHANNEL List List of channels

```
    The  list in the same format as the FOR-NEXT loop. Up to 20 channels
    at a time may be listed. The information written is (1) the observa-
    tion number, (2,3) the offsets in current angle units, (4,5,...) the
    channels value.

  MOMENT List       Moments of the observations

    The list in the same format as the FOR-NEXT loop, for example v1  v2
    v3  v4,  or  v1 to v4 by 1. The moments (area, mean velocity, width)
    are computed in the ranges [v1,v2] [v3,v4] etc... Up  to  10  ranges
    (20  values  in list) may be listed at a time. Current unit is used.
    The information written is (1) the  observation  number,  (2,3)  the
    offsets  in  current  angle  units,  (4,5,6)  area, position, width,
    (7,8,9) etc...  If a velocity boundary do  not  fall  exactly  on  a
    channel  boundary,  it is rounded to the nearest integer channel, so
    that no channel fraction is involved in the computations.

  POINTING          Pointing fit results

    Results of CONTINUUM method fit are printed in a format  adapted  to
    pointing constants measurements. See subtopic for details.

  FLUX              Flux measurements results

    Results of CONTINUUM method fits printed in a format adapted to flux
    measurements. The information written is (1,2) The  observation  and
    scan number, (3) a code for scan direction, (4,5) Azimuth and Eleva-
    tion in degrees, (6) Time in hours, (7,8) Position and  error  along
    drift  direction,  (9)  the antenna number, (10,11) Width and error,
    (12,13) Intensity and rms noise, (14) Image gain ratio, (15,16) Sig-
    nal and Image frequencies, (17) Source name, (18) Observing date.

/OUTPUT File
  Write  the  results  in  a formatted file named File instead of on the
  screen. The file is suited for later processing by GreG, in particular
  for contour maps.

/TABLE File
  Write  the  results  in a table named File. This table can also be ac-
  cessed by GreG, in a much faster (50 times) way than a formatted file.
  Tables  can  also  handle  many more columns than formatted files, and
  they are not limited in precision by the formatting. Mathematical  op-
  erations  can  be  done directly on the table columns. However the FIT
  results cannot be written to a table.
```

PRINT FIT

```
    ANALYSE\PRINT FIT

The information written is the following ("current angle  unit"  is  the
current SET ANGLE setting):
  - 1 Number of lines fitted,
  - 2 Observation number,
  - 3 Lambda offset [current angle unit],
  - 4 Beta offset [current angle unit],

Then  follows  the  fit  results for the current selected method (METHOD
command), in the same order as in DISPLAY command. "X"  means  SET  UNIT
unit at time of the fit.

METHOD GAUSS:
  -  5 Fitted line area [K.X],
  -  6 Error on fitted line area [K.X],
  -  7 Fitted position [X],
  -  8 Error on fitted position [X],
  -  9 Fitted width [X],
  - 10 Error on fitted width [X],
  - 11 Intensity (Area/Width/1.064467) [K],
  - 12 Base RMS [K],
  - 13 Line RMS [K]

METHOD CONTINUUM:
  -  1 Observation number,
  -  2 Azimuth [degrees],
  -  3 Elevation [degrees],
  -  4 Fitted line area [counts.current angle unit],
  -  5 Error on fitted line area [counts.current angle unit],
  -  6 Fitted position [current angle unit],
  -  7 Error on fitted position [current angle unit],
  -  8 Fitted width [current angle unit],
  -  9 Error on fitted width [current angle unit],
  - 10 Fitted intensity (Area/Width/1.064467) [counts],
  - 11 Base RMS [counts],
  - 12 Line RMS [counts],
  - 13 Source name
```

PRINT POINTING

```
    ANALYSE\PRINT POINTING

The information written is:
  -  1 Observation number
  -  2 Scan number,
```

- 3 Code for scan direction according to R%HEAD%DRI%APOS [0 drift along elevation, +1 drift along azimuth, -1 other],
- 4 Azimuth [degrees],
- 5 Elevation [degrees],
- 6 Sidereal time [hours],
- 7 Fitted collimation along drift direction [arcsec]
  R%HEAD%DRI%COLLE + fitted position + R%HEAD%POS%BETOF (code 0), or
  R%HEAD%DRI%COLLA + fitted position + R%HEAD%POS%LAMOF (code 1), or
  fitted position (code -1)
- 8 Fitted collimation error (= error on fitted position) [arcsec],
- 9 Antenna number,
- 10 Station code,
- 11 Fitted width [arcsec],
- 12 Error on fitted width [arcsec],
- 13 Fitted intensity (Area/Width/1.064467) [counts],
- 14 RMS noise on line [counts],
- 15 Source name

5.2.15 REDUCE

```
ANALYSE\REDUCE
```

This command reduces a SKYDIP observation, by fitting the sky emission using atmospheric parameters written in the observation, and displays the results : Water vapor content and Telescope Forward Efficiency

5.2.16 RESAMPLE

```
ANALYSE\RESAMPLE Nx Xref Xval Xinc Unit [Shape] [Width] [/[NO]FFT]
ANALYSE\RESAMPLE /LIKE GDFFile [/FFT]
```

This command resamples a given spectrum on a specified X axis in given unit. The arguments are:

Nx the number of channels desired,

Xref the new reference channel,

Xval the value at the reference channel. Changing the reference frequency or velocity has non trivial effects on the axis. Use '=' when possible (see below), else change the value with caution,

Xinc the new channel separation. In case of FREQUENCY resampling, the resolution is expected to be given in the restf frame. However, RESAMPLE does not modify the current frame of the spectrum (i.e. the Doppler factor is left unchanged), so the input frequency resolution will be corrected to the current frame.

Unit VELOCITY or FREQUENCY for spectroscopic data, or ANGLE for continuum drifts, to indicate the units used for Xval and Xinc.

Shape (optional) a keyword for the frequency response of the synthesized channels (see subtopic for details). By default the shape of the output channels is the same as the shape of the input

channels.
Width (optional) the width of the output channels in units of the channel separation. The default (and minimum value) is 1.

Using an equal sign '=' for any of these arguments indicates to use the same value as before. Using a wildcard '*' will choose an automatic default:
Nx is set such as the last input and output channels are aligned,
Xref is set such as the first input and output channels are aligned,
Xval the reference frequency and velocity are left unchanged (same as =)
Xinc same as =
Shape same as =
Width same as =

/NOFFT is the default behavior. In this case, the output spectrum is a subset of the original one if their axes are aligned to a given precision. If not, the spectrum is resampled by direct interpolation. In this mode the exact channel separation requested will be used. The counterpart to this technique is that correlation is introduced between adjacent channels and thus affects the noise intensity (e.g. it is divided by sqrt(2) if the axis is shifted by half a channel).

The Width parameter is used to provide an oversampled output spectrum if needed.

RESAMPLE /FFT

```
RESAMPLE /FFT
```

The resampling is done in the delay domain and the data are Fourier transformed. The Fourier transform is divided by the transform of the input channel shape, then extrapolated by zeroes (if interpolation in the frequency domain is required), multiplied by the transform of the desired channel shape, and finally transformed back to frequency domain. The output channel separation usually needs to be slightly rounded (to enable use of the FFT, the input spectrum must contain a round number of output channels).

RESAMPLE /LIKE

```
ANALYSE\RESAMPLE /LIKE GDFFile
```

/LIKE option can be used to resample automatically the spectrum on the spectral axis used by the named GDF file (can be a cube or UV table).

RESAMPLE SHAPE

Four keywords are used to describe the frequency response of the synthesized channels:
- TBOX means a rectangular function in the delay domain, as for unsmoothed correlator channels (sin(pi.x)/(pi.x) in the frequency domain)
- TPAR means a parabolic function in the delay domain, as for smoothed correlator channels (smoothing function used at Plateau de Bure)
- FBOX means a rectangular function in the frequency domain (as for filterbank channels)
- FTRIANGLE means a triangular function in the frequency domain (as for filterbank channels, Hanning smoothed)

The shape of the input channels is derived from the backend number read in the telescope name for IRAM-30m spectra: TBOX for autocorrelator, FBOX for other (filter) backends. Otherwise, rectangular filter channels are assumed.

5.2.17 RETRIEVE

ANALYSE\RETRIEVE Name

Retrieve the observation in memory "Name" and copy it in the R observation.

5.2.18 SMOOTH

ANALYSE\SMOOTH [Arg...]

SMOOTH copies R into T, then degrades the frequency/velocity resolution of R according to the selected method. If the R buffer also provides Associated Arrays, the smoothing method is also applied to them, except for the method NOISE where this is not relevant. It is not yet possible to provide a custom smoothing method for the Associated Arrays.

The arguments are used to specify the method:

- HANNING (no argument, default)
  The new spectrum has twice less channels and twice less resolution. Each output channel RY[J] is evaluated from the inputs channels as TY[I-1]/2 + TY[I] + TY[I+1]/2.

- HANNING Width Spacing [Shift]
  The Hann window width is ruled by the "Width" variable, and the output channel spacing by the "Spacing" variable. By default, the left boundaries of the first input and output channels are aligned, but user can shift the output grid by any fraction of channel using a custom shift ("Shift" variable). All values are expected in lower axis unit. Invoking SMOOTH HANNING 4*res 2*res 0.5*res (where "res"

is the lower axis resolution) is equivalent to the default SMOOTH HANNING (no argument, as described above), except it uses the generic (less efficient) engine.

- GAUSS Width
  Convolves (by multiplication in the Fourier plane) by a gaussian of specified width in current X units. Because this method does not support bad values in the Fourier plane, bad channels are interpolated before use.

- BOX Nchan
  Averages Nchan adjacent channels to produce a spectrum with Nchan less resolution and channels.

- NOISE Flux Nc
  For each channel, sums at most Nc neighbouring channels until a total flux Flux is reached in the sum. Then attributes the average value (sum divided by number of channels added) to the channel. This smoothing is non-uniform, strictly positive, and has an obvious tendency to produce wings...

5.2.19 STAMP

ANALYSE\STAMP NX [NY] [/LABEL SCAN|NUMBER|SOURCE|LINE [...]]

This command is intended to have a quick look at all the observations in the index. It makes a plot of all the observations in the index, arranged in a "map" of NX by NY observations. (If the index contains many observations, they will be very small, hence the name of the command). The command does not require that X and Y scales be fixed.

Command POPUP allows a selective zooming of any observation displayed by command STAMP.

STAMP /LABEL

ANALYSE\STAMP NX [NY] /LABEL SCAN|NUMBER|SOURCE|LINE [...]

The option /LABEL writes the required additional information (scan number, observation number, source or line name) in some part of the box to help identification. Several additional informations can be displayed. number of each observation in the upper left part of its box.

5.2.20 STRIP

ANALYSE\STRIP File

STRIP creates a GILDAS 2-D image which can be used later by GILDAS to produce Velocity-Position plots using command RGMAP. STRIP works on the current index, and checks that it defines a true strip along one of the main directions (according to the SET MATCH tolerance). That means you should have build the current index with FIND /OFFSET Value * or FIND /OFFSET * Value. The current X unit is used. The name of the output file must be given.

To produce a strip along another direction than X or Y, change the offsets of your spectra using command MODIFY OFFSETS, and write the modified spectra in a new output file.

STRIP can also produce a GILDAS 2-D image from a set of parallel continuum drifts. The drifts must have the same steps, and be regularly spaced. The index must define a coherent map.

PV cuts might also be done easily in GREG from a GILDAS table or cube with more versatility.

5.2.21 TABLE

ANALYSE\TABLE Filename [OLD|NEW] [/MATH Expression1 ... ExpressionN] [/RANGE rmin rmax unit] [/RESAMPLE Nc Ref Val Inc Unit [Shape] [Width]] [/FFT] [/LIKE GDFFile] [/FREQUENCY name rest-freq] [/WEIGHT TIME|SIGMA|EQUAL] [/NASMYTH] [/NOCHECK [SOURCE|POSITION|LINE|SPECTROSCOPY]]

Build a GILDAS table from current index entries. No gridding is done. This table may be then regridded on a Regular Grid to build a datacube for further use in GREG.

OLD Append to table 'filename'
NEW Create a new table

/MATH
Compute one or more mathematical expressions instead of storing a spectrum. See subtopic /MATH for details.

/RANGE
By default, the whole X axis is stored in the GILDAS table. A range can be imposed, in which case the unit must be specified.

/RESAMPLE
By default, the spectra are resampled, on-the-fly while building the TABLE, on the first spectrum X axis. This option allows to enforce that resampling is done on the given X axis. See HELP RESAMPLE for possible uses and caveats.

/LIKE

```
    Same as /RESAMPLE, except that the X axis is guessed from a refer-
    ence GDF file (can be a cube or a UV table).

  /FFT
    To specify that resampling is done in time domain.

  /FREQUENCY
    To change the line name and/or the rest frequency.

  /WEIGHT
    Specify which weight is to be saved in the dedicated column in the
    output table. Default is ruled by SET WEIGHT.

  /NASMYTH
    Specify that the spectra offsets (X and Y leading columns) written
    to the table should be rotated to offsets expressed in the Nasmyth
    plane. Beware it is the responsibility of the user to remember if
    this conversion was applied or not, as the data format does not keep
    track of it.

  /NOCHECK
    By default, the consistency of the index is checked against SOURCE,
    POSITION, LINE and SPECTROSCOPY parameters (see LAS\CONSISTENCY) be-
    fore building the table. If the SPECTROSCOPY section appears incon-
    sistent, the spectra are resampled on-the-fly on the first spectrum
    X axis. Other sources of inconsistency are not recoverable and the
    table is not built. This option without keyword allows to disable
    all consistency checks. With keyword(s), disable the corresponding
    checks.

Warnings:
   1. /MATH, /RANGE, /RESAMPLE and /LIKE are exclusive
   2. Continuum data not supported
```

TABLE /MATH

```
    TABLE /MATH Expression1 [... ExpressionN]

Instead of storing the spectrum (channels) for each entry in the current
index, you can use any Sic mathematical formula (e.g. involving some R
buffer variables).

The following functions are also predefined (see e.g. HELP FUNCTION SDTV
for details):
    STDV(dummy)   integrated area (K.Km/s)
    TPEAK(dummy)  peak temperature (K)
    VPEAK(dummy)  velocity at peak (Km/s)
    TDV(v1,v2)    integrated area (K.Km/s) between velocities V1 and V2
```

```
    TPEAKV(v1,v2) peak temperature (K) between velocities V1 and V2
    VPEAKV(v1,v2) velocity at peak (Km/s) between velocities V1 and V2
```

For example, to create a table containing the integrated area of each spectra, you can use:

```
    TABLE Filename /MATH STDV(0)
```

To create a table of the RMS of each spectrum measured during BASElining, use:

```
    TABLE Filename /MATH R%HEAD%BAS%SIGFI
```

And to create a table of the area, position, and width of the line fitted with the command MINIMIZE for all the spectra, use:

```
    TABLE Filename /MATH R%HEAD%GAU%RESULT[1] R%HEAD%GAU%RESULT[2] -
      R%HEAD%GAU%RESULT[3]
```

Once used in XY_MAP, these tables will result in maps of integrated area, RMS, line width, etc.

5.3 FIT Language Internal Help

5.3.1 Language

```
        FIT\ Language Summary

DISPLAY             Prints the results of the profile fit of R.
VISUALIZE [N]       Plots the current fitted profile.
MINIMIZE            Performs a profile fit.
ITERATE             Iterates the profile fit.
KEEP                Saves Gaussian fit results in the output file.
LINES [N]           Enters the initial values for the profile fit.
METHOD Arg          Selects the line profile for fits.
RESIDUAL [N]        Computes the residuals of the profile fit.
RESULT [N]          Computes the profile fit.
```

5.3.2 DISPLAY

```
    FIT\DISPLAY [Yoff] [/NOPLOT] [/METHOD]
```

Displays the results of the profile fit of R for the current method, in the terminal and in the plot header. Yoff indicates the Y offset for the results to be printed on the plot; default value is 0.

For the Gauss (default) method (see HELP METHOD), the command displays (1) the Line number, (2) the Area, (3) the Position, (4) the Width and (5) the Intensity , each with their associated errors, and resulting RMS in the Base and Line windows.

Note that the values are also available in the R% variable, for use in procedures. See HELP MINIMIZE for details.

/NOPLOT: if present, print the fit result but do not display them on the plot.
/METHOD: if present, print the fit result for the corresponding method.

5.3.3 ITERATE

FIT\ITERATE [N] [/NOCHECK BASELINE]

ITERATE tries the GRADIENT technique to improve the last results of the profile fit. Optional argument N indicates the number of iterations. Do not expect spectacular results if the absolute minimum is far away! Typically, this command should be used if the previous fit did not converge, but seems close enough. If you are using the GAUSS method, be cautious not to change the X unit in the mean time.

ITERATE /NOCHECK

FIT\ITERATE [N] /NOCHECK BASELINE

Except for the CONTINUUM method, all the fitted models tend to zero at the edges, so the command checks by default if a baseline has been removed before fitting.

If the option /NOCHECK BASELINE is present, the command will not raise an error if a baseline has not been removed. This assumes the spectrum has intrinsicly a zero-level background.

5.3.4 KEEP

FIT\KEEP

Save profile fit results in the output file. KEEP is equivalent to UPDATE, but only profile fit results are saved; KEEP is subject to the same restrictions as UPDATE.

5.3.5 LINES

FIT\LINES [N [Guesses]] [/NOCURSOR] [/INPUT File_Name] [/SHOW]

Enter the initial values for the profile fit. If argument "AUTO" is given instead of a number, then the number of lines found in the last read spectrum will be used for ITERATE command.

If N is not specified, then the last value for N is used. If N is 0, MINIMIZE will attempt to guess initial values for a single line by computing the moments of the spectrum. If N is not 0, the user has to pro-

```
vide the initial values for line parameters. N is limited to  10.  These
values  should  be separated by spaces and can be entered as SIC expres-
sions as follows:

Method      Code,Value     Code,Value   Code,Value     Code,Value

GAUSS       Temperature    Position     Width

SHELL       Area (K.MHz)   Offset (MHz) Width (MHz)    Horn/Center

NH3         Temperature    Position     Width (km/s)   Opacity

HFS         Temperature    Position     Width (km/s)   Opacity

ABSORPTION  Opacity        Position     Width
  (no initial guess is needed for the continuum level)

The code is an integer interpreted as follows :
        0       adjustable parameter
        1       fixed parameter
        2       adjustable parameter (head of group)
        3       parameter fixed with respect to parameter coded 2 or 4
        4       fixed parameter (head of group)
Codes 2 3 and 4 are used to fit dependent lines (e.g. HCN, for which the
displacements  are  4.842  and -7.064 km/s, or -1.431 and 2.088 MHz, and
line ratios 1:0.6:0.2). Codes 0 and 1 only are allowed for  NH3  method.
The  value for a parameter with code 3 should be either the ratio to the
head of group value (for both intensity and width), or the  offset  from
head of group (for the position).

There are 4 possibilities for the user to provide the line parameters:
    LINES /INPUT File_Name
        Guesses  are  read from the input file. "Free" format is used to
        read in this input file. The first line must contain the  number
        of lines, others are as the input at the keyboard.
    LINES N "Code Val ..." "Code Val ..." ...
        The  guesses  can also be entered from the command line, as dou-
        ble-quoted strings (one string per fitted line), e.g.
            LINES 2 "0 0.20  0 0.00  0 25.00" "0 0.10  0 0.00  0 50.00"
    LINES N
        If the cursor is available, you will have to  set  interactively
        on the plot the boundaries of the N lines. The program then com-
        putes the moments of the spectra between these boundaries as in-
        put values. The codes are all set to 0 (free parameters) in this
        way which is meaningfull ONLY FOR FREE  and  INDEPENDENT  lines.
        This  cursor  selection is not used if you specify the /NOCURSOR
        option.
```

```
    LINES N /NOCURSOR
        If none of the 3 methods above is used, then the user is prompt-
        ed  in  the terminal for the guesses. The parameters must be en-
        tered together, separated by blanks,  one  set  of  guesses  per
        prompt line.
The option /SHOW can be used for a feedback of what initial guesses have
just been defined.

Command LINES is not supported for method CONTINUUM.
```

5.3.6 METHOD

```
    FIT\METHOD Arg [Parameters ...]

Select the line profile for fits. Five type of profiles are available.

  - GAUSS     Select gaussian profiles.
    Up to 10 Gaussian, dependent or independent can be fitted in a spec-
    trum, according to command lines. Fitted parameters are:
        1) Area
        2) Velocity, and
        3) Width (FWHM)

  - SHELL     Horn-type profiles for Circumstellar Shells.
    Fitted parameters are:
        1) Area [K.MHz]
        2) Center offset [in MHz]
        3) Full Width at zero level, and
        4) Horn-to-center ratio (-1 for parabolic optically thick lines,
           0 for flat-topped lines, Infinity for perfectly double peaked
           lines).
    Profiles  are symmetric. Up to 10 dependent or independent lines may
    be fitted. X axis must be frequency.

  - NH3(1,1) or NH3(2,2) or NH3(3,3)
    Compute ammonia line profiles, with the assumptions of gaussian  ve-
    locity distribution and equal excitation temperatures. Fitted param-
    eters are:
        1) Main group opacity times ( Excitation temperature minus Back-
           ground temperature),
        2) Velocity,
        3) Width (FWHM = sqrt(8.0*alog(2.0)) * line 2nd order moment),
        4) Main group opacity.
    Up  to  10  independent lines may be fitted. The hyperfine structure
    description can be found in the structure HFS% right after the  com-
    mand METHOD is invoked.

  - HFS FileName    for HyperFine Structure
```

This is very similar to the NH3 method, but the coefficients which describe the hyperfine structure are read in the file FileName. It must contain:
- the number of hyperfine components (< 40, first line),
- the velocity offset and relative intensity (repeated for each component, one per line),

The hyperfine structure description read from the this file can be found in the structure HFS% right after the command METHOD is invoked. The fitted parameters are the same as for the NH3 method, except for the main group opacity.

In the HFS method the main group opacity refers to the total opacity of the combined hyperfine components present in the description file divided by the sum of their relative intensities. i.e. if the given relative intensities are normalised (their sum is 1) the resulting main group opacity will be the sum of the opacities of all hyperfine components present in the description file. This behaviour is useful for example in the case where we have only a part of the hyperfine components of a transition available in a spectrum. In this case one may fit only the hyperfine components present on the spectrum and still recover the total opacity of the main transition. The safest way to avoid mistakes is to always use the statistical weight of the upper hyperfine level divided by the total statistical weight of the upper level of the main transition. In this manner one will always recover the total opacity of the main transition over all hyperfine components.

- ABSORPTION Filename for hyperfine structures in absorption
Fitted parameters are:
    1) One continuum level
and for each line:
    2) Opacity
    3) Velocity, and
    4) Width (FWHM)

- CONTINUUM [Width [Area_Ratio [Width_Ratio]]] for Continuum drifts
Fit a single gaussian and a linear baseline at the same time, with automatic determination of the parameters. Uses beam-switched information for two-component fitting if required. The command METHOD CONTINUUM accept additional arguments, which can be used to customize the fit:
    - Width is the width of the beam in arc seconds. By default, Width is a free parameter.
    - Area_Ratio is the reference to main beam area ratio, used for beam-switch observing. By default, Area_Ratio is 1.
    - Width_Ratio is the reference to main beam width ratio, used for beam-switch observing. By default, Width_Ratio is 1.

A * can be used instead of a numerical value for each parameter, and indicates that the parameter is free.

- BASECONT
  Same as CONTINUUM, except that VISUALIZE will plot the baselined fitted model. This makes sense if the plotted data is the baselined drift while the model parameters still provide the baseline (offset+slope) information.

Note that in command LINES, one enters the peak line antenna temperature instead of the first variable in the guesses, except for SHELL method. Command LINES has no effect for CONTINUUM.

CONTINUUM and GAUSS methods only are relevant for continuum data, while GAUSS, SHELL, NH3, HFS and ABSORPTION methods can be used for spectroscopic data.

5.3.7 MINIMIZE

FIT\MINIMIZE [/NOCHECK BASELINE]

Performs a fit of a theoretical profile in the R spectrum using initial values specified by the command LINES. The profile is determined by the METHOD command (GAUSS, SHELL or NH3 lines are currently implemented, and a special method, CONTINUUM, may be used for continuum data).

MINIMIZE performs the fit only on the part of the spectrum selected by SET MODE X. It also ignores the regions under the masks defined by SET MASK.

If LINES is 0, an attempt to guess initial values for a single line is done by computing the moments of the spectrum. A first minimisation is performed using the Simplex technique; the results are improved and the uncertainties computed using the Gradient technique. If MINIMIZE did not converge, the errors quoted in the results are meaningless, and you should use command ITERATE to try to do better.

In the fit result output, the fit is said:
- "optimistic" if R%HEAD%GAU%RMS_BASE > R%HEAD%GAU%RMS_LINE * 1.5,
- "bad" if R%HEAD%GAU%RMS_BASE < R%HEAD%GAU%RMS_LINE / 1.5

where RMS_LINE and RMS_BASE are the RMS of residuals on the line range(s) and the remaining base, respectively. This comment is purely informative.

For CONTINUUM method, although a linear baseline is fitted with the gaussian, only the gaussian parameters are displayed. If the observation was made using beam-switching method and the reference beam appears in the drifts, a two-component gauss fit is done using separation speci-

```
fied in the data.

The fitting results are saved in the R% variable, for e.g. reuse in pro-
cedures:
  Method      Structure
  GAUSS       R%HEAD%GAU%
  SHELL       R%HEAD%SHE%
  NH3/HFS     R%HEAD%HFS%
  ABSORPTION  R%HEAD%ABS%
  CONTINUUM   R%HEAD%POI%
See subtopic for details.
```

MINIMIZE GAUSSIAN

```
    FIT\MINIMIZE (parameters of the GAUSSIAN profile)

The gaussian fit parameters can be found in:

  Parameter       Variable

  Area            R%HEAD%GAU%RESULT['(ILINE-1)*3+1']
  Position        R%HEAD%GAU%RESULT['(ILINE-1)*3+2']
  Width           R%HEAD%GAU%RESULT['(ILINE-1)*3+3']

  Error(Area)     R%HEAD%GAU%ERROR['(ILINE-1)*3+1']
  Error(Position) R%HEAD%GAU%ERROR['(ILINE-1)*3+2']
  Error(Width)    R%HEAD%GAU%ERROR['(ILINE-1)*3+3']

where ILINE is the number of one of the fitted lines, between 1 and
R%HEAD%GAU%NLINE.
```

MINIMIZE SHELL

```
    FIT\MINIMIZE (fit parameters of the SHELL profile)

The shell fit parameters can be found in:

  Parameter             Variable                              Unit

  Area                  R%HEAD%SHE%RESULT['(ILINE-1)*4+1']  K.MHz
  Central frequency     R%HEAD%SHE%RESULT['(ILINE-1)*4+2']  MHz, offset from
  Half width at 0-level R%HEAD%SHE%RESULT['(ILINE-1)*4+3']  MHz
  Horn/Center           R%HEAD%SHE%RESULT['(ILINE-1)*4+4']  -

  Error(Area)           R%HEAD%SHE%ERROR['(ILINE-1)*4+1']   K.MHz
  Error(Central Freq)   R%HEAD%SHE%ERROR['(ILINE-1)*4+2']   MHz, offset from
  Error(HW at 0-level)  R%HEAD%SHE%ERROR['(ILINE-1)*4+3']   MHz
  Error(Horn/Center)    R%HEAD%SHE%ERROR['(ILINE-1)*4+4']   -
```

where ILINE is the number of one of the fitted lines, between 1 and R%HEAD%SHE%NLINE. See Class documentation for a detailed description of the fitting formula.

MINIMIZE HFS

```
    FIT\MINIMIZE (parameters of the NH3/HFS profile)
```

The NH3/HFS fit parameters can be found in:

Parameter	Variable
Tant * Tau	R%HEAD%HFS%RESULT['(ILINE-1)*4+1']
V lsr	R%HEAD%HFS%RESULT['(ILINE-1)*4+2']
Delta V	R%HEAD%HFS%RESULT['(ILINE-1)*4+3']
Tau main	R%HEAD%HFS%RESULT['(ILINE-1)*4+4']
Error(Tant*Tau)	R%HEAD%HFS%ERROR['(ILINE-1)*4+1']
Error(V lsr)	R%HEAD%HFS%ERROR['(ILINE-1)*4+2']
Error(Delta V)	R%HEAD%HFS%ERROR['(ILINE-1)*4+3']
Error(Tau main)	R%HEAD%HFS%ERROR['(ILINE-1)*4+4']

where ILINE is the number of one of the fitted lines, between 1 and R%HEAD%NFS%NLINE. See Class documentation for a detailed description of the fitting formula.

MINIMIZE CONTINUUM

```
    FIT\MINIMIZE (parameters of the CONTINUUM profile)
```

The continuum drift fit parameters can be found in:

Parameter	Variable	Unit
Baseline Offset	R%HEAD%POI%NCONT[1]	counts
Baseline Slope	R%HEAD%POI%NCONT[2]	counts/rad
Area	R%HEAD%POI%NCONT['(ILINE-1)*3+3']	counts.rad
Position	R%HEAD%POI%NCONT['(ILINE-1)*3+4']	rad
Width	R%HEAD%POI%NCONT['(ILINE-1)*3+5']	rad
Error(Base Off)	R%HEAD%POI%ECONT[1]	counts
Error(Base Slo)	R%HEAD%POI%ECONT[2]	counts/rad
Error(Area)	R%HEAD%POI%ECONT['(ILINE-1)*3+3']	counts.rad
Error(Position)	R%HEAD%POI%ECONT['(ILINE-1)*3+4']	rad
Error(Width)	R%HEAD%POI%ECONT['(ILINE-1)*3+5']	rad

where ILINE is the number of one of the fitted lines, between 1 and

R%HEAD%POI%LCONT. Note that these values are available in radians angles, but they are displayed or plotted in CLASS using the current SET ANGLE setting.

MINIMIZE /NOCHECK

FIT\MINIMIZE /NOCHECK BASELINE

Except for the CONTINUUM method, all the fitted models tend to zero at the edges, so the command checks by default if a baseline has been removed before fitting.

If the option /NOCHECK BASELINE is present, the command will not raise an error if a baseline has not been removed. This assumes the spectrum has intrinsicly a zero-level background.

5.3.8 RESIDUAL

FIT\RESIDUAL [N]

Compute the residuals of the last MINIMIZE fit. Copies R into T, then subtract the line profile number N from the spectrum in R. If N is 0, the sum of all components is subtracted.

5.3.9 RESULT

FIT\RESULT [N]

Compute the last MINIMIZE fit. Copies R into T, then save the line profile number N into R. If N is 0, the sum of all components is computed.

5.3.10 VISUALIZE

FIT\VISUALIZE [N] [/PEN [i]]

Plot the current fitted profile. The Nth line obtained the last time MINIMIZE was executed, or recovered by DISPLAY, is plotted on the screen. If N is 0, the sum of all lines is plotted, and if N is not given, the last value of N is used.

/PEN [ipen]: if ipen is present, use the given pencil 'ipen' properties (defined with "pen ipen"), otherwise use the last defined pencil. Without option, use pencil 2 properties.

5.4 MAP Language Internal Help

5.4.1 Language

```
XY_MAP              Build an LMV spectra cube from an XY table
```

5.4.2 XY_MAP

```
MAP\XY_MAP [TableName [CubeName]] [/NOGRID] [/TYPE LMV|VLM]
```

Build an LMV spectra cube and an associated weight image from an XY table (usually produced with the command TABLE).

The input XY table name is given as first input. If its extension is omitted, ".bat" table is selected if present, else ".tab" table is assumed. If the XY table name is omitted, the name used at the last command call is reused.
The output cube (.lmv, .vlm, ...) and the weight image (.wei) names are given as second input. If omitted, they share their names with the XY table one.

By default, data is gridded with a convolution (general case), unless /NOGRID option is invoked (typically when the data is already regularly placed). See subtopic for details.

By default, the spatial grid of the LMV cube will be large enough to use all the spectra of the XY table. The pixel size will ensure Nyquist sampling (i.e. at least 2 pixels per beam). The grid position reference and position angle will be taken from the XY table header. The convolution kernel is a Gaussian of size one-third the telescope beamwidth, the convolution being computed on a limited support (large enough to ensure excellent accuracy). The gridding can be customized with a set of parameters, see subtopic TUNING for details.

For large datasets, XY_MAP will process the data by slices of channels. This introduces reading, computing, and writing overheads, but there are ways to limit those. See subtopic MEMORY for details.

Note that the output LMV cube can be then analysed with the standard GreG image tools like GO VIEW. However, it is also possible to reimport the cube in Class as a set of spectra thanks to the command ANALYSE\LMV (see HELP for details).

XY_MAP /NOGRID

```
MAP\XY_MAP [Filename] /NOGRID
```

XY_MAP /NOGRID builds an LMV spectra cube and an associated weight image from an XY table, but does not regrid the data with a convolution.

The command checks that it encounters points which are regularly spaced (i.e., on a grid), and it deduces from these the typical parameters of

the spatial grid (pixel size, map size, ...). The grid alignment is checked with a tolerance. By default the tolerance is a tenth of the beam, if the beam is known (see HELP XY_MAP TUNING for the possible ways to declare the beam). If the beam is not known, the tolerance can be explicitly set in the variable MAP%TOLE (in arcsec).

The parameters of the output cube can still be overridden by user's MAP%LIKE and MAP% objects.

XY_MAP /TYPE

```
MAP\XY_MAP [Filename] /TYPE LMV|VLM
```

By default, XY_MAP produces an LMV (position-position-velocity) cube. This can be overridden by the option /TYPE to reorder the dimensions as VLM (velocity-position-position).

VLM is the natural ordering of the XY_MAP internal buffers, i.e. it should be faster to create the cube in this order. VLM cubes are also useful when data has to be accessed by spectrum (e.g. to use the cube with FILE IN mycube.vlm), as opposed to the LMV cubes which are efficient for access by channel (image plane).

XY_MAP TUNING

```
MAP\XY_MAP: tuning parameters
```

The gridding process may be finely tuned through user defined values of the MAP structure.

The header of an existing LMV cube may be used as a template for the grid definition. To do this, the user must set the variable

```
MAP%LIKE       [string    ] Reference header for spatial parameters
```

When MAP%LIKE is set to an empty string, the user may define the grid by hand through the following parameters

```
MAP%CELL[2]    [arcsec    ] X and Y pixel size
MAP%SIZE[2]    [pixel     ] X and Y map size
```

Each MAP%CELL sign indicates the axis orientation: East is left if MAP%CELL[1]<0 and North is top if MAP%CELL[2]>0.

The projection parameters can be customized with:

```
MAP%SHIFT      [logical   ] Reproject the grid?
MAP%PTYPE      [code      ] Grid projection type
MAP%RA         [  HH:MM:SS] Grid reference position
MAP%DEC        [+DDD:MM:SS] Grid reference position
MAP%ANGLE      [degree    ] Grid position angle
```

These parameters are applied only if MAP%SHIFT is YES. MAP%PTYPE is one

```
of the SIC%CODE%PROJ% codes (or H%PTYPE from the DEFINE HEADER command);
null  code stands for automatic choice (projection taken from the table,
except radio which is converted to azimuthal). Empty MAP%RA  or  MAP%DEC
are  taken from the table header. MAP%ANGLE is the target angle (not the
angle to be added).

The natural beamwidth of the telescope is  derived  from  (in  order  of
precedence):
    TAB%MAJOR      [radian    ] The resolution in the table header
    MAP%TELE       [string    ] The telescope name (if no resolution in tabl
    MAP%DIAM       [m         ] The telescope diameter (if MAP%TELE = "")
    MAP%BEAM       [arcsec    ] The beam size (if MAP%DIAM = 0)
Because  of  the convolution, the spatial resolution of the gridded data
will by default be 5.4% wider than the telescope beamwidth at the obser-
vation  frequency.  It is however possible to grid the data with a wider
Gaussian in order to smooth the data (e.g. to  increase  the  signal-to-
noise ratio of extended emission). On the other hand, it is by construc-
tion impossible to request a final resolution smaller than the intrinsic
resolution  (beam  size). The user can define the required resolution of
the gridded data through
    MAP%RESO[2]    [arcsec    ] Final resolution of the gridded data
The resolution can not be smaller than the beam size  (default  0  means
automatic  resolution). The command will deduce the size of the gridding
kernel from the telescope beamwidth and the required resolution.

In /NOGRID mode (see subtopic for details), XY_MAP will guess the X  and
Y  increments  from  the spectra positions, but it needs some tolerance.
The default is 1/10 of the beam, which is  enough  for  Nyquist  sampled
grids. However, if the beam is not known, or if the grid is oversampled,
the tolerance can be customized through the variable
    MAP%TOLE       [arcsec    ] Tolerance for X and Y increments (/NOGRID mo

It is advised to avoid changing the following more technical parameters.
However those parameters are tunable through
    MAP%COL[2]     [integer   ] Range of columns to grid
    MAP%WCOL       [integer   ] Column of weights
    MAP%CTYPE      [integer   ] Convolution kernel type
    MAP%SUPPORT[2] [arcsec    ] Convolution kernel support size

For  ease of use, a sensible value of each gridding parameter will auto-
matically be computed from the context if the value of the corresponding
MAP structure component is set to its default value, i.e. zero value and
empty string. The only exception is the projection  definition  position
(changing them requires in addition to set MAP%SHIFT to YES).
```

XY_MAP MEMORY

```
    MAP\XY_MAP: memory and efficiency
```

```
All the considerations below about memory consumption and file access
efficiency are described in details in the document "class-gridding.pdf"
in the CLASS documentation.

XY_MAP relies on the Sic logical VOLATILE_MEMORY as an upper limit for
its temporary memory comsuption. The current value of VOLATILE_MEMORY
can be checked and modified as follows:
  LAS> sic log volatile_memory        ! Current value
  volatile_memory = 90%
  LAS> sic log volatile_memory 50%    ! % of total RAM
  LAS> sic log volatile_memory 512MB  ! Explicit size
  LAS> sic log volatile_memory 4GB    !
90% is a correct value if you are not sharing the resources with several
people, or if you are not running other memory consuming software at the
same time.

If the problem (input table and output cube) fits in memory, the context
is ideal: XY_MAP will process all the channels in a single run. This is
the most efficient from a file access point of view. This means that you
should set VOLATILE_MEMORY as large as possible.

If the problem does not fit in memory, the good point is that XY_MAP is
still able to process the data. However, this implies processing the
channels by slices (intermediate smaller velocity ranges).

When slicing is enabled, the default ordering of the table is not ideal,
and each slice implies traversing the whole input table (knowing that
disk accesses are usually slow). If you want to avoid this, it may be
worth producing in advance a transposed ".bat" table with the command:
  LAS> TRANSPOSE mytable.tab mytable.bat 21
This transposition takes also time, but the overall time will be smaller
than letting XY_MAP dealing with a ".tab" table. On the other hand, this
duplicates the table data on disk, and the .bat table must be regenerat-
ed each time .tab table is updated.

Also when slicing is enabled, the default LMV order for the cube is ide-
al. Trying to produce directly a VLM cube (/TYPE VLM) is still possible
to some point, but this enlarges the memory consumption (one more buf-
fer) and increases the number of slices and the processing time. Prefer
LMV order if possible.
```

5.5 DSB2SSB Language Internal Help

5.5.1 Language

```
DSB2SSB\ Language Summary
```

```
INITIALIZE
DECONVOLVE
```

5.5.2 INITIALIZE

```
    DSB2SSB\INITIALIZE [Continuum_Offset]
```

Builds the data array for deconvolution from all spectra in the index. A continuum offset can be provided.

This instruction MUST be called before your first call to DECONVOLVE.

5.5.3 DECONVOLVE

```
    DSB2SSB\DECONVOLVE Tolerance  Itmax  [Lambda1  [Lambda2  [Lambda3]]]
[/GAIN [First_Guess]] [/KEEP]
```

Deconvolves DSB dataset loaded with INITIALIZE into SSB spectrum. Minimizes the function:

```
        Chi^2 - lambda1 F1 - lambda2 F2
```

until either the difference in Chi^2 between the iterations is less than the Tolerance, or the maximum number of iterations Itmax has been reached.

Here:

```
  Chi^2 = Sum (dsb - ssb_lsb*gain_lsb - ssb_usb*gain_usb)^2
          /(weight * nu)

  F1 : entropy of the channels.
  F1 = - Sum ssb/SSB * log ssb/SSB

  F2 : entropy of the gains.
  F2 = - Sum gain_ssb/e * log gain_ssb/e
```

and

```
  dsb          - double sideband data
  ssb          - single sideband data
  gain         - gain of single sideband data
  weight       - T_sys^2/(integrationime * bandwidth)
  nu           - number of free parameters:
  dsb_channels - ssb_channels - sideband_gains
  SSB          - Sum ssb
```

If Lambda1 or Lambda2 or Lambda3 are not specified, they are set to 0 and the algorithm does NOT try to maximize the entropy term.

By default the deconvolution makes no attempt at fitting the sideband gains, unless the option /GAIN is specified.

DECONVOLVE /GAIN

```
DSB2SSB\DECONVOLVE /GAIN [First_Guess]
```

When this option is present, the deconvolution will try to adjust the sideband gains, fitting one constant value per sideband for every LO setting.

--> make sure you have run the deconvolution without gain fitting first! Here's a typical sequence of commands to turn on the gain fitting during the deconvolution:

```
> initialize
> deconvolve 1e-5 100 /keep
> deconvolve 1e-5 100 /gain
```

First_Guess can be set to 1 or 0.

By default, First_Guess = 1 (the initial guess for signal and image sideband gains is that they are equal to 1 for every LO setting).

Should any information on the sideband ratio be available in the header variable GAINI, the user can enforce this as initial guess for the gain fitting by setting First_Guess = 0.

DECONVOLVE /KEEP

```
DSB2SSB\DECONVOLVE /KEEP
```

Keeps the outcome of the deconvolution in memory and uses it as initial guess for the next (if any) run. The initial guess is reset to 0 whenever the command INITIALIZE is used.

5.6 EXPERIMENTAL Language Internal Help

5.6.1 Language

```
EXPERIMENTAL\ Language Summary
```

These commands are experimental and their name, calling sequence, and behavior are subject to changes without advise.

```
DIFF            List the differences between 2 observation headers
FILTER          Apply a filter on the R spectrum
MEDIAN          Compute the median and MAD of the R spectrum
RMS             Compute the RMS of the index
SUBTRACT        Compute T-R
UNBLANK         Remove blanks in a Class TABLE
VARIABLE        Create sic variables mapping the observation(s)
```

```
UV_ZERO          Create a zero spacing UV table
WAVELET          Decompose the R spectrum into wavelets
```

5.6.2 DIFF

```
EXPERIMENTAL\DIFF [Num1 Num2]
```

List the differences between 2 observation headers (the data is not diff'ed). Without argument, the command compares the R and T buffers. If 2 observation numbers are given, the corresponding entries (latest version) from the current input file are compared.

As of today, the command compares the following sections:
- General (observation number and version ignored),
- Position,
- Spectroscopy,
- Calibration,
- Switching.

The values are compared to all their precision. No tolerance is used.

5.6.3 FILTER

```
EXPERIMENTAL\FILTER [/STORE]
```

Compute the Mean Absolute Deviation of the R spectrum, and store the result in R%HEAD%BAS%SIGFI if the option /STORE is present.

5.6.4 MEDIAN

```
EXPERIMENTAL\MEDIAN [Width] [Sampling]
```

Compute the Median and the Median Absolute Deviation along the R spectrum and save them in the 2 associated arrays named BASELINE and RMS (available in R%ASSOC%BASELINE% and R%ASSOC%RMS% respectively).

These quantities are computed in windows of the given Width (MHz, default 20). Windows are set every Sampling space (MHz, default width/2). Intermediate values are then interpolated, so that the final Associated Arrays are of size Nchan. Using a sampling of 1 channel is possible but at larger computation cost.

For the first and last half windows (boundary conditions), flat values are used (no extrapolation).

If the input channel in RY is blanked, the resulting channels in BASELINE and RMS are also blanked. Blank channels do not contribute to the surrounding windows.

5.6.5 RMS

EXPERIMENTAL\RMS [/RESAMPLE [NX Xref Xval Xinc Unit]] [/NOCHECK [SOURCE|POSITION|LINE|SPECTROSCOPY|CALIBRATION]] [/WEIGHT TIME|SIGMA|EQUAL|ASSOCIATED]

Produce the RMS spectrum of the index.

5.6.6 SUBTRACT

EXPERIMENTAL\SUBTRACT

Compute the difference T-R. Consistency of the input spectroscopic axes is checked. Output header is NOT IMPLEMENTED: T header is reused.

5.6.7 UNBLANK

EXPERIMENTAL\UNBLANK in.tab out.tab [/MODE REJECT]
EXPERIMENTAL\UNBLANK in.tab [out.tab] /MODE NOISE|INTERPOLATE|VALUE [Value]

Patch the blank values from a Class table (i.e. produced by the command ANA\TABLE). This is useful for later use in XY_MAP which does not ignore blanks for efficiency purpose.

4 modes are offered:
- REJECT (default):
  Create a new table from the input one, rejecting spectra which have one or more blank channels. As the table size will differ in the end, this action can not be performed in place.
- VALUE [Value]:
  Create a new table, or update in place if no output table name is provided, patching the blank channels with a custom value (default 0).
- NOISE:
  Create a new table, or update in place if no output table name is provided, patching the blank channels with random noise. The noise level is computed from the weight column found in the table itself (one weight per spectrum).
- INTERPOLATE:
  Create a new table, or update in place if no output table name is provided, patching the blank channels with interpolated values between the previous and next valid channels. At the spectrum boundaries, the values are extrapolated from the two nearest valid channels.

The command supports Velocity-Position (.tab) or Position-Velocity (.bat) tables, except for the REJECT mode. Note that the table(s) must fit in memory.

5.6.8 UV_ZERO

```
EXPERIMENTAL\UV_ZERO Name [/LIKE GDFFile] [/VERBOSE]
```

Create a Zero Spacing UV Table from the R spectrum, using channels in the current SET MODE X range.

The channel weights are set according to the current SET WEIGHT tuning. If weight is EQUAL, the value 1 is used.

/LIKE option can be used to produce a UV table using the same frequency/velocity axis as another reference GDF file (can be a cube or a UV table).

5.6.9 VARIABLE

```
EXPERIMENTAL\VARIABLE Section1 [Section2 ...] [/MODE READ|WRITE|OFF]
[/INDEX]
```

VARIABLE /INDEX will read (from file) and save (in memory) in Sic arrays the named sections of all entries in the current index. The Sic arrays will be available in the structure IDX%HEAD%, under each named section. Any further call to FIND and DROP (i.e. commands which modify the current index) will destroy the Sic arrays. For efficiency purpose, they are not recreated implicitly, the command VARIABLE /INDEX must be called again instead.

VARIABLE USER /INDEX will load the user section for all entries, if the associated hooks to the command are declared in Class. The Sic arrays will be available in the structure IDX%USER%<owner>%<title>%

The option /MODE indicates to create read-only (READ) or read-write (WRITE) variables. The keyword OFF will delete the variables of the named sections and free their associated memory. WRITE is not allowed on /INDEX arrays.

5.6.10 WAVELET

```
EXPERIMENTAL\WAVELET [/BASE [N]] [/PLOT [Ipen]]
```

Compute the wavelets of the R spectrum, and store all the orders in the Sic array named WAVELET.

If the option /BASE N is present, the N-th order is subtracted from the R spectrum (default N is 5). This baseline can be plotted with the option /PLOT (default pen is 1).

Index

ACCUMULATE, 20, 21, 58
    /NOCHECK, 58
    /RESAMPLE, 59
ASSOCIATE, 59
AVERAGE, 20, 21, 60
    /NOCHECK, 60
    /RESAMPLE, 61

BASE, 19, 24, 62
    /PLOT, 19
    LAST, 19
    SINUS, 19
BOX, 18, 63
    /INDEX, 19
    /UNIT, 18

CATALOG, 63
    /SORT, 64
COMMENT, 107
CONSISTENCY, 64
COPY, 64

DECONVOLVE, 139
    /GAIN, 140
    /KEEP, 140
DIFF, 141
DISPLAY, 24, 126
DIVIDE, 24, 107
DRAW, 27, 28, 107
    FILL, 29, 108
    KILL, 28, 108
    LOWER, 28, 108
    MASK, 28, 108
    MOLECULE, 108
    TEXT, 28, 109
    UPPER, 28, 108
    WINDOW, 28, 109
DROP, 64
DUMP, 65

EXTRACT, 65

FFT, 24, 109
    /KILL, 110
    /REMOVE, 110
FILE, 13, 66
    BOTH, 67
    IN, 66
    OUT, 66
    UPDATE, 67
FILL, 110
FILTER, 141
FIND, 14, 16, 67
    /ALL, 15
    /MASK, 69
    /POSITION, 70
    /SWITCHING, 70
    APPEND, 68
    NEW_DATA, 68
    UPDATE, 69
FITS, 71
    READ, 71
    WRITE, 71
FOLD, 20, 72

GET, 13, 72
GET n, 15
GREG, 27, 110

HEADER, 73

IGNORE, 74
INITIALIZE, 139
ITERATE, 24, 127
    /NOCHECK, 127

KEEP, 24, 127

Language, 57, 106, 126, 134, 138, 140
LINES, 22, 127
LIST, 74
    /TOC, 75
LMV, 111

FAXIS, 112
LOAD, 15, 76

MAP, 18, 113
/BASE, 114
/CELL, 18, 114
/GRID, 18, 114
/NOLABEL, 18, 114
/NUMBER, 18, 114
MEDIAN, 141
MEMORIZE, 114
MERGE, 76
METHOD, 21, 129
CONTINUUM, 22
GAUSS, 21
HFS, 22
NH3, 21
SHELL, 21, 22
MINIMIZE, 24, 131
/NOCHECK, 134
CONTINUUM, 133
GAUSSIAN, 132
HFS, 133
SHELL, 132
MODEL, 115
MODIFY, 76
BANDS, 76
BEAM_EFF, 77
BLANKING, 76
DOPPLER, 77
ELEVATION_GAIN, 79
FREQUENCY, 79
IMAGE, 79
LINENAME, 79
OFFSETS, 79
PARALLACTIC_ANGLE, 80
POSITION, 80
PROJECTION, 80
RECENTER, 80
SCALE, 81
SOURCE, 81
SWITCH_MODE, 82
SYSTEM, 81
TELESCOPE, 81
VELOCITY, 81
WIDTH, 81
MULTIPLY, 82

NEW_DATA, 82
NOISE, 24, 116

PLOT, 18, 82
/INDEX, 19
POPUP, 18, 116
PRINT, 26, 117
/OUTPUT, 26
/TABLE, 27
AREA, 26
CHANNEL, 26
FIT, 26, 119
FLUX, 26
MOMENT, 26
POINTING, 26, 119

REDUCE, 26, 120
RESAMPLE, 25, 120
/FFT, 121
/LIKE, 121
SHAPE, 121
RESIDUAL, 24, 134
RESULT, 134
RETRIEVE, 122
RMS, 142

SAVE, 83
SET, 14, 83
ACTION, 83
ALIGN, 20, 84
ANGLE, 14, 84
BAD, 84
BAD Mode, 21
BASE, 19
BASELINE, 84
BOX, 85
CALIBRATION, 21, 85
CHECK, 85
COORDINATE, 14
DEFAULT, 85
ENTRY, 85
EXTENSION, 86
FIND, 86
FORMAT, 18, 86
FREQUENCY, 86
LEVEL, 87
LINE, 14, 87
MAP, 88
MASK, 24, 88

MATCH, 15, 21, 88
MODE, 18, 26, 88
NOCHECK, 89
NOMATCH, 21, 89
NUMBER, 14, 89
OBSERVATORY, 90
OBSERVED, 14, 90
OFFSET, 14
OFFSETS, 90
PLOT, 17, 90
POSANGLE, 91
PROCESSING, 91
QUALITY, 91
RANGE, 14, 91
REDUCED, 14, 92
SCALE, 92
SCAN, 15, 92
SELECTION, 92
SORT, 15, 93
SOURCE, 15, 93
SUBSCAN, 93
SYSTEM, 93
TELESCOPE, 15, 94
TYPE, 15, 25, 94
UNIT, 17, 18, 94
VARIABLE, 16, 17, 94
VELOCITY, 100
WEIGHT, 20, 100
WINDOW, 19, 101
SET EXTENSION .my_extension, 13
SET MODE, 19
SHOW, 102
SMOOTH, 25, 122
AUTO, 25
SPECTRUM, 18, 102
/INDEX, 19
STAMP, 18, 123
/LABEL, 123
STITCH, 103
/NOCHECK, 103
/RESAMPLE, 104
STRIP, 15, 123
SUBTRACT, 142
SWAP, 15, 104

TABLE, 124
/MATH, 125
TAG, 104
TITLE, 18, 105
/INDEX, 19

UNBLANK, 142
UPDATE, 13, 105
UV_ZERO, 143

VARIABLE, 143
VISUALIZE, 24, 134

WAVELET, 143
WRITE, 13, 105

XY_MAP, 135
/NOGRID, 135
/TYPE, 136
MEMORY, 137
TUNING, 136