IRAM Memo 2023-1

Review of the spatial projections support in GILDAS
Compatibility with FITS

S. Bardeau[1], J. Pety[1,2]

1. IRAM (Grenoble)
2. Observatoire de Paris

22-nov-2023
version 1.0

Abstract

This memo reviews the spatial projections implemented in GILDAS and their compatibility with the FITS conventions defined by the AIPS memos and by Calabretta & Greisen (2002). It focuses on the *radio* projection used at the IRAM 30m telescope, explains why its legacy GLS description is obsolete in FITS, and shows how the same data can be represented with the CG02-compliant SFL projection through a header translation only. It also summarizes the related corrections introduced in GILDAS to restore reliable FITS import and export for the affected projections.

Keywords: projection, FITS
Related documents: *GreG documentation*

Contents

1 Description

1.1 IRAM context

At the IRAM 30m telescope, the observations are traditionally performed so that the spectra coordinates, and the resulting gridded cubes created from on-the-fly projects, are expressed as an absolute position on the celestial sphere plus relative offsets projected on a 2D plane using what we call the *radio projection*. This projection is correctly supported within the GILDAS frame (including the GILDAS Data Format), but also when exporting cubes to FITS using the equivalent *GLobal-Sinusoidal (hereafter GLS)* projection inherited from AIPS legacy.

However, recent versions of third-party software (CASA, CARTA) have broken support of the GLS projection in FITS files. This memo analyzes the current FITS standard regarding spatial projections, makes a complete review of the various projections supported in GILDAS, and describes the actions performed to ensure GILDAS users have no problem when exporting or importing their data to/from FITS.

1.2 FITS formalism

The AIPS Memos 27[1] and 46[2] introduced a number of projections and the equations to convert between the 2D projection plane coordinates and 3D spherical celestial coordinates. The Calabretta & Greisen 2002 paper[3] (hereafter CG02) extended the AIPS conventions and introduced a fully generic formalism to describe many kind of projections in the FITS file headers. Given the parameters reproduced in the table 1, the CG02 general idea, as described in their section 2, going from the projected plane to the celestial sphere consists in 3 major steps:

Definition: the 2D projected plane is defined as a regular coordinate system with a reference point (x_0, y_0) located at $(0, 0)$ by design. Its extents might be finite or infinite depending on the projection kind.

Unprojection: one converts from the (x, y) projected coordinates to the unprojected *native* spherical coordinate: (x, y) are converted to (ϕ, θ). Unprojection obviously depends on the projection kind and parameters. The *native* sphere is defined such as the reference point (x_0, y_0) transforms to the sphere fiducial point (ϕ_0, θ_0). This point is typically either on the Equator (coordinates $(0, 0)$ for *e.g.* cylindrical projections) or located on a pole (coordinates $(0, \pm 90°)$ for *e.g.* zenithal projections); this is an intrinsic property of the projection kind.

Euler rotation: then one rotates the native coordinates (ϕ, θ) to celestial coordinates (α, δ) by an Euler rotation of the native sphere. The Euler angles are defined thanks to (ϕ_0, θ_0) matching (α_0, δ_0), plus a possible inclination angle.

The Figure 1 offers a visual representation of these transformations. The opposite conversion (from absolute celestial coordinates to projected plane offsets) is done by reverting these operations.

[1] *Non-linear Coordinate Systems in AIPS* by Eric W. Greisen, May 20th, 1983, revised on January 16th, 1993; link `http://www.aips.nrao.edu/aipsmemo.html`

[2] *Additional Non-linear Coordinates* by Eric W. Greisen, May 20, 1986, revised on January 16, 1993

[3] *Representations of celestial coordinates in FITS*, M. R. Calabretta and E. W. Greisen, A&A 395, 1077-1122 (2002)

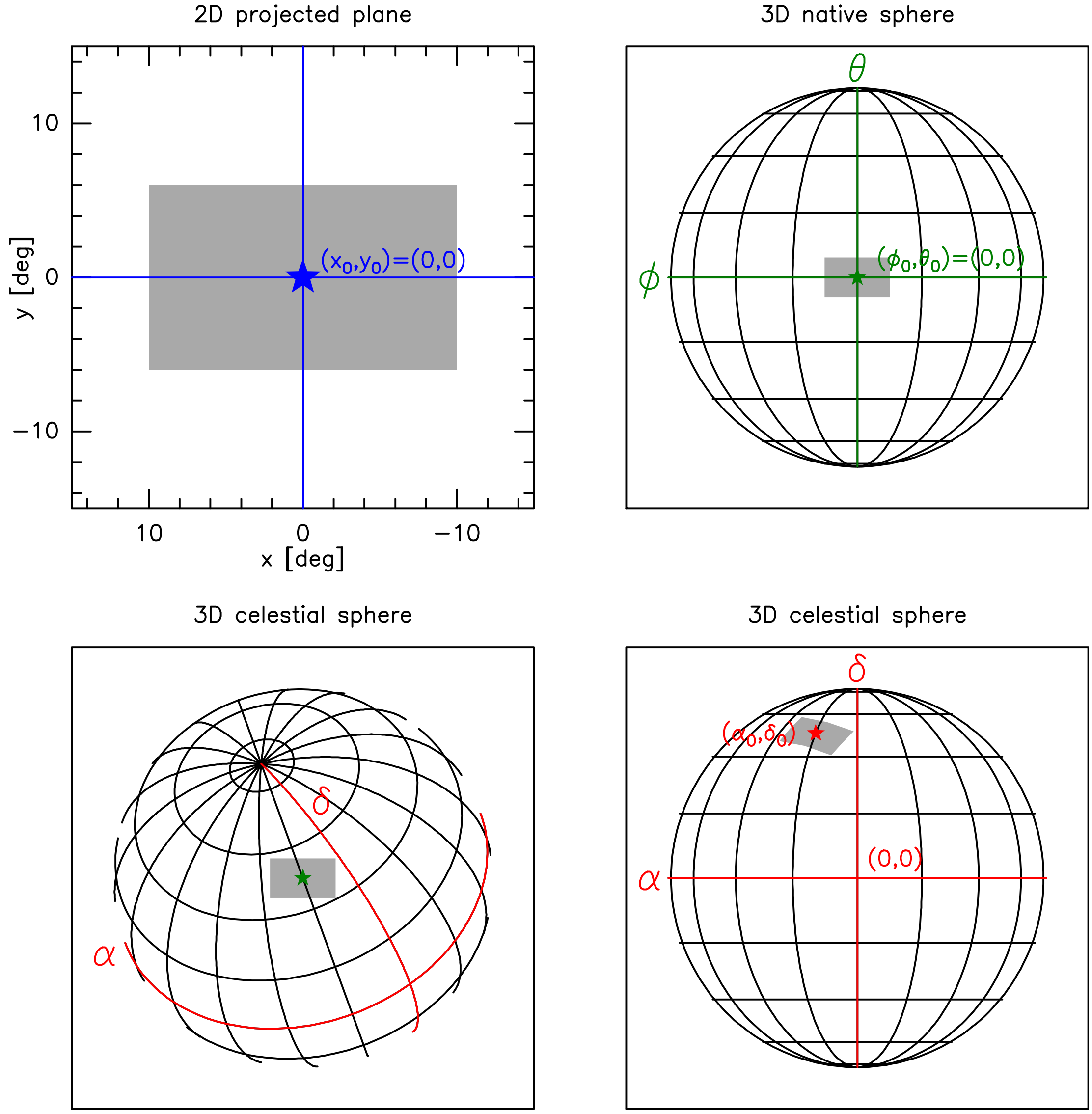

Figure 1: Visual representation of the CG02 formalism. In all the plots, the grey region and the star marker are at the same position on sky. **Top left**: the grey region symbolizes a 20×12 square degrees in the (x, y) 2D projected plane. Blue lines show the abscissa and ordinate origins. For simplicity the region is centered on the reference point, but it could be anywhere allowed in the plane. **Top right**: the same area is unprojected to (ϕ, θ) spherical coordinates on the 3D native sphere (the *unprojection* kind chosen for this example is not detailed here). The equator and zero reference meridian are shown in green. In this example, the fiducial point (ϕ_0, θ_0) has coordinates $(0, 0)$. **Bottom left**: the whole sky orientation is kept the same as before, but the coordinate system represented is now the 3D celestial sphere, with arbitrary reference point and inclination angle for the example. The equator and zero reference meridian are shown in red. **Bottom right**: Same as previous plot but the celestial sphere is rotated and displayed with its origin of coordinates in front. In this process, the coordinates have to be Euler-rotated from (ϕ, θ) to (α, δ) to be expressed in this new system.

Table 1: CG02 coordinate definitions (from their Table 1).

(x, y)	Projection plane coordinates
(ϕ, θ)	Native longitude and latitude
(α, δ)	Celestial longitude and latitude
(x_0, y_0)	Projection plane coordinates of the fiducial point
(ϕ_0, θ_0)	Native longitude and latitude of the fiducial point
(α_0, δ_0)	Celestial longitude and latitude of the fiducial point (`CRVALia`)
(ϕ_p, θ_p)	Native longitude and latitude of the celestial pole (`LONPOLEa`)
(α_p, δ_p)	Celestial longitude and latitude of the native pole

1.3 Compatibility between FITS and GILDAS

The notable projections supported by GILDAS are summarized in the Table 2.

Table 2: List of the projections implemented in GILDAS and subject to export or import to/from FITS data format. The top group was defined in the AIPS memo 27, the middle group in the memo 46, and the bottom group in the CG02 paper.

Name	FITS	GILDAS	Note
Gnomonic	TAN	gnomonic	Full compatibility GILDAS/FITS
Slant orthographic	SIN	orthographic	Basic compatibility (GILDAS does not implement the additional SIN projection parameters)
Zenithal equidistant	ARC	azimuthal	Full compatibility GILDAS/FITS
North Celestial Pole	NCP	ncp	Deprecated in FITS, kept in GILDAS as this would require additional projection parameters of the SIN projection
Stereographic	STG	stereographic	Full compatibility GILDAS/FITS since aug23 (see section 2.5)
Hammer-Aitoff	AIT	aitoff	Compatibility GILDAS/FITS as long as the reference is on the Equator
Global-Sinusoidal	GLS	radio	Compatibility GILDAS/FITS but deprecated in FITS, translated to SFL with translation
Sanson-Flamsteed	SFL	sfl	Full compatibility GILDAS/FITS since oct23 (see section 2.8)

1.3.1 Direct compatibility

A number of projections defined in the AIPS memos are fully compatible with the new CG02 formalism, which allows an immediate compatibility of the GILDAS projections with the FITS files. According to the CG02 section 6.1.1, *The SIN [orthographic] projection defined by Greisen (1983) is here generalized by the addition of projection parameters. However, these parameters assume default values which reduce to the simple orthographic projection of the AIPS convention. Therefore no translation is required.* And in their section 6.1.3, *The TAN [gnomonic], ARC [azimuthal] and STG [stereographic] projections defined by Greisen (1983, 1986) are directly equivalent to those defined here and no translation is required.*

1.3.2 Incompatibilities

On the other hand, CG02 section 6.1.4 stress that *the AIPS convention cannot represent oblique celestial coordinate graticules such as the one shown in [their] Fig. 2.* `CRVALi` *for these projections [(AIT/Aitoff, GLS/radio, MER)] in AIPS does not correspond to the celestial coordinates* (α_0, δ_0) *of the reference point, as understood in this formalism, unless they are both zero in which case no translation is required.* In other words, these 3 projections do not fit into the CG02 general formalism (unless the reference point is on the Equator). See next subsections for details.

1.3.3 The GLS/radio projection in details

The AIPS memo 46 defines the *global-sinusoidal* projection (GLS) in its section 2.3, which writes as

$$x = (\alpha - \alpha_0) \cos\delta, \quad (1)$$
$$y = \delta - \delta_0. \quad (2)$$

In this case, there is no rotation of the 2D plane compared to the celestial system (*e.g.* ICRS). (α, δ) are the celestial coordinates, and (α_0, δ_0) the coordinates of the reference point. This can not be described in the CG02 formalism, which makes this projection obsolete for writing in new FITS files.

Fortunately, the GLS projection at any declination and no rotation is equivalent to a CG02 compliant Sanson-Flamsteed (SFL) projection at zero declination (*i.e.* on the Equator)[4]. It is therefore possible to recast a cube in GLS projection as a cube in SFL projection just by patching the header and without any change on the data! The demonstration and the exact translation to be performed are detailed in the appendix section A.3. The Fig. 2 gives a visual representation of these transformations.

1.3.4 The Hammer-Aitoff case

The Hammer-Aitoff projection (hereafter AIT) is similar to the GLS case, *i.e.* it is possible to convert from the old (AIPS) to the new (CG02) formalism by converting the projection description. But, while the GLS projection is known to have been used with non-zero declination reference (hence a new name for the projection), it is less obvious for AIT. Knowing that both AIPS-AIT and FITS-AIT formalisms are compatible at $\delta_0 = 0$, CG02 have chosen to keep the same name for the new convention.

2 GILDAS implementation

GILDAS implements a variety of projections. They are in place since more than 30 years and used by the community. Here we compare them to the AIPS memos and the CG02 paper, in order to check their compatibility with the current FITS standard.

[4]Quoting CG02: *A translation into the new formalism exists for non-zero* `CRVALi` *but only if* `CROTAi` *is zero. It consists of setting* `CRVALi` *to zero and adjusting* `CRPIXj` *and* `CDELTi` *accordingly in the AIPS header whereupon the above situation is obtained. The corrections to* `CRPIXj` *are obtained by computing the pixel coordinates of* $(\alpha, \delta) = (0, 0)$ *within the AIPS convention. [...] Of the three projections only GLS is known to have been used with non-zero* `CRVALi`*. Consequently we have renamed it as SFL as a warning that translation is required.* CG02 chose the name *Sanson-Flamsteed* (SFL) for the new formalism because GLS has been used with non-zero declination, and these cases have no general equivalent in the new formalism.

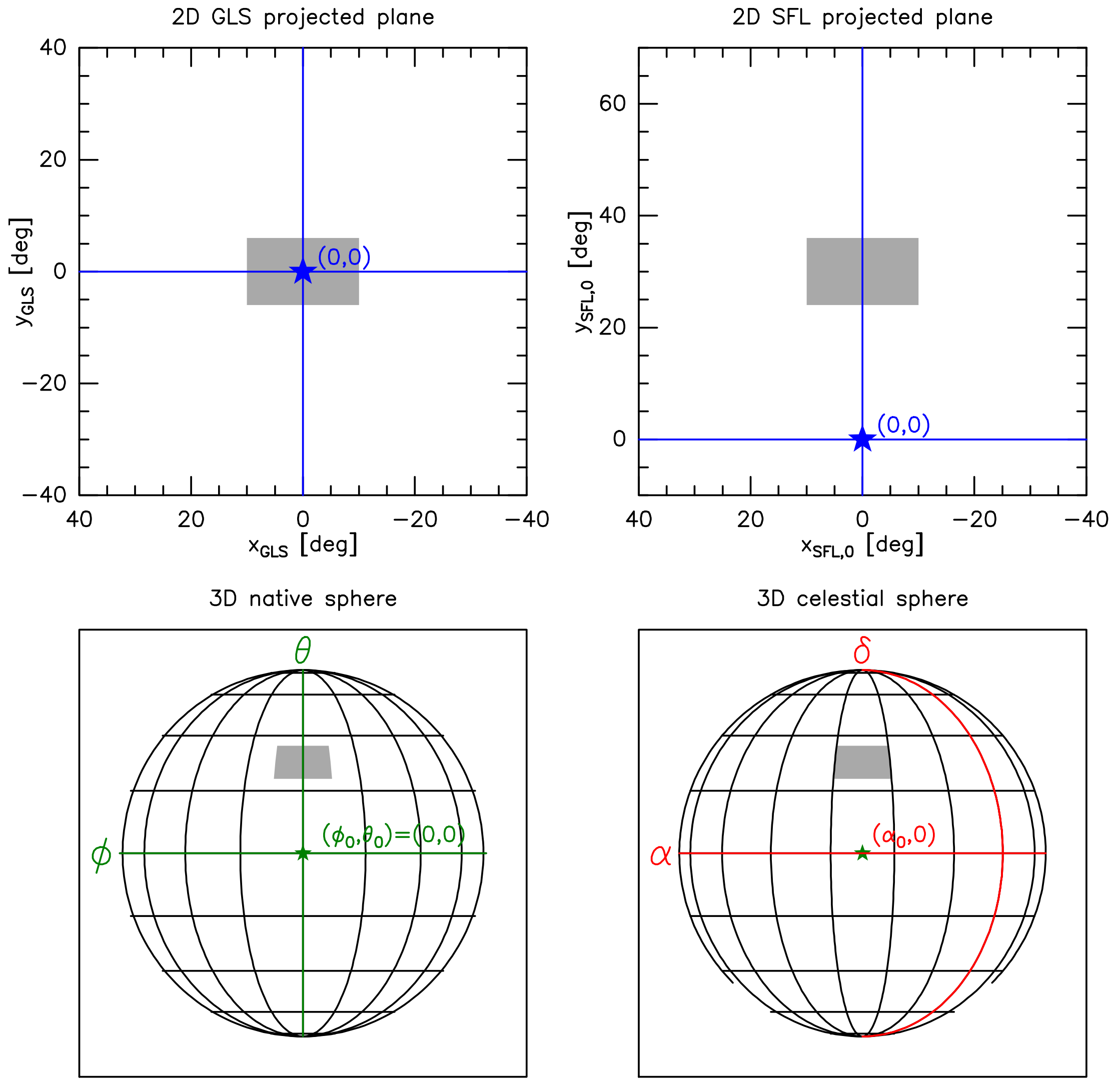

Figure 2: Visual representation of the GLS-to-SFL transformation. In all the plots, the grey region covers the same area on sky. **Top left**: the grey region symbolizes a 20×12 square degrees in the (x, y) 2D AIPS-GLS projected plane. Blue lines show the abscissa and ordinate origins. There is no transformation possible to go to a native sphere in the CG02 formalism. **Top right**: the reference point is shifted to the Equator, *i.e.* by a δ_0 shift (30° in this example). The equations now match the FITS-SFL formalism at $\delta_0 = 0$. **Bottom left**: the area is SFL-unprojected to (ϕ, θ) spherical coordinates on the 3D native sphere. The equator and zero reference meridian are shown in green. The fiducial point (ϕ_0, θ_0) of the SFL projection has coordinates $(0, 0)$ by design. **Bottom right**: the whole sky orientation is kept the same as before, but the coordinate system represented is now the 3D celestial sphere. The equator and zero reference meridian are shown in red. α_0 is 50° in this example.

2.1 Gnomonic projection

GILDAS implementation is fully compatible with the AIPS-TAN convention, hence also with the CG02 one.

2.2 Orthographic projection

The CG02 paper has augmented the definition of the SIN projection with more parameters in order to describe more projections, remaining compatible with the AIPS-SIN by default. GILDAS implementation is only compatible with the AIPS-SIN convention, and can not describe the augmented FITS-SIN projection.

2.3 Azimuthal projection

GILDAS implementation is fully compatible with the AIPS-ARC convention, hence also with the CG02 one.

2.4 North Celestial Pole projection

GILDAS implementation is fully compatible with the AIPS-NCP convention. NCP is deprecated in the CG02 formalism and should be converted to an augmented SIN projection[5], but since GILDAS is not able to describe an augmented SIN, NCP is kept in GILDAS.

2.5 Stereographic projection

Up to the release jul23, the GILDAS implementation was:

$$z = \sin\delta\cos\delta_0 - \cos\delta\sin\delta_0\cos(\alpha-\alpha_0) \tag{3}$$
$$\theta = \cos^{-1}((\sin\delta - z\cos\delta_0)/\sin\delta_0) \tag{4}$$
$$r = \tan\frac{\theta}{2} \tag{5}$$
$$x = r\sin p \tag{6}$$
$$y = r\cos p \tag{7}$$

where (α, δ) are the celestial coordinates and (x, y) the projection plane coordinates. z is an intermediate quantity. r is the distance from the reference point in the projection plane, and p the inclination angle (not detailed here). After a few computations one can show that the θ angle is compatible with the AIPS definition (AIPS memo 46 Eq. 3):

$$\cos\theta = \sin\delta\sin\delta_0 + \cos\delta\cos\delta_0\cos\Delta\alpha \tag{8}$$

But, on the other hand, the coordinate modulus in the AIPS convention is (AIPS memo 46 section 2.1.2):

$$r_{\mathrm{AIPS}} = 2\frac{\sin\theta}{1+\cos\theta} \tag{9}$$

which can be written as

$$r_{\mathrm{AIPS}} = 2\tan\frac{\theta}{2} \tag{10}$$

[5] see CG02 section 6.1.2

thanks to traditional trigonometric identities [6], *i.e.* twice the r modulus used in GILDAS. In other words, the spatial scaling in the GILDAS and AIPS conventions differ by a factor 2. Of these two definitions, the GILDAS convention was the unsatisfying one because angles from the reference point are projected as twice less distance on the projection plane ($\tan(\theta/2) \simeq \theta/2$ for small angles, *i.e.* $r \simeq \theta/2$ for the GILDAS definition, while one would prefer $r \simeq \theta$ for correct distance measurement in the projected plane). This has been considered as a bug in the implementation and this was fixed with the appropriate factor in the aug23 GILDAS release. Doing so, the stereographic projection implementation in GILDAS is now fully compatible with the AIPS one, and thus to the CG02 one.

2.6 Aitoff projection

GILDAS implements the AIT projection only for $\delta_0 = 0$; non-zero reference declinations are always ignored when projecting or unprojecting positions, even if the data format usually saves any declination value. This implementation is compatible[7] with the AIPS (and thus the CG02) convention *in the* $\delta_0 = 0$ *case.*

Note, however, that when data were exported to FITS files, GILDAS versions earlier than aug23 were using the possibly non-zero declinations as the `CRVAL2` value. This is incorrect and was leading to wrong interpretation of the maps by third-party software. This is fixed in the aug23 release by always setting `CRVAL2` parameter to 0 for the AIT projection.

If, at some point, the non-zero declination case is to be implemented in GILDAS, we should choose the newest formalism, *i.e.* the CG02 one[8]

2.7 Radio projection

The GILDAS kernel implements the so-called *radio* projection with the same formalism as AIPS GLS (Eqs. 1 and 2). In particular, this projection system is used natively at the IRAM-30m telescope and is preserved to describe the position of pointed spectra, or on-the-fly observations and the resulting reconstructed position-position-velocity cubes. Up to GILDAS version jul23, the radio-projection description was exported under the `-GLS` suffix in FITS header files (GILDAS command `V\FITS output.fits FROM input.gdf`).

However, it appeared that this could cause issues when the exported FITS are interpreted by some third-party software. Many FITS-based software implement an implicit GLS-to-SFL conversion as suggested by CG02. However, some seem to implement it incorrectly (the translation is not correctly applied), while some other just fail in this case (CASA, CARTA). Because of these various troubles, it is decided not to export any more FITS data out of GILDAS with the radio/GLS description.

2.7.1 Radio-projected cubes

The cubes using a radio-projection in GILDAS typically look like:

```
File : 13co10.lmv                                                    REAL*4
Size          Reference Pixel           Value                  Increment
```

[6] $\tan x = \frac{\sin x}{\cos x} = \frac{2\sin x\cos x}{2\cos^2 x} = \frac{\sin 2x}{1+\cos 2x}$

[7] The formulae have not been checked, but the maps compatibility with other software is confirmed.

[8] Since the GILDAS data format used to save any declination value while the projection was defined for $\delta_0 = 0$, we should then introduce a new internal projection code to distinguish with the backward compatible case where the reference declination is ignored.

```
       61   44.5561196503335       0.00000000000000     -5.411193342297338E-05
      101   55.4797321532119       0.00000000000000      5.411193342297338E-05
      921   578.812988281250       2.50000000000000     -0.106259830296040
Source name         NGC7023
Axis type           RA          DEC          VELOCITY
Coordinate system   EQUATORIAL         Velocity    LSR
Right Ascension   21:01:32.24000       Declination       68:10:29.5500
Equinox            2000.0000
Projection type     RADIO              Angle     0.000000000000000
Axis 1     A0     21:01:32.24000       Axis 2     D0     68:10:29.5500
[...]
```

Note the RADIO projection type. Starting from GILDAS version aug23, this kind of header will be implicitely exported with the SFL projection by the V\FITS command. The resulting header will be:

```
NAXIS   =                    3         /
NAXIS1  =                   61         /
NAXIS2  =                  101         /
NAXIS3  =                  921         /
CTYPE1  = 'RA---SFL    '               /
CRVAL1  =  0.3153843333333E+03         /
CDELT1  = -0.3100385406429E-02         /
CRPIX1  =  0.4455611965033E+02         /
CUNIT1  = 'deg         '               /
CTYPE2  = 'DEC--SFL    '               /
CRVAL2  =  0.0000000000000E+00         /
CDELT2  =  0.3100385406429E-02         /
CRPIX2  = -0.2193368163425E+05         /
CUNIT2  = 'deg         '               /
OBJECT  = 'NGC7023     '               /
RADESYS = 'FK5'                        / Coordinate system
RA      =  0.3153843333333E+03         / Right Ascension
DEC     =  0.6817487500002E+02         / Declination
[...]
```

Note 1) the -SFL suffix on the spatial axes, 2) the null CRVAL2 value, 3) the reference pixel shifted far away, to the Equator (CRPIX2 is about −22000 pixels). Note also that the indicative source position (RA,DEC) is preserved.

2.7.2 30m CLASS cubes

In order to avoid changing the projection once the date were gridded, we modified it one step earlier in the data processing. In practice, the XY_MAP command in CLASS (which grids a collection of spectra into a cube) will now implicitly convert radio-projected spectra positions into azimuthal projection *before* gridding. Other projections are left unchanged. The user can not customize the projection of the output cube in XY_MAP, but he can still do this another step earlier by reprojecting (MODIFY PROJECTION) all the individual spectra in the .30m file.

Note that NOEMA cubes are not concerned as they are natively constructed with the azimuthal projection.

2.8 SFL projection

Since the dec07 GILDAS release, the SFL projection was implemented as:

$$x = (\alpha - \alpha_0) \times cos(\delta - \delta_0) \tag{11}$$

$$y = \delta - \delta_0 \tag{12}$$

This convention is very near to the radio/GLS one, except x depends on the cosine of $\delta - \delta_0$, not δ itself. Unfortunately, this convention does not match the GC02 one, in particular the stage 2 described in section 1 where the native sphere is Euler-rotated to the celestial sphere. This is not as simple as the simple δ shift in the above formula.

Choice is made to discard the above implementation and go for the CG02 one, without changing the GILDAS name *SFL* and the associated internal identifier for this projection. This assumes this projection was seldom used on the GILDAS side (import/export from/to FITS was anyway not available). This may break the interpretation of old GILDAS cubes using this projection, if any, but introduces full compatibility with FITS-SFL files. The implementation is detailed in Appendix A and is done since GILDAS release oct23.

A New SFL implementation in GILDAS

This appendix is based on the formalism of the CG02 paper, applied to the particular case of the SFL projection. See Table 1 for the coordinates involved here. According to the CG02 section 5.3, *the pseudocylindricals are constructed with the native coordinate system origin at the reference point. Accordingly we set* (their eq. 89):

$$(\phi_0, \theta_0) = (0, 0) \tag{13}$$

The native longitude of the celestial pole ϕ_p is defined as (their section 2.2):

$$\phi_p = \begin{cases} 0^\circ & \text{for } \delta_0 \geq 0 \\ 180^\circ & \text{for } \delta_0 < 0 \end{cases} \tag{14}$$

This implies $\sin \phi_p = 0$. Their eq. 8 is simplified as:

$$\delta_{p,1} = \operatorname{atan2}(0, \cos \phi_p) + \cos^{-1}(\sin \delta_0) \tag{15}$$

$$\delta_{p,2} = \operatorname{atan2}(0, \cos \phi_p) - \cos^{-1}(\sin \delta_0) \tag{16}$$

where $\operatorname{atan2}(y, x)$ is the inverse tangent function that returns the correct quadrant. $\delta_{p,1}$ and $\delta_{p,2}$ are two solutions for δ_p. The choice between the two is ruled by several conditions detailed in the CG02 paper section 2.4. The rule 4 states that the northernly solution is the default if the native latitude of the celestial pole (θ_p, `LATPOLEa` in FITS headers) is not defined, which we implement as:

$$\delta_p = \begin{cases} \delta_{p,1} & \text{if } \delta_{p,1} > 0 \text{ and } \delta_{p,1} \leq 90^\circ \\ \delta_{p,2} & \text{otherwise.} \end{cases} \tag{17}$$

If $\delta_p = \pm 90^\circ$, their rule 2 applies to determine α_p, otherwise their generic equations 9 and 10 are used:

$$\alpha_p = \begin{cases} \alpha_0 + \phi_p - 180^\circ & \text{for } \delta_p = +90^\circ \\ \alpha_0 - \phi_p & \text{for } \delta_p = -90^\circ \\ \alpha_0 - \operatorname{atan2}(0, -\frac{\sin \delta_p \sin \delta_0}{\cos \delta_p \cos \delta_0}) & \text{otherwise.} \end{cases} \tag{18}$$

A.1 From projection plane to celestial coordinates

(x, y) projection planes coordinates can be converted to the absolute celestial coordinates (α, δ) as follows. The CG02 equations 92 and 93 for the SFL projection are written:

$$\phi = \frac{x}{\cos y} \tag{19}$$

$$\theta = y \tag{20}$$

Using their eq. 2 with the parameters computed before, the coordinates are:

$$\alpha = \alpha_p + \operatorname{atan2}(-\cos \theta \sin(\phi - \phi_p), \sin \theta \cos \delta_p - \cos \theta \sin \delta_p \cos(\phi - \phi_p)) \tag{21}$$

$$\delta = \sin^{-1}(\sin \theta \sin \delta_p + \cos \theta \cos \delta_p \cos(\phi - \phi_p)) \tag{22}$$

A.2 From celestial to projection plane coordinates

The reverse operation is done with the generic Euler rotation (their equation 5)

$$\phi = \phi_p + \text{atan2}(-\cos\delta\sin(\alpha-\alpha_p), \sin\delta\cos\delta_p - \cos\delta\sin\delta_p\cos(\alpha-\alpha_p)) \quad (23)$$
$$\theta = \sin^{-1}(\sin\delta\sin\delta_p + \cos\delta\cos\delta_p\cos(\alpha-\alpha_p)) \quad (24)$$

and the projection specific to SFL (their equations 90 and 91):

$$x = \phi\cos\theta \quad (25)$$
$$y = \theta \quad (26)$$

A.3 Special case of SFL with reference on the Equator

The case where the reference point is chosen on the Equator ($\delta_0 = 0$) is interesting because of the conversion from GLS to SFL suggested by CG02 and described in section 1. Injected in the previous formulas and rules, this gives:

$$\phi_p = 0^\circ \quad (27)$$
$$\delta_p = +90^\circ \quad (28)$$
$$\alpha_p = \alpha_0 - 180^\circ \quad (29)$$

The $\phi_{\text{SFL},0}$ longitude simplifies as:

$$\phi_{\text{SFL},0} = \text{atan2}(-\cos\delta\sin(\alpha-\alpha_0-180^\circ), -\cos\delta\cos(\alpha-\alpha_0-180^\circ)) = \alpha-\alpha_0 \quad (30)$$

because $\text{atan2}(-y,-x) = \text{atan2}(y,x)+180^\circ$ (opposite quadrant). Similarly the $\theta_{\text{SFL},0}$ latitude is in this case:

$$\theta_{\text{SFL},0} = \sin^{-1}(\sin\delta) = \delta \quad (31)$$

Finally:

$$x_{\text{SFL},0} = (\alpha-\alpha_0)\cos\delta \quad (32)$$
$$y_{\text{SFL},0} = \delta \quad (33)$$

In this case (x, y) have almost the same definition as the GLS definition provided in Eqs. 1 and 2. If we write a y-shifted version of the GLS equations as:

$$x'_{\text{GLS}} = x_{\text{GLS}} = (\alpha-\alpha_0)\cos\delta \quad (34)$$
$$y'_{\text{GLS}} = y_{\text{GLS}} + \delta_0 = \delta \quad (35)$$

i.e. $(x'_{\text{GLS}}, y'_{\text{GLS}}) = (x_{\text{SFL},0}, y_{\text{SFL},0})$. This shows we can write the GLS projection at any declination as a SFL projection at zero declination with the relationship $x_{\text{GLS}} = x_{\text{SFL},0}$ and $y_{\text{GLS}} = y_{\text{SFL},0} - \delta_0$.

B Consistency checks

Here we compare the behaviour of GILDAS and other astronomical software regarding GLS and SFL projections. Comparisons are done using the GREG `CONVERT` command, `astropy.wcs` Python module version 5.2.2, and the FITS graphical viewer `ds9` version 8.2.1. The test data is created as a large scale map, with all values blanked except 3 single pixels for easy identification:

$(3600, -1800), (3600, +1800)$ and $(-200, 0)$ arcseconds relative offsets on the projection plane, and checking what the various software return as corresponding absolute celestial position on sky. Note that the ambiguity is about 1 pixel (5 arcsec in the test data) which is good enough to detect problems on such large maps.

B.1 Support for TAN / gnomonic projection

This projection is used to confirm the 3 software behave the same on a friendly/non-ambiguous projection, and serves to give correct absolute coordinates to be compared with the other consistently-reprojected data.

B.2 Support for GLS / radio projection

- `astropy.wcs` and `G\CONVERT` are consistent and compatible with the gnomonic absolute coordinates. Note that `astropy.wcs` claims the projection is SFL, so this means it implicitly translated the reference pixel as requested by CG02.
- `ds9` disagree and gives improper coordinates. It seems it acts as if the projection was SFL without translating the reference (which is obviously wrong).

B.3 Support for SFL (properly shifted) projection

As explained in the section A.3, the header is consistently modified to use a SFL projection with the reference pixel shifted on the Equator, and data does not need to be reprojected. All the 3 software are consistent.

B.4 Support for SFL (no shift) projection

In this case, only the projection kind is forced to SFL instead of GLS/radio, but not the reference. Data is not modified. We do not expect the absolute positions of the pixels to be correct, but we can compare if the 3 software consistently return the same positions for their generic SFL implementation at $\delta_0 \neq 0$.

- `ds9` and `astropy.wcs` are consistent: they have the same interpretation of the header,
- `G\CONVERT` in GILDAS version jul23 disagree,
- after reimplementation of the SFL projection, `G\CONVERT` in GILDAS version oct23 is consistent with the 2 other ones.