A&A manuscript no.
(will be inserted by hand later)

Your thesaurus codes are:
03 (03.13.2 03.20.1 04.01.1 05.01.1)

ASTRONOMY
AND
ASTROPHYSICS
9.9.1996

Representations of celestial coordinates in FITS

Eric W. Greisen(1) **and Mark Calabretta**(2)

[1] National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, VA, USA 22903-2475
[2] Australia Telescope National Facility, P. O. Box 76, Epping, NSW 2121 Australia

September 9, 1996

Abstract. The initial descriptions of the FITS format provided a simplified method for describing the physical coordinate values of the image pixels, but deliberately did not specify any of the detailed conventions required to convey the complexities of actual image projections. Building on conventions in wide use within astronomy, this paper proposes changes to the simple methods for describing coordinates and proposes detailed conventions for describing most of the methods by which spherical coordinates may be projected onto a two-dimensional plane. Simple methods for converting from the existing coordinate conventions are described. This paper does not attempt to address the complex questions related to frequency/velocity coordinates and time systems, nor does it directly address other kinds of coordinates. It does, however, provide a clear framework and pattern through which agreements in these areas may someday be achieved.

Key words: methods: data analysis — astrometry

1. Introduction

The Flexible Image Transport System, or FITS format, was first described by Wells, Greisen, and Harten (1981). This format is characterized by a fixed logical record length of 2880 bytes, and the use of an unlimited number of character-format "header" records with an 80-byte, keyword-equals-value substructure. The header is followed by the header-specified number of binary data records, which are optionally followed by extension records of the specified length, but, at that time, of unspecified format. Since then, a number of authors have suggested various types of extensions (*e.g.*, Greisen and Harten (1981), Grosbøl, *et al.* (1988) and Harten, *et al.* (1988)). Because of its great flexibility, the FITS format has been, and continues to be, very widely used in astronomy. In fact, the FITS tape format was recommended (resolution C1) for use by all observatories by Commission 5 at the 1982 meeting of the IAU at Patras (1983) and the General Assembly of the IAU adopted (resolution R11) the recommendations of its commissions, including the FITS resolution.

The original paper anticipated the need to specify the coordinates to be attached to each pixel. It viewed each axis of the n-dimensional image as having a coordinate type and a reference point for which the pixel coordinate, a coordinate value, and an increment were given. Note that this reference point was not required to occur at integer pixel locations nor even to occur within the image. An undefined "rotation" parameter was also provided for each axis. Since there are, in general, more coordinates to be attached to a pixel than there are "real" axes in the n-dimensional image, the convention of declaring axes with a single pixel was also established in both examples given by Wells, *et al.* The keywords defined were

`CRVAL`n	coordinate value at reference point
`CRPIX`n	array location of the reference point in pixels
`CDELT`n	coordinate increment at reference point
`CTYPE`n	axis type (8 characters)
`CROTA`n	rotation from stated coordinate type

A list of suggested values for `CTYPE`n was provided with few of the details actually required to specify coordinates. The units were specified to be The International System of Units "SI" (meters, grams, seconds) with the addition of degrees for angles.

The simplicity of this initial description was deliberate. It was felt that a detailed specification of coordinate types was a lengthy and complicated business, well beyond the scope intended for the initial paper. In addition, the authors felt that a detailed specification would probably be somewhat controversial and thus likely to compromise the possibility of wide-spread agreement on, and use of, the basic structures of the format. Hindsight also suggests that we were rather naive at the time concerning coordinates and it is fortunate that the detailed specification was postponed until greater experience could be obtained.

While participating in the development of the AIPS software package of the National Radio Astronomy Ob-

Send offprint requests to: E. Greisen

servatory, Greisen (1983 and 1986) found it necessary to supply additional details to the coordinate definitions for both velocity and celestial coordinates. The latter have been widely used for imagery from the VLA and a number of other instruments (Moshir, *et. al*, 1992 for IRAS). Because of this wide usage, we require that any future specification support the AIPS description of coordinates wherever possible. Greisen (1983) assumed that there is no skew in the image coordinates and that the only rotation applied to the celestial coordinate pair. For that pair, he introduced left-handed, linear coordinates in the tangent plane, which are given by

$$\begin{aligned} x &= (i - \mathtt{CRPIX}i)\,\mathtt{CDELT}i \\ y &= (j - \mathtt{CRPIX}j)\,\mathtt{CDELT}j, \end{aligned} \tag{1}$$

where i, j are pixel numbers in the relevant two axes. The pixel numbers are counted from one through NAXISi and NAXISj, respectively, with the location of each pixel taken to be its center. Therefore, the actual image area runs from $(i,j) = (0.5, 0.5)$ to $(i,j) = (\mathtt{NAXIS}i + 0.5, \mathtt{NAXIS}j + 0.5)$. The direction cosines, (L, M) are then given by

$$\begin{aligned} L &= x\cos\rho - y\sin\rho \\ M &= y\cos\rho + x\sin\rho\,, \end{aligned} \tag{2}$$

where L is in the direction of constant latitude and M is in the direction of constant longitude. The rotation ρ is given, by convention, in the CROTAn parameter for the latitude-like axis. Note that these equations define ρ as the angle from the $+y$ axis to the $+M$ axis in the direction of the $+x$ axis, *i.e.*, counter-clockwise for left-handed systems and clockwise for right-handed systems.

The type of longitude-latitude was specified by the first four characters of CTYPEn as 'RA--' and 'DEC-' for equatorial coordinates, 'GLON' and 'GLAT' for Galactic coordinates, and 'ELON' and 'ELAT' for ecliptic coordinates. The reference point was taken to be the tangent point and the reference point coordinate values were taken to be the longitude and latitude in the specified system. The type of projection to the tangent plane was specified in the second four characters of CTYPEn. The projections in the first paper were '-SIN' for the orthographic projection commonly used in radio aperture synthesis, '-TAN' for the gnomonic projection commonly used in optical telescopes, '-ARC' for the zenithal equidistant projection approximately used by Schmidt telescopes, and '-NCP' for the orthographic projection used by east-west radio interferometers. The second paper provided descriptions for the stereographic, "global-sinusoidal", Hammer-Aitoff, and Mercator geometries. These descriptions were useful, but the last two employed ad-hoc meanings for reference values other than the coordinate origin. These ad-hoc meanings will be replaced by more general interpretations to be given later in this paper.

The descriptions of coordinates in the initial FITS paper are simply inadequate. They provide no description of the meaning of the physical coordinates and suggest a rather incomplete list of coordinate types. The use of a single rotation per axis cannot describe any general rotation of more than two axes. Furthermore, there is no provision for skew. The AIPS specifications are better, although there is still no skew and only a single kind of rotation. Furthermore, AIPS attempted to solve the problem of going from (x, y) to latitude and longitude in a single set of equations for each projection. This led to a limited view of coordinates which this paper will attempt to generalize. In particular, a proper separation of the steps in the computations will yield a clear meaning for the reference point in all coordinate types.

2. Basic concepts

2.1. Coordinate definition and computation

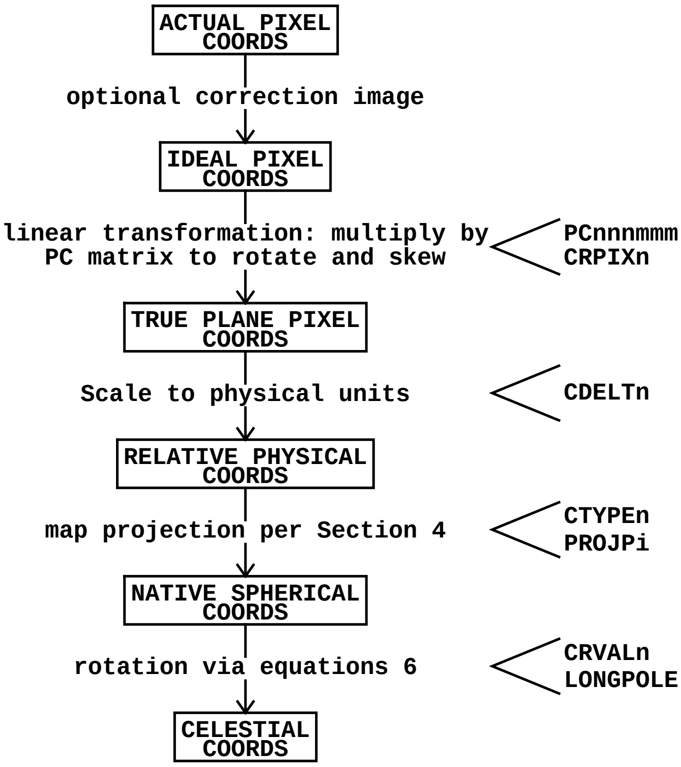

Fig. 1. Conversion of pixel to celestial spherical coordinates

In the current proposal, we regard the conversion from simple pixel counts to a full coordinate description as a multi-step process containing one optional and four required steps. These steps are indicated conceptually in Fig. 1. The first and optional step is used to correct the actual image pixel numbers into those which would have been recorded by an ideal instrument. This correction will be driven by an optional "pixel regularization image" described in Appendix A. The corrections in this image are intended to be sufficiently small and sufficiently smoothly varying across the image as to be negligible except for high-precision computations. Software systems may choose to ignore such images, when present, without making serious errors for most uses. Thus, the corrections should never be used to describe one standard non-linear coordinate as another more "desirable" one "(with corrections)." Nor should these corrections be used to make the sort of corrections readily handled by scaling, rotation and skew, which are to be handled with the `PC` matrix described next. Furthermore, the presence of this correction image within the FITS conventions should not be construed as discouraging instrument teams from designing more ideal instruments or instruments whose data may be regridded without serious loss of information into ideal form before transmission away from the instrument site.

For all types of coordinates, the second and third (first and second required) steps are a *linear* matrix multiplication to convert from pixel numbers $i, j, k, \ldots$ to offsets from the reference point along physical axes but still in pixel units and then to convert from pixel units on these axes to relative physical units. Thus,

$$\begin{pmatrix} x \\ y \\ z \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathtt{CDELT1} & 0 & 0 & \ldots \\ 0 & \mathtt{CDELT2} & 0 & \ldots \\ 0 & 0 & \mathtt{CDELT3} & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \times \begin{pmatrix} \mathtt{PC001001} & \mathtt{PC001002} & \mathtt{PC001003} & \ldots \\ \mathtt{PC002001} & \mathtt{PC002002} & \mathtt{PC002003} & \ldots \\ \mathtt{PC003001} & \mathtt{PC003002} & \mathtt{PC003003} & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \times \begin{pmatrix} i - i_0 \\ j - j_0 \\ k - k_0 \\ \vdots \end{pmatrix}, \quad (3)$$

where $i_0, j_0, k_0, \ldots$ are the pixel coordinates of the reference point (given by the `CRPIX`n), `PC`$nnnmmm$ is a pixel coordinate translation matrix, where nnn and mmm are the axis numbers with leading zeros as needed to make three decimal digits each, `CDELT`n is a diagonal matrix to convert from pixel to physical coordinates, and $(x, y, z, \ldots)$ are relative coordinates in physical units. The default values for `PC`$nnnmmm$ shall be 1.0 for $nnn = mmm$ and 0.0 for $nnn \neq mmm$. There is no need for additional offsets to $(x, y, z, \ldots)$ since, by definition, they are all zero at the reference point. Note that integer pixel numbers refer to the center of the pixel in each axis, so that, for example, the first pixel runs from pixel number 0.5 to pixel number 1.5 on every axis. Note also that the reference point location need not be integer nor need it even occur within the image. The original FITS paper (Wells, *et al.*, 1981) defined the pixel numbers to be counted from 1 to `NAXIS`n (≥ 1) on each axis in a Fortran-like order as presented in the FITS image.[1]

This is a very general proposal! The full `PC`$nnnmmm$ matrix allows for skew and fully general rotations. The reader should note that this allows dissimilar axes to be rotated into one another. This is meaningful in imaging, for example, a spectral-line cube where one may wish to re-sample the cube from some special viewing angle in the three-space of two celestial coordinates and one frequency coordinate. Such rotations are, however, forbidden into axes whose coordinate values are, by convention, only integral. The `STOKES` axis is one such coordinate. The `PC`$nnnmmm$ matrix could also be used to represent images which have been transposed, *e.g.*,

$$\mathtt{PC} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} .$$

This is a legal usage, but strongly discouraged. In this example, the FITS user will read in the header that the first axis is `CTYPE1`, whereas, in fact, the most rapidly varying coordinate is of type `CTYPE2`. Note that keywords `NAXIS1`, `CRPIX1`, `PC`iii`001`, and `PC001`jjj, for example, all refer to the first *pixel* axis in the image, while `CTYPE1`, `CRVAL1`, and `CDELT1` all refer to the first physical ("x") axis. They must produce a correct result when Eq. 3 is applied. Thus, x is of type `CTYPE1` even if it does not change with i (to use the nomenclature of Eq. 3). Therefore, it is good form to transpose the header parameters along with the image so that the on-diagonal terms in the `PC` matrix predominate. Furthermore, we recommend strongly that the *sign* of the coordinate increment be represented in the `CDELT`n keywords rather than in the `PC` matrix. If the `PC` matrix is essentially diagonal and positive, then the human reader of the FITS header will have a better chance of understanding the coordinate representation. Good form in FITS always extends to include the human as well as the software readers.

Two other forms of the `PC`$nnnmmm$ keyword, labeled `CD`n_m and `CD`$nnnmmm$, have also been used and should be recognized as the product of the `PC`$nnnmmm$ matrix

---

[1] The authors are well aware that this convention differs from the usual practice in computer graphics where the pixels are counted from zero with pixel centers as half integers (*e.g.*, Adobe Systems, Inc., 1990). The convention proposed here has been used extensively in FITS since the format was invented and no argument has been advanced sufficiently compelling to invalidate the many thousands of tapes written with that convention. Furthermore, we regard our image samples as "voxels" in real physical space rather than pixels in two-dimensional display space. As such, they may be viewed from any angle via transposition and rotation. The only point within the voxel that remains invariant under those operations is its center and we argue, therefore, that it is the center of the voxel which we should count.

and the CDELTn diagonal matrix. These divergent forms should no longer be written.

The fourth (third required) step in the process of finding the true coordinates depends on the type of axis given in CTYPEn. For simple linear axes, the true coordinate is found by adding the offset found above to the coordinate value at the reference point given by CRVALn. Otherwise, some function of the offset(s), the CRVALn, and, perhaps, other parameters must be established by convention and agreement. Greisen (1983) gives examples of the considerations which might apply to defining astronomical "velocity" axes.

In the present manuscript, we will treat only the conventions to be applied to pairs of astronomical angular coordinates, *i.e.*, to celestial longitudes and latitudes with respect to various systems and represented with various spherical projections. For this case, the fourth step in the computation of coordinate values is to convert the linear (x, y) offsets into longitudes and latitudes in the "native" coordinate system of the particular spherical projection. The type of spherical projection will be specified by characters 5 through 8 of the CTYPEn keyword and must be the same for both axes of the coordinate pair. The particular characters to be used will be established by the conventions described in Sections 4 and 7. Any type not so specified should be taken to indicate that the conventions of this manuscript are not being used. The formulæ for the conversion from (x, y) to (ϕ, θ) will also be described in Section 4. The nature of the "native" system is also determined entirely by the projection. For example, for zenithal projections, the native system has its North pole at the tangent point. Some projections require additional parametric information. These parameters will be given with FITS keywords PROJPm, for $m = 0, 1, 2, \ldots$ as specified in the following descriptions of the projections.

The fifth (fourth required) step in the computation of the coordinate value of the angle pair consists of a spherical coordinate rotation from the native longitude and latitude to a recognized celestial coordinate system. The type of the standard system is indicated in the first four characters of the CTYPEn as 'RA--' and 'DEC-' for equatorial coordinates, 'xLON' and 'xLAT' for longitude-latitude systems, where $x =$ G for Galactic, E for Ecliptic, H for helioecliptic, and S for supergalactic coordinates.[2] Other values of x will undoubtedly be defined; FITS readers should note the general form and handle the coordinate pair even if the particular value of x is not recognized. The CRVALn give the coordinates in the standard system of the reference point in the native system (*e.g.*, the tangent point). Exactly what the reference point is will be defined for each of the projections described in Section 4. A full coordinate rotation requires a third angle as well. This third angle was avoided in AIPS by Greisen's (1983) requirement that the native system be locally parallel to the standard system. The third angle will be indicated by the new FITS keyword LONGPOLE and is understood to have a default value of 0° or 180° where possible (see below). The formulæ for the coordinate rotation are

$$\begin{aligned} \sin\theta &= \sin\delta\sin\delta_P + \cos\delta\cos\delta_P\cos(\alpha-\alpha_P) \\ \phi-\phi_P &= \arg\left[\sin\delta\cos\delta_P - \cos\delta\sin\delta_P\cos(\alpha-\alpha_P),\right. \\ &\qquad \left. -\cos\delta\sin(\alpha-\alpha_P)\right], \end{aligned} \tag{4}$$

where arg is an inverse tangent function that returns the correct quadrant, *i.e.*, if $(x, y) = (r\cos\alpha, r\sin\alpha)$, then $\arg(x, y) = \alpha$. The longitude and latitude in the native system are given by (ϕ, θ) and in the standard system are given by (α, δ). The coordinates (α_P, δ_P) are the standard-system coordinates of the North Pole of the (ϕ, θ) system and ϕ_P is the longitude in the native system of the North Pole of the standard system (specified with keyword LONGPOLE). Note that, if $\delta_p = 90°$, Eq. 4 becomes

$$\begin{aligned} \theta &= \delta \\ \phi &= \phi_p + \alpha - \alpha_p + 180° \end{aligned}$$

which may be used to define a simple change in the origin of longitude. A useful variant of Eqs. 4 is

$$\begin{aligned} &\sin\theta = \sin\delta\sin\delta_P + \cos\delta\cos\delta_P\cos(\alpha-\alpha_P) \\ &\cos\theta\sin(\phi-\phi_P) = -\cos\delta\sin(\alpha-\alpha_P) \\ &\cos\theta\cos(\phi-\phi_P) = \sin\delta\cos\delta_P \\ &\qquad\qquad - \cos\delta\sin\delta_P\cos(\alpha-\alpha_P)\,. \end{aligned} \tag{5}$$

The inverse equations are

$$\begin{aligned} &\sin\delta = \sin\theta\sin\delta_P + \cos\theta\cos\delta_P\cos(\phi-\phi_P) \\ &\cos\delta\sin(\alpha-\alpha_P) = -\cos\theta\sin(\phi-\phi_P) \\ &\cos\delta\cos(\alpha-\alpha_P) = \sin\theta\cos\delta_P \\ &\qquad\qquad - \cos\theta\sin\delta_P\cos(\phi-\phi_P)\,. \end{aligned} \tag{6}$$

A matrix method of handling this rotation is described in Appendix D.

In some of the projections, the reference point is not at the North pole of the native coordinates, but instead at some other latitude, namely $(\phi, \theta) = (0, \theta_0)$, which we will represent by (α_0, δ_0) in the standard system. The default value of LONGPOLE is 0° if the latitude in the standard system of the reference point of the projection is greater than the latitude of that point in the native system; otherwise the default value of LONGPOLE is 180°. This is the condition for the celestial latitude to increase in the same direction as the native latitude at the reference point. Thus, for example, in zenithal projections the default is always 180° since $\theta_0 = 90°$. In cylindrical and standard projections, where $\theta_0 = 0°$, the default value for LONGPOLE is 180° for $\delta_0 < 0$, but it is 0° for $\delta_0 > 0$.

The coordinates of the pole may be computed using,

$$\delta_P = \arg\left(\cos\theta_0\cos\phi_p,\, \sin\theta_0\right) \pm$$

[2] "New" Galactic coordinates are assumed here, 1960. Users of the older system or future systems should adopt a different value of x and document its meaning.

$$\cos^{-1}\left(\frac{\sin\delta_0}{\sqrt{\sin^2\theta_0+\cos^2\theta_0\cos^2\phi_P}}\right) \quad (7)$$

$$\sin(\alpha_0-\alpha_P) = \sin\phi_P\cos\theta_0/\cos\delta_0 \quad (8)$$

$$\cos(\alpha_0-\alpha_P) = \frac{\sin\theta_0-\sin\delta_P\sin\delta_0}{\cos\delta_P\cos\delta_0}, \quad (9)$$

after which Eqs. 4 or 5 may be used to determine the native coordinates. Note that Eq. 7 contains an ambiguity in the sign of the inverse cosine and that all three indicate that some combinations of α_0, δ_0, θ_0, and ϕ_P are not allowed. For these projections, we are therefore required to adopt the additional conventions:

1. Equations. 8 and 9 indicate that α_p is undefined when $\delta_0 = \pm 90°$. This simply represents the longitude singularity at the pole and forces us to define $\alpha_p = \alpha_0$ in this case.
2. If the North pole of the native system approaches the North or South pole of the standard system, namely $\delta_p = \pm 90°$, then $\delta_0 = \theta_0$. The longitude at the native pole is then $\alpha_p = \alpha_0 + \phi_p - 180°$ for $\delta_p = 90°$ and $\alpha_p = \alpha_0 - \phi_p$ for $\delta_p = -90°$.
3. Some combinations of θ_0, δ_0, and ϕ_p produce an invalid argument for the $cos^{-1}()$ in Eq. 7. In this case there is no solution for δ_p. Otherwise Eq. 7 produces two solutions for δ_p. Valid solutions are ones that lie in the range $-90°$ to $+90°$. For some combinations of θ_0, δ_0, and ϕ_p, neither solution is valid. Note, however, that when `LONGPOLE` ($\equiv \phi_p$) takes its default value of $0°$ or $180°$ (depending on θ_0) then *any* combination of δ_0 and θ_0 produces a valid argument to the $\cos^{-1}()$ in Eq. 7, and at least one of the solutions is valid.
4. Where Eq. 7 has two valid solutions the one closest to the value of the new FITS keyword `LATPOLE` is chosen. It is acceptable to set `LATPOLE` to a number greater than +90 to choose the northerly solution (the default if `LATPOLE` is omitted), or a number less than −90 to select the southern solution.
5. Eq. 7 often only has one valid solution (because the other lies outside the range $-90°$ to $+90°$). In this case `LATPOLE` is ignored.
6. For the special case where $\theta_0 = 0$, $\delta_0 = 0$, and $\phi_p = \pm 90$ then δ_p is not determined and `LATPOLE` specifies it completely. `LATPOLE` has no default value in this case.

In light of the above points, users should check their values of θ_0, δ_0, and ϕ_p against Eqs. 7, 8, and 9 to ensure their validity.

This proposal eliminates the need for the previously specified keyword `CROTA`n. In Section 5 we will discuss considerations in the translation of older FITS images to the new system and provisions that updated FITS-writing programs should make to support older FITS-reading programs.

The keywords proposed above and throughout the main body of this manuscript apply to the relatively simple images stored in the main FITS image data and in `IMAGE` extensions. When coordinates are used to describe image fragments in `BINTABLE` extension tables, there are additional nomenclature difficulties. A proposed solution to these is given in Appendix B.

2.2. Improved axis description

The original FITS paper (Wells, *et al.*, 1981) naively assumed that the units along each axis could be implied simply by the contents of the `CTYPE`n keyword and that they would be in the basic SI units. Outside of celestial coordinates, both of these assumptions have apparently failed in practice. There has been little agreement on the contents of the keyword and even less on the units. We propose, therefore, to provide a mechanism whereby responsible FITS writers may at least document the units which they choose to use. The obvious, character-valued keyword for this is `CUNIT`n to describe the units used in the values given for *both* `CRVAL`n and `CDELT`n. In celestial coordinates, under the conventions of this manuscript, the units for angles on the celestial sphere are restricted to degrees and the value of `CUNIT`n for such an axis is therefore restricted to `'deg'`; `CUNIT`n for these axes will be ignored. If this keyword is omitted, FITS readers should assume degrees for angular measures of all kinds and simple SI units appropriate to the axis type indicated. By simple, we mean meters, seconds, Kelvins, and kilograms and explicitly exclude scaled measures such as cm and days. We hope that FITS writers will not take the addition of this keyword as encouragement to use arbitrary scaled or otherwise complex units. Such practice interferes with the communication of the physics of astronomical images.

2.3. Secondary axis descriptions

In some cases, the axis of an image may be described as having more than one coordinate. An example of this would be the frequency, velocity, and wavelength along a spectral axis (only one of which, of course, could be linear). One can also describe the position on a photographic plate in meters as well as in degrees on the sky. To allow up to 8 additional descriptions of each axis, we propose the addition of the following optional, but now reserved, keywords.

`C`m`VAL`n	coordinate value at reference point
`C`m`PIX`n	pixel coordinate of the reference point
`C`m`ELT`n	coordinate increment at reference point
`C`m`YPE`n	axis type (8 characters)
`C`m`NIT`n	units of `C`m`VAL`n and `C`m`ELT`n (character)

The value of m will be 2 through 9 for the second through ninth description of axis n. The alternate coordinate descriptions are computed in the same fashion as the primary coordinates. Note that the type of coordinate depends on the `CTYPE`n or `C`m`YPE`n and may be linear in one of the alternate descriptions and non-linear in another. (Conventions implied by special values for `CTYPE`n and

CmYPEn for describing non-celestial non-linear coordinates will need to be developed, but are beyond the scope of the present manuscript.) Note too that we do not propose any defaults for these keywords; if a secondary axis description is given, all of these keywords must be specified.

2.4. Image display conventions

It is very helpful to adopt a convention for the display of images transferred via the FITS format. Many of the current image processing systems have converged upon such a convention. Therefore, we recommend that FITS writers order the pixels so that the first pixel in the FITS file (for each image plane) be the one that would be displayed in the lower-left corner (with the first axis increasing to the right and the second axis increasing upwards) by the imaging system of the FITS writer. This convention would not preclude a program from looking at the axis descriptions and overriding this convention, or preclude the user from requesting a different display. This convention also does not alleviate FITS writers from the expectation that they will provide complete and correct descriptions of the image coordinates, allowing the user to determine the meaning of the image. The ordering of the image for display is simply a convention of convenience, whereas the coordinates of the pixels are part of the physics of the observation.

3. Coordinate reference frames

In order to define rigorously the coordinate systems being used, several additional keywords are required. For example, equatorial coordinates (right ascension and declination) are relative to a given equinox and epoch, and relative to a fundamental coordinate system. For this last, we introduce the new keyword RADECSYS which gives a character string to specify the frame of reference for the equatorial coordinate system. The allowed values are

RADECSYS	Definition
'FK4 '	mean place, old (pre-IAU 1976) system
'FK4-NO-E'	mean place, old system but without e-terms
'FK5 '	mean place, new (post-IAU 1976) system
'GAPPT '	geocentric apparent place, post-IAU 1976 system

The initial FITS paper (Wells, *et al*, 1981) used the keyword EPOCH to mean the epoch of the mean equator and equinox. Since the word *epoch* has a conventional and different use in astrometry, we propose replacing EPOCH with EQUINOX. If the old keyword is given, it should be replaced with the new one. For FK4 and FK4-NO-E, any stated equinox is Besselian and, if neither EQUINOX nor EPOCH are given, a default of 1950.0 is to be taken. For FK5, any stated equinox is Julian and, if neither keyword is given, a default of 2000.0 is to be taken. If RADECSYS is not given, but an EQUINOX is, then, if EQUINOX $<$ 1984.0, the equinox is taken as Besselian of type FK4. Alternatively, if EQUINOX $\geq$ 1984.0, then the default system is Julian of type FK5. (There is really no acceptable reason for a modern FITS-writing program to write an EQUINOX without a RADECSYS keyword.) Note that these defaults, while probably true of older files using the EPOCH keyword, are not required of them. Note that EQUINOX is understood to apply to ecliptic coordinates in the standard way as well as to equatorial coordinates.

The FK4 reference system is not inertial; there is a small, but significant, rotation relative to distant objects. Therefore, we require an epoch to specify when the mean place was correct as well as the equinox which specifies the reference frame. The old keyword DATE-OBS may be used for this purpose. To provide a more accurate, and easier to use, specification, we also propose the new floating-valued keyword MJD-OBS to specify the time of of observation in the form of a Modified Julian Date (JD $-$ 2400000.5). Of necessity, this time must represent some average of the times actually entering the observation, but should be used in preference to DATE-OBS when available. Comments explaining the nature of this "average" time would be quite appropriate. Future agreements may clarify the time system used to determine modified Julian dates, but the usage of MJD-OBS should not require such detailed accuracy due to the "average" nature of the time given. The GAPPT system of coordinates does not require an EQUINOX, but only an MJD-OBS.

The combination of the EQUINOX, RADECSYS, and CTYPEn FITS header cards define the coordinate system in which the CRVALn are specified. It is also understood that the FITS image has been resampled so as to contrive that application of the algorithm for computing image coordinates (Fig. 1) will produce coordinates of the specified type at all points of the image. This resampling will be important for applications requiring astrometric accuracy (for example intercomparison of optical and VLBI data) and where, as will almost always be the case, the chosen reference frame is not the one in which the instrument observed.

For example synthesis radio telescopes produce maps which are an orthographic (SIN) projection of the celestial sphere. Therefore, if B1950.0 coordinates are used, then, strictly speaking, they should be styled as "FK4-NO-E." However, in common usage, the CRVALn for such maps are usually given as FK4 coordinates. By doing so, the E-terms ($\leq$ 343 milliarcsec) are effectively corrected to first order only, although the error thereby introduced will generally be much smaller than instrumental effects.

Note that we do not propose to add to the FITS standard any stipulations regarding the various astrometric transformations involved (precession, nutation, aberration and gravitational deflection) beyond requiring that they be done properly, using modern formulations, and to an accuracy commensurate with the data.

4. Spherical image projections

There is considerable confusion in the astronomical community in the area of units, including the units to be used in the FITS format. It was the intention of the original FITS paper (Wells, *et al.*, 1981) that all units be simply SI units or compounds (such as meters squared) of SI units, with the exception that all angular measures be expressed in degrees. Due to the widespread disregard for this original intention, it is probably not practical to try to enforce uniformity (or communication) of units at this time. However, because of the original papers's intentions and the very extensive use of Greisen's (1983) conventions for celestial coordinates, we deem that the units for all measures of angles on the celestial sphere purporting to be within the conventions of the present manuscript shall be (are required to be, must be) degrees.

The units of the `CDELT`*n* are degrees per pixel for angular axes, making degrees the units of (x, y), and R_θ. The units of (ϕ, θ) and A_ϕ are also degrees simply because they are angles. From the reverse point of view, (x, y), and R_θ are in units of fractions of the unit sphere, multiplied by $\frac{180^\circ}{\pi}$ to bring the units to the FITS-conventional degrees rather than radians. In the formulæ developed below, it is assumed that all trigonometric functions require arguments in degrees and all inverse trigonometric functions return values in degrees. The arguments of inverse trigonometric functions and the results returned by trigonometric functions are assumed to take their normal values. Whenever a conversion between radians and degrees is required in the equations below, it is explicitly shown.

The equations below describe a *left-handed* coordinate system, called (x, y), which will be centered either on the north pole of the native coordinate system ((longitude,latitude) $\equiv (\phi, \theta) = (0, 90^\circ)$) or on the origin of the native system ($(\phi, \theta) = (0, 0)$). These are illustrated in Fig. 2.

In most of the projections below, we have not included the infinite variety of oblique variations on the basic projections such as equatorial cases of the zenithal projections, which are included in many books on map projections. We believe that this is a desirable omission, which will improve the simplicity and clarity of the data communication, while the later rotation from the native coordinates of the projection to standard, recognized systems of coordinates provides all needed obliquity. There are innumerable other projections used in cartography which we have also omitted. Evenden (1991) offers lovely displays (with little mathematics) for 73 different projections, some of which we propose either directly or obliquely.

Calabretta (1995) has written, and made available under a GNU license, a package of C and of Fortran subroutines implementing all projections and coordinate conversions proposed in this manuscript.

4.1. Zenithal projections

Zenithal or *azimuthal* projections are a class of projections in which the surface of projection is a plane. The native coordinate system is defined to have the polar axis orthogonal to the plane of projection at the *reference* point. Then, the meridians (of native longitude) are projected as rays emanating from the reference point and the parallels (of native latitude) are mapped as concentric circles centered on the same point. By convention, therefore, the reference point in all zenithal projections is the native coordinate system north pole, $(\phi, \theta) = (0, 90^\circ)$.

Defining (ϕ, θ) as the native longitude and latitude, respectively, zenithal projections are then characterized by a cylindrical coordinate radius R_θ and an angle A_ϕ equal to the longitude ϕ. Our linear, rectangular coordinates, (x, y), defined by Eq. 3, are then given by

$$x = R_\theta \sin\phi \qquad (10)$$

$$y = -R_\theta \cos\phi\,, \qquad (11)$$

which may be inverted as

$$R_\theta = \sqrt{x^2 + y^2} \qquad (12)$$

$$\phi = \arg(-y, x)\,. \qquad (13)$$

4.1.1. AZP (perspective)

Perspective zenithal projections are generated from a point and carried through the unit sphere to the plane of projection. We consider here only those cases where the plane of projection is tangent to the sphere at its pole since it may be shown that having the plane intersect the sphere at another (native) latitude causes only a simple rescaling of the projection. If the source of the projection is at a distance μ from the center of the sphere ($\mu > 0$ in the direction away from the plane of projection), then it is straightforward to show that

$$R_\theta = \frac{180^\circ}{\pi} \cos\theta \left(\frac{\mu + 1}{\mu + \sin\theta}\right)\,. \qquad (14)$$

The inverse of this equation is

$$\theta = \arg(\rho, 1) - \sin^{-1}\left(\frac{\rho\mu}{\sqrt{\rho^2 + 1}}\right)\,, \qquad (15)$$

where

$$\rho \equiv \frac{\pi}{180^\circ} \frac{R_\theta}{\mu + 1}\,.$$

This projection is illustrated in Fig. 3. The FITS keyword `PROJP1` will be used to specify μ and the FITS projection type will be represented by the code `AZP`. Values of $\mu > 10^{14}$ will be taken to be infinite, but if $\mu >>$ the number of pixels, then it may be taken to be infinite with very little

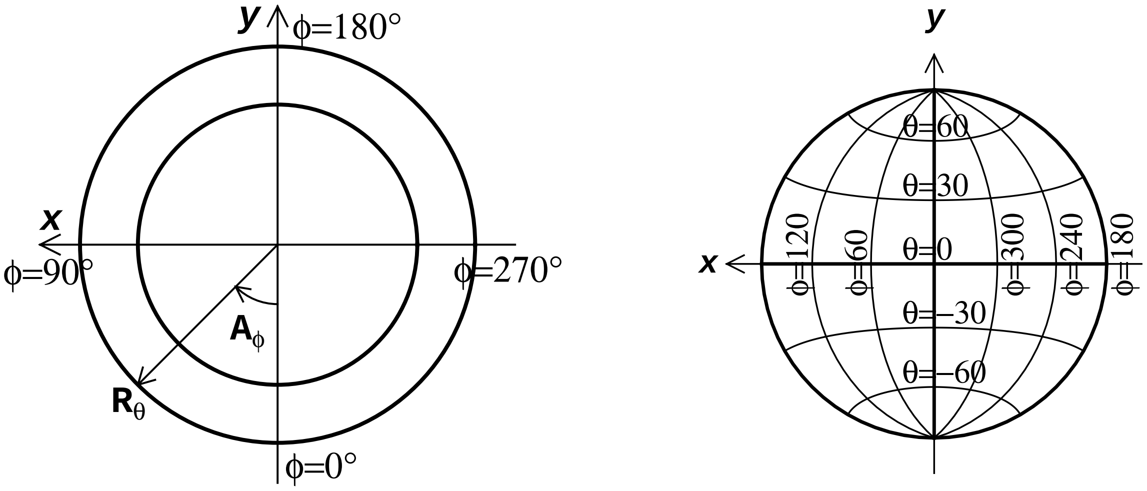

Fig. 2. Coordinate definitions for origins at (longitude, latitude) $\equiv (\phi, \theta) = (0, 90°)$ (left) and $(\phi, \theta) = (0, 0)$ (right)

error. This projection correctly represents the image of a sphere, such as a planet, when viewed from a distance $\mu - 1$ times the planetary radius. The sub-viewer point is then the south pole of the projection, $\theta = -90°$, and the coordinates of the reference point may be expressed in planetary longitude and latitude, λ, β. The `PC` matrix should even allow for first-order correction for planetary oblateness.

4.1.2. TAN (gnomonic)

There are three special values of μ in perspective zenithal projections which are in wide use in astronomy and which have come to have conventional codes for the projection types. If you are using one of the special values of μ, you should use the conventional code rather than `AZP`. The first of these, $\mu = 0$, is widely used in optical astronomy and is called `TAN`. Equation 14 becomes

$$R_\theta = \frac{180°}{\pi} \cot\theta \tag{16}$$

$$\theta = \tan^{-1}\left(\frac{180°}{\pi R_\theta}\right) . \tag{17}$$

This projection is illustrated in Fig. 4. The `TAN` projection (with no `PROJP1` keyword) will be considered equivalent to the `AZP` projection with keyword `PROJP1` set to zero. Since this projection is from the center of the sphere, great circles are projected as straight lines. Thus, the shortest distance between two points on the sphere is represented as a straight line interval, which, however, is not uniformly divided. A single `TAN` projection diverges at $\theta = 0$, but one may use a tangent-like projection on six square faces to display the whole sky. See Section 4.4.9 for details.

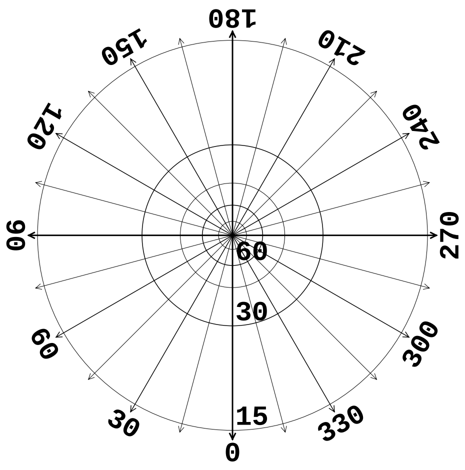

Fig. 4. `TAN` projection, diverges at $\theta = 0$

4.1.3. SIN (orthographic)

In radio aperture synthesis, the perspective zenithal projection which arises naturally has $\mu = \infty$. In this case, Eq. 14 becomes

$$R_\theta = \frac{180°}{\pi} \cos\theta$$

$$\theta = \cos^{-1}\left(\frac{\pi}{180°} R_\theta\right) .$$

This projection is illustrated in the upper portion of Fig. 5. The `SIN` projection (with no `PROJP1` keyword) will be considered equivalent to the `AZP` projection with keyword `PROJP1` set to a very large number. This projection gives a representation of the visual appearance of a sphere (*e.g.*, a planet) when seen from a great distance. Radio interferometers which have all baselines oriented in a plane can use a "slant orthographic" projection which Greisen (1983) called the `NCP` projection for the particular case of

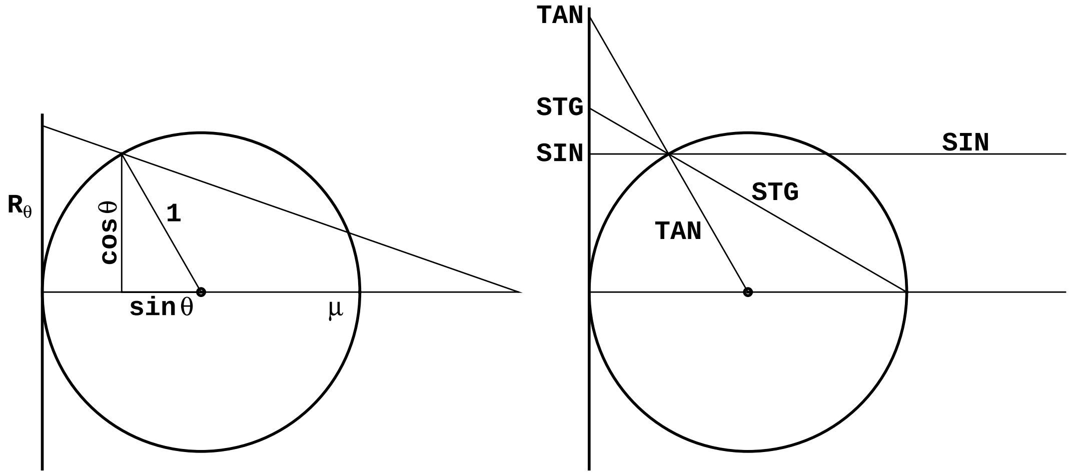

Fig. 3. Geometry of azimuthal projections

an East-West interferometer. A simple extension of the formulæ above, allows all such cases to be handled naturally. We, therefore, propose that the `SIN` projection actually be defined as

$$x = \frac{180^\circ}{\pi}\left[\cos\theta\sin\phi + \alpha\,(\sin\theta - 1)\right] \tag{18}$$

$$y = -\frac{180^\circ}{\pi}\left[\cos\theta\cos\phi + \beta\,(\sin\theta - 1)\right]\,, \tag{19}$$

where α and β will be given by keywords `PROJP1` and `PROJP2`, respectively. For a planar interferometer array, we may write

$$N_u u + N_v v + N_w w = 0\,,$$

where (N_u, N_v, N_w) are the direction cosines of the normal to the plane in the same coordinate system used for the "interferometer coordinates" (u, v, w). Then, the projection parameters are

$$\alpha = N_u/N_w$$
$$\beta = N_v/N_w\,.$$

For an East-West interferometer, $(\alpha, \beta) = (0, \cot\delta_0)$, while Cornwell and Perley (1991) give, for any instant of time, $(\alpha, \beta) = (\tan Z\sin\chi, \tan Z\cos\chi)$, where Z is the zenith angle and χ is the parallactic angle. Two cases of the oblique projection are illustrated in the lower portion of Fig. 5.

4.1.4. STG (stereographic, zenithal orthomorphic)

The third perspective zenithal projection in general use has $\mu = 1$, for which Eq. 14 becomes

$$R_\theta = 2\frac{180^\circ}{\pi}\frac{\cos\theta}{1+\sin\theta} \tag{20}$$

$$= 2\frac{180^\circ}{\pi}\tan\left(\frac{90^\circ - \theta}{2}\right)$$

$$\theta = 90^\circ - 2\tan^{-1}\left(\frac{\frac{\pi}{180^\circ}R_\theta}{2}\right)\,. \tag{21}$$

This projection is illustrated in Fig. 6. The `STG` projection (with no `PROJP1` keyword) will be considered equivalent to the `AZP` projection with keyword `PROJP1` set to one. This projection has the property that all circles on the generating sphere are projected as circles. It is also *orthomorphic*, meaning that the ϕ and θ scales are the same at each point and, therefore, that shape is preserved over small regions.

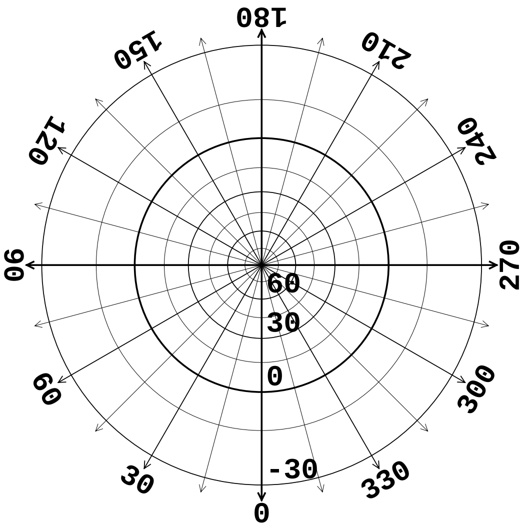

Fig. 6. `STG` projection, diverges at $\theta = -90°$

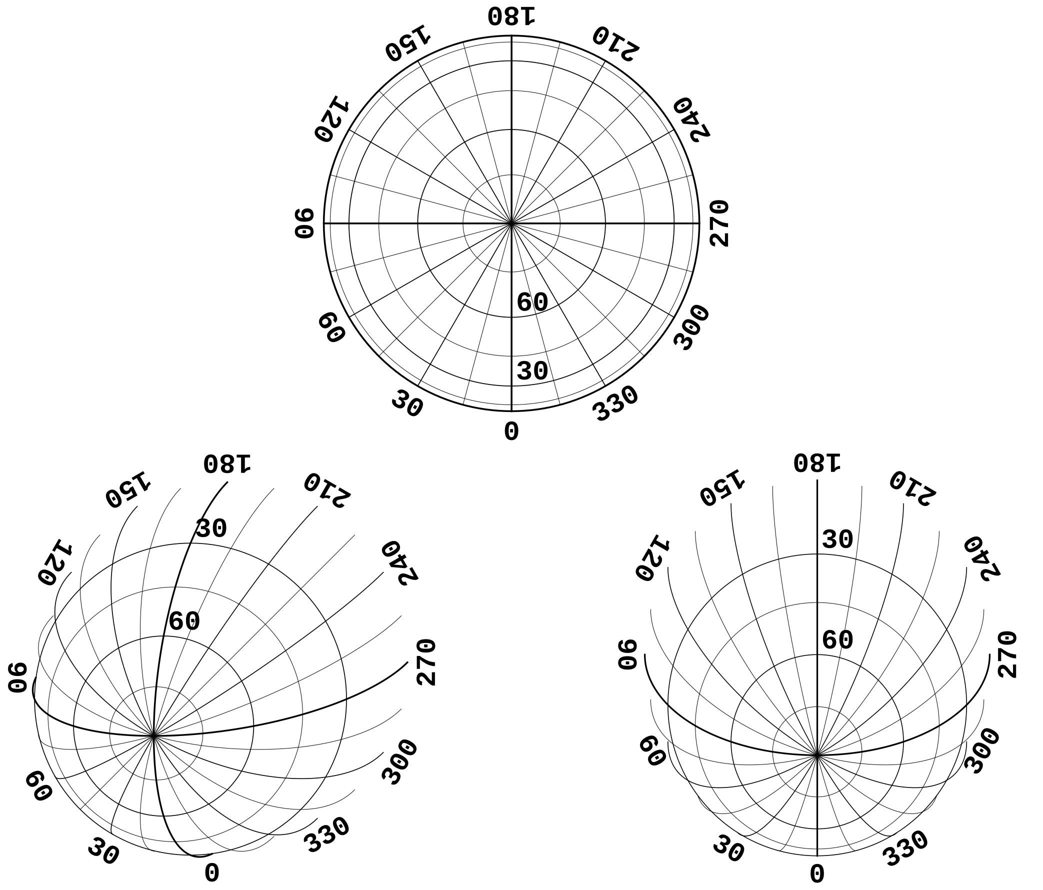

Fig. 5. SIN projection, (top) $\alpha = 0, \beta = 0$, North and South sides begin to overlap at $\theta = 0$: (bottom left) $Z = 30°, \chi = 45°$, (bottom right) East-West array at $\delta_0 = 60°$.

4.1.5. ARC (zenithal equidistant)

There are also some non-perspective zenithal projections of interest. The first appeared in Greisen (1983) as `ARC` and is widely used as the approximate projection of Schmidt telescopes. The native meridians are uniformly divided to give uniformly spaced parallels, or

$$R_\theta = (90° - \theta) \tag{22}$$

$$\theta = 90° - R_\theta \,. \tag{23}$$

This projection is illustrated in Fig. 7. No `PROJP1` keyword is required.

4.1.6. ZPN (zenithal polynomial)

An extension of the `ARC` projection would be to allow terms up to ninth order in the zenith distance. We call this `ZPN` and represent it as

$$R_\theta = \frac{180°}{\pi} \sum_{i=0}^{9} P_i \left(\frac{\pi}{180°} (90° - \theta) \right)^i \tag{24}$$

where the P_i are given by the keywords `PROJP0`, `PROJP1`, …, `PROJP9`, all of which have default values of zero. Note

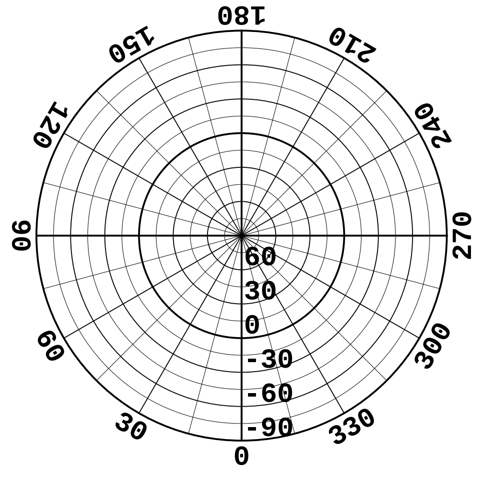

Fig. 7. ARC projection, no limits

the units implied by the $\frac{\pi}{180°}$. Since it does not have an exact inverse, this projection should only be used when the geometry of the observations require it. In particular, it should never be used as an $n^{\rm th}$-order expansion of one of the standard zenithal projections. An example of this projection is illustrated in Fig. 8.

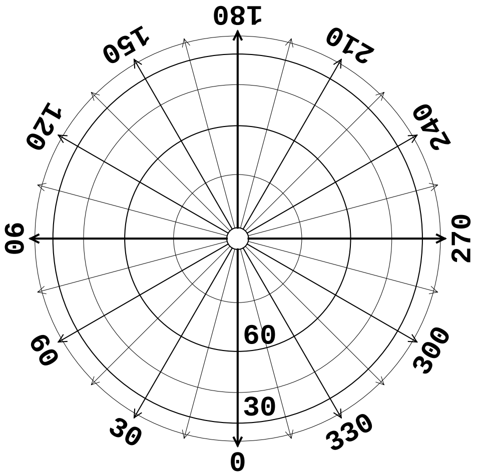

Fig. 8. `ZPN` projection with parameters, 0.05, 0.95, -0.025, -0.15833, 0.00208, 0.00792, -0.00007, -0.00019; limits depend upon the parameters

4.1.7. ZEA (zenithal equal-area)

An equal-area zenithal projection, which we will henceforth call `ZEA`, may be generated using

$$R_\theta = \frac{180^\circ}{\pi}\sqrt{2(1-\sin\theta)} \tag{25}$$

$$= 2\frac{180^\circ}{\pi}\sin\left(\frac{90^\circ-\theta}{2}\right) \tag{26}$$

$$\theta = 90^\circ - 2\sin^{-1}\left(\frac{\frac{\pi}{180^\circ}R_\theta}{2}\right). \tag{27}$$

This projection is illustrated in Fig. 9. No `PROJP1` keyword is required.

4.1.8. AIR (Airy projection)

The Airy projection is a projection which minimizes the error for the region within latitude θ_b (Evenden, 1991). It is defined by

$$R_\theta = -\frac{180^\circ}{\pi}\left(\frac{\ln(\cos\xi)}{\tan\xi} + \frac{\ln(\cos\xi_b)}{\tan^2\xi_b}\tan\xi\right), \tag{28}$$

where

$$\xi \equiv \frac{90^\circ-\theta}{2}$$

$$\xi_b \equiv \frac{90^\circ-\theta_b}{2}.$$

When θ_b approaches 90°, the second term of Eq. 28 approaches its asymptotic value of -1/2. For all θ_b, the projection is unbounded at the native south pole. The inversion of Eq. 28 may not be expressed so simply, since it is

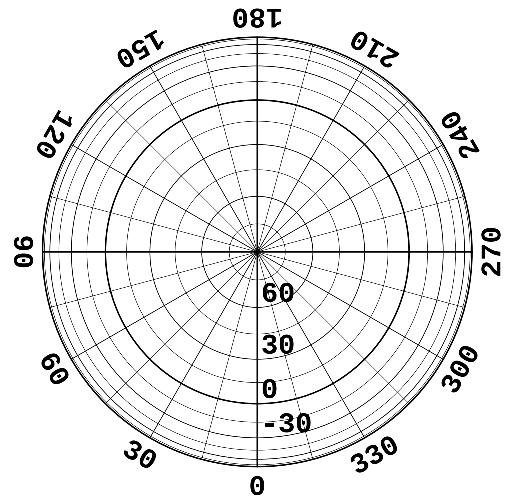

Fig. 9. `ZEA` projection, no limits

a transcendental equation in ξ or θ. One may determine a solution iteratively with Newton's method (see Section 6). Perhaps more simply, one can use Eq. 28 to find an interval in *e.g.*, $\cos\xi$ such that $R_{\theta 2} > R_\theta > R_{\theta 1}$. Then, one iterates toward the solution always finding an interval surrounding the correct answer until convergence is achieved. See Appendix D.2 for an example of this technique.

The FITS keyword `PROJP1` will be used to specify θ_b in degrees with a default of 90°. This projection is illustrated in Fig. 10 with two different values of θ_b.

4.2. Cylindrical projections

Cylindrical projections are a class of projections in which the surface of projection is a cylinder. The native coordinate system is chosen to have its polar axis coincident with the axis of the cylinder. Meridians and parallels are mapped onto a rectangular grid so that cylindrical projections are described by formulæ which return x and y. By convention, the reference point in all cylindrical projections is the native coordinate system origin, $(\phi,\theta) = (0,0)$. Furthermore, all cylindrical projections have

$$x \propto \phi\,. \tag{29}$$

4.2.1. CYP (general perspective)

For general cylindrical perspective, we project the unit sphere on a cylinder of radius λ from points in the equatorial plane of the native system at a distance μ measured from the center of the sphere in the direction opposite the projected surface. It is straightforward to show that

$$x = \lambda\phi \tag{30}$$

$$y = \frac{180^\circ}{\pi}\left(\frac{\mu+\lambda}{\mu+\cos\theta}\right)\sin\theta\,. \tag{31}$$

This may be inverted as

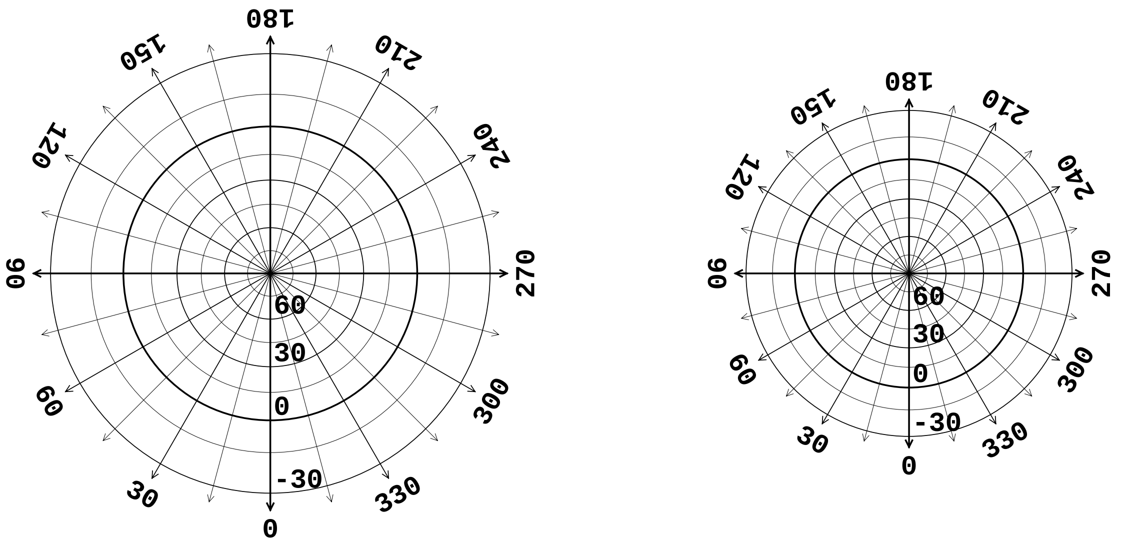

Fig. 10. AIR projection: (left) $\theta_b = 90°$ (right) $\theta_b = -10°$. Both are limited to $\theta > -35°$ and diverge at $\theta = -90°$. The two plots are on the same scale.

$$\phi = \frac{x}{\lambda} \tag{32}$$

$$\theta = \arg(1, \eta) + \sin^{-1}\left(\frac{\eta\mu}{\sqrt{\eta^2+1}}\right), \tag{33}$$

where

$$\eta \equiv \frac{\pi}{180°}\frac{y}{\mu+\lambda}. \tag{34}$$

Note that all values of μ are allowable *except* $\mu = -\lambda$. This projection is illustrated in Fig. 11. For FITS purposes, we define the keyword `PROJP1` to convey μ and the keyword `PROJP2` to convey λ. Values of $\mu > 10^{14}$ will be taken to be infinite, but if $\mu >>$ the number of pixels, then it may be taken to be infinite with very little error.

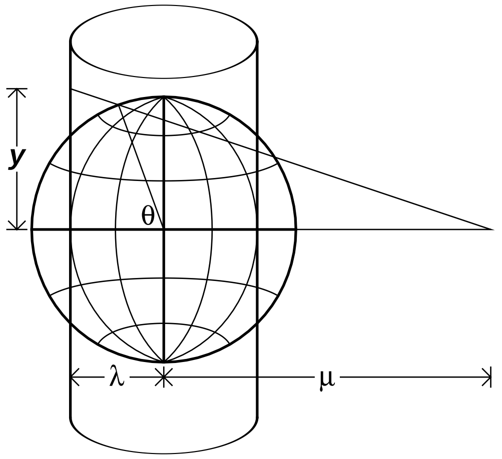

Fig. 11. Geometry of cylindrical projections

Since perspective cylindrical projections have not been in wide use in astronomy, we will define no special-case projections for these as we did for the zenithal perspective projections. We note that the simplest case puts the point of projection at the center of the sphere and uses a unit cylinder (*i.e.*, $\mu = 0, \lambda = 1$). In that case, $y = \frac{180°}{\pi}\tan\theta$ as illustrated in Fig. 12. Gall's projection

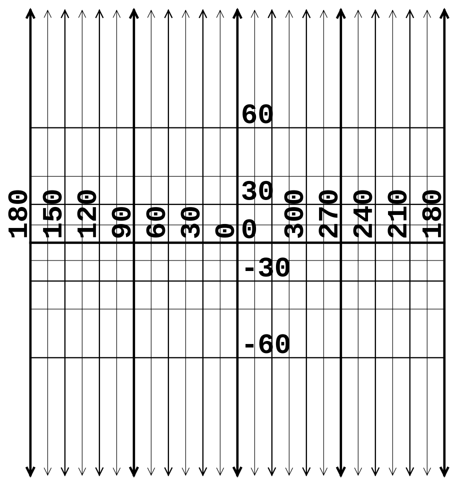

Fig. 12. CYP projection with $\mu = 0, \lambda = 1$, diverges at $\theta = \pm 90°$

minimizes distortions in the equatorial regions with parameters $\mu = 1, \lambda = \sqrt{1/2}$, giving

$$x = \phi/\sqrt{2}$$

$$y = \frac{180°}{\pi}\left(1+\frac{\sqrt{2}}{2}\right)\tan\left(\frac{\theta}{2}\right),$$

illustrated in Fig. 13. Lambert's equal-area projection is a special case of both the general perspective and general equal-area cylindrical projections. It has parameters $\mu = \infty, \lambda = 1$, equations

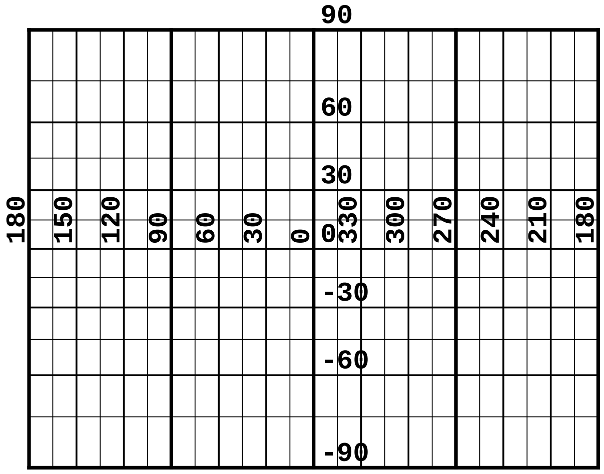

Fig. 13. Gall's CYP projection with $\mu = 1, \lambda = \sqrt{\frac{1}{2}}$, no limits

$$x = \phi$$
$$y = \frac{180^\circ}{\pi} \sin\theta \,,$$

and is illustrated in Fig. 14.

4.2.2. CAR (Cartesian)

The Cartesian, or Plate Carrée, projection is the absolutely simplest projection one could propose. Its formulæ are

$$x = \phi \tag{35}$$
$$y = \theta \,. \tag{36}$$

This projection is illustrated in Fig. 15. The keywords `PROJP1` and `PROJP2` are not used. If no projection type is specified, then FITS readers should *not* assume the linear Cartesian `CAR`. They must assume, instead, simple linear coordinates with none of the celestial-coordinate conventions of this manuscript.

4.2.3. MER (Mercator)

The Mercator projection has been widely used in navigation since it has the property that lines of constant bearing (*rhumb lines*) are projected as straight lines. Since this is the result of a differential equation, this has the nasty formulation

$$x = \phi \tag{37}$$
$$y = \frac{180^\circ}{\pi} \ln\tan\left(\frac{90^\circ + \theta}{2}\right) \,, \tag{38}$$

which reverses to

$$\theta = 2\tan^{-1}\left(e^{y\pi/180^\circ}\right) - 90^\circ \,. \tag{39}$$

This projection is illustrated in Fig. 16. The keywords `PROJP1` and `PROJP2` are not used.

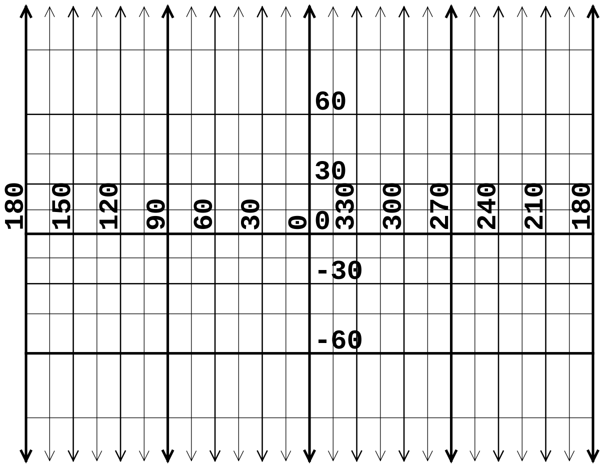

Fig. 16. MER projection, diverges at $\theta = \pm 90^\circ$

4.2.4. CEA (equal area)

Parallels are spaced so as to make the projection equal area and scaled so as to make it conformal at latitudes $\theta = \pm\theta_x$. Defining $\lambda \equiv \cos^2\theta_x$, the formulæ are

$$x = \phi \tag{40}$$
$$y = \frac{180^\circ}{\pi} \frac{\sin\theta}{\lambda} \,, \tag{41}$$

which reverses to

$$\theta = \sin^{-1}\left(\frac{\pi}{180^\circ} y\lambda\right) \,. \tag{42}$$

The keywords `PROJP1` is used to give λ which must satisfy $0 < \lambda \leq 1$. Lambert's projection is the case with $\lambda = 1$ (see Fig. 14).

4.3. Conic projections

Conic projections are a class of projections in which the surface of projection is a cone and parallels are projected as arcs of circles. In "standard" conic projections the axis of the cone is coincident with the polar axis of the native coordinate system, allowing them to be characterized by the two (native) latitudes at which the cone intersects the unit sphere. In "one-standard" conic projections, the cone is exactly tangent to the sphere and the two parametric latitudes become equal. (See the top portions of

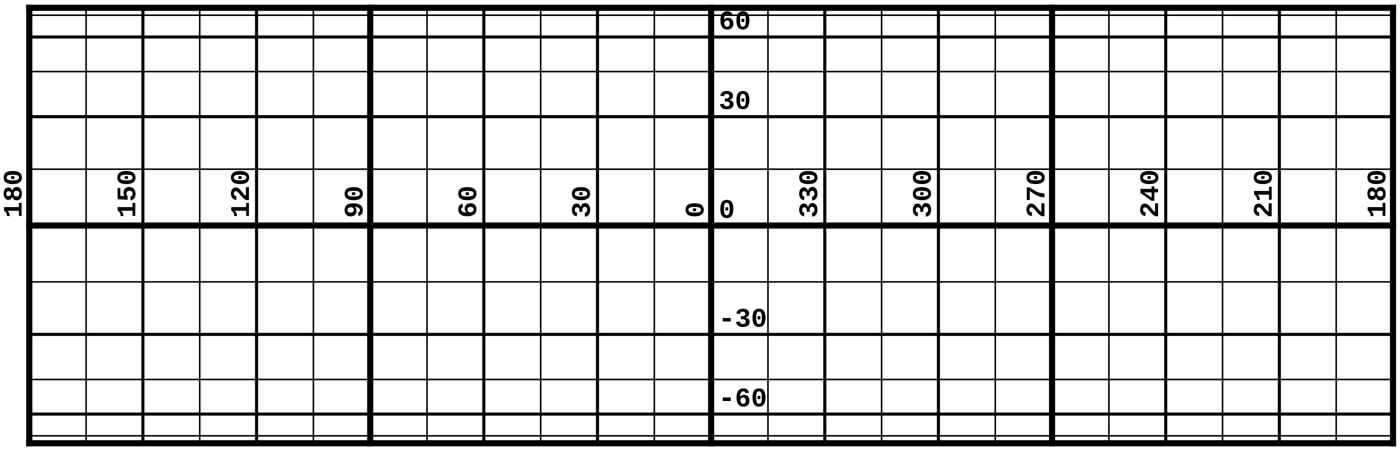

Fig. 14. Lambert's equal area `CYP` projection with $\mu = \infty, \lambda = 1$, no limits

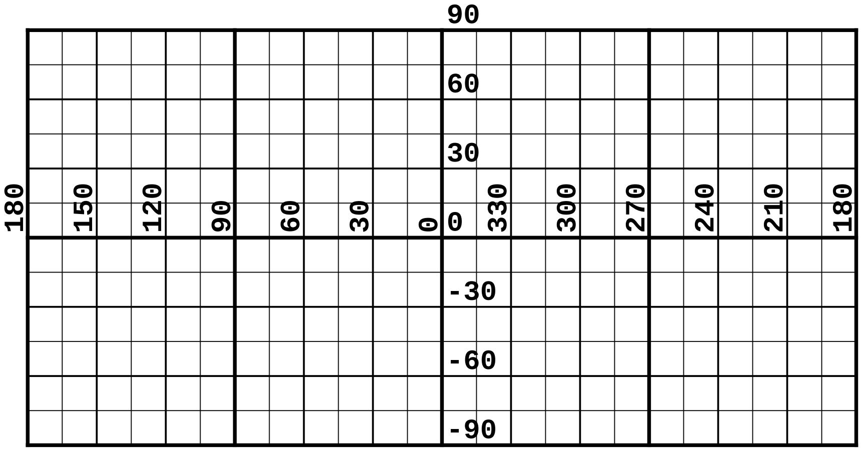

Fig. 15. `CAR` projection, no limits

Fig. 17.) In all of the conic projections presented below, the parallels are concentric and may be described by R_θ, the radius for latitude θ, and A_ϕ, the angle for longitude ϕ, but an offset in y is required to force $(x, y) = (0, 0)$ at the reference point. Then

$$x = R_\theta \sin A_\phi \tag{43}$$

$$y = Y_0 - R_\theta \cos A_\phi \,. \tag{44}$$

By convention, the reference point in all conical projections is the native coordinate system central latitude, $(\phi, \theta) = (0, \theta_a)$, where $\theta_a \equiv (\theta_1 + \theta_2)/2$. None of the equations below require $\theta_2 > \theta_1$ and, if we allow the generalization

$$R_\theta = \mathrm{sign}\theta_a \sqrt{x^2 + (Y_0 - y)^2} \tag{45}$$

$$A_\phi = \arg\left(\frac{Y_0 - y}{R_\theta}, \frac{x}{R_\theta}\right) \,. \tag{46}$$

then none require that $\theta_1 \geq 0$.

One should note that the zenithal projections are just special cases of the conicals (with $\theta_1 = \theta_2 = 90°$). Likewise, the cylindrical projections correspond to conicals in the limit $\theta_1 = \theta_2 = 0$. We present the zenithal and cylindrical projections separately because of their general usage and because of the mathematical complexity of the conical projection formulæ in these limits. We regard it as poor form to describe projections as conicals at these limits.

4.3.1. COP (perspective)

The two-standard perspective projection is characterized by two angles, $-90° \leq \theta_1 \leq \theta_2 \leq 90°$. The projection formulæ are

$$\begin{aligned} R_\theta &= \frac{180°}{\pi} \cos\alpha \left[\cot\theta_a - \tan(\theta - \theta_a)\right] \\ &= \frac{180°}{\pi} \frac{\cos\alpha \cos\theta}{\sin\theta_a \cos(\theta - \theta_a)} \end{aligned} \tag{47}$$

$$A_\phi = \phi \sin\theta_a \tag{48}$$
$$Y_0 = \frac{180^\circ}{\pi}\cos\alpha\cot\theta_a\ , \tag{49}$$
where
$$\theta_a \equiv (\theta_1+\theta_2)/2 \tag{50}$$
$$\alpha \equiv (\theta_2-\theta_1)/2\,. \tag{51}$$
The inverse may be computed using Eqs. 45 and 46 with
$$\phi = \frac{A_\phi}{\sin\theta_a} \tag{52}$$
$$\theta = \theta_a + \tan^{-1}\left(\cot\theta_a - \frac{\pi}{180^\circ}\frac{R_\theta}{\cos\alpha}\right)\,. \tag{53}$$
This projection is illustrated in Fig. 17 with two different parameterizations. The keyword `PROJP1` will be used to give θ_a in degrees and the keyword `PROJP2` will be used to give α in degrees. The default value for `PROJP2` is 0 while `PROJP1` has no default. It is recommended that both keywords always be given. The case $\theta_1 = -\theta_2$ gives a divergent equation for R_θ, but not for (x, y). This case should be written as a cylindrical projection (which it is) rather than a conical one in order to avoid posing any mathematical difficulties. To do otherwise is, we believe, worse than simply "bad form."

4.3.2. COD (equidistant)

The two-standard equidistant projection is characterized by two angles, $-90^\circ \le \theta_1 \le \theta_2 \le 90^\circ$. The projection formulæ are
$$R_\theta = \theta_a - \theta + \alpha\cot\alpha\cot\theta_a \tag{54}$$
$$A_\phi = \left(\frac{180^\circ}{\pi}\frac{\sin\theta_a\sin\alpha}{\alpha}\right)\phi \tag{55}$$
$$Y_0 = \alpha\cot\alpha\cot\theta_a\ , \tag{56}$$
where
$$\theta_a \equiv (\theta_1+\theta_2)/2 \tag{57}$$
$$\alpha \equiv (\theta_2-\theta_1)/2\,. \tag{58}$$
Using Eqs. 45 and 46, the inverse equations are then
$$\phi = \left(\frac{\pi}{180^\circ}\frac{\alpha}{\sin\theta_a\sin\alpha}\right)A_\phi \tag{59}$$
$$\theta = \theta_a + \alpha\cot\alpha\cot\theta_a - R_\theta\,. \tag{60}$$
These expressions are well behaved when $\theta_1 = \theta_2$, but are evaluated more simply with
$$R_\theta = \theta_a - \theta + \frac{180^\circ}{\pi}\cot\theta_a \tag{61}$$
$$A_\phi = \phi\sin\theta_a \tag{62}$$
$$Y_0 = \frac{180^\circ}{\pi}\cot\theta_a\ , \tag{63}$$
and, using Eqs. 45 and 46,
$$\phi = \frac{A_\phi}{\sin\theta_a} \tag{64}$$
$$\theta = \theta_a + \frac{180^\circ}{\pi}\cot\theta_a - R_\theta\,. \tag{65}$$
This projection is illustrated in Fig. 18 with two different parameterizations. The keyword `PROJP1` will be used to give θ_a in degrees and the keyword `PROJP2` will be used to give α in degrees. The default value for `PROJP2` is 0 while `PROJP1` has no default; it is recommended that both keywords always be given.

The Murdoch conical projection (Pearson, 1984) has the form
$$R_\theta = \theta_a - \theta + \left(\frac{180^\circ}{\pi}\right)^2\frac{\sin\alpha}{\alpha}\cos\theta_a$$
$$A_\phi = \phi\sin\theta_a$$
$$Y_0 = \left(\frac{180^\circ}{\pi}\right)^2\frac{\sin\alpha}{\alpha}\cos\theta_a\ ,$$
which is so similar to the conical equidistant that we reject it as a separate FITS-standard projection.

4.3.3. COE (equal area)

The two-standard equal area projection is characterized by two angles, $-90^\circ \le \theta_1 \le \theta_2 \le 90^\circ$. The projection formulæ are
$$R_\theta = \frac{180^\circ}{\pi}\frac{2}{\gamma}\sqrt{1+\sin\theta_1\sin\theta_2-\gamma\sin\theta} \tag{66}$$
$$A_\phi = \phi\gamma/2 \tag{67}$$
$$Y_0 = \frac{180^\circ}{\pi}\frac{2}{\gamma}\sqrt{1+\sin\theta_1\sin\theta_2-\gamma\sin((\theta_1+\theta_2)/2)}\,, \tag{68}$$
where
$$\gamma \equiv \sin\theta_1+\sin\theta_2\ , \tag{69}$$
or, using the average and difference angles,
$$R_\theta = \frac{180^\circ}{\pi}\frac{\sqrt{\cos^2\alpha+\sin^2\theta_a-2\sin\theta_a\cos\alpha\sin\theta}}{\sin\theta_a\cos\alpha} \tag{70}$$
$$A_\phi = \phi\sin\theta_a\cos\alpha \tag{71}$$
$$Y_0 = \frac{180^\circ}{\pi}\frac{\sqrt{\cos^2\alpha+\sin^2\theta_a(1-2\cos\alpha)}}{\sin\theta_a\cos\alpha} \tag{72}$$
where
$$\theta_a \equiv (\theta_1+\theta_2)/2 \tag{73}$$
$$\alpha \equiv (\theta_2-\theta_1)/2\,. \tag{74}$$
Using Eqs. 45 and 46, the inverse equations become
$$\phi = \frac{2}{\gamma}A_\phi \tag{75}$$
$$\theta = \sin^{-1}\left(\frac{1}{\gamma}+\frac{\sin\theta_1\sin\theta_2}{\gamma} - \gamma R_\theta^2\left(\frac{\pi}{360^\circ}\right)^2\right)\ , \tag{76}$$
or
$$\phi = \frac{A_\phi}{\sin\theta_a\cos\alpha} \tag{77}$$
$$\theta = \sin^{-1}\left(\frac{\cos\alpha}{2\sin\theta_a}+\frac{\sin\theta_a}{2\cos\alpha}\right.$$

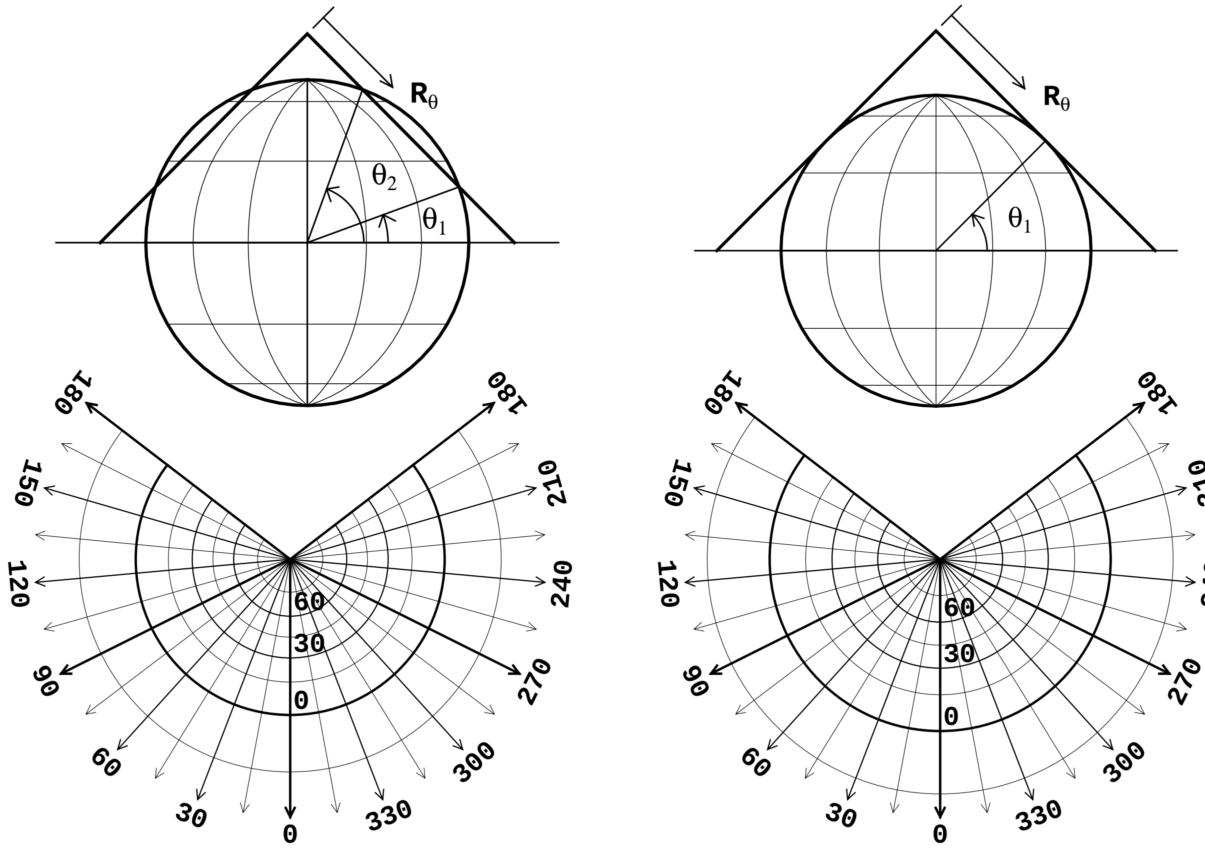

Fig. 17. (top) Perspective conic diagrams, (bottom) COP projections: (left) $\theta_1 = 20°, \theta_2 = 70°$; (right) $\theta_1 = \theta_2 = 45°$; both have the same θ_a and both diverge at $\theta = \theta_a \pm 90°$

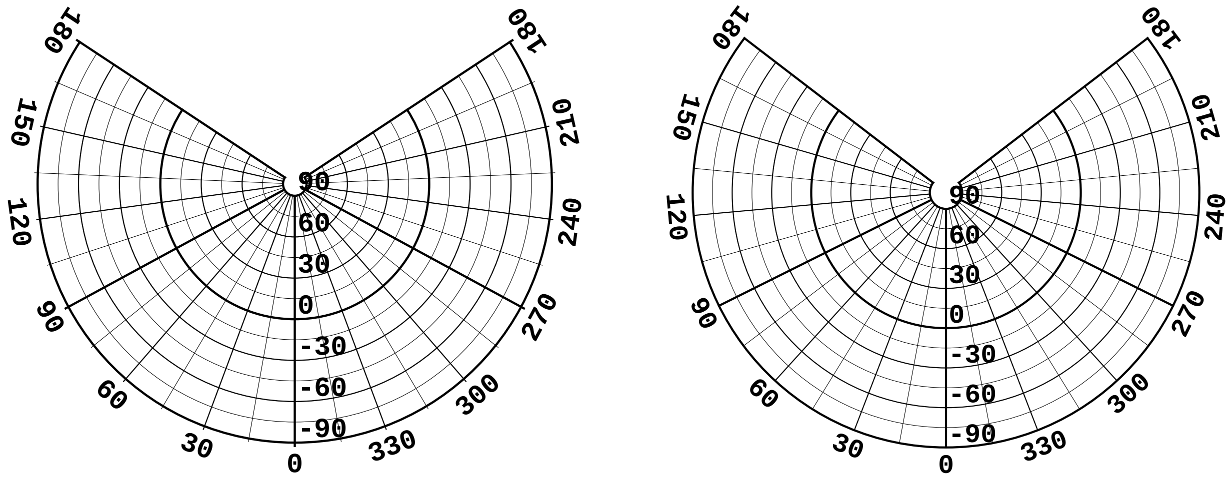

Fig. 18. COD projection: (left) $\theta_1 = 20°, \theta_2 = 70°$; (right) $\theta_1 = \theta_2 = 45°$; both have the same θ_a and no limits

$$-\frac{\sin\theta_a \cos\alpha R_\theta^2}{2}\left(\frac{\pi}{180°}\right)^2\Bigg) . \tag{78}$$

This projection is illustrated in Fig. 19 with two different parameterizations. These expressions remain useful when $\theta_1 = \theta_2$ and require no other form. The keyword `PROJP1` will be used to give θ_a in degrees and the keyword `PROJP2` will be used to give α in degrees. The default value for `PROJP2` will be 0 while `PROJP1` has no default; it is recommended that both keywords always be given.

4.3.4. COO (orthomorphic)

The two-standard orthomorphic projection is characterized by two angles, $-90° \leq \theta_1 \leq \theta_2 \leq 90°$. The projection formulæ are

$$R_\theta = \psi \left[\tan\left(\frac{90° - \theta}{2}\right)\right]^C \tag{79}$$

$$A_\phi = C\phi \tag{80}$$

$$Y_0 = \psi \left[\tan\left(\frac{90° - \theta_a}{2}\right)\right]^C \tag{81}$$

where

$$\psi \equiv \frac{180°}{\pi}\frac{\cos\theta_1}{C\left[\tan\left(\frac{90°-\theta_1}{2}\right)\right]^C} \tag{82}$$

$$= \frac{180°}{\pi}\frac{\cos\theta_2}{C\left[\tan\left(\frac{90°-\theta_2}{2}\right)\right]^C} \tag{83}$$

$$C \equiv \frac{\ln\left(\frac{\cos\theta_2}{\cos\theta_1}\right)}{\ln\left[\frac{\tan\left(\frac{90°-\theta_2}{2}\right)}{\tan\left(\frac{90°-\theta_1}{2}\right)}\right]} . \tag{84}$$

Using Eqs. 45 and 46, the inverse equations are

$$\phi = \frac{A_\phi}{C} \tag{85}$$

$$\theta = 90° - 2\tan^{-1}\left(\left[\frac{R_\theta}{\psi}\right]^{(1/C)}\right) . \tag{86}$$

The expression for C becomes unwieldy when $\theta_1 = \theta_2$ and may be replaced with $C = \sin\theta_1$. Expressions in the average and difference angles appear less than useful in this case. This projection is illustrated in Fig. 20 with two different parameterizations. The keyword `PROJP1` will be used to give θ_a in degrees and the keyword `PROJP2` will be used to give α in degrees. The default value for `PROJP2` will be 0 while `PROJP1` has no default; it is recommended that both keywords always be given.

4.4. Conventional

Conventional projections are mathematical constructions designed to map the entire sphere with minimal distortion. By convention, the reference point in all of the conventional projections is the native coordinate system origin, $(\phi, \theta) = (0, 0)$.

4.4.1. BON (Bonne's equal area)

Bonne's equal-area polyconic projection is characterized by a single angle, $-90° \leq \theta_1 \leq 90°$. Parallels are concentric equidistant arcs of circles of true length. The projection formulæ are

$$x = R_\theta \sin A_\phi \tag{87}$$

$$y = Y_0 - R_\theta \cos A_\phi . \tag{88}$$

where

$$Y_0 = \frac{180°}{\pi}\cot\theta_1 + \theta_1$$

$$R_\theta = Y_0 - \theta$$

$$A_\phi = \frac{\phi\cos\theta}{\frac{\pi}{180°}R_\theta} .$$

The inverse formulæ are then

$$R_\theta = \mathrm{sign}\,(\theta_1)\sqrt{x^2 + (Y_0 - y)^2}$$

$$A_\phi = \arg\left(\frac{Y_0 - y}{R_\theta}, \frac{x}{R_\theta}\right)$$

$$\theta = Y_0 - R_\theta \tag{89}$$

$$\phi = \frac{\pi}{180°}A_\phi R_\theta \,/\cos(\theta) . \tag{90}$$

This projection is illustrated in Fig. 21. The keyword `PROJP1` will be used to give θ_1 in degrees and has no default. The case $\theta_1 = 0$ gives a divergent equation for R_θ, but not for (x, y). This case should be written as the `GLS` projection (which it is) rather than a `BON` in order to avoid posing any mathematical difficulties. To do otherwise is, we believe, worse than simply "bad form."

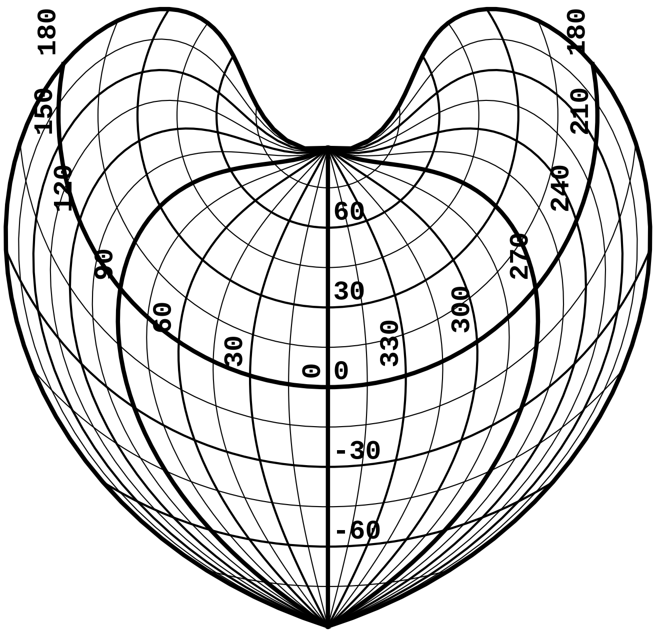

Fig. 21. `BON` projection: $\theta_1 = 45°$, no limits

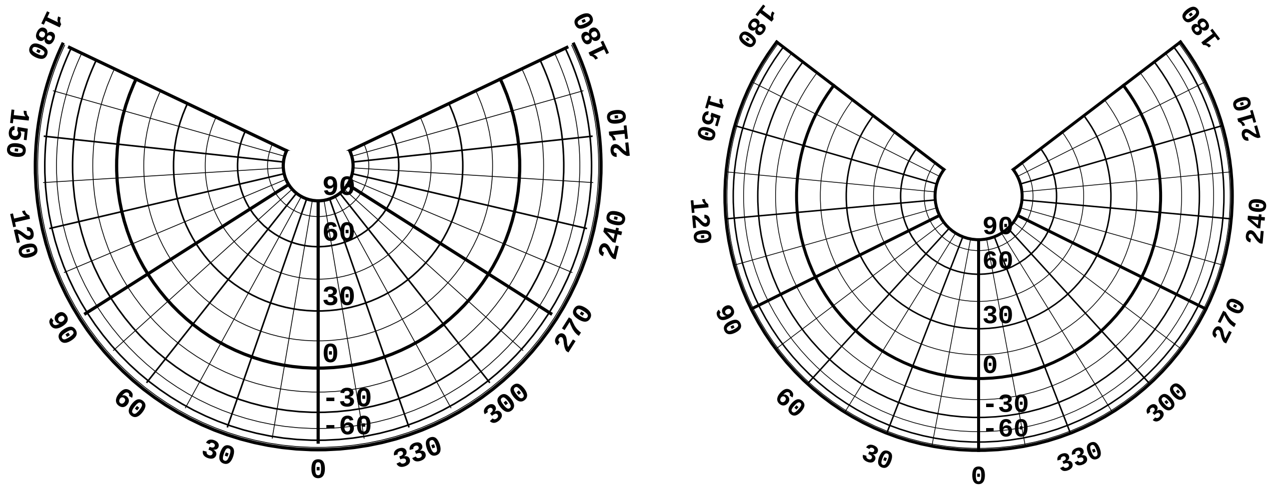

Fig. 19. COE projection: (left) $\theta_1 = 20°, \theta_2 = 70°$; (right) $\theta_1 = \theta_2 = 45°$; both have the same θ_a and no limits

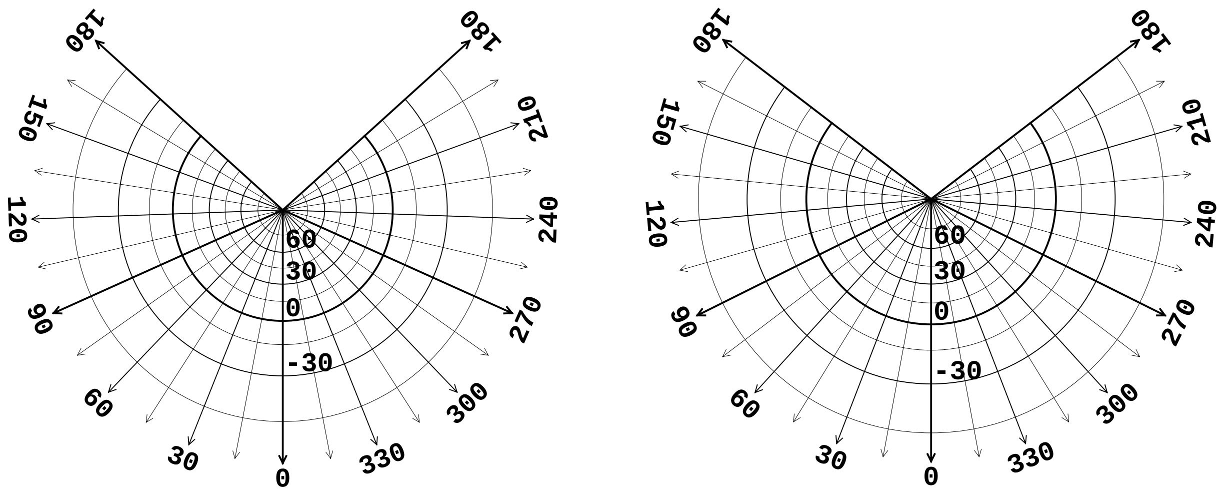

Fig. 20. COO projection: (left) $\theta_1 = 20°, \theta_2 = 70°$; (right) $\theta_1 = \theta_2 = 45°$; both have the same θ_a and diverge at $\theta = -90°$

4.4.2. PCO (polyconic)

In the polyconic projection, every parallel is projected as arcs of circles of radius $r \cot\theta$ at their true length, $2\pi r \cos\theta$, and correctly divided. The scale along the central meridian is true and consequently the parallels are not concentric. The projection formulæ must then be expressed in x and y and are

$$x = \frac{180°}{\pi} \cot\theta \sin(\phi \sin\theta) \tag{91}$$

$$y = \theta + \frac{180°}{\pi} \cot\theta \left[1 - \cos(\phi\sin\theta)\right] . \tag{92}$$

The inverse may not be expressed so simply, since it requires the solution of the transcendental equation

$$x^2 - \frac{360°}{\pi}(y - \theta)\cot\theta + (y - \theta)^2 = 0 \tag{93}$$

One may determine a solution iteratively with Newton's method (see Section 6) or, perhaps more simply, by assuming

$$\theta_0 = 1° \operatorname{sign}(y)$$

and computing

$$\theta_{i+1} = \tan^{-1}\left(\frac{\frac{360°}{\pi}(y - \theta_i)}{x^2 + (y - \theta_i)^2}\right)$$

for iteration i. Once θ is known, then

$$\phi = \frac{1}{\sin\theta} \arg\left(\frac{180°}{\pi} - (y - \theta)\tan\theta, x\tan\theta\right) .$$

A method of doing this iteration with better convergence properties is detailed in Appendix D.2. The polyconic projection is illustrated in Fig. 22.

4.4.3. GLS (Sanson-Flamsteed, sinusoidal)

Parallels are equispaced and their length is chosen to make an equal area projection. It is, in fact, the equatorial case

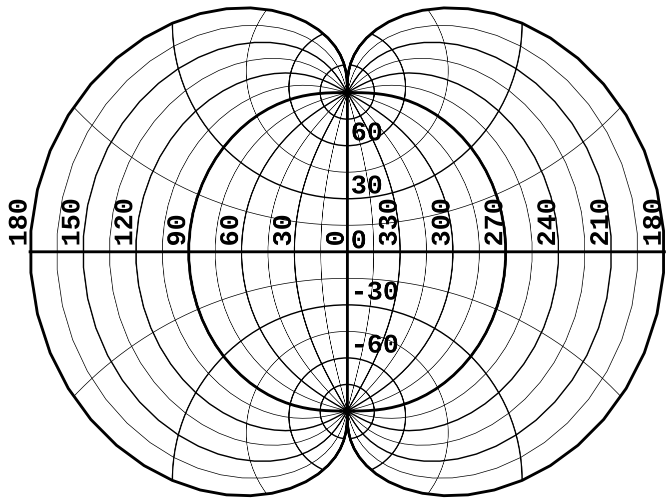

Fig. 22. `PCO` projection, no limits

($\theta_1 = 0$) of Bonne's projection. The formulæ are

$$x = \phi \cos\theta \tag{94}$$
$$y = \theta\,, \tag{95}$$

which reverse into

$$\phi = \frac{x}{\cos y} \tag{96}$$
$$\theta = y\,. \tag{97}$$

This projection is illustrated in Fig. 23.

4.4.4. PAR (parabolic projection)

In the parabolic or Craster projection, parallels are straight lines parallel to the equator while meridians are parabolic arcs. The scaling is set to make this also an equal-area projection. The formulæ are

$$x = \phi\left(2\cos\frac{2\theta}{3} - 1\right) \tag{98}$$
$$y = 180^\circ \sin\frac{\theta}{3}\,, \tag{99}$$

which has inverse formulæ

$$\theta = 3\sin^{-1}\left(\frac{y}{180^\circ}\right) \tag{100}$$
$$\phi = \frac{180^\circ}{\pi}\frac{x}{1 - 4(y/180^\circ)^2}\,. \tag{101}$$

This projection is illustrated in Fig. 24.

4.4.5. AIT (Hammer-Aitoff)

This projection is developed from the zenithal equal area projection by rotating the coordinates to the equator and then doubling the equatorial scale and longitude coverage (Calabretta, 1992). The whole sphere is mapped thereby while preserving the equal area property. The formulæ for the projection and its inverse are derived in Greisen (1986) and Calabretta (1992) among others. They are

$$x = 2\alpha\cos\theta\sin\frac{\phi}{2} \tag{102}$$
$$y = \alpha\sin\theta \tag{103}$$

where

$$\alpha \equiv \frac{180^\circ}{\pi}\sqrt{\frac{2}{1 + \cos\theta\cos(\phi/2)}}\,. \tag{104}$$

The reverse equations are

$$\phi = 2\arg\left(2Z^2 - 1, \frac{\pi}{180^\circ}\frac{Z}{2}x\right) \tag{105}$$
$$\theta = \sin^{-1}\left(\frac{\pi}{180^\circ}yZ\right) \tag{106}$$

where

$$Z \equiv \sqrt{1 - \left(\frac{\pi}{180^\circ}\frac{x}{4}\right)^2 - \left(\frac{\pi}{180^\circ}\frac{y}{2}\right)^2}\,. \tag{107}$$

This projection is illustrated in Fig. 25.[3]

4.4.6. MOL (Mollweide's)

In Mollweide's projection, meridians are projected as ellipses, parallels as straight lines with spacing chosen to make the projection equal area. The formulæ are

$$x = \frac{2\sqrt{2}}{\pi}\phi\cos\alpha \tag{108}$$
$$y = \sqrt{2}\frac{180^\circ}{\pi}\sin\alpha\,, \tag{109}$$

which reverse to

$$\phi = \pi x / \left(2\sqrt{2 - (\frac{\pi}{180^\circ}y)^2}\right) \tag{110}$$
$$\theta = \sin^{-1}\left(\frac{1}{90^\circ}\sin^{-1}\left(\frac{\pi}{180^\circ}\frac{y}{\sqrt{2}}\right) + \frac{y}{180^\circ}\sqrt{2 - (\frac{\pi}{180^\circ}y)^2}\right) \tag{111}$$

where α is defined as the solution of the transcendental equation

$$\sin\theta = \frac{\alpha}{90^\circ} + \frac{\sin 2\alpha}{\pi}\,. \tag{112}$$

This projection is illustrated in Fig. 26.

4.4.7. CSC (Cobe quadrilateralized spherical cube)

The quad-sphere projection was developed to represent the full sphere in 6 separate square faces, minimizing distortions while maintaining the equal area property. A new axis type is required to describe which of the 6 faces is

[3] This projections should be attributed to Hammer rather than Aitoff according to Jones (1993). The string `AIT` has been in use too long within the FITS community, however, for it to be replaced now.

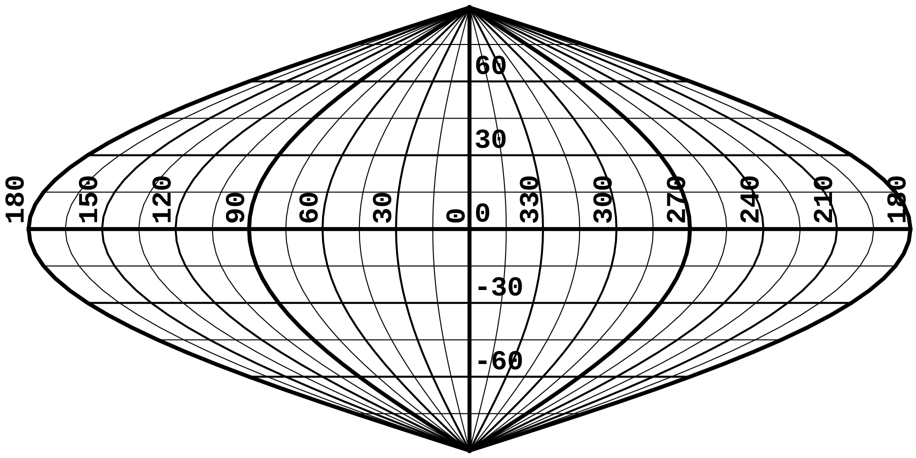

Fig. 23. GLS projection, no limits

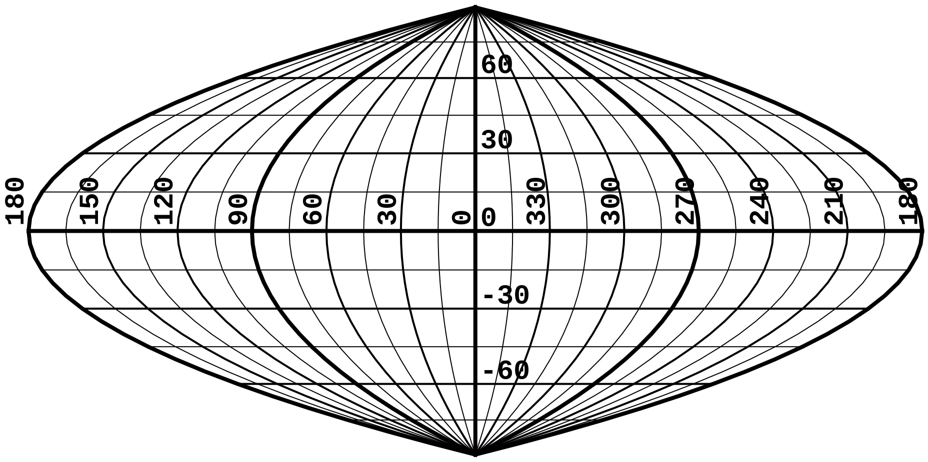

Fig. 24. PAR projection, no limits

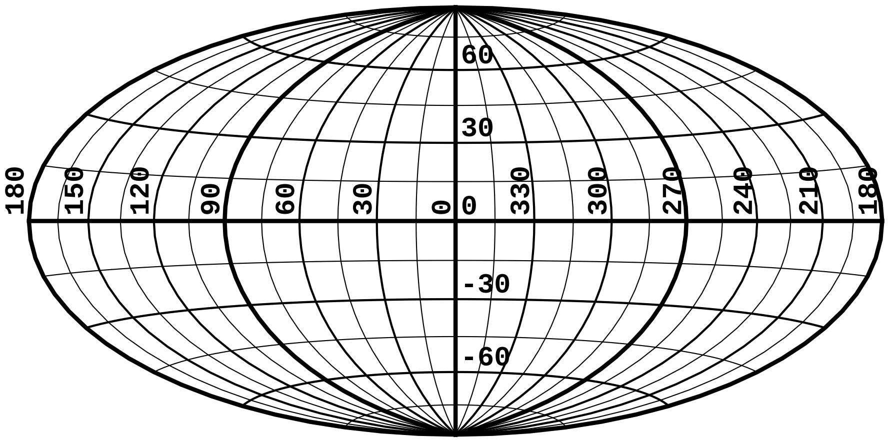

Fig. 25. AIT projection, no limits

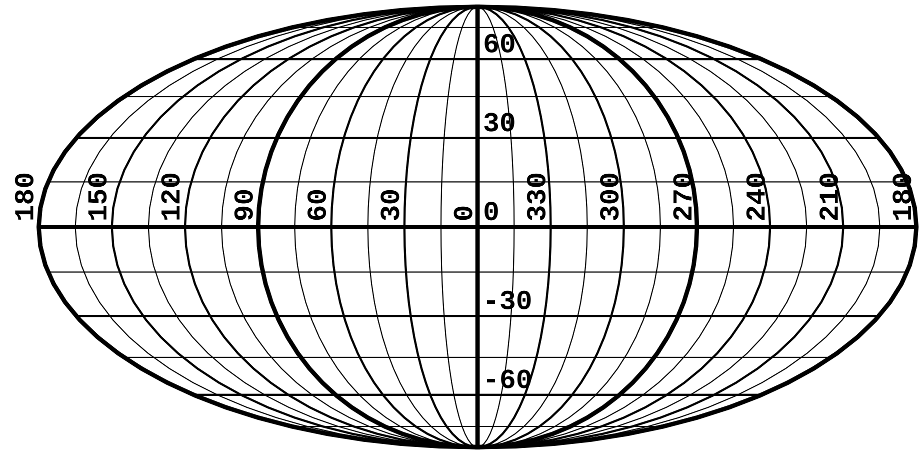

Fig. 26. `MOL` projection, no limits

being viewed in the data cube. The value used will be `'CUBEFACE'` for `CTYPE`n. By convention, the coordinate values at the reference point on the longitude and latitude axes are for the origin of the native coordinates of the projection which is located in the central face (1). The faces along the native equator are numbered 4, 3, 2, and 1 from left to right, while the northernmost is face number 0 and the southernmost is number 5 (Jack Saba, 1993, private communication). If the full globe is represented, then the axis dimensions alone determine the pixel coordinates of the reference point (in the numeric center of each face $= (N + 1.0)/2.0$) and the axis increments are, within a sign, $360°/(\pi\sqrt{3}(N-1))$, where N is the number of pixels on the axis.

O'Neill and Laubscher (1976) suggest a useful method to compute the projection. From the native longitude and latitude, (ϕ, θ), one computes the direction cosines

$$\begin{aligned} l &= \cos\theta\cos\phi \\ m &= \cos\theta\sin\phi \\ n &= \sin\theta\,. \end{aligned} \tag{113}$$

If ρ is the largest of $n, l, m, -l, -m$, and $-n$, then the face number is 0 through 5, respectively, and the parametric variables ξ, η and the central values of the native longitude and latitude, (ϕ_c, θ_c), may be found from Table 1. The scaled variables

$$\alpha = \xi/\rho \tag{114}$$

$$\beta = \eta/\rho \tag{115}$$

are used to find the (linear) relative array coordinates with

$$x = 45°\ F(\alpha, \beta) \tag{116}$$

$$y = 45°\ F(\beta, \alpha)\,, \tag{117}$$

where the function F is given by

$$\begin{aligned} F(\alpha,\beta) = {} & \alpha\gamma^* + \alpha^3\left(1-\gamma^*\right) \\ & +\alpha\beta^2\left(1-\alpha^2\right)\Bigg[\Gamma + (M-\Gamma)\alpha^2 \\ & \quad + \left(1-\beta^2\right)\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} C_{ij}\alpha^{2i}\beta^{2j}\Bigg] \\ & +\alpha^3\left(1-\alpha^2\right)\left[\Omega_1 - \left(1-\alpha^2\right)\sum_{i=0}^{\infty} D_i\alpha^{2i}\right]. \end{aligned} \tag{118}$$

The C_{ij} and D_i are derived from the c^*_{ij} and d^*_i given by Chan and O'Neill (1975), while the other parameters are given by exact formulæ developed by O'Neill and Laubscher (1976). The latter authors provide lists of the numeric values of their parameters in Appendix B and in software listings in Appendix E. Both disagree with their formulæ, but the software listings do contain the actual numeric parameters still in use for the Cobe Project (Immanuel Freedman, private communication, 1993). They are

$$\begin{aligned} \gamma^* &= 1.37484847732 \\ M &= 0.004869491981 \\ \Gamma &= -0.13161671474 \\ \Omega_1 &= -0.159596235474 \\ D_0 &= 0.0759196200467 \\ D_1 &= -0.0217762490699 \\ C_{00} &= 0.141189631152 \\ C_{10} &= 0.0809701286525 \\ C_{01} &= -0.281528535557 \\ C_{20} &= -0.178251207466 \\ C_{11} &= 0.15384112876 \\ C_{02} &= 0.106959469314\,, \end{aligned}$$

and should be considered to define the actual transformation as used by the Cobe Project (hence the name `CSC`). The projection so defined is illustrated in Fig. 27 for the full sphere in the natural coordinates. Note that, for normal observations, it is Eq. 118 which determines how accu-

Table 1. Assignment of parametric variables and central longitude and latitude by face number for quadrilateralized spherical cube projections.

Face	ξ	η	ρ	ϕ_c	θ_c	α	β
0	m	$-l$	n	$0°$	$90°$	m/n	$-l/n$
1	m	n	l	$0°$	$0°$	m/l	n/l
2	$-l$	n	m	$90°$	$0°$	$-l/m$	n/m
3	$-m$	n	$-l$	$180°$	$0°$	m/l	$-n/l$
4	l	n	$-m$	$270°$	$0°$	$-l/m$	$-n/m$
5	m	l	$-n$	$0°$	$-90°$	$-m/n$	$-l/n$

rately this projection satifies the equal-area requirement. Extensive tests show that this consideration is not important. A very high order inverse to Eq. 118 yields the same measures of equal-area conformance as we obtain for Eq. 121. Both are equal-area projections with an rms of 1.06% over the full field and 0.6% over the inner 64% of the area of each square. The maximum discrepancies from equal area are +13.7% and −4.1% at the edges of the field and only ±1.3% within the inner 64% of the area.

Chan and O'Neill (1975) first defined the "forward" direction of this conversion. (Note that, in the nomenclature of this paper, this is actually the inverse direction, *i.e.*, projected position to sky position.) Their formulation may be rewritten in a more convenient form which is now the current usage in the Cobe Project (Immanuel Freedman, private communication, 1993). This new form is

$$\alpha \equiv f(x,y) \tag{119}$$

$$\beta = f(y,x) \tag{120}$$

$$f(x,y) = X + X\left(1-X^2\right)\sum_{j=0}^{N}\sum_{i=0}^{N-j} P_{ij}X^{2i}Y^{2j} \tag{121}$$

with, X and Y redefined as

$$X \equiv \frac{x}{45°}$$

$$Y \equiv \frac{y}{45°},$$

and N as the limit to the series. For Cobe, $N = 6$ and the best-fit parameters have been taken to be

$$\begin{aligned}
P_{00} &= -0.27292696 & P_{04} &= 0.93412077\\
P_{10} &= -0.07629969 & P_{50} &= 0.25795794\\
P_{01} &= -0.02819452 & P_{41} &= 1.71547508\\
P_{20} &= -0.22797056 & P_{32} &= 0.98938102\\
P_{11} &= -0.01471565 & P_{23} &= -0.93678576\\
P_{02} &= 0.27058160 & P_{14} &= -1.41601920\\
P_{30} &= 0.54852384 & P_{05} &= -0.63915306\\
P_{21} &= 0.48051509 & P_{60} &= 0.02584375\\
P_{12} &= -0.56800938 & P_{51} &= -0.53022337\\
P_{03} &= -0.60441560 & P_{42} &= -0.83180469\\
P_{40} &= -0.62930065 & P_{33} &= 0.08693841\\
P_{31} &= -1.74114454 & P_{24} &= 0.33887446\\
P_{22} &= 0.30803317 & P_{15} &= 0.52032238\\
P_{13} &= 1.50880086 & P_{06} &= 0.14381585\,.
\end{aligned}$$

Since we now have the face number, α, and β, we have, with the help of Table 1, the ratio of two of the direction cosines to the third and may, thereby, determine the native latitude and longitude.

Although this projection produces pleasing full-sky images, it is clearly very difficult to use in practice. These functions are clearly inexact and, in fact, have not converged very well with only these terms. One may evaluate the error produced by converting (x,y) to (α,β) with Eq. 121 followed by converting the (α,β) found back to (x,y) with Eq. 118,*i.e.*,

$$\begin{aligned}
E_{ij}^2 \equiv\ & \left(F(f(x_i,y_j),f(y_j,x_i)) - x_i\right)^2\\
& + \left(F(f(y_j,x_i),f(x_i,y_j)) - y_j\right)^2 .
\end{aligned}$$

The Cobe parameterization produces an average error of 4.7 arc seconds over the full field. The root mean square and peak errors are 6.6 and 24 arc seconds, respectively. In the central parts of the image ($|X|,|Y| \leq 0.8$), the average and root mean square errors are 5.9 and 8.0 arc seconds, larger than for the full field. One may also write Eq. 118 in the form of Eq. 121 and then determine optimal parameters equivalent to the P_{ij} to minimize, say, the sum of E_{ij}^2 over the full field. The Cobe formulæ produce errors similar to $N = 4$. Limiting ourselves to $N = 6$, as in the Cobe usage of Eq. 121, changes the error to average 3.1,

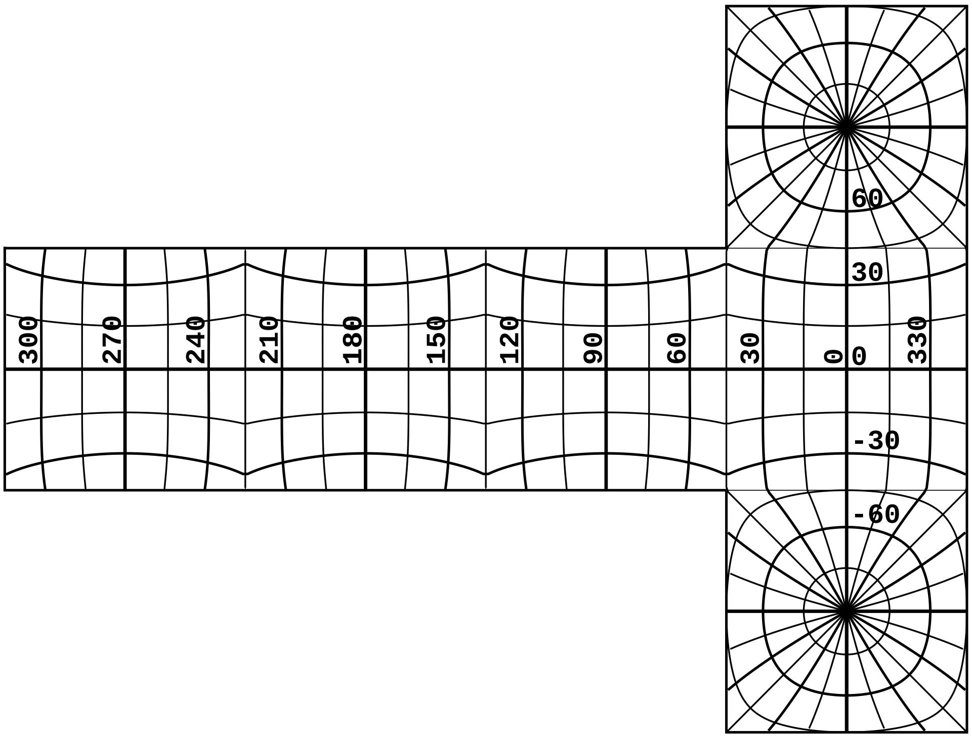

Fig. 27. `CSC` projection, no limits

root mean square 5.4, and peak 22 arc seconds over the full field. Allowing $N = 9$, reduces these numbers to 0.74, 0.92, and 9.6 arc seconds, respectively. In the central regions, the peak errors drop to 14 and 2 arc seconds for $N = 6$ and 9.

4.4.8. QSC (quadrilateralized spherical cube)

O'Neill and Laubscher (1976) derived an exact expression for an equal-area transformation from a sphere to the six faces of a cube. At that time, their formulation was thought to be computationally intractable, but, today, with modern computers and telescopes of higher angular resolution than Cobe, their formulation has come into use. Fred Patt (1993, private communication) has provided us with the inverse of the O'Neill and Laubscher formula and their expression in cartesian coordinates.

As with the `CSC` projection, the new axis type `'CUBEFACE'` as the value for `CTYPE`n is required to describe which of the 6 faces is being viewed in the data cube. By convention, the coordinate values at the reference point on the longitude and latitude axes are for the origin of the native coordinates of the projection which is located in the central face (1). The faces along the native equator are numbered 4, 3, 2, and 1 from left to right, while the northernmost is face number 0 and the southernmost is number 5 (Jack Saba, 1993, private communication). If the full globe is represented, then the axis dimensions alone determine the pixel coordinates of the reference point (in the numeric center of each face $= (N + 1.0)/2.0$) and the axis increments are, within a sign, $360^\circ/(\pi\sqrt{3}(N-1))$, where N is the number of pixels on the axis.

O'Neill and Laubscher (1976) suggest a useful method to compute the projection. From the native longitude and latitude, (ϕ, θ), one computes the direction cosines by Eq. 113. Then, if ρ is the largest of $n, l, m, -l, -m$, and $-n$, the face number is 0 through 5, respectively, and the parametric variables α and β may be found from Table 1. The position variables within each face are (x_f, y_f), with $(0, 0)$ at the center, x_f increases to the left, and y_f increases upward, in the usual manner. O'Neill and Laubscher's derivation applies only in the quadrant $-45^\circ \leq \phi \leq 45^\circ$ and must be reflected into the other quadrants. This has the affect of making the projection non-differentable along the diagonals. Patt's formulæ in our nomenclature are

$$u = 45S\sqrt{\frac{1-\rho}{1-1/\sqrt{2+\zeta^2}}} \tag{122}$$

$$v = \frac{u}{15}\left[\tan^{-1}(\zeta) - \sin^{-1}\left(\frac{\zeta}{\sqrt{2(1+\zeta^2)}}\right)\right], \tag{123}$$

where

$$u \equiv \begin{cases} x & \text{if } |\alpha| > |\beta| \\ y & \text{if } |\alpha| \leq |\beta| \end{cases}$$

$$v \equiv \begin{cases} y & \text{if } |\alpha| > |\beta| \\ x & \text{if } |\alpha| \le |\beta| \end{cases}$$

$$\zeta \equiv \begin{cases} \beta/\alpha & \text{if } |\alpha| > |\beta| \\ \alpha/\beta & \text{if } |\alpha| \le |\beta| \end{cases}$$

$$S \equiv \begin{cases} +1 & \text{if } \alpha > |\beta| \text{ or } \beta > |\alpha| \\ -1 & \text{if } -\alpha > |\beta| \text{ or } -\beta > |\alpha| \end{cases}$$

The inverse formulæ are then

$$\rho = 1 - \left(\frac{u}{45}\right)^2 \left(1 - \frac{1}{\sqrt{2+\zeta^2}}\right) \qquad (124)$$

$$\zeta = \frac{\sin(15v/u)}{\cos(15v/u) - 1/\sqrt{2}} \qquad (125)$$

The projection is illustrated in Fig. 28 for the full sphere in the natural coordinates.

Although this projection produces pleasing full-sky images, it is clearly difficult to use in practice. The `QSC` has most of the pleasing aspects of the `CSC` projection, plus it has an exact inverse and is exactly equal area. However, it has discontinuous derivative along the diagonals of each face.

4.4.9. TSC (tangential spherical cube)

In a manner similar to the quad-sphere projections (Sections 4.4.7 and 4.4.8), one may represent the full sphere in 6 separate square faces with a gnomonic projection in each face. The new axis type `'CUBEFACE'` as the value for `CTYPE`n is required to describe which of the 6 faces is being viewed in the data cube. By convention, the coordinate values at the reference point on the longitude and latitude axes are for the origin of the native coordinates of the projection which is located in the central face (1). The faces along the native equator are numbered 4, 3, 2, and 1 from left to right, while the northernmost is face number 0 and the southernmost is number 5 (Jack Saba, 1993, private communication). If the full globe is represented, then the axis dimensions alone determine the pixel coordinates of the reference point (in the numeric center of each face $= (N + 1.0)/2.0$) and the axis increments are, within a sign, $360°/(\pi\sqrt{3}(N-1))$, where N is the number of pixels on the axis.

O'Neill and Laubscher (1976) suggest a useful method to compute the projection. From the native longitude and latitude, (ϕ, θ), one computes the direction cosines by Eq. 113. Then, if ρ is the largest of $n, l, m, -l, -m$, and $-n$, the face number is 0 through 5, respectively, and the parametric variables ξ and η may be found from Table 1. The position variables within each face are (x_f, y_f), with $(0,0)$ at the center, x_f increases to the left, and y_f increases upward, in the usual manner. They are given by

$$x_f = R_{\theta_f} \sin A_{\phi_f} \qquad (126)$$

$$y_f = -R_{\theta_f} \cos A_{\phi_f} , \qquad (127)$$

where

$$R_{\theta_f} = 45 \cot\left(\sin^{-1}(\rho)\right) \qquad (128)$$

$$A_{\phi_f} = \arg(-\eta, \xi) . \qquad (129)$$

A less intuitive, but simpler, form of these is

$$x_f = 45\,\frac{\xi}{\rho} \qquad (130)$$

$$y_f = 45\,\frac{\eta}{\rho} . \qquad (131)$$

The projection is illustrated in Fig. 29 for the full sphere in the natural coordinates. To do the inverse, we take the face number, R_{θ_f}, and A_{ϕ_f}, and, with the help of Table 1, we get one of the direction cosines and the ratio of the other two may, thereby, determine the native latitude and longitude.

Although this projection produces pleasing full-sky images, it is clearly difficult to use in practice. The `TSC` has some of the pleasing aspects of the `CSC` projection, plus it has an exact inverse and projects great circles into piecewise straight lines. However, it does not conserve area, length, or shape.

5. Converting previous formats

There exists a large number of FITS images written using the old AIPS conventions and a substantial body of software which understands those conventions. Although the NRAO and other institutions will probably alter their software systems to understand and generate the new conventions proposed here, many of their user sites will not be able to update their versions of AIPS, *et al.* in so timely a manner. Thus, for a few years, FITS-writing programs which use the new conventions should also write the `CROTA` keyword whenever possible. And, of course, FITS-reading programs will need to understand the old conventions virtually forever. If *no* `PC`$nnnmmm$ keywords are given in the header, then, for longitude axis i and latitude axis j, reading programs should set

$$\begin{pmatrix} \mathtt{PC}iiiiii & \mathtt{PC}iiijjj \\ \mathtt{PC}jjjiii & \mathtt{PC}jjjjjj \end{pmatrix} = \begin{pmatrix} \cos(\mathtt{CROTA}j) & -\sin(\mathtt{CROTA}j)\,S \\ \sin(\mathtt{CROTA}j)\,/\,S & \cos(\mathtt{CROTA}j) \end{pmatrix} \qquad (132)$$

where $S = \mathtt{CDELT}j/\,\mathtt{CDELT}i$. We note that the expressions in Hanisch and Wells (1988) and Geldzahler and Schlesinger (1992) are for the product of the `PC`$nnnmmm$ matrix and the `CDELT`n matrices. They yield the same results as Eq. 132 for the usual left-handed sky coordinates and right-handed pixel coordinates, but can lead to an incorrect interpretation (namely, possible sign errors for the cross terms in the `PC` matrix) for other configurations of the coordinate systems. The Hanisch and Wells draft and the coordinate portions of Geldzahler and Schlesinger are superceded by the present manuscript. In the presence of rotation or skew, modern FITS-writing programs should always write the `PC`$nnnmmm$ keywords. To assist older

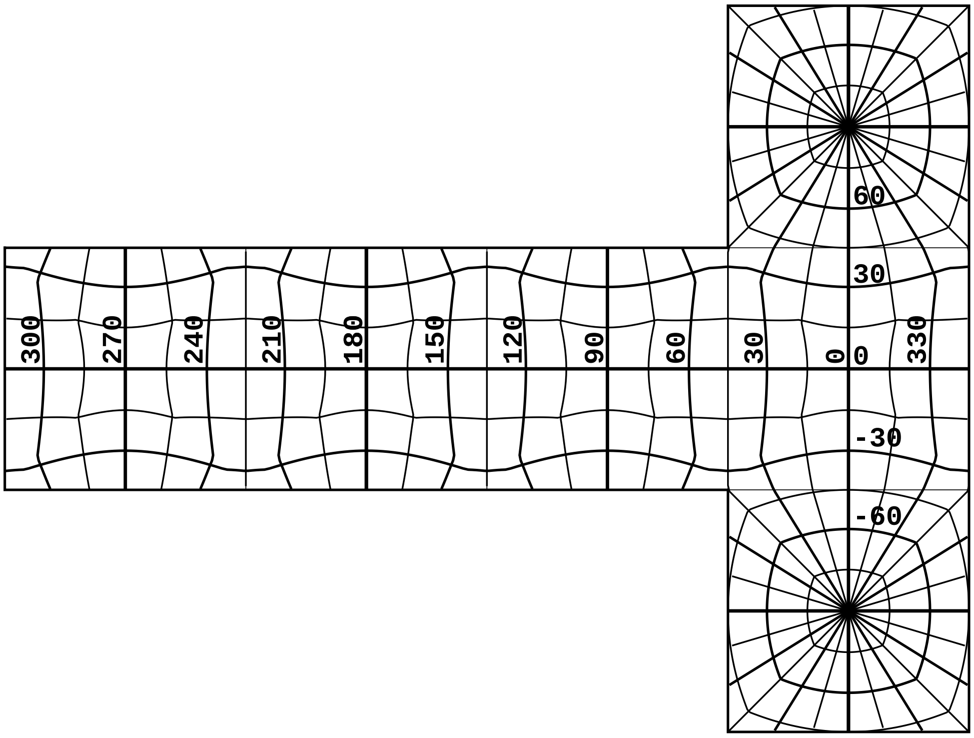

Fig. 28. `QSC` projection, no limits

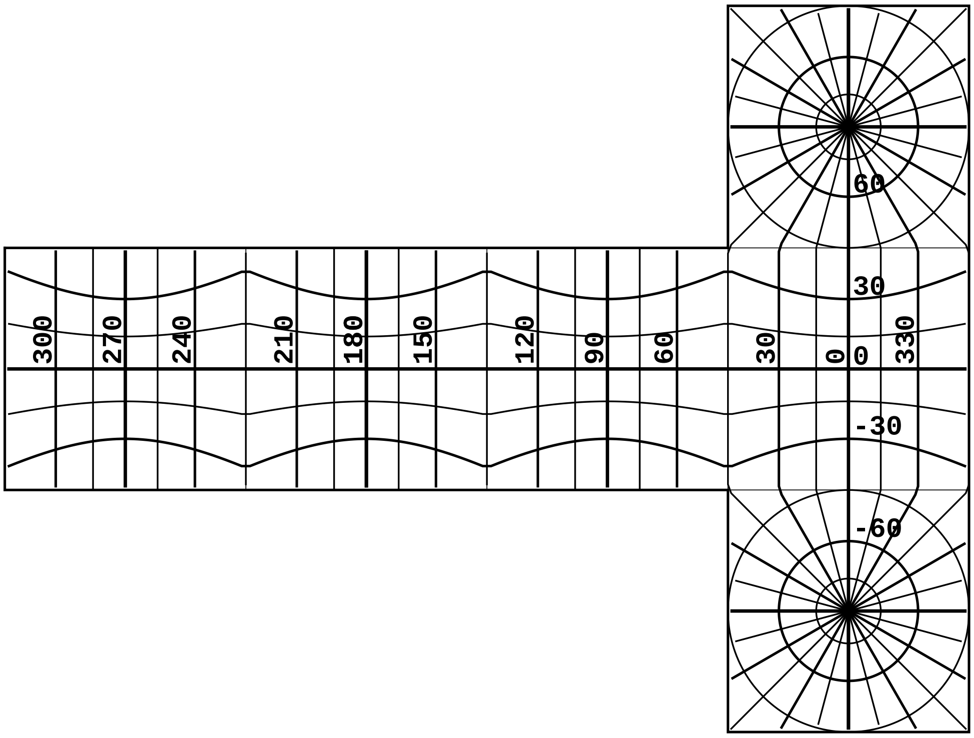

Fig. 29. `TSC` projection, no limits

FITS-reading programs, for a few years, these modern FITS-writing programs should also write CROTAj for latitude axis j when the longitude/latitude pair fits Eq. 132. The fit to Eq. 132 may be determined by evaluating both

$$\begin{aligned}\rho_i &= \arg(\mathtt{PC}iiiiii, \mathtt{PC}jjjiii\, S)\\ \rho_j &= \arg(\mathtt{PC}jjjjjj, -\mathtt{PC}iiijjj/S)\ .\end{aligned}$$

If $\rho_i = \rho_j$ within reasonable precision (Geldzahler and Schlesinger, 1992), then the older keyword should be written using

$$\mathtt{CROTA}j = (\rho_i + \rho_j)/2 \qquad (133)$$

where i is the longitude axis and j the latitude axis.

The "north-celestial-pole" coordinate representation established by Greisen (1983) is not one of the projections proposed above, should no longer be written, and should be translated by modern FITS-reading programs. There are two possible translations of this old system. The simpler regards the '-NCP' system as a slant orthographic projection. In this case, if the longitude and latitude axis numbers are m, n, respectively, then the parameters to be translated and the new parameters are

$$\begin{aligned}\mathtt{CTYPE}m &= \mathtt{'RA---SIN'}\\ \mathtt{CTYPE}n &= \mathtt{'DEC--SIN'} \qquad (134)\\ \mathtt{PROJP1} &= 0\\ \mathtt{PROJP2} &= \cot\delta_{NCP}\end{aligned}$$

One may also regard the '-NCP' system as an orthographic projection with the north equatorial pole as the tangent point, but with the center of the observation declared to be the reference point. From this point of view, the parameters which need to be translated are

$$\begin{aligned}\mathtt{CTYPE}m &= \mathtt{'RA---SIN'}\\ \mathtt{CTYPE}n &= \mathtt{'DEC--SIN'}\\ \mathtt{CRVAL}n &= 90\,\mathrm{sign}(\delta_{NCP})\\ \mathtt{CRPIX}m &= m_{NCP} + \frac{180^\circ}{\pi}\cot\delta_{NCP}\sin\rho/\Delta_m\\ \mathtt{CRPIX}n &= n_{NCP} + \frac{180^\circ}{\pi}\cot\delta_{NCP}\cos\rho/\Delta_n \qquad (135)\\ \mathtt{PC}mmmmmm &= \cos\rho\\ \mathtt{PC}mmmnnn &= -\Delta_n\sin\rho/\Delta_m\\ \mathtt{PC}nnnmmm &= \Delta_m\sin\rho\,|\sin\delta_{NCP}|/\Delta_n\\ \mathtt{PC}nnnnnn &= \cos\rho\,|\sin\delta_{NCP}|\\ \mathtt{CDELT}m &= \Delta_m\\ \mathtt{CDELT}n &= \Delta_n\ ,\end{aligned}$$

where m_{NCP}, n_{NCP} (old values for keywords CRPIXm and CRPIXn) are the original pixel coordinates of the reference point on the longitude and latitude (right ascension and declination) axes, $\alpha_{NCP}, \delta_{NCP}$ (old values for keywords CRVALm and CRVALn) are the coordinate values at the reference point, and Δ_m, Δ_n (old values for keywords CDELTm and CDELTn) are the increments on those axes, ρ = the old value for the old keyword CROTAn. Note that this translation requires the PC nomenclature when $\rho \neq 0$ since the rotation is offset from the tangent point. Note that, when $\rho = 0$, one can also set the PC matrix to the unity matrix and CDELTn to $\Delta_n \sin\delta_{NCP}$.

6. Tutorial discussion

The material so far presented has been quite general and all encompassing. For tutorial purposes, let us consider a concrete example of a "normal" optical image. Its header could look like Table 2. The NAXIS and NAXISn keywords tell us that we have a four-dimensional image consisting of 512 columns, 512 rows, 196 planes, and one cube. The last axis is used simply to convey another coordinate value even though the entire image has that value. One-point axes should be put at the end of the list for simplicity, but they are allowed to be anywhere within the list. The CRPIXn keywords tell us that the reference point is at the center of the pixel in the 256$^{\text{th}}$ column, 257$^{\text{th}}$ row, first plane, and first cube. The PC keywords all have their default values implying no rotation or skew in the coordinates.

Thus, the relative coordinate vector may be computed as shown in Table 3 from Eq. 3. Since 'VELOCITY' and 'STOKES' are linear axes, the actual velocity and Stokes of each point are found simply by adding the coordinate value
at the reference point to the relative coordinate. Thus,

Velocity	=	$500000. + 7128.3(k-1)$	meters/sec
Stokes	=	$1.0 + (l-1)$	= I polariz.

The spherical coordinates are not as simple. The CTYPEn keywords tell us that the projection is of type TAN which is zenithal projection with the source of the projection at the center of the sphere. The reference point is the tangent point which is the north pole of the native coordinate system. The native longitude and latitude are then given by

$$\begin{aligned}\phi &= \arg(-y, x)\\ \theta &= \tan^{-1}\left(\frac{180^\circ}{\pi}\frac{1}{\sqrt{x^2+y^2}}\right)\ ,\end{aligned}$$

which, on substitution, become

$$\begin{aligned}\phi &= \arg(j-257, i-256) + 180^\circ\\ \theta &= \tan^{-1}\left(\frac{20626.48062}{\sqrt{(i-256)^2+(j-257)^2}}\right)\ ,\end{aligned}$$

using Eqs. 136, 12, 17 and 13. The recognized coordinate system is equatorial since the CTYPEn begin with RA and DEC and the reference point has equatorial coordinates 45.°83 in right ascension and 63.°57 in declination from the CRVALn. The equatorial north pole has a longitude of

Table 2. Example FITS header with coordinates

```
NAXIS    =                    4      / 4-dimensional cube
NAXIS1   =                  512      / x axis (fastest)
NAXIS2   =                  512      / y axis (2nd fastest)
NAXIS3   =                  196      / z axis (planes)
NAXIS4   =                    1      / dummy to give a coordinate
CTYPE1   = 'RA---TAN'                / TAN projection used
CRVAL1   =                45.83      / RA at reference point
CRPIX1   =                  256      / pixel i of reference point
CDELT1   =          -0.00277777      / 10 arcsec per pixel
PC001001 =                  1.0      / no rotation, skew
CTYPE2   = 'DEC--TAN'                / TAN projection used
CRVAL2   =                63.57      / Dec at reference point
CRPIX2   =                  257      / pixel j of reference point
CDELT2   =           0.00277777      / 10 arcsec per pixel
PC002002 =                  1.0      / no rotation, skew
CTYPE3   = 'VELOCITY'                / each plane at a velocity
CRVAL3   =             500000.0      / velocity in m/sec
CRPIX1   =                    1      / pixel k of reference point
CDELT3   =               7128.3      / 7128.3 m/sec
PC003003 =                  1.0      / no rotation, skew
CTYPE4   = 'STOKES  '                / Polarization
CRVAL4   =                    1      / unpolarized
CRPIX4   =                    1      / at our dummy plane
CDELT4   =                    1      / or anything here.
PC004004 =                  1.0      / no rotation, skew
LONGPOLE =                  180      / native longitude of equatorial pole
```

Table 3. Example FITS header coordinate computation

$$\begin{pmatrix} x \\ y \\ z \\ s \end{pmatrix} = \begin{pmatrix} \mathtt{CDELT1} & 0 & 0 & 0 \\ 0 & \mathtt{CDELT2} & 0 & 0 \\ 0 & 0 & \mathtt{CDELT3} & 0 \\ 0 & 0 & 0 & \mathtt{CDELT4} \end{pmatrix} \begin{pmatrix} \mathtt{PC001001} & \mathtt{PC001002} & \mathtt{PC001003} & \mathtt{PC001004} \\ \mathtt{PC002001} & \mathtt{PC002002} & \mathtt{PC002003} & \mathtt{PC002004} \\ \mathtt{PC003001} & \mathtt{PC003002} & \mathtt{PC003003} & \mathtt{PC003004} \\ \mathtt{PC004001} & \mathtt{PC004002} & \mathtt{PC004003} & \mathtt{PC004004} \end{pmatrix} \begin{pmatrix} i - i_0 \\ j - j_0 \\ k - k_0 \\ l - l_0 \end{pmatrix}$$

which becomes

$$\begin{pmatrix} x \\ y \\ z \\ s \end{pmatrix} = \begin{pmatrix} -0.00277777 & 0 & 0 & 0 \\ 0 & 0.00277777 & 0 & 0 \\ 0 & 0 & 7128.3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i - 256 \\ j - 257 \\ k - 1 \\ l - 1 \end{pmatrix} . \tag{136}$$

180° in the native coordinate system from the `LONGPOLE` keyword. Since this is a zenithal projection, `LATPOLE` is not given. Thus, Eqs. 6 for the right ascension and declination become

$$\begin{aligned} \sin\delta &= \sin\theta\sin(63.57) - \cos\theta\cos\phi\cos(63.57) \\ \cos\delta\sin(\alpha - 45.83) &= \cos\theta\sin\phi \\ \cos\delta\cos(\alpha - 45.83) &= \sin\theta\cos(63.57) \\ &\quad + \cos\theta\cos\phi\sin(63.57)\,. \end{aligned}$$

Therefore the coordinates of, for example, the first and last corners of the image are

parameter	units	first corner	last corner
(i,j)	pixels	(0.5, 0.5)	(512.5, 512.5)
(k,l)	pixels	(0.5, 1.0)	(196.5, 1.0)
ϕ	degrees	44.888	225.112
θ	degrees	88.9944412	88.9944412
α	degrees	47.3852040	44.1887934
δ	degrees	62.8489681	64.2704912
Velocity	m/s	496435.85	1893582.65
Stokes	type	$1.0 \equiv$ I	$1.0 \equiv$ I

Even in this large ($\sim 2°$ diagonal) image, the effect of the projection non-linearity is only 0.04 pixels at the corners. If the pixel separation were ten times larger, then the projection non-linearity would be 3.4 pixels at the corners which is a thousand times larger. The right ascension offset from the reference point is affected by the $\cos\delta$ term and is, therefore, quite nonlinear even in this fairly small image. The two $|\alpha - \alpha_P|$ differ by 20.66 seconds of time (309.8 arc seconds). This coordinate rotation nonlinearity is small for reference points near the equator and large for points near the poles.

It is clear that non-linearities are very important in the all-sky and other large-field projections, but it is less clear that the projective non-linearities matter for small fields. The `SIN` projection always has an R_θ less than the the other zenithal projections. Therefore, we have plotted the difference between the R_θ for various projections and that for the `SIN` projection as a function of natural latitude θ in Fig. 30 with the small angles displayed on a different scale in Fig. 31. We note that the `TAN` projection blows up at $\theta = 0$, while the differences for the other projections approach some 10's of degrees. The difference for all projections exceeds 1 arc second for $\theta < 88°$ and the difference for the `TAN` projection exceeds one milliarcsecond only 440 arc seconds from the natural pole. Such nonlinearities would be significant in radio VLB and other very-high resolution observations.

The role of the coordinate values at the reference point and of `LONGPOLE` depends on the sort of imaging being done. The zenithal projections called `TAN`, `ARC` and `SIN` are the natural result of certain techniques of observation and the coordinate values at the reference point are the tangent point of those observations (usually the telescope pointing position). The longitude of the pole parameter then conveys any rotation of the scene which the imaging instrument may have caused. Geometries which cover large areas of the sky have some regions which are little distorted in either area, angles, or shape and other areas which are highly distorted in two or more of these. By selecting the reference point location and the rotation of the scene, one can alter which portions of the sky are highly distorted and which are represented fairly reliably. Figure 32 shows the zenithal equal area projection with four different choices of reference parameters. The same parameter choices are illustrated for the conical equidistant projection in Fig. 33, the Hammer-Aitoff projection in Fig. 34, and Cobe's Quadrilateralized Spherical Cube in Fig. 35.

In writing computer programs to handle coordinates, it is relatively simple to write ones which do the conversions from pixel numbers (i,j) to standard celestial (α,δ) coordinates and the inverse using the four steps described in Section 4. However, in a variety of circumstances such as the plotting of tick marks on image displays, the program knows i and δ, for example, and requires j and α. For such cases, it would be convenient to combine the four steps into two equations in the four variables, so that they may be solved for any two given the other two. For some of the projections, this may be done analytically, but others are not so simple. We present some of the details of such computations in Appendix C. For the more difficult cases, iterative techniques will be needed. We remind readers that the simplest powerful iterative technique is Newton's Method, namely, at the $i+1$ iteration,

$$\begin{aligned} x_{i+1} &= x_i - \left[\frac{f g_y - g f_y}{f_x g_y - g_x f_y}\right]_{(x_i,y_i)} \\ y_{i+1} &= y_i - \left[\frac{g f_x - f g_x}{f_x g_y - g_x f_y}\right]_{(x_i,y_i)} , \end{aligned} \tag{137}$$

where we are seeking the roots of

$$\begin{aligned} f(x,y) &= 0 \\ g(x,y) &= 0 , \end{aligned}$$

and the x and y subscripts refer to differentiation of the function by x and y, respectively. In one dimension, this simplifies to

$$x_{i+1} = x_i - \left[\frac{f(x_i)}{f_x(x_i)}\right] , \tag{138}$$

where we are seeking the roots of

$$f(x) = 0 .$$

Another useful relationship to remember is that, if

$$0 = \beta \sin a + \gamma \cos a - Z ,$$

then

$$a = \sin^{-1}\left(\frac{Z}{\sqrt{\beta^2+\gamma^2}}\right) - \arg(\beta,\gamma) .$$

7. Summary

The changes to FITS-header keywords are summarized in Table 4. As described in Section 5, some of the deprecated keywords should be used along with the new keywords so that the coordinate information may be understood by software systems which have yet to be converted to these new conventions.

The last three characters of `CTYPE`i keywords give the type of projection used for spherical coordinates. The 3-letter codes proposed in this paper are listed in Table 5.

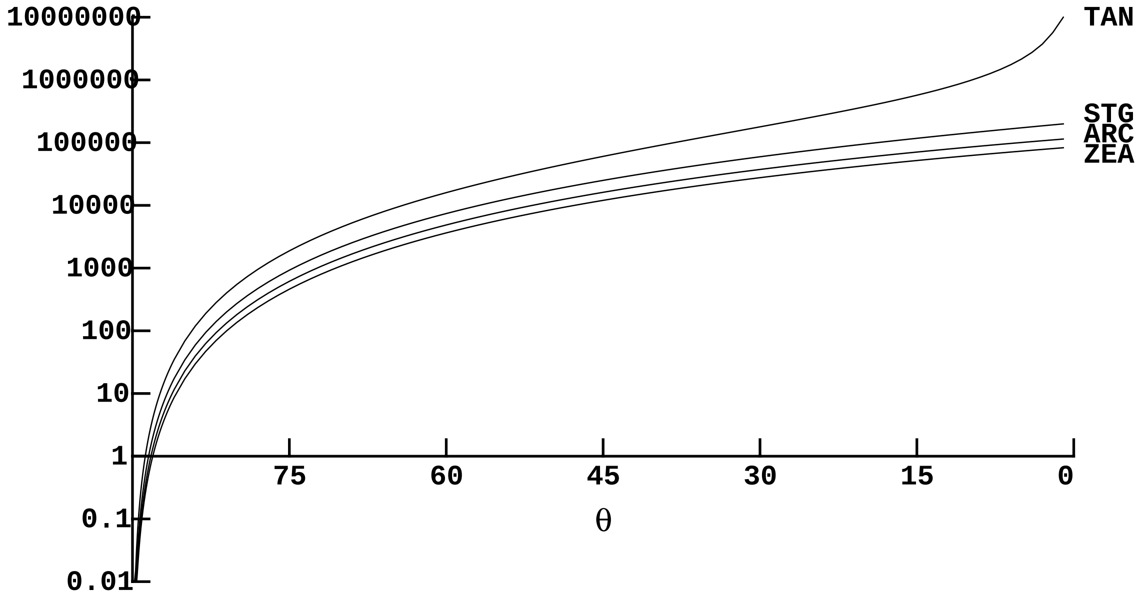

Fig. 30. $R_\theta(projection) - R_\theta(\texttt{SIN})$ in arc seconds plotted versus θ for various projections.

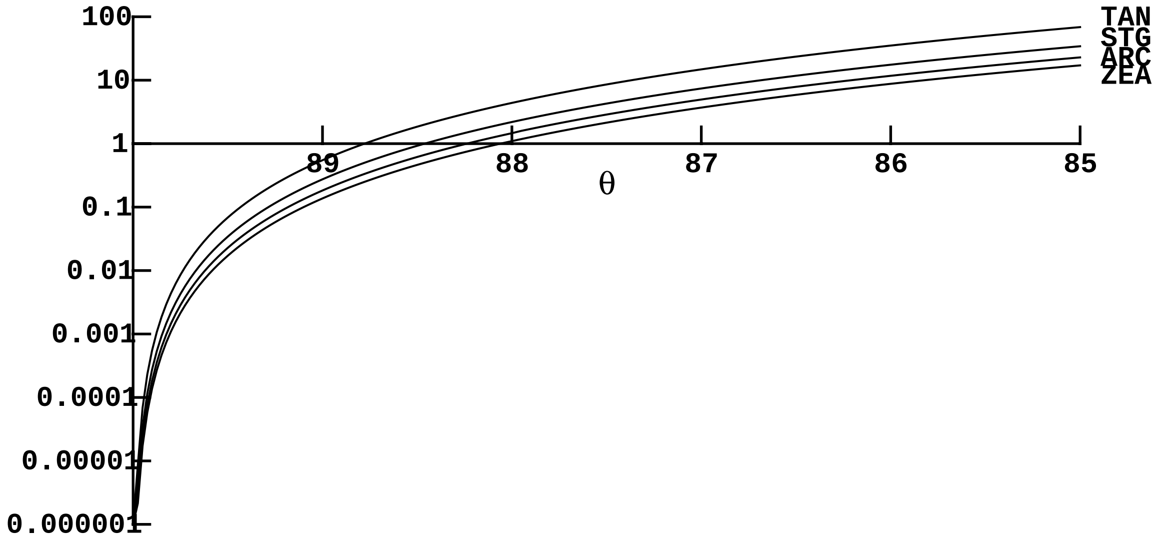

Fig. 31. $R_\theta(projection) - R_\theta(\texttt{SIN})$ in arc seconds plotted versus θ for various projections (detail).

The column labeled RP gives the native longitude of the reference point in degrees. The required projection parameters are listed in the nomenclature used previously. When no projection parameter is listed, none should be given. Some of the "standard" projections which are simply special cases of these projections are listed in Table 6. Evenden (1991) was particularly useful in making this list of synonyms.

Calabretta (1995) has written, and made available under a GNU license, a package of C and of Fortran subroutines implementing all projections and coordinate conversions proposed in this manuscript.

8. Acknowledgments

The authors thank Patrick Wallace (U.K. Starlink) for his helpful comments which included the text for some of the paragraphs of Section 3. We also thank William Pence, Arnold Rots, and Lorella Angelini (NASA Goddard Space Flight Center) for contributing the text of Appendix B. We are also most grateful to Rick Balsano (Los Alamos National Laboratory) for his detailed comments and corrections. Mr. Balsano has developed a complete set of scripts in IDL for converting between celestial coordinates and the (x, y) coordinates of the image pro-

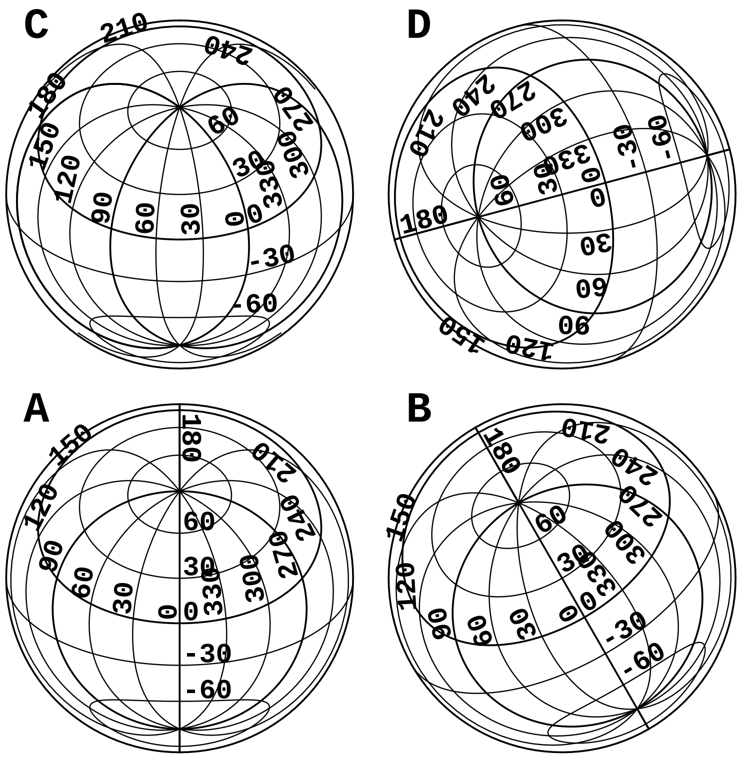

Fig. 32. `ZEA` projection plotted with lines in α, δ with parameters (bottom left: A) $\alpha_P = 0$, $\delta_P = 30°$, $\phi_P = 180°$, ($\alpha_0 = 0$, $\delta_0 = -60°$); (bottom right: B) $\alpha_P = 0$, $\delta_P = 30°$, $\phi_P = 150°$, ($\alpha_0 = 49.1°$, $\delta_0 = -48.6°$); (top left: C) $\alpha_P = 45°$, $\delta_P = 30°$, $\phi_P = 180°$, ($\alpha_0 = 45°$, $\delta_0 = -60°$); (top right: D) $\alpha_P = 0$, $\delta_P = 30°$, $\phi_P = 75°$, ($\alpha_0 = 97.6°$, $\delta_0 = 13.0°$).

jections described in this manuscript under the direction of J. Bloch. The authors would also like to acknowledge the general assistance of the basic references by Pearson (1984) and Kellaway (1946). The National Radio Astronomy Observatory is a facility of the (U.S.) National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

A. Pixel regularization image

In some circumstances it is not possible to describe the celestial coordinates of an image adequately solely by means of a linear transformation combined with one of the spherical projections defined in this paper. For example, in optical astrometry, plate distortions may be introduced by imperfect optics or by unequal shrinkage of the photographic emulsion. Thus coordinates on optical plates are often defined by means of an analytic "plate solution" which is derived via an analysis of the positions of reference stars measured from the plate. Similar distortions may arise in radio maps, caused, for example, by effects such as ionospheric refraction.

The "Pixel Regularization Image" is defined here as a general method of correcting for residual coordinate offsets within FITS. It should be stressed that the pixel regularization image is only to be used to account for the (typically small) offsets which may remain after the image coordinates have been described by an appropriate linear transformation and projection. For example, for an optical plate the linear transformation (`PC` matrix) should account for rotation and skewness, and a zenithal polynomial projection (`ZPN`) might be used to account for any radially symmetric component. The pixel regularization image would then be used to correct for residual offsets.

We envisage that the pixel regularization image will be used in any of three ways by FITS-reading software packages:

1. Ignore it. The peak and RMS error so introduced will be commensurate with the peak and RMS error of the pixel regularization image values. (The error in angular displacement will be weighted by variable scaling across the image).
2. Apply it once and for all by re-gridding the image.
3. Attach the pixel regularization image to the image for use whenever coordinate conversions are required.

This third usage implies that the pixel regularization image must be amenable to sub-imaging operations, while the first two emphasize that the corrections in the pixel regularization image must be small.

The role of the pixel regularization image in the chain of algorithms is shown in Fig. 1. It provides a correction

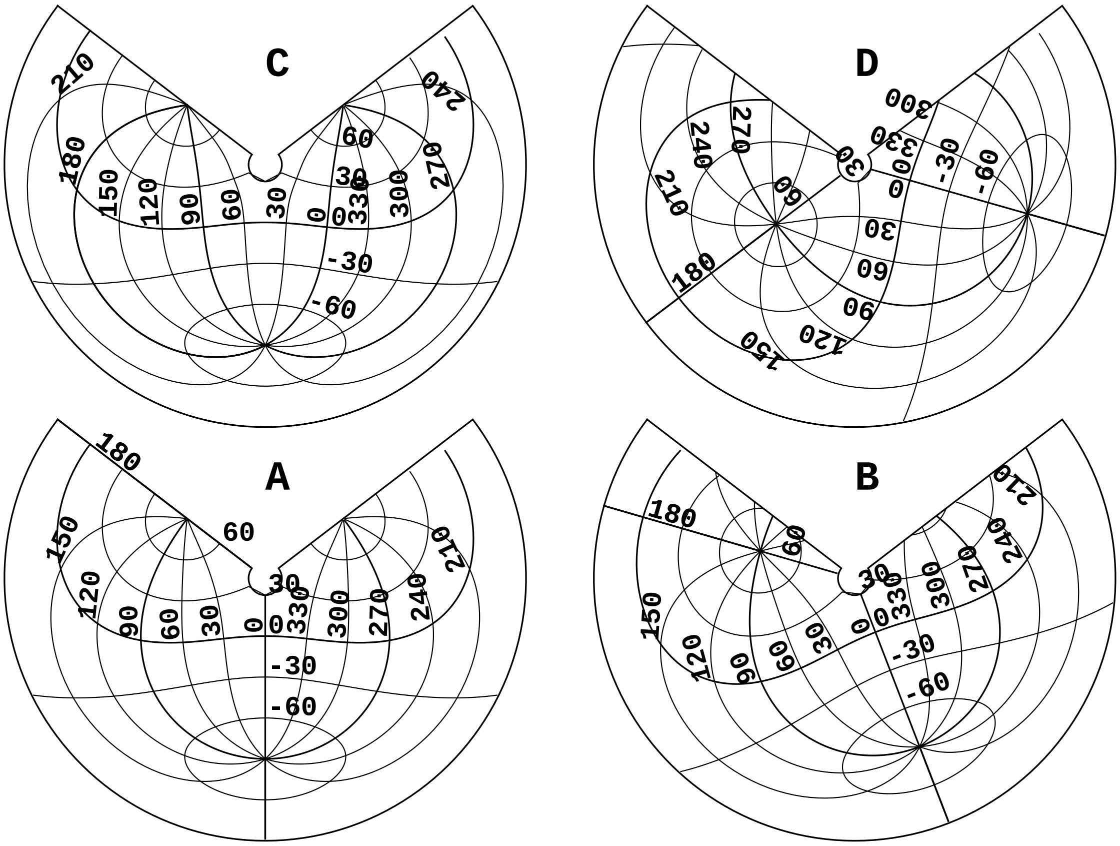

Fig. 33. `COD` projection plotted with lines in α, δ with parameters (bottom left: A) $\alpha_P = 0$, $\delta_P = 30°$, $\phi_P = 180°$, $(\alpha_0 = 0, \delta_0 = -60°)$; (bottom right:B) $\alpha_P = 0$, $\delta_P = 30°$, $\phi_P = 150°$, $(\alpha_0 = 49.1°$, $\delta_0 = -48.6°)$; (top left: C) $\alpha_P = 45°$, $\delta_P = 30°$, $\phi_P = 180°$, $(\alpha_0 = 45°$, $\delta_0 = -60°)$; (top right: D) $\alpha_P = 0$, $\delta_P = 30°$, $\phi_P = 75°$, $(\alpha_0 = 97.6°$, $\delta_0 = 13.0°)$.

(δ_i, δ_j) to be applied to the image pixel coordinates (i', j') to convert them to ideal pixel coordinates (i, j). That is

$$(i, j) = (i', j') + (\delta_i, \delta_j) \tag{A1}$$

As stated in Sect. 2.1, integral pixel coordinates correspond to the center of the pixel.

At one extreme, the pixel regularization image could simply store a correction for each pixel in the image. In practice though, sufficient accuracy should be obtainable by storing corrections for a regularly spaced sub-grid and using bilinear interpolation. For example, if a correction were provided for every 10^{th} image pixel in either direction then the pixel regularization image would have a size only 2 per cent that of the image.

The pixel regularization image stores values on a regular N_i by N_j grid with a spacing of (s_i, s_j) image pixels between grid elements in either dimension. Note that s_i and s_j are not required to be integers. Suppose that $P1 : (i'_1, j'_1)$, $P2 : (i'_1 + s_i, j'_1)$, $P3 : (i'_1, j'_1 + s_j)$, and $P4 : (i'_1 + s_i, j'_1 + s_j)$ are the image pixel coordinates of the four grid elements surrounding (i', j'). Let $(\delta_{i1}, \delta_{j1}), \ldots (\delta_{i4}, \delta_{j4})$ be the corresponding offsets. Then

$$\delta_i = W_1\delta_{i1} + W_2\delta_{i2} + W_3\delta_{i3} + W_4\delta_{i4} \tag{A2}$$
$$\delta_j = W_1\delta_{j1} + W_2\delta_{j2} + W_3\delta_{j3} + W_4\delta_{j4} \tag{A3}$$

where

$$\Delta'_i = (i' - i'_1)/s_i \tag{A4}$$
$$\Delta'_j = (j' - j'_1)/s_j \tag{A5}$$
$$W_1 = (1 - \Delta'_i) * (1 - \Delta'_j) \tag{A6}$$
$$W_2 = \Delta'_i * (1 - \Delta'_j) \tag{A7}$$
$$W_3 = (1 - \Delta'_i) * \Delta'_j \tag{A8}$$
$$W_4 = \Delta'_i * \Delta'_j \tag{A9}$$

For the inverse problem, where (i, j) is known and it is desired to compute (i', j'), inversion of the above formulae may be done by iteration.

If a pixel regularization image is provided, it should cover the whole of the image region; no extrapolation methods are defined by this standard. FITS-reading software packages will provide implementation-specific behavior in the event that a coordinate calculation is required in a part of an image not covered by a pixel regularization image. This might consist of simply returning a fatal error

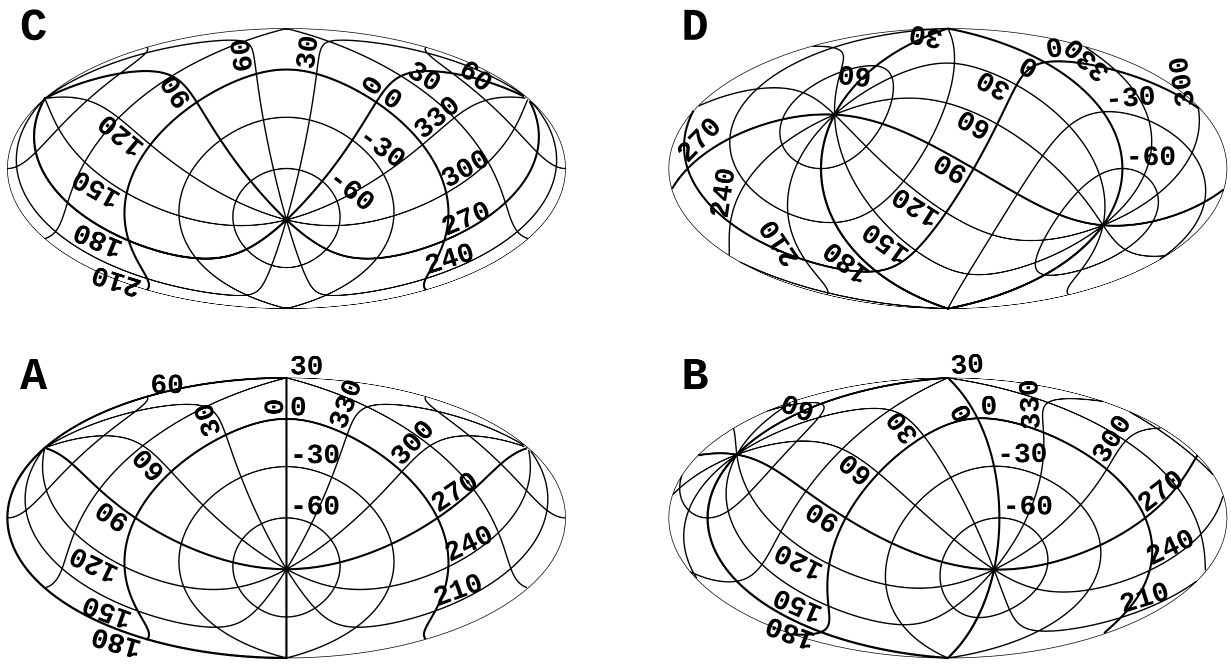

Fig. 34. AIT projection plotted with lines in α, δ with parameters (bottom left: A) $\alpha_0 = 0$, $\delta_0 = -60°$, $\phi_P = 180°$, ($\alpha_P = 0$, $\delta_P = 30°$); (bottom right: B) $\alpha_0 = 49.1°$, $\delta_0 = -48.6°$, $\phi_P = 150°$, ($\alpha_P = 0$, $\delta_P = 30°$); (top left: C) $\alpha_0 = 45°$, $\delta_0 = -60°$, $\phi_P = 180°$, ($\alpha_P = 45°$, $\delta_P = 30°$); (top right: D) $\alpha_0 = 97.6°$, $\delta_0 = 13.0°$, $\phi_P = 75°$, ($\alpha_P = 0$, $\delta_P = 30°$).

or of assuming a zero correction; the choice may depend on the positional accuracy required with respect to the RMS error indicated by the extant portion of the pixel regularization image.

The pixel regularization image will be implemented as a FITS image extension (Ponz *et al.*, 1994). The pixel regularization image will contain two image planes of N_i columns and N_j rows each. The first plane will contain the δ_i and the second the δ_j. The storage order will be as for FITS images so that the first row of the pixel regularization image corresponds (loosely speaking) to the first row of the image.

The pixel regularization image will contain sufficient information to associate correction grid coordinates with image pixel coordinates. This will be implemented via `CRPIX`n, `CRVAL`n, and `CDELT`n keywords in the pixel regularization image header. The `CRPIX`n define a location (i''_0, j''_0) on the correction grid which corresponds to the image pixel coordinate (i'_0, j'_0) defined by the `CRVAL`n, and the `CDELT`n specify the grid compression ratio, (s_i, s_j) defined above. The relationship between correction grid coordinates and image pixel coordinates is therefore

$$i' = s_i * (i'' - i''_0) + i'_0 \qquad \text{(A10)}$$

$$j' = s_j * (j'' - j''_0) + j'_0 \qquad \text{(A11)}$$

A pixel regularization image may be associated with a sub-image extracted from an image by suitably changing the `CRPIX`n, `CRVAL`n, and `CDELT`n. If size is important, this could also be accompanied by a reduction in the number of columns and rows.

The header shown in Table 7 describes a pixel regularization image with image values given in terms of single precision (4-byte) floating-point numbers. Angular brackets denote formulae expressed in terms of the variables defined above for quantities which must be supplied in numerical form.

B. Alternate FITS image representations: pixel list and vector column elements[4]

In addition to the image format discussed in the main body of this paper (*i.e.*, an N-dimensional array in a FITS primary array or FITS `IMAGE` extension) there are two other FITS image representations that are used commonly by the astronomical community:

1. a multi–dimensional vector in a single element of a FITS binary table, and
2. a tabulated list of pixel coordinates in a FITS ASCII or binary table.

The purpose of this appendix is to define a naming convention for the coordinate system keywords to be used with these alternate image formats.

[4] Contributed by William Pence, Arnold Rots, and Lorella Angelini of the NASA Goddard Space Flight Center, Greenbelt, MD 20771.

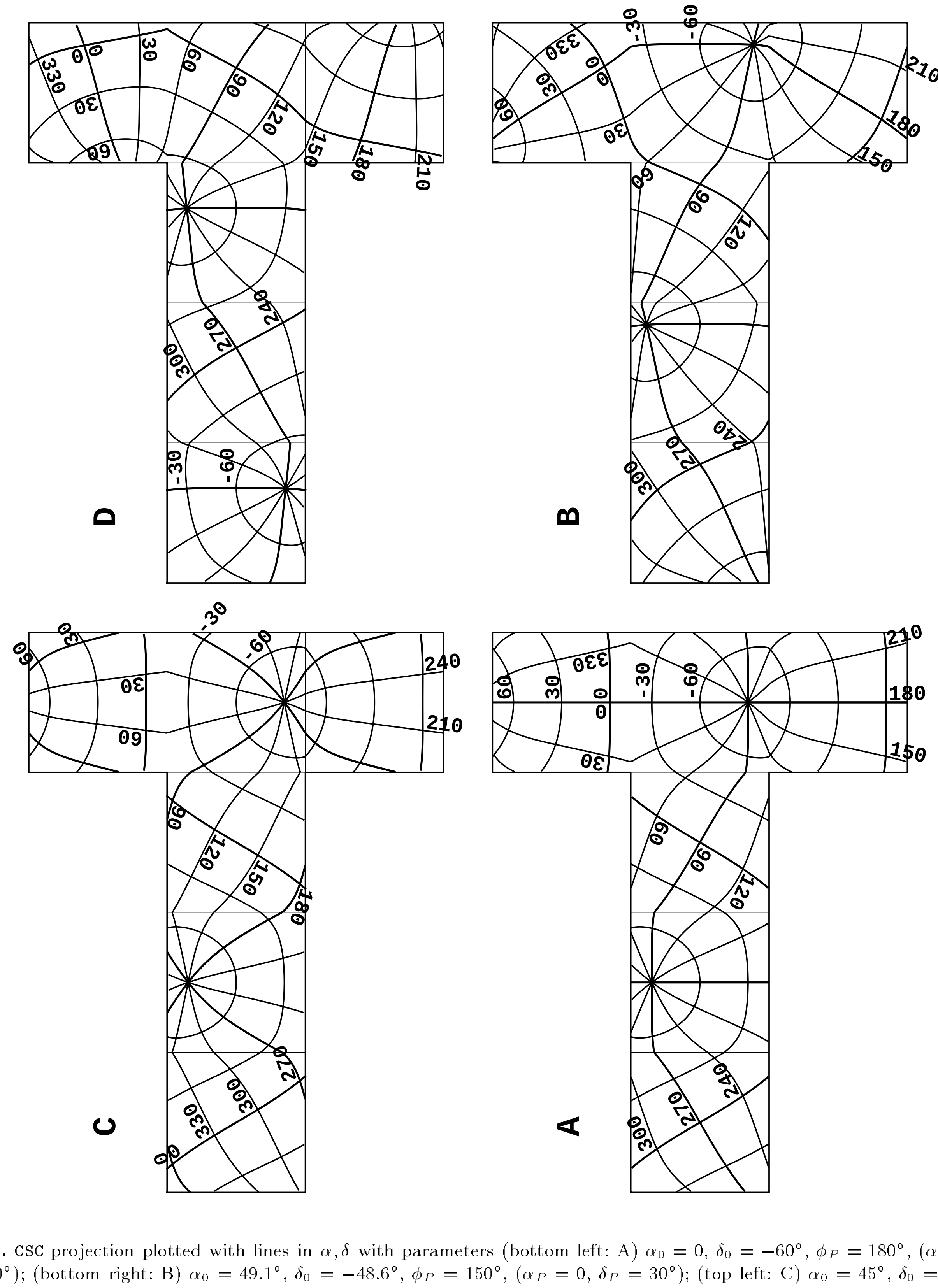

Fig. 35. CSC projection plotted with lines in α, δ with parameters (bottom left: A) $\alpha_0 = 0$, $\delta_0 = -60°$, $\phi_P = 180°$, ($\alpha_P = 0$, $\delta_P = 30°$); (bottom right: B) $\alpha_0 = 49.1°$, $\delta_0 = -48.6°$, $\phi_P = 150°$, ($\alpha_P = 0$, $\delta_P = 30°$); (top left: C) $\alpha_0 = 45°$, $\delta_0 = -60°$, $\phi_P = 180°$, ($\alpha_P = 45°$, $\delta_P = 30°$); (top right: D) $\alpha_0 = 97.6°$, $\delta_0 = 13.0°$, $\phi_P = 75°$, ($\alpha_P = 0$, $\delta_P = 30°$).

Table 4. Coordinate keywords

Keyword	Use	Status	Comments
`CRVAL`i	value at reference point	clarified	meaning of reference point forced by projection; no default.
`CRPIX`i	pixel of reference point	clarified	meaning of reference point forced by projection; no default.
`CDELT`i	increment at ref. point	clarified	meaning of reference point forced by projection; no default.
`CROTA`i	rotation at ref. point	deprecated	replaced by `PC`*iiijjj*.
`CTYPE`i	coordinate/projection type	clarified	for spherical coordinates, first 4 characters give "standard system" used in `CRVAL`n, second 4 characters give type of projection as in Table 5; no default.
`CUNIT`i	units of coordinate values	new	character-valued; keep it simple please; ignored for angles which are always degrees
`PC`*iiijjj*	coordinate increment	new	converts pixel number to pixels along true coordinates; default $= 0 (iii \neq jjj), = 1 (iii = jjj)$.
`CD`*iiijjj*	coordinate increment	defined deprecated	synonym for `PC`*iiijjj* times `CDELT`n diagonal matrix; no default; — should not be written
`CD`*i_j*	coordinate increment	defined deprecated	synonym for `PC`*iiijjj* times `CDELT`n diagonal matrix; no default; — should not be written
`LONGPOLE`	coordinate rotation	new	longitude in the native coordinate system of the standard system's North pole; default $= 0°$ if $\delta_0 > \theta_0$, $= 180°$ otherwise.
`LATPOLE`	coordinate rotation	new	latitude in the native coordinate system of the standard system's North pole; default (= 999) equivalent given by Eq. 7 with + taken.
`PROJP`m	projection parameter m	new	parameters required in some projections, see Table 5; no default for $m = 1$, otherwise 0.
`EPOCH`	coordinate epoch	deprecated	replaced by `EQUINOX`.
`EQUINOX`	coordinate epoch	new	epoch of the mean equator and equinox in years; (Besselian if `FK4`, Julian if `FK5`; see Section 3 for defaults
`MJD-OBS`	date of observation	new	Modified Julian Date (JD - 2400000.5) of observation in days; default = `DATE-OBS` or, if missing, `EQUINOX`.
`RADECSYS`	frame of reference	new	string identifying the frame of reference of the equatorial coordinates; default = `'FK4'` for `EQUINOX` < 1984.0 and `'FK5'` for $\geq$ 1984.0
`C`m`VAL`i	value at reference point	new	$(m = 2, 3, \ldots 9)$ secondary coordinate for axis i; no default.
`C`m`PIX`i	pixel of reference point	new	secondary coordinate description; no default.
`C`m`ELT`i	increment at ref. point	new	secondary coordinate description; no default.
`C`m`YPE`i	coordinate/projection type	new	secondary coordinate description; no default.
`C`m`NIT`i	units of coordinate values	new	secondary coordinate description; no default except angles are in degrees.

B.1. Multi-dimensional vector in a binary table

A vector column in a binary table (`BINTABLE`) extension can be used to store a multi-dimensional image in each element (*i.e.*, each row) of the column. In the simple case in which all the images have the same fixed size, the `TDIM`n keyword can be used to specify the dimensions. In the more general case, a variable length vector may be used to store different-sized images within the same column.

Because two or more columns in a binary table can contain images, the naming convention for these coordinate system keywords must encode the column number containing the image to which the keyword applies as well as the axis number within the image. The naming convention described here uses the keyword prefix to specify the axis number and the keyword suffix to specify the column number containing the image (*e.g.*, the `2CDLT15` keyword applies to the second axis of the image in column 15 of the table).

B.2. Tabulated list of pixels

An image may also be represented as a list of i and j pixel coordinates in a binary or ASCII table extension. This rep-

Table 5. Projection names and required parameters

FITS code	RP	Name	`PROJP1`	`PROJP2`	`PROJP`n
`AZP`	90	Zenithal perspective	μ		
`TAN`	90	Gnomic (`AZP` w $\mu = 0$)			
`SIN`	90	Orthographic (`AZP` w $\mu = \infty$)	α	β	
`STG`	90	Stereographic (`AZP` w $\mu = 1$)			
`ARC`	90	Zenithal equidistant			
`ZPN`	90	Zenithal polynomial	P_1	P_2	P_n for $n = 0, \dots 9$
`ZEA`	90	Zenithal equal-area			
`AIR`	90	Airy	θ_b		
`CYP`	0	Cylindrical perspective	μ	λ	
`CAR`	0	Cartesian			
`MER`	0	Mercator			
`CEA`	0	Cylindrical equal area	λ		
`COP`	90	Conical perspective	θ_a	α	
`COD`	90	Conical equidistant	θ_a	α	
`COE`	90	Conical equal-area	θ_a	α	
`COO`	90	Conical orthomorphic	θ_a	α	
`BON`	90	Bonne's equal area	θ_1		
`PCO`	0	Polyconic			
`GLS`	0	Sinusoidal			
`PAR`	0	Parabolic			
`AIT`	0	Hammer-Aitoff			
`MOL`	0	Mollweide			
`CSC`	0	Cobe Quadrilateralized Spherical Cube			
`QSC`	0	Quadrilateralized Spherical Cube			
`TSC`	0	Tangential Spherical Cube			

resentation is frequently used in high-energy astrophysics as a way of recording the position and other properties of individually detected photons. This image format requires a minimum of 2 table columns which give the i (axis 1) and j (axis 2) pixel coordinate of the corresponding event in the virtual 2-D image; any number of other columns may be included in the table to store other parameters associated with each event such as arrival time or photon energy. This virtual image may be converted into a real image by computing the 2-dimensional histogram of the number of listed events that occur in each pixel of the image (*i.e.*, the intensity value assigned to each pixel (i, j) of the image is equal to the number of rows in the table which have axis 1 coordinate $= i$ and axis 2 coordinate $= j$).

A variation on this pixel list format may be used to explicitly specify the intensity value of each image pixel. This case requires at least three table columns which specify the axis 1 coordinate, the axis 2 coordinate, and the value of the pixel at that coordinate. In this representation each pixel coordinate would only be listed at most once in the table; pixels with a value $= 0$ may be omitted entirely from the table to conserve space.

Each axis of the image in this representation translates into a separate column of the table, so the suffix of the coordinate system keywords all refer to a column number rather than an axis number (*e.g.*, the `TCDLT12` keyword applies to the coordinates listed in the 12^{th} column of the table).

Note that the keyword naming convention given here does not define a mechanism for explicitly pairing together the coordinate axis columns. A convention to do this could be developed, but in practice, the use of distinctive column names (*e.g.*, `'DETX'` and `'DETY'`, or `'X'` and `'Y'`) is usually sufficient to indicate which columns are associated with each other.

B.3. Keyword naming convention

Table 8 lists the corresponding set of coordinate system keywords for use with each type of FITS image representation. The allowed values for these keywords are identical for all three types of images as defined in the main body of this paper. The following notes apply to the naming conventions used in Table 8:

- The i, j prefix or suffix characters are integers referring to an axis number of the array. When used as a keyword suffix the image dimension may range from 1 to 999, but when used as a prefix the integer is limited to a single digit to conform to the 8-character keyword

Table 6. Other standard projections

Name	Synonym for
Near-sided perspective	AZP with $\mu = 1.35$
Clarke's (first)	AZP with $\mu = 1.35$
Clarke's (second)	AZP with $\mu = 1.65$
James'	AZP with $\mu = 1.367$
La Hire's	AZP with $\mu = 1.71$
Approximate equidistant zenithal perspective	AZP with $\mu = 1.7519$
Approximate equal area zenithal perspective	AZP with $\mu = 2.4142$
Central	TAN
Postel	ARC
Equidistant	
Globular	
Lambert azimuthal equivalent	ZEA
Lambert azimuthal equal area	
Lambert polar azimuthal	
Lorgna	
Simple cylindrical	CYP with $\mu = 0, \lambda = 1$
Central cylindrical	
Cylindrical Central Perspective	
Gall's orthographic	CYP with $\mu = 1, \lambda = 1/\sqrt{2}$
Peter's	
Behrmann equal area	CYP with $\mu = 1, \lambda = \cos 30°$
Lambert's cylindrical	CYP with $\mu = \infty, \lambda = 1$
Lambert's equal area	
Miller	CAR with x scaled by $2/\pi$
Plate Carrée	CAR
Equidistant cylindrical	
Cassini	CAR transverse case
Gauss conformal	MER transverse case
Gauss-Krüger	
Transverse cylindrical orthomorphic	
Lambert's equal area	CEA with $\lambda = 1$
One-standard conic	COP, COD, COE, COO with $\theta_1 = \theta_2$
Two-standard conic	COP, COD, COE, COO with $\theta_1 \neq \theta_2$
Murdoch conic	of same general form as COD
Alber's	COE
Alber's equal area	
Lambert equal area	COE with $\theta_2 = 90°$
Lambert Conformal conic	for spherical earth = COO
Werner's	BON with $\theta_1 = 90°$
Mercator equal-area	GLS
Sanson-Flamsteed	
Sanson's	
Craster	PAR
Hammer equal area	AIT
Aitov	
Bartholomew's nordic	AIT oblique case
Mollweide's homolographic	MOL
Homolographic	
Homalographic	
Babinet	
Elliptical	
Bartholomew's atlantis	MOL oblique case

Table 7. Example pixel regularization image header

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
--------------------------------------------------------------------------------
XTENSION= 'IMAGE   '           / image extension
EXTNAME = 'WCS_PIXREG'         / WCS pixel regularization image
BITPIX  =                  -32 / IEEE floating-point
NAXIS   =                    3 / 3-dimensional binary image
NAXIS1  =                <N_i> / number of image columns
NAXIS2  =                <N_j> / number of image rows
NAXIS3  =                    2 / delta i, then delta j
PCOUNT  =                    0 / special data area of size zero
GCOUNT  =                    1 / one data group
CRPIX1  =               <i"_0> / correction grid, I-coordinate
CRVAL1  =               <i'_0> / image pixel I-coordinate
CDELT1  =                <s_i> / grid step size in I
CRPIX2  =               <j"_0> / correction grid, J-coordinate
CRVAL2  =               <j'_0> / image pixel J-coordinate
CDELT2  =                <s_j> / grid step size in J
END
```

Table 8. Alternate coordinate keywords

Keyword Description	Primary Array	BINTABLE Vector	Pixel List
Reference Value	CRVAL*i*	*i*CRVL*n*	TCRVL*n*
Reference Pixel	CRPIX*i*	*i*CRPX*n*	TCRPX*n*
Coordinate Increment	CDELT*i*	*i*CDLT*n*	TCDLT*n*
Axis Rotation	CROTA*i*	deprecated	deprecated
Axis Type	CTYPE*i*	*i*CTYP*n*	TCTYP*n*
Axis Units	CUNIT*i*	*i*CUNI*n*	TCUNI*n*
Rotation Matrix	PC*iiijjj*	*ij*PC*n*	TC*nnnmmm*
Coord Rotation	LONGPOLE	LONGP*n*	LONGPOLE
Coord Rotation	LATPOLE	LATP*n*	LATPOLE
Proj Parmeters	PROJP*m*	P*m*PRO*n*	PROJP*m*
Coord Epoch	EQUINOX	EQUIN*n*	EQUINOX
Date of Obs	MJD-OBS	MJDOB*n*	MJD-OBS
Reference Frame	RADECSYS	RADEC*n*	RADECSYS
Secondary Type	C*m*YPE*i*	*i*C*m*YP*n*	TC*m*YP*n*
Secondary Units	C*m*NIT*i*	*i*C*m*NI*n*	TC*m*NI*n*
Second Ref Pix	C*m*PIX*i*	*i*C*m*PX*n*	TC*m*PX*n*
Second Ref Val	C*m*VAL*i*	*i*C*m*VL*n*	TC*m*VL*n*
Second Pix Incr	C*m*ELT*i*	*i*C*m*LT*n*	TC*m*LT*n*

name limit so the image may only contain up to 9 dimensions.

- iii, jjj are 3-digit integer axis numbers including leading zeros if necessary (001–999).
- m is a single digit integer. In the case of the projection parameters it can range from 0 through 9, and in the case of the secondary axis description keywords it can range from 2 through 9.
- n is an integer table column number without any leading zeros (1–999).
- mmm, nnn are 3-digit integer table column numbers including leading zeros if necessary (001–999).
- The guidelines given Section 2.2 must be applied to the the value of the `CUNIT`i keyword and its derivitives. In particular the value is restricted to `'deg'` when referring to celestial coordinates.

When using the `BINTABLE` vector image format, if the table only contains a single image column or if there are multiple image columns but they all have the same value for any of the last 8^{th} (`LONGPOLE`) through 13^{th} (`RADECSYS`) keywords in Table 8 then the simpler form of the keyword name, as used for primary arrays, may be used. For example, if all the images in the table have the same epoch then one may use a single `EQUINOX` keyword rather than multiple `EQUIN`n keywords. The other keywords, however, must always be specified using the more complex keyword name with the column number suffix and the axis number prefix.

In principle, more than one pixel list image can be stored in a single FITS table by defining more than one pair of i and j pixel coordinate columns. Under the convention defined here, however, all the images must share the same values for the 8^{th} through 13^{th} keywords in Table 8 (`LONGPOLE` through `RADECSYS`).

B.4. Multiple images and the "Greenbank Convention"

In the case of the binary table vector representation, all the images contained in a given column of the table may not necessarily have the same coordinate transformation values. For example, the pixel location of the reference point may be different for each image/row in the table, in which case a single `1CRPX`n keyword in the header is not sufficient to record the individual value required for each image. In such cases, the keyword must be replaced by a column with the same name (*i.e.*, `TTYPE`m `= '1CRPX`n`'`) which can then be used to store the pixel location of the reference point appropriate for each row of the table. This convention for expanding a keyword into a table column (or conversely, collapsing a column of identical values into a single header keyword) is commonly known as part of the 'Greenbank Convention' for FITS keywords and is illustrated in the example header shown in Table 10.

B.5. Example FITS Headers

The following tables show how the same image may be represented using each of the three different FITS image formats. In each case the images are 300 × 200 pixels in size and contain a tangent plane projection rotated 30 degrees with respect to the RA and DEC celestial coordinate system.

Table 9 shows a listing of the FITS header in which the image is stored as a conventional FITS primary array. This header contains the standard set of celestial coordinate system keywords as defined in the main body of this paper.

Table 10 shows the FITS header for a set of similar images which are stored in the binary table vector column format. In this example the images are stored as 2-D vectors in column 3 of the table and each row of the table contains a 300 × 200 pixel image of a different region on the sky. This might represent a set of smaller images extracted from a single larger CCD image. In this case all the coordinate system parameters except for the pixel coordinate of the reference point are the same for each image and are given as header keywords. The pixel coordinates of the reference point are different for each image, therefore they are given in the `1CRPX3` and `2CRPX3` columns of the table.

Finally, Table 11 shows the header for the same image when it is given in the pixel list format. There are 10000 rows in this table, each one listing the pixel coordinates (XPOS, YPOS) of every detected "event" or photon in the image. For illustration purposes, this table also contains an optional `'DATA_QUALITY'` column that could be used to flag the reliability or statistical significant of each event. A real image can be constructed from this virtual image by constructing the 2-dimensional histogram of the number of events that occur within each pixel. The additional `TLMIN`n and `TLMAX`n keywords shown here are used to specify the minimum and maximum legal values in XPOS and YPOS columns and are useful for determining the range of each axis when constructing the image histogram.

This keyword naming convention is optional and will not preclude other conventions.

C. Coordinate representation combined

To solve the general case problem in which any two of the four pixel and celestial coordinates are known and the other two desired, we must combine Eqs. 3 and 4 (or 5 or 6) with the equations for the particular projection relating the natural longitude and latitude (ϕ, θ) with the linear displacements (x, y). In all equations below, we refer to the longitude-like axis with symbols i, x, α and the latitude-like axis with symbols j, y, δ for pixel, relative linear, and standard spherical coordinates, respectively.

For the zenithal, perspective projection (`AZP`), we have

$$\begin{aligned} x &= (\mathtt{PC}xxxxxx(i-i_0) + \mathtt{PC}xxxyyy(j-j_0))\mathtt{CDELT}x \\ y &= (\mathtt{PC}yyyxxx(i-i_0) + \mathtt{PC}yyyyyy(j-j_0))\mathtt{CDELT}y\,, \end{aligned} \quad \text{(C1)}$$

and

$$\begin{aligned} x &= R_\theta \sin\phi \\ y &= -R_\theta \cos\phi \end{aligned}$$

or

$$\begin{aligned} x &= \frac{180^\circ}{\pi}\left(\frac{\mu+1}{\mu+\sin\theta}\right)\cos\theta\sin\phi \\ y &= -\frac{180^\circ}{\pi}\left(\frac{\mu+1}{\mu+\sin\theta}\right)\cos\theta\cos\phi\,. \end{aligned}$$

Substituting Eq. 5, we obtain

$$\begin{aligned} &A(i-i_0) + B(j-j_0) = \\ &\quad -(\mu+1)\frac{\cos\delta\sin(\alpha-\alpha_P)}{\mu+\sin\delta\sin\delta_P+\cos\delta\cos\delta_P\cos(\alpha-\alpha_P)} \\ &C(i-i_0) + D(j-j_0) = \end{aligned} \quad \text{(C2)}$$

Table 9. Example primary array image header

```
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                   32 / number of bits per data pixel
NAXIS   =                    2 / number of data axes
NAXIS1  =                  300 / length of data axis   1
NAXIS2  =                  200 / length of data axis   2

CTYPE1  = 'RA---TAN'           / TAN projection used
CRPIX1  =                  150 / pixel of reference point
CRVAL1  =                45.83 / RA at the reference point
CDELT1  =          -0.00277777 / increment per pixel (degrees)
CUNIT1  = 'deg     '           / physical units of axis 1

CTYPE2  = 'DEC--TAN'           / TAN projection used
CRPIX2  =                  100 / pixel at reference point
CRVAL2  =                63.57 / DEC at the reference point
CDELT2  =           0.00277777 / increment per pixel (degrees)
CUNIT2  = 'deg     '           / physical units of axis 2
CROTA2  =                 30.0 / image rotation  - DEPRECATED

PC001001=          0.866025403 / Coord. Descrp. Matrix:  cos(CROTA2)
PC002002=                 -0.5 / Coord. Descrp. Matrix: -sin(CROTA2)
PC001002=                  0.5 / Coord. Descrp. Matrix:  sin(CROTA2)
PC002001=          0.866025403 / Coord. Descrp. Matrix:  cos(CROTA2)

EQUINOX =               2000.0 / coordinate epoch
MJD-OBS =        44258.7845612 / Date of the observation
END
```

$$-(\mu+1)\frac{\cos\delta\sin\delta_P\cos(\alpha-\alpha_P)-\sin\delta\cos\delta_P}{\mu+\sin\delta\sin\delta_P+\cos\delta\cos\delta_P\cos(\alpha-\alpha_P)}\,,$$

where

$$\begin{aligned}
A &\equiv \frac{\pi}{180^\circ}\big(\cos\phi_P \mathtt{PC}xxxxxx\,\mathtt{CDELT}x \\
&\qquad + \sin\phi_P \mathtt{PC}yyyxxx\,\mathtt{CDELT}y\big) \\
B &\equiv \frac{\pi}{180^\circ}\big(\cos\phi_P \mathtt{PC}xxxyyy\,\mathtt{CDELT}x \\
&\qquad + \sin\phi_P \mathtt{PC}yyyyyy\,\mathtt{CDELT}y\big) \\
C &\equiv \frac{\pi}{180^\circ}\big(\sin\phi_P \mathtt{PC}xxxxxx\,\mathtt{CDELT}x \\
&\qquad - \cos\phi_P \mathtt{PC}yyyxxx\,\mathtt{CDELT}y\big) \\
D &\equiv \frac{\pi}{180^\circ}\big(\sin\phi_P \mathtt{PC}xxxyyy\,\mathtt{CDELT}x \\
&\qquad - \cos\phi_P \mathtt{PC}yyyyyy\,\mathtt{CDELT}y\big)\,.
\end{aligned} \tag{C3}$$

The reduction of these equations to the special cases for `TAN`, `SIN`, and `STG` projections ($\mu = 0, \infty$, and 1, respectively) are obvious and need not be written out here.

The situation deteriorates rapidly, however. For the zenithal equidistant projection (`ARC`), Eqs. C2 become

$$\begin{aligned}
A(i-i_0)+B(j-j_0) &= \\
&\left[\frac{\pi}{180^\circ}\frac{\theta-90^\circ}{\cos\theta}\right]\cos\delta\sin(\alpha-\alpha_P) \\
C(i-i_0)+D(j-j_0) &= \\
&\left[\frac{\pi}{180^\circ}\frac{\theta-90^\circ}{\cos\theta}\right](\cos\delta\sin\delta_P\cos(\alpha-\alpha_P)-\sin\delta\cos\delta_P)\,,
\end{aligned} \tag{C4}$$

where A, B, C, and D are defined by Eqs. C3. There is no simple formula for the $\beta/\sin(\beta)$ term, so θ will have to be computed (or estimated in iterative schemes) at each step in the solution. One may use Eq. C1 with

$$|\theta| \;=\; \sqrt{x^2+y^2}$$

or

$$\theta \;=\; \sin^{-1}\left[\sin\delta\sin\delta_P+\cos\delta\cos\delta_P\cos(\alpha-\alpha_P)\right]\,,$$

whichever is more convenient. The zenithal equal area projection (`ZEA`) ends up with similar equations, replacing the $\frac{\pi}{180^\circ}(90^\circ-\theta)$ with $2\sin(\frac{90^\circ-\theta}{2})$.

Table 10. Example BINTABLE vector header

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =               240008 / width of table in bytes
NAXIS2  =                    4 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)

TFIELDS =                    3 / number of fields in each row
TTYPE1  = '1CRPX3  '           / reference point pixel along 1st coordinate
TFORM1  = '1J      '           / data format of the field: I*4 integer
TTYPE2  = '2CRPX3  '           / reference point pixel along 2nd coordinate
TFORM2  = '1J      '           / data format of the field: I*4 integer
TTYPE3  = 'Image   '           / 2-D image vector
TFORM3  = '60000J  '           / data format of the field: I*4 vector
TDIM3   = '(300,200)'          / dimension sizes of the vector

COMMENT  The following keywords define the coordinate system of the images
COMMENT  contained in Column 3 of the table

1CTYP3  = 'RA---TAN'           / TAN projection used in axis 1
1CRVL3  =                45.83 / RA at the reference point
1CDLT3  =          -0.00277777 / increment per pixel
1CUNI3  = 'deg     '           / physical units of axis 1

2CTYP3  = 'DEC--TAN'           / TAN projection used in axis 2
2CRVL3  =                63.57 / DEC at the reference point
2CDLT3  =           0.00277777 / increment per pixel
2CUNI3  = 'deg     '           / physical units of axis 2

11PC3   =          0.866025403 / Coord. Descrp. Matrix:  cos(rotation)
22PC3   =                 -0.5 / Coord. Descrp. Matrix: -sin(rotation)
12PC3   =                  0.5 / Coord. Descrp. Matrix:  sin(rotation)
21PC3   =          0.866025403 / Coord. Descrp. Matrix:  cos(rotation)

EQUINOX =               2000.0 / coordinate epoch
MJD-OBS =        44258.7845612 / Date of the observation
END
```

D. Mathematical methods and other usage notes

D.1. Coordinate rotation with matrices

The coordinate rotations represented in Eqs. 4 or 5 may be represented by a matrix multiplication of a vector of direction cosines. The matrix and its inverse (which is simply the transpose) may be pre-computed and applied repetitively to a variety of coordinates, improving performance. Thus, we have

$$\begin{pmatrix} l \\ m \\ n \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} l' \\ m' \\ n' \end{pmatrix} , \qquad \text{(D1)}$$

where

$$\begin{aligned} l' &= \cos\delta\cos\alpha \\ m' &= \cos\delta\sin\alpha \\ n' &= \sin\delta \\ l &= \cos\theta\cos\phi \\ m &= \cos\theta\sin\phi \\ n &= \sin\theta \\ r_{11} &= -\sin\alpha_P\sin\phi_P - \cos\alpha_P\cos\phi_P\sin\delta_P \\ r_{12} &= \cos\alpha_P\sin\phi_P - \sin\alpha_P\cos\phi_P\sin\delta_P \\ r_{13} &= \cos\phi_P\cos\delta_P \end{aligned}$$

Table 11. Example pixel list header

```
XTENSION= 'BINTABLE'          / binary table extension
BITPIX  =                   8 / 8-bit bytes
NAXIS   =                   2 / 2-dimensional binary table
NAXIS1  =                   5 / width of table in bytes
NAXIS2  =               10000 / number of rows in table
PCOUNT  =                   0 / size of special data area
GCOUNT  =                   1 / one data group (required keyword)

TFIELDS =                   3 / number of fields in each row
TTYPE1  = 'DATA_QUALITY'      / quality flag value of the photon
TFORM1  = '1B      '          / data format of the field: 1-byte integer
TTYPE2  = 'XPOS    '          / axis 1 pixel coordinate of the photon
TFORM2  = '1I      '          / data format of the field: I*2 integer
TTYPE3  = 'YPOS    '          / axis 2 pixel coordinate of the photon
TFORM3  = '1I      '          / data format of the field: I*2 integer

TCTYP2  = 'RA---TAN'          / TAN projection used
TCRPX2  =                 150 / reference point pixel coordinate
TCRVL2  =               45.83 / RA at the reference point
TCDLT2  =          -.00277777 / increment per pixel (degrees)
TCUNI2  = 'deg     '          / physical units of axis in col 2
TLMIN2  =                   1 / lower limit of axis in col 2
TLMAX2  =                 300 / upper limit of axis in col 2

TCTYP3  = 'DEC--TAN'          / TAN projection used
TCRPX3  =                 100 / reference point pixel coordinate
TCRVL3  =               63.57 / DEC at the reference point
TCDLT3  =           .00277777 / increment per pixel (degrees)
TCUNI3  = 'deg     '          / physical units of axis in col 3
TLMIN3  =                   1 / lower limit of axis in col 3
TLMAX3  =                 200 / upper limit of axis in col 3

TC002002=         0.866025403 / Coord. Descrp. Matrix:  cos(rotation)
TC003003=                -0.5 / Coord. Descrp. Matrix: -sin(rotation)
TC002003=                 0.5 / Coord. Descrp. Matrix:  sin(rotation)
TC003002=         0.866025403 / Coord. Descrp. Matrix:  cos(rotation)

EQUINOX =              2000.0 / coordinate epoch
MJD-OBS =       44258.7845612 / Date of the observation
END
```

$$
\begin{aligned}
r_{21} &= \sin\alpha_P \cos\phi_P - \cos\alpha_P \sin\phi_P \sin\delta_P \\
r_{22} &= -\cos\alpha_P \cos\phi_P - \sin\alpha_P \sin\phi_P \sin\delta_P \\
r_{23} &= \sin\phi_P \cos\delta_P \\
r_{31} &= \cos\alpha_P \cos\delta_P \\
r_{32} &= \sin\alpha_P \cos\delta_P \\
r_{33} &= \sin\delta_P \,.
\end{aligned}
$$

The inverse equation is

$$
\begin{pmatrix} l' \\ m' \\ n' \end{pmatrix} = \begin{pmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{pmatrix} \begin{pmatrix} l \\ m \\ n \end{pmatrix} . \tag{D2}
$$

D.2. Iterating the inverse of PCO (polyconic)

The inverse of the `PCO` projection requires the solution of the transcendental equation

$$x^2 - \frac{360^\circ}{\pi}(y-\theta)\cot\theta + (y-\theta)^2 = 0$$

An iterative solution of this problem, based on dividing the interval with some bias toward the end which is closer to the solution (a modified form of the *regula falsi* algorithm), goes as follows. Begin by setting

$$\begin{aligned}
\theta_p &= 90^\circ\,\mathrm{sign}(y)\\
\theta_n &= 0\\
f_p &= x^2 - \frac{360^\circ}{\pi}(y-\theta_p)\cot\theta_p + (y-\theta_p)^2\\
f_n &= -999\,,
\end{aligned}$$

and then computing the loop

$$\begin{aligned}
\lambda &= \begin{cases} 0.5 & \text{if } f_n < -100\\ \frac{f_p}{f_p - f_n} & \text{otherwise}\end{cases}\\
\lambda' &= \max(0.1, \min(0.9, \lambda))\\
\theta &= (1-\lambda')\theta_p + \lambda'\theta_n\\
f &= x^2 - \frac{360^\circ}{\pi}(y-\theta)\cot\theta + (y-\theta)^2\\
f_p &= f \qquad (f > 0)\\
f_n &= f \qquad (f < 0)\\
\theta_p &= \theta \qquad (f > 0)\\
\theta_n &= \theta \qquad (f < 0)\,.
\end{aligned}$$

The loop terminates when $|f| < \epsilon_f$ or $|\theta_p - \theta_n| < \epsilon_\theta$. This method has obvious applicability to other iterative problems as well.

D.3. Random groups visibility data

The random-groups extension to FITS (Greisen and Harten, 1981) has been used to transmit interferometer fringe visibility sample data. It has been customary among users of this format to convey the suggested projection type as the last four characters of the random parameter types (`PTYPE`i) of the Fourier plane coordinates (traditionally called (u, v, w)). Any rotation of these coordinates was carried, however, by the `CROTA`j associated with the one-pixel array axis used to convey the field declination. We suggest that, under this new agreement, users of random groups for visibility data be prepared to read, carry forward, and use as needed, the new keywords `LONGPOLE`, `LATPOLE`, `PROJP`n, `MJD-OBS`, `EQUINOX`, `RADECSYS`, and `PC`$iiijjj$. In particular, any rotation of the (u, v) coordinate system should be represented in the `PC`$nnnmmm$ for the two one-point axes used to convey the celestial coordinates of the field.

References

Adobe Systems, Inc, 1990, *PostScript Language Reference Manual*, Second Edition, Adison-Wesley Publishing Company, Inc., New York, (see especially page 215).

Blaauw, A., Gum, C. S., Pawsey, J. L., Westerhout, G., 1960, The New I.A.U. System of Galactic Coordinates (1958 Revision), MNRAS, 121, 123.

Calabretta, M., 1992, Spherical Map Projections, AIPS++ Memo No. 107, The AIPS++ Consortium, c/o National Radio Astronomy Observatory, Charlottesville, Virginia, December 15, Available from the World-Wide Web using `http://www.cv.nrao.edu/aips++/docs/memos/107.ps`.

Calabretta, M., 1995, WCSlib, Version 2, available from the World-Wide Web beginning at `http://fits.cv.nrao.edu/` or via `ftp` from fits.cv.nrao.edu : `fits/src/wcs/wcslib-2.3.tar.gz`.

Chan, F. K., O'Neill, E. M., 1975, Feasibility Study of a Quadrilateralized Spherical Cube Earth Data Base, EPRF Technical Report 2–75 (CSC), Computer Sciences Corporation, Silver Springs, Maryland.

Cornwell, T. J., Perley, R. A., 1991, Radio-interferometric Imaging of Very Large Fields: The problem of non-coplanar arrays, A&A, 261, 353.

Evenden, G. I., 1991, Cartographic Projection Procedures for the UNIX Environment — A User's Manual, Open-File Report 90-284, United States Department of the Interior Geological Survey. Anonymous ftp site: kai.er.usgs.gov: `pub/PROJ.4/OF90-284.ps.gz` with accompanying code and documents; also fits.cv.nrao.edu: `fits/documents/wcs/USGS/doc*.ps.Z`.

Geldzahler, B., Schlesinger, B. M., 1992, A Users Guide for the Flexible Image Transport System (FITS), NASA/OSSA Office of Standards and Technology, Greenbelt, Maryland, Version 3.0 (October). Available from the World-Wide Web using `http://www.gsfc.nasa.gov/astro.fits/fits_home.html`.

Greisen, E. W., Harten, R. H., 1981, An Extension of FITS for Groups of Small Arrays of Data, A&AS, 44, 371.

Greisen, E. W., 1983, Non-linear Coordinate Systems in AIPS, AIPS Memo No. 27, National Radio Astronomy Observatory, Charlottesville, Virginia, November, fits.cv.nrao.edu : `fits/documents/wcs/aips27.ps`.

Greisen, E. W., 1986, Additional Non-linear Coordinates, AIPS Memo No. 46, National Radio Astronomy Observatory, Charlottesville, Virginia, May 20, fits.cv.nrao.edu: `fits/documents/wcs/aips46.ps.Z`.

Grosbøl, P., Harten, R. H., Greisen, E. W., Wells, D. C., 1988, Generalized Extensions and Blocking Factors for FITS, A&AS, 73, 359.

Hanisch, R. J., Wells, D. C., 1988, World Coordinate Systems Representations Within the FITS Format, draft based on notes from a meeting sponsored by the National Aeronautics and Space Administration, Code EZ, held in Charlottesville, Virginia in January 1988, fits.cv.nrao.edu: `/fits/documents/wcs/wcs88.ps`.

Harten, R. H., Grosbøl, P., Greisen, E. W., Wells, D. C., 1988, The FITS Tables Extension, A&AS, 73, 365.

IAU Inf. Bull., 1983, No. 49, 14.

Jones, M. H., 1993, On the Projection Commonly Attributed to Aitoff, The Observatory, 113, 213.

Kellaway, G. P., 1946, *Map Projections*, Methuen & Co. Ltd., London, England.
Moshir, M., Kopan, G., Conrow, T., McCallon, H., Hacking, P., Gregorich, D., Rohrback, G., Melnyk, M., Rice, W., Fullmer, L., White, J., Chester, T., 1992, *Explanatory Supplement to the IRAS Faint Source Survey, Version 2*, JPL D-10015, California Institue of Technology, Pasadena, California.
O'Neill, E. M., Laubscher, R. E., 1976, Extended Studies of the Quadrilateralized Spherical Cube Earth Data Base, NEPRF Technical Report 3–76 (CSC), Computer Sciences Corporation, Silver Springs, Maryland.
Pearson, F., 1984, *Map Projection Methods*, Sigma Scientific, Inc., Blacksburg, Virginia.
Ponz, J. D., Thompson, R. W., Muñoz, 1994, A&AS, 105, 53.
Wells, D. C., Greisen, E. W., Harten, R. H., 1981, FITS: A Flexible Image Transport System, A&AS, 44, 363.

This article was processed by the author using Springer-Verlag LaTeX A&A style file *L-AA* version 3.