Astronomy & Astrophysics manuscript no. dcs
(DOI: will be inserted by hand later)

April 22, 2004

Representations of distortions in FITS world coordinate systems

M. R. Calabretta[1], F. G. Valdes[2], E. W. Greisen[3], and S. L. Allen[4]

[1] Australia Telescope National Facility, PO Box 76, Epping, NSW 1710, Australia
[2] National Optical Astronomy Observatory, PO Box 26732, Tucson, AZ 85726-6732, USA
[3] National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, VA 22903-2475, USA
[4] UCO/Lick Observatory, University of California, Santa Cruz, CA 95064, USA

Draft dated 2004/04/22

Abstract. Greisen & Calabretta (2002) developed a generalized method for specifying coordinate systems for FITS images such as might be obtained from ideal astronomical instruments. This paper extends that work by providing methods to describe the distortions inherent in the image coordinate systems of real astronomical data.

Key words. methods: data analysis – techniques:image processing – astronomical data bases: miscellaneous – astrometry – techniques: spectroscopic

1. Introduction

Astronomical instrumentation typically produces a distorted representation of its subject matter. Instrument makers labour to minimize such defects, and may succeed to within quite narrow limits, but no measurement is ever entirely free of systematic error. Astrometry, spectroscopy and other areas of astronomy that seek to extract the highest accuracy from data must compensate for the effect of such imperfections, either by theoretical or empirical means.

As foreshadowed in Greisen & Calabretta (2002), Calabretta & Greisen (2002) and Greisen et al. (2004), hereinafter referred to as Papers I, II & III, this paper, Paper IV in the series, addresses the problem of describing distortions as they apply to image coordinate systems. It builds on the foundation provided by Paper I by interposing additional steps in the chain of operations by which world coordinates are computed from pixel coordinates, and vice versa, as shown schematically in Fig. 1. This figure may be compared with the corresponding diagram in Papers I & II.

Paper IV is a logical step in the historical development of the representation of world coordinates in FITS, the Flexible Image Transport System introduced by Wells, Greisen & Harten (1981) and most recently codified by Hanisch et al. (2001). While the foundation paper took a rather simplified view of coordinates, providing only the mechanics for describing linear coordinate systems, it was soon augmented by Greisen (1983) to include a small selection of non-linear celestial and spectral coordinate systems. Greisen's work, referred to as the "AIPS convention", soon become a de facto FITS standard. It was put on a sound footing and significantly extended

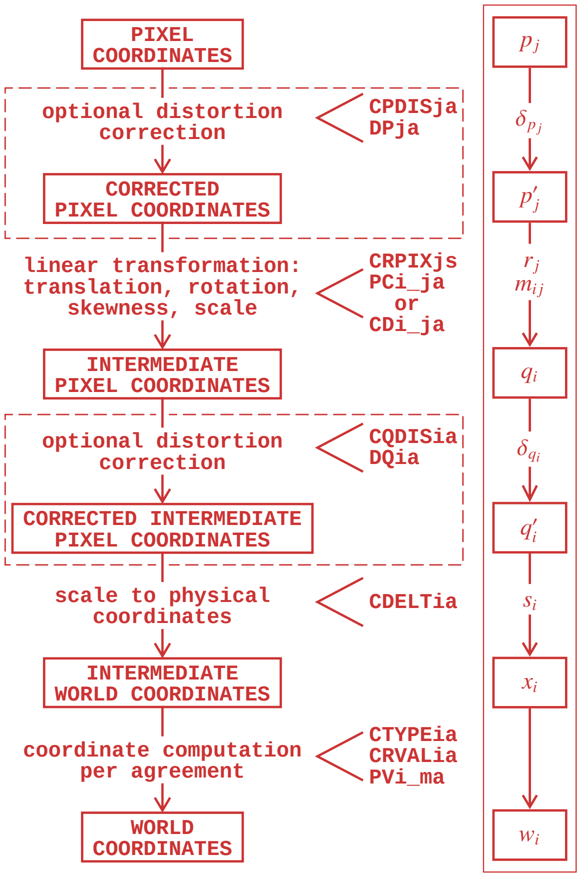

Fig. 1. Conversion of pixel coordinates to world coordinates showing optional distortion corrections enclosed in the dashed boxes. For later reference, the mathematical symbols associated with each step are shown in the box at right.

Send offprint requests to: M. Calabretta by Papers I, II & III. However, these papers dealt only with idealized coordinates in the sense that the non-linear transformations defined are what would be expected of perfect astronomical instruments.

For example, while the ARC celestial projection of Paper II can broadly describe the geometry of a Schmidt plate with an oblique celestial coordinate graticule, it cannot account for the small-scale distortions introduced by imperfect optics or uneven shrinkage of the photographic emulsion. In some circumstances it may be adequate, but in others totally inadequate. Typically, distortions are described by high-order polynomials or spline functions with many empirically-derived coefficients. One celestial projection, ZPN, went part-way in this regard, but in practice its assumption of radial symmetry is rather limiting.

This work provides general methods to be used in describing distorted image coordinates. In fact, these methods are so general that they could be used by themselves to describe a coordinate system to any required degree of accuracy. However, of necessity they are more complex, bulky and computationally intensive than those of the idealized coordinate systems. Therefore, we consider that, wherever possible, an image coordinate system should be described as accurately as possible by one of the ideal representations and that the methods of this paper be used to supply small, residual corrections. By doing so we seek to provide WCS (world coordinate system) interpreters with the option of ignoring the residual correction and accepting the accompanying error. To this end we will define methods for specifying the magnitude of that error. This will help to identify situations which may arise, for example in defining a coordinate system for the surface of an irregularly shaped object, where the distortion correction is not small and cannot be ignored.

Keeping the distortion correction small will also assist when it comes to inverting the coordinate transformation. Iterative methods will normally be required because distortion functions will not usually have analytic inverses. Small displacements in relatively smoothly varying functions should promote rapid convergence.

The presence of correction methods within the FITS conventions is not intended to discourage instrument makers from minimizing imperfections, nor recalibration of data (e.g. by resampling or regridding without serious loss of information) before transmission away from the instrument site.

2. Basic concepts

By analogy with Paper I, the framework for representing distortions consists of specifying an algorithm and defining its parameters via a set of FITS header keywords.

2.1. Distortion corrections

In Paper I the linear transformation was split into two separate steps,

$$q_i = \sum_{j=1}^{N} m_{ij}(p_j - r_j), \tag{1}$$

which transforms vector $\boldsymbol{p}$ of *pixel coordinate* elements p_j to element q_i of *intermediate pixel coordinate* vector $\boldsymbol{q}$, and

$$x_i = s_i q_i, \tag{2}$$

which transforms $\boldsymbol{q}$ to vector $\boldsymbol{x}$ of *intermediate world coordinate* elements x_i. The parameters r_j, m_{ij}, and s_i are given by the FITS header cards[1] CRPIX*ja*, PC*i_ja*, and CDELT*ia*. Alternatively, the product $s_i m_{ij}$ may be given by CD*i_ja* cards.

Figure 1 shows that a distortion correction may be applied before or after the matrix multiplication of Eq. (1). Where it appears before (a *prior distortion correction*) it has the mathematical form

$$p'_j = p_j + \delta_{p_j}(\boldsymbol{p}), \tag{3}$$

where $\delta_{p_j}(\boldsymbol{p})$ is a prior distortion *function*, and p'_j is an element of the *corrected pixel coordinate* vector $\boldsymbol{p}'$ which then replaces p_j in Eq. (1). Where it appears after (a *sequent distortion correction*) it has the form

$$q'_i = q_i + \delta_{q_i}(\boldsymbol{q}), \tag{4}$$

where q'_i is an element of the *corrected intermediate pixel coordinate* vector $\boldsymbol{q}'$ and replaces q_i in Eq. (2). Equations (3) and (4) may be written in vector form

$$\boldsymbol{p}' = \boldsymbol{p} + \boldsymbol{\delta_p}(\boldsymbol{p}), \tag{5}$$
$$\boldsymbol{q}' = \boldsymbol{q} + \boldsymbol{\delta_q}(\boldsymbol{q}), \tag{6}$$

a useful notational convenience when referring to the collection of individual distortion corrections and functions.

Note that the prior distortion functions, $\boldsymbol{\delta_p}(\boldsymbol{p})$, operate on pixel coordinates (i.e. $\boldsymbol{p}$ rather than $\boldsymbol{p}-\boldsymbol{r}$), and that the independent variables of the distortion functions are the *uncorrected* pixel or intermediate pixel coordinates. That is, for example, we do not allow the possibility of

$$q'_3 = q_3 + \delta_{q_3}(q'_1, q'_2). \tag{7}$$

To do so would risk introducing circular dependencies. Even if these were solvable it would place too great a burden on WCS interpreters to disentangle them, probably also involving significant computational overhead.

Prior and sequent distortion corrections should not normally appear together in a well constructed coordinate representation, though such *mixed distortion corrections* are not precluded. In fact, since distortion corrections are applied on an axis-by-axis basis, it would be quite reasonable to have a mixed correction when the linear transformation matrix is diagonal. In such a case it might be possible to achieve the effect of Eq. (7) above by correcting the pixel coordinates from which q_1 and q_2 were computed.

Mathematically speaking, there is no significant difference between prior and sequent distortion corrections; generally it should be possible to translate one usage into the other. The dichotomy is provided as a matter of convenience depending on where the distortion arises in the instrumentation.

[1] Usage in this paper of the keyword indices i, j, m, and a is consistent with their definition in Paper I.

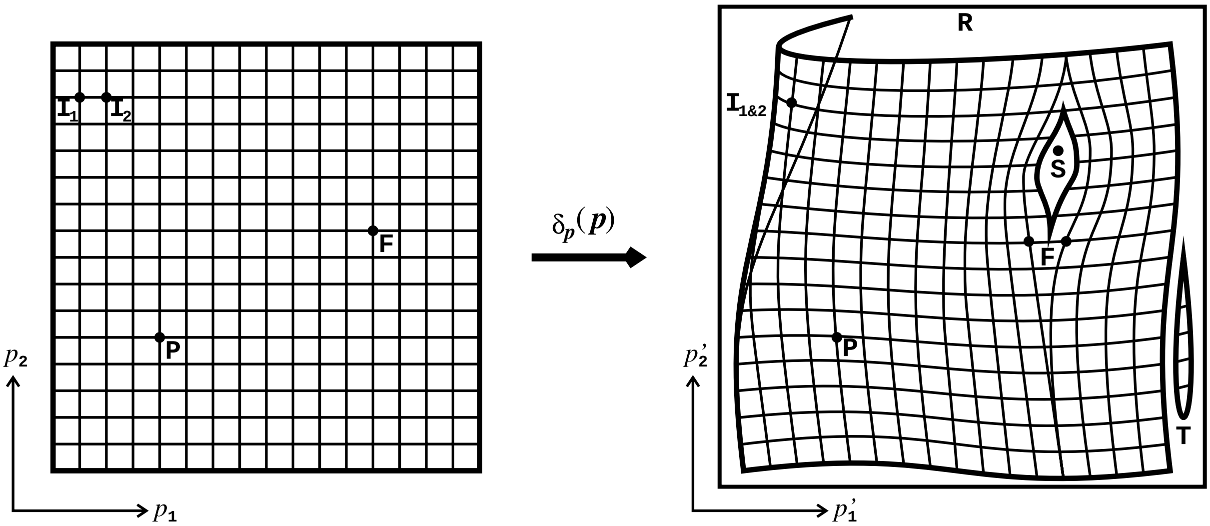

Fig. 2. Pathology of the inversion of a two-dimensional distortion function. (a) Injectivity is violated where points $\mathbf{I}_1$ and $\mathbf{I}_2$ both map to the same point in the distorted grid which thus does not have a unique inverse. (b) Point **S** in the range of $\boldsymbol{\delta_p}$ has no counterpart in its domain, thus violating surjectivity. In this example the tear that gave rise to **S** also creates duplicate mappings such as point **F** so that the mapping is not even a *function* in the strict mathematical sense. (c) The "tearaway" in the boundary at **T**, and the overlap in the upper-left corner, would severely test inversion algorithms. Even where inversion is possible in a mathematical sense, it may be difficult to achieve in practice at points in the vicinity of these irregularities since inversion algorithms, such as those based on interval dissection, necessarily operate over finite spans. Practical algorithms would also normally assume that the distorted grid is defined within a rectangular region **R** enclosing the distorted image boundary. This may even affect inversion at points such as **P** that otherwise might be thought to be relatively straightforward.

Where the major part of the distortion arises in the detector, as may be the case with spatial distortion in a CCD (charge-coupled device) chip, the correction should be pretty-well static for each chip and could be determined once and for all (this is why $\boldsymbol{\delta_p}(\boldsymbol{p})$ operates on the raw pixel coordinates, refer to Sect. 2.5.1 for subimaging options in this case). Thus it would best be applied to the pixel coordinates. The transformation matrix would then account for any linear errors - rotation, scale and skewness - arising from the placement of the detector in a particular observation.

On the other hand, where the distortion arises outside the detector, as might be the case where imperfections in a telescope's optics give rise to distortions that are pretty-well fixed in the focal plane, it would be best to account for the placement of the photographic plate first via the linear transformation matrix, and then correct for the distortion induced by the optics.

Bear in mind that computation of the distortion correction may present a significant overhead, particularly in inverting the transformation, so WCS composers should avoid constructing mixed distortion corrections unless there are compelling reasons to do so.

2.2. Invertibility

While the majority of algorithms defined in Papers II & III have analytic inverses, we do not expect this of distortion functions because of their greater complexity. Thus, as mentioned above, iterative methods will normally be required to invert them. For example, the distortion functions in Sect. 3 are defined only in one direction and no attempt is made to find their analytic inverse. Accordingly, whatever its location in the algorithm chain, we require that the distortion function be defined so that its natural direction is from pixel coordinates to world coordinates (the direction of the arrows in Fig. 1). This requirement is made primarily to simplify matters for software written to implement these methods.

In order to be invertible, distortion functions must satisfy the mathematical properties of *injectivity* and *surjectivity* shown schematically in Fig. 2. A transformation is injective (or *one-to-one*) if each point in its range is mapped to by *no more than one* point in its domain. A transformation is surjective (or *onto*) if each point in its range is mapped to by *at least one* point in its domain. For our purposes the domain and range are defined by the set of points on and interior to the boundary of the image. We assume that the boundary in pixel coordinates is mapped to a continuous and closed curve at each of the first three steps of Fig. 1 (but not the fourth and last step).

Regarding this last step, Paper II notes that since image planes are rectangular and the boundary of many celestial spherical projections is curved, an image may contain pixels that are outside the boundary of the projection. In other words, the transformation is not even defined at some points and therefore is not invertible. This is unavoidable but manageable because WCS interpreters can identify such points as being outside the normal range of native longitude or latitude. However, header interpretation example 3 of Paper II illustrates the subtle problems that can arise in such cases for improperly constructed WCS.

Distortion functions differ somewhat from the celestial case because there is no obvious method for identifying rogue points. However, surjectivity is not likely to be a limiting requirement for real astronomical instrumentation; we don't expect any gaps or holes in an output image and even if there were the distortion function could probably be interpolated across them. On the other hand, injectivity could conceivably be violated in the presence of severe distortions, thus reflecting a real and probably unrecoverable loss of information. This would lead to the situation where a world coordinate had a non-unique inverse value. In such a case the solution chosen by the WCS interpreter would have to be implementation dependent.

Another aspect of the invertibility requirement is that the distortion function must be defined at all points in the image; or to put it another way, no extrapolation methods are defined for distortion functions. WCS interpreting software will provide implementation-specific behavior in the event that a coordinate calculation is required in a part of an image not covered by a distortion function. This might consist simply of signalling an error or of assuming a zero correction; the choice may depend on the accuracy required with respect to the error indicated by keywords defined in Sect. 2.5.1.

Establishing injectivity and surjectivity for a particular distortion function may have to be done empirically. Where either is violated within the domain of the image the WCS writer should verify that this is a real artifact of the astronomical instrument.

2.3. Application

Distortion corrections may be used in any of three ways by FITS-reading software packages:

1. Ignore them and accept the accompanying error. Section 2.5.1 defines keywords that record the magnitude of such error.
2. Apply them once and for all by re-regridding the image.
3. Associate distortion corrections with the image in some implementation-dependent way for use whenever coordinates need to be computed.

An important consequence of the first option is that neither $\boldsymbol{\delta}_p(\boldsymbol{p})$ nor $\boldsymbol{\delta}_q(\boldsymbol{q})$ may introduce a rescaling. While $\boldsymbol{p}$, $\boldsymbol{p}'$, $\boldsymbol{q}$, and $\boldsymbol{q}'$ are all nominally pixel coordinates, if a CD*i_ja* matrix is used and has scaled $\boldsymbol{q}$ to physical units, then $\boldsymbol{q}'$ *must* have the same units.

Use of the first and second options relies on the corrections being small (enough) and the third implies that the distortion correction must be amenable to subimaging and image transposition operations and must be in a form suitable for incorporation into image analysis software packages.

Although regridding may seem to solve the distortion problem "once and for all", it may not always be an option since it often introduces correlation into the noise in the map and can lead to problems in error estimation. Also, many regridding algorithms are not flux conservative on censored data (e.g. when pixels have been lost due to CCD defects, cosmic rays, etc.) and may not be an option.

2.4. Distortion correction keywords

2.4.1. Distortion function keywords

Standard "4-3" form for CTYPE*ia*, defined in Paper I for encoding non-linear algorithms, is not simply extensible for the needs of this paper principally because the *i* in CTYPE*ia* always refers to q_i and hence is not suitable for prior distortion corrections which are based on p_j. Moreover, an extension of "4-3" form, say to "4-3-3" form, would result in an awkward syntax for linear coordinate types such as 'FREQ', leading to keyvalues[2] like 'FREQ-----XYZ'. This may also cause problems for software implementations that key on the fifth character being '-' when identifying non-linear coordinate types. Instead we here introduce two new keywords

CPDIS*ja*	(string-valued),	and
CQDIS*ia*	(string-valued)	

to record distortion function codes for prior and sequent distortion corrections respectively. Several *distortion codes* will be defined in Sect. 3. These are not limited to three characters and so may be more mnemonic.

2.4.2. Distortion parameter keywords

Paper I introduced the PV*i_ma* FITS keywords to record the parameters required by the formulæ encoded by the CTYPE*ia* algorithm codes. For example, the standard parallels for conic projections are recorded as PV*i_1a* and PV*i_2a* (in sum and difference form), where *i* corresponds to the latitude axis. Up to 100 parameters are catered for with the *m* subscript ranging from 0 to 99 and this is ample for the non-linear functions encountered. Thus far, only one non-linear algorithm has been defined (the ZPN projection of Paper II) that requires more than a single digit for *m*.

However, 100 parameters will generally be insufficient for the larger, more complex distortion functions so *m* would extend to at least three digits. This presents a significant problem in devising a suitable keyword for use in binary tables because of the eight-character limit on FITS keyword names. For example, the BINTABLE form of PV*i_ma* is given in Paper I Table 2 as *i*V*n_ma*. If *m* consumed three characters, with *i*, V, the underscore which is required to separate two groups of digits, and *a*, all using one character, then only one character would remain for the column number, *n*, and this is much too restrictive. Furthermore, the keyword is so skeletal that only one character, V, serves to identify it. If many of the 26 possible variants of this type of keyword started to proliferate they would quickly become non-mnemonic.

Another property of the distortion function parameters is that they will often most naturally be described by a data structure. Ideally they should be represented in FITS as named elements of such a data structure.

These considerations have led us to develop a class of FITS keywords that have a new semantic type but still conform to conventional syntax as codified by Hanisch et al. (2001). In short, the keywords are string-valued, but the string is to be interpreted as a definition giving (1) a record field name, and (2) its floating point value. Such keywords will be described as *record-valued*. In a FITS header they have the following syntax

keyword= '*field-specifier*:␣*float*'

[2] A *keyvalue* is the value associated with a keyword.

where *keyword* is a standard eight-character FITS keyword name, *float* is the standard FITS ASCII representation of a floating point number, and these are separated by a single blank denoted here by ␣. As with string-valued keywords, when record-valued keywords are recorded in a column of a FITS ASCII or binary table the single quotes that are used as delimiters for string-valued header keyvalues must be omitted.

The grammar for *field-specifier* is

field-specifier:
  field
  field-specifier `.` *field*

field:
  identifier
  identifier `.` *index*

where *identifier* is a sequence of letters (upper or lower case), underscores, and digits of which the first character must not be a digit, and *index* is a sequence of digits. No blank characters may occur in the field-specifier; its definition is a required part of the value of a record-valued keyword. The index is provided primarily for defining array elements though it need not be used for that purpose.

Multiple record-valued keywords of the same name but differing values may be present in a FITS header. These will be referred to in this paper using the abbreviation

keyword · *field-specifier* ,

which recognizes that the field-specifier may be viewed as part of the keyword name, or alternatively that the keyword name may be considered to be the first part of the field-specifier.

Record-valued keywords differ from the informal `HIERARCH` keyword[3] in that they conform to standard FITS syntax and they may have a name other than `HIERARCH`, specifically in that they may be indexed. This allows keyword names to be defined so that all keywords for a particular coordinate representation may be identified solely via the keyword name, without needing to examine any keyvalues. Thus we here introduce the

`DP`*ja* (record-valued), and
`DQ`*ia* (record-valued)

keywords to record the parameters required by the prior and sequent distortion functions respectively. The corresponding forms for `BINTABLE` image arrays are

j`DP`*na* (record-valued), and
i`DQ`*na* (record-valued)

and for pixel lists they are

`TDP`*na* (record-valued), and
`TDQ`*na* (record-valued).

[3] Used by the European Southern Observatory (ESO), see `http://archive.eso.org/dicb/dicd/dic-1-1.4.html#30006`.

2.4.3. Distortion parameter arrays

Paper I introduced the *parameter-array* syntax for *i*V*n*_*ma*, i.e. *i*V*n*_X*a*, to allow an array of parameters to occupy a single column of a binary table; all parameters up to the maximum dimension of the column given by its `TFORM`*n* keyword must be specified. This was developed mainly to circumvent the restriction on index values imposed by the eight-character limit on the keyword name. However, it is also quite a useful storage convention, primarily for binary tables containing image arrays for which the parameters differ from row to row.

This convention may be extended to record-valued parameters such as *j*`DP`*na* and *i*`DQ`*na* simply by allowing the table column containing such keywords to contain more than one of them. Since record-valued keywords are implemented as strings this may be done in either of two ways:

– Using the "Substring array" convention of either fixed or variable length substrings based on qualifying the `TFORM`*n* keyword for a character array field (`TFORM`*n* = `'`*r*`A:SSTR`*w*/*ddd*`'`) as defined in Appendix C of Cotton et al. (1995). This is the preferred method.
– Using the "Multidimensional array" convention based on the `TDIM`*n* keyword defined in Appendix B of Cotton et al. (1995). The table column would consist of a two-dimensional character array. The first dimension of `TDIM`*n* would be set to the maximum length of any of the keyvalues, and the second dimension would correspond to their number.

Unlike an *i*V*n*_X*a* array, there is no need to store the full set of parameters because the field-specifier in each value provides the necessary context. Hence distortion parameters with appropriate default values need not be specified. Where necessary for fixed-length substrings or multidimensional character arrays, null substrings may be specified as a keyvalue consisting solely of blanks.

This convention is not relevant to pixel lists. Each row of a pixel list stores a pixel coordinate and value that comprise one particular image and it would not make sense to store the full set of parameters for each pixel.

2.5. Distortion parameter keyvalues

Paper I discussed the problem of grouping image axes. It left open the precise method by which this was to be accomplished but pointed forward to a construct defined in Paper II that was to serve as a model. Paper II formalized the simple '*x*`LON`/*x*`LAT`' and '*yz*`LN`/*yz*`LT`' conventions based on the first four characters of `CTYPE`*ia* for associating longitude/latitude coordinate pairs. In fact, this was simply a generalization of the usage established by the AIPS convention. However, as far as distortions are concerned, the axis grouping problem is generally too complex to be handled by such simple methods and a more rigorous formalism is required. The method adopted here is to define the grouping via several `DP`*ja* and `DQ`*ia* parameters.

In this and subsequent sections, where usage is described in terms of the `DP`*ja* keywords, exactly the same applies for `DQ`*ia*, and vice versa, unless stated otherwise.

Firstly, we need to consider that some astronomical instruments may introduce dependencies between axes of different physical types. A good example is a Fabry-Perot interferometer which produces a data cube with two spatial and one spectral axis. Wavelength away from the image centre varies in a well-defined way as a function of position. Ideally

$$\lambda(r) = \lambda_0 \cos(\tan^{-1}(r/C)), \tag{8}$$

where r is the arc length from the optical axis to the field point and C is a scale angle characteristic of the interferometer. While such a simple non-linearity could be represented via the methods of Paper I, in a real Fabry-Perot interferometer the spatial plane and the spectral axis may contain additional complex, possibly interdependent distortions. One could imagine wanting to describe the distortion in the spectral axis as a polynomial in all three coordinates, with the spatial distortion, independent of wavelength, given by two-dimensional B-splines.

2.5.1. Axis coupling parameters

If we take the ideal Fabry-Perot interferometer of Eq. (8) as a concrete example, the spectral coordinate, which is linear in wavelength, is dependent on the two spatial coordinates. Assume that wavelength is on the third axis and we are constructing the primary coordinate description. We set DPja·NAXES, where j corresponds to the axis being corrected, to record the number, $\hat{N}$, of coordinate axes that will form the independent variables of the distortion function; we use $\hat{N}$ to distinguish this number from the number N of axes in the image. If omitted from the header this value defaults to zero, i.e. no distortion correction. We then set DPja·AXIS.$\hat{\jmath}$, with default value $\hat{\jmath}$, for $\hat{\jmath} = 1, \ldots, \hat{N}$, to record the coordinate axes that axis j depends on. In our example DP3·NAXES = 2, DP3·AXIS.1 = 1, and DP3·AXIS.2 = 2 for the two spatial axes. Subsequently in this paper these will be denoted by $\hat{p}_{\hat{\jmath}}$ or $\hat{q}_{\hat{\imath}}$ ($\hat{\jmath}, \hat{\imath} = 1, \ldots, \hat{N}$) where the *hat* is used to indicate a different indexing scheme from that of the j or i indices.

So far we have used $1 + \hat{N}$ parameters to define the independent variables of the distortion function. These also have an explicit ordering which is a useful feature that simplifies image transposition - the parameters can readily be permuted to suit. Continuing the Fabry-Perot example, to transpose the two spatial axes in the data array, leaving the spectral axis as is, the header keywords and values on the left below would have to change to those on the right:

```
CRPIX1 = 129              CRPIX2 = 129
CRPIX2 = 513              CRPIX1 = 513
CRPIX3 = 772              CRPIX3 = 772
CDELT1 = -0.000277778     CDELT2 = -0.000277778
CDELT2 =  0.000277778     CDELT1 =  0.000277778
CDELT3 =  0.0000000003    CDELT3 =  0.0000000003
CRVAL1 = 150.0000000      CRVAL2 = 150.0000000
CRVAL2 = -35.0000000      CRVAL1 = -35.0000000
CRVAL3 =   0.000000530    CRVAL3 =   0.000000530
CTYPE1 = 'RA---TAN'       CTYPE2 = 'RA---TAN'
CTYPE2 = 'DEC--TAN'       CTYPE1 = 'DEC--TAN'
CTYPE3 = 'WAVE'           CTYPE3 = 'WAVE'
CPDIS3 = 'Polynomial'     CPDIS3 = 'Polynomial'
   DP3 = 'NAXES: 2'          DP3 = 'NAXES: 2'
   DP3 = 'AXIS.1: 1'         DP3 = 'AXIS.1: 2'
   DP3 = 'AXIS.2: 2'         DP3 = 'AXIS.2: 1'
   DP3 = ...                 DP3 = ...
     ⋮                         ⋮
```

While the *names* of the keywords have changed because $j = 1$ swaps with $j = 2$, and likewise for i, the keywords are in the same semantic order as before. Thus the *values* to the right of the equals signs are the same as before except for DP3·AXIS.1 and DP3·AXIS.2 which refer to axis numbers.

In this context we note that Paper I discourages permutation of the transformation matrix to handle such transpositions, i.e. by using off-diagonal elements, PC1_1 would become PC1_2, etc. Although simpler to implement (because only CRPIXja and PCi_ja need to be changed), such techniques produce representations that are too confusing for human interpretation. Moreover they would not work for prior distortion corrections.

2.5.2. Renormalization parameters

In practical applications, it is often desirable to renormalize the independent variables of the distortion function prior to calculation as this may help to avoid computational effects such as rounding errors and overflows or underflows. This is particularly the case when CDi_ja is used to define the transformation matrix with a sequent distortion correction, since CDi_ja incorporates the scaling of pixel coordinates to physical quantities.

Provision of an offset also allows subimaging applications. For example, if a CCD array has a fixed and well-known distortion correction, but only every second in the inner quarter of the CCD is recorded, then the standard distortion function may be used with offset and scale parameters chosen to match the subimage recorded.

Two values are required for each of $\hat{N}$ independent variables; an offset, applied first by subtraction, and then a multiplicative scale. DPja·OFFSET.$\hat{\jmath}$ (default value 0.0) and DPja·SCALE.$\hat{\jmath}$ (default value 1.0) will record the offset and scale for the variable indicated by DPja·AXIS.$\hat{\jmath}$.

These parameters, together with those of Sect. 2.5.1, bring the total number of parameters thus far defined to $1 + 3\hat{N}$.

2.5.3. Maximum correction keywords

As discussed in Sect. 1, typically the distortion correction will be used to provide small corrections over the best representation that can be obtained using an ideal coordinate system. WCS interpreters will have the option of ignoring the residual correction and accepting the accompanying error. However, in some cases the corrections will not be small enough to ignore. A method is therefore required to record what this error amounts to.

On an axis-by-axis basis the

`CPERR`*ja* (floating-valued)

and

`CQERR`*ia* (floating-valued)

keywords record a number that *exceeds* the maximum absolute value of the correction computed by Eq. (3) or (4) over the whole domain of the image.

Each `CPDIS`*ja* and `CQDIS`*ia* keyword in the header should have a matching `CPERR`*ja* or `CQERR`*ia* keyword; normally their values would be available as a by-product of the derivation of each distortion function. Typically, the number recorded will *equal* the maximum value but the looser condition facilitates subimaging without its recomputation. There is no default value for `CPERR`*ja* or `CQERR`*ia* if missing from the header but the action of WCS interpreters will be implementation dependent in such cases.

A keyword without axis number

`DVERR`*a* (floating-valued)

records the maximum error, Δ, of the combined distortion functions as an offset in pixel coordinates. This error is specified in pixel coordinates rather than world coordinates, or intermediate world coordinates, for a number of reasons:

- In general, a distortion defined by Eqs.(3) or (4) could give rise to a displacement in intermediate world coordinates, $(\Delta_{x_1}, \Delta_{x_2}, \ldots)$ in which the Δ_{x_i} are not commensurable (e.g. a mix of spatial and spectral coordinate elements, this is particularly the case where the `CD`*i*`_`*ja* matrix is used) and there is no simple way to combine these into a single error estimate.
- Pixel coordinates are linear across the image whereas world coordinates may be very non-linear; a small displacement in the image plane may lead to a large offset in world coordinates near points where the world coordinates are singular (e.g. near the native south pole of a zenithal equal area projection).
- Knowledge of the error in a world coordinate element (e.g. right ascension) may not be very useful in the absence of other information (e.g. the declination).
- If required, an offset in pixel coordinates is readily transformed to an offset in intermediate world coordinates by means of Eqs. (1) and (2) and this will often be interpretable in terms of the world coordinates (e.g. an offset in (x, y) coordinates in the plane of a celestial projection).
- Image data are often sampled at or near the Nyquist limit, so an offset in pixel coordinates may be interpretable in that context.

If there are only prior distortion corrections then

$$\Delta = \max\left(\sqrt{\sum_{j=1}^{N}\left(\delta_{p_j}(\boldsymbol{p})\right)^2}\right), \tag{9}$$

where the maximum is computed over the whole image. In the general case, where there are one or more sequent distortion corrections, the displacement in pixel coordinates may be found by computing $\boldsymbol{q}$ *without* any prior or sequent distortion corrections, and $\boldsymbol{q'}$ *with* all prior and sequent distortion corrections and applying the inverse of Eq. (1) to $\boldsymbol{q'} - \boldsymbol{q}$. Then Δ is the maximum value of this displacement computed for each point in the image.

`DVERR`*a* records a number that *exceeds* Δ. Normally, it will *equal* Δ but the looser condition facilitates subimaging without its recomputation. Use of `DVERR`*a*, while strongly recommended, remains optional. It has no default value.

3. Distortion functions

In this section we define a number of general purpose distortion functions that we expect will cover most situations although perhaps not as efficiently as functions specially tailored to their purpose. We need to be proactive in this because of the time involved in informing the FITS community of their definition and in distributing software to interpret them.

Of course, special-purpose distortion functions can still be defined but there is no guarantee that anyone will know what to do with them, other than ignore them and accept the residual error.

As previously, we use the circumflex (hat) accent to distinguish the different number of coordinate axes and indices used for the full set of pixel and intermediate pixel coordinates, p_j and q_i $(j, i = 1, \ldots, N)$ on the one hand, from the subset of these used as the independent variables of the distortion function, $\hat{p}_{\hat{j}}$ and $\hat{q}_{\hat{\imath}}$ $(\hat{j}, \hat{\imath} = 1, \ldots, \hat{N})$, on the other.

Also, where prior distortion functions are defined, exactly the same definition applies for sequent distortion functions, and vice versa.

As we have seen, there are $1 + 3\hat{N}$ distortion parameters that have a fixed interpretation for all distortion functions; the rest depend on the particular function.

3.1. `Polynomial`

Polynomial functions are widely used for fitting measured offsets and in this section we will consider how to encode them in a set of `DP`*ja* (likewise `DQ`*ia*) keywords. Conventional polynomials are commonly used, though particular sets of orthogonal polynomials are also often employed in order to reduce the correlation between best-fit coefficients in a least-squares regression. Legendre and Chebyshev polynomials, the latter of which minimize the maximum error of the fit, are suitable for rectangular fields, though as they are univariate functions, multi-dimensional fits must be handled as a product of one-dimensional fits. Zernike polynomials, which are based on the unit circle and thus inherently two-dimensional, are well suited to describe aberrations in optical systems.

When an orthogonal basis set of polynomials with independent variable x is used for fitting, the result can always be re-expressed simply as a conventional polynomial in x. Since we are concerned here only with defining the correction and not with the method by which it was originally determined, it is of no consequence that the factorization in terms of basis polynomials is not recorded explicitly (though it should be recoverable if the nature of the basis set is known).

However, problems may arise when x does not correspond to an element of pixel coordinate $\boldsymbol{p}$ when defining $\delta_{p_j}(\boldsymbol{p})$, or of intermediate pixel coordinate $\boldsymbol{q}$ in defining $\boldsymbol{\delta}_{\boldsymbol{q}_i}(\boldsymbol{q})$. For example, Zernike polynomials are usually expressed in terms of the polar coordinates (ρ, ϕ). In fact, as will be seen in Sect. 4.1, it is often useful to have the radial distance from the origin, ρ, as an independent variable when doing higher dimensional fits. Consequently, the formulation presented here will be sufficiently general to deal with ρ directly. However, we will not consider an azimuthal dependence based on coordinate ϕ explicitly since it has no general extension to three or more dimensions. Nevertheless, where ϕ appears as the argument of a trigonometric function it will often be possible to handle it indirectly. For example, the Zernike polynomials have the general form $R_n^m(\rho)\cos(m\phi)$ and $R_n^m(\rho)\sin(m\phi)$ for the even and odd polynomials respectively, where the radial functions, $R_n^m(\rho)$, are themselves orthogonal polynomials. Since

$$(\cos(m\phi), \sin(m\phi)) = (T_m(\chi), \eta U_{m-1}(\chi)), \tag{10}$$

where $(\chi, \eta) = (x/\rho, y/\rho)$ and T_m and U_m are Chebyshev polynomials of the first and second kind, it is apparent that a Zernike polynomial can always be re-expressed as a polynomial in ρ, χ, and η. (In fact, Zernike polynomials can always be re-expressed in the form $\sum_{ij} x^i y^j$, otherwise they wouldn't be called "polynomials".)

We now set DPja·NAUX, default value 0, to define a number, $K \geq 0$, of auxiliary variables (such as ρ for example). Denoting the $\hat{N}$ pixel coordinates that form the independent variables as $(\hat{p}_1, \hat{p}_2, \ldots)$, and the coefficients for the kth auxiliary variable, ρ_k, as $(a_{k_0}, a_{k_1}, a_{k_2}, \ldots a_{k_{\hat{N}}})$, and their powers as $(b_{k_0}, b_{k_1}, b_{k_2}, \ldots b_{k_{\hat{N}}})$, then

$$\rho_k = \left(a_{k_0} + a_{k_1}\hat{p}_1^{\,b_{k_1}} + a_{k_2}\hat{p}_2^{\,b_{k_2}} + \ldots\right)^{b_{k_0}}. \tag{11}$$

The coefficients will be encoded as DPja·AUX.k.COEFF.$\hat{\jmath}$ with default value 0.0, and the powers as DPja·AUX.k.POWER.$\hat{\jmath}$ with default value 1.0. For example, for $\hat{N} = 2$ the radial variable, ρ, encoded as the one and only auxiliary variable, would have parameters

```
DPja = 'NAUX: 1'
DPja = 'AUX.1.COEFF.1: 1'
DPja = 'AUX.1.POWER.1: 2'
DPja = 'AUX.1.COEFF.2: 1'
DPja = 'AUX.1.POWER.2: 2'
DPja = 'AUX.1.COEFF.0: 0'
DPja = 'AUX.1.POWER.0: 0.5'
```

The only restriction on the values of these coefficients is that ρ_k must be defined at all points within the image. In practice this means that care must be exercised to avoid the possibility of raising a negative number to a fractional power, or zero to a negative power.

Having defined the auxiliary independent variables, DPja·NTERMS with default value 0 defines the number of terms, $M \geq 0$, of the polynomial, each of the form

$$\kappa \hat{p}_1^{\lambda_1} \hat{p}_2^{\lambda_2} \ldots \hat{p}_{\hat{N}}^{\lambda_{\hat{N}}} \rho_1^{\mu_1} \ldots \rho_K^{\mu_K}. \tag{12}$$

These will be encoded as DPja·TERM.m.COEFF with default value 1.0 for κ, DPja·TERM.m.VAR.$\hat{\jmath}$ with default value 0.0 for $\lambda_{\hat{\jmath}}$, and DPja·TERM.m.AUX.k for μ_k also with default value 0.0.

As before, the only restriction on the values of these coefficients is that the polynomial must be defined at all points within the image. However, in order to accomodate terms of the form x/ρ which is indeterminate when $x = \rho = 0$, we specify that if any of the factors in Eq. (12) is zero, then the term is zero.

A total of $(1 + 3\hat{N}) + 1 + K(2\hat{N} + 2) + 1 + M(1 + \hat{N} + K)$ coefficients are required to define the distortion function which in fact is much more general than a simple polynomial. This consists of an overhead of $3(\hat{N} + 1) + K(2\hat{N} + 2)$ coefficients mostly used to define the independent and auxiliary variables, and an increment of $1 + \hat{N} + K$ coefficients for each term of the polynomial. For example, a 40-term polynomial in (x, y, ρ) would require 174 coefficients, rather more than if the DPja had been given static definitions as coefficients of predefined combinations of integer powers of the independent variables. However, this formalism permits the definition of an arbitrary number of auxiliary independent variables and does not restrict the degree of the polynomial. The admission of negative and fractional powers is also potentially very powerful.

Use of polynomial distortion functions will be signalled by setting CPDISja (likewise CQDISia) to 'Polynomial'.

3.2. Cubic spline

Use of polynomial distortion functions will be signalled by setting CPDISja or CQDISia to 'Cubic-spline'.

3.3. B-spline

Use of polynomial distortion functions will be signalled by setting CPDISja or CQDISia to 'B-spline'.

3.4. Lookup

Some coordinate systems are so irregular that they cannot be described with sufficient accuracy via analytic mathematical expressions employing a reasonable number of coefficients. Such systems are usually handled by some sort of table lookup method. Although Paper III has already defined a general-purpose '-TAB' algorithm of this nature, this is intended for use as the complete definition of coordinate systems that are very irregular, or indeed discontinuous, as illustrated by the examples given in Paper III. In the present context, the coordinate system is presumed to be defined quite well by one of the standard linear or non-linear coordinate descriptions, with only a

Table 1. Example distortion array header (blank lines have been inserted for clarity). Names used in the text for the pronumerals associated with each keyword are indicated in square brackets.

123456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789	
`XTENSION= 'IMAGE   '           / Image extension`	
`BITPIX  =                  -32 / IEEE floating-point`	
`NAXIS   =                    2 / 2-D image`	
`NAXIS1  =                  129 / Number of image columns`	$[\hat{N}_1]$
`NAXIS2  =                  129 / Number of image rows`	$[\hat{N}_2]$
`PCOUNT  =                    0 / Special data area of size zero`	
`GCOUNT  =                    1 / One data group`	
`EXTNAME = 'WCSDVARR'           / WCS distortion array`	
`EXTVER  =                    1 / Distortion array version no.`	
`CRPIX1  =                 65.0 / Distortion array reference pixel`	$[\hat{r}_1]$
`CDELT1  =                  8.0 / Grid step size in 1st coordinate`	$[\hat{s}_1]$
`CRVAL1  =                513.0 / Image array pixel coordinate`	$[\hat{w}_1]$
`CRPIX2  =                  1.0 / Distortion array reference pixel`	$[\hat{r}_2]$
`CDELT2  =            7.9921875 / Grid step size in 2nd coordinate`	$[\hat{s}_2]$
`CRVAL2  =                  1.0 / Image array pixel coordinate`	$[\hat{w}_2]$
`END`	

small residual correction to be applied. Use of `'-TAB'` in such cases would disguise the basic nature of the coordinate system, and might also be relatively inefficient.

For example, celestial coordinates in optical photographic plates are often adequately represented by a `'-TAN'` projection with a small residual correction that can be described by the lookup method to be described here. Using `'-TAB'` in this case would preclude the use of `'-TAN'` thereby obscuring the fundamental nature of the coordinate system, and probably requiring a larger lookup table. Also, the assumption that distortion corrections are small and smoothly varying allows a rather simpler lookup method to be defined here. In particular it omits the powerful, though complicated indexing method employed by `'-TAB'`. As a practical example of the distinction between between the two methods, a map of the Earth might use a standard spherical projection with a `'Lookup'` distortion correction to correct for the Earth's mild oblateness, whereas a map of the very irregular surface of the asteroid Eros would probably gain little by modelling it as a sphere and could use `'-TAB'` directly to good advantage.

In the one-dimensional case, the usual approach is to define a "lookup table" with dependent and independent variables occupying separate columns. However, in the general multi-dimensional case it is more appropriate to speak of an "array" for which the correction, stored as the array value, is indexed by a number of independent variables. Such an array may be treated as an ordinary FITS image, subject to the usual forms of display and analysis. Having two arrays provides great scope for confusion so we will be careful always to distinguish between the image array and the distortion array. In particular, in this section, all pronumerals associated with the distortion array will be distinguished by a circumflex ("hat") accent, e.g. $\hat{r}_{\hat{\jmath}}$.

At one extreme, the distortion array could simply store a correction for each pixel in the image. In practice, though, sufficient accuracy should be obtainable by interpolating on a regularly spaced sub-grid. For example, if a correction were provided for every 10$^{\text{th}}$ image pixel in either direction in a two-dimensional image then the distortion array would have a size only 1% of that of the image.

Distortion array values will be stored as a FITS image extension (Ponz et al. 1994) for which `EXTNAME` is set to `'WCSDVARR'`. `EXTVER` will be used to distinguish between multiple distortion arrays within a single FITS file; each must define `EXTVER` with a unique value and `DP`ja·`EXTVER` will be used to select one. Hanisch et al. (2001) specify that `EXTVER` has a default value of 1 if omitted from the FITS header, and so likewise for the corresponding `DP`ja keyword.

The dimensionality of the distortion array, denoted by $\hat{N}$, is defined by `DP`ja·`NAXES`. The first axis of the distortion array, $\hat{\jmath} = 1$, corresponds to image array pixel axis j_1 (or intermediate pixel axis i_1) defined by `DP`ja·`AXIS.1`; the second axis, $\hat{\jmath} = 2$, corresponds to j_2 given by `DP`ja·`AXIS.2`, and so on, so that in general, distortion array axis $\hat{\jmath}$ matches image array axis $j_{\hat{\jmath}}$. There is no need for the image and distortion array axes to match, either in number or order, nor for any of the $j_{\hat{\jmath}}$ to match the j that appears in `DP`ja. However, no two axes in the distortion array are allowed to match the same axis in the image array. Hence, $\hat{N} \leq N$, where N is the dimensionality of the image array.

Standard WCS keywords, `CRPIX`$\hat{\jmath}$, `CDELT`$\hat{\imath}$, and `CRVAL`$\hat{\imath}$, in the distortion array header will define the association between distortion array pixel coordinates and image array pixel coordinates (or intermediate pixel coordinates - but henceforth we will consider only $\boldsymbol{\delta_p}(\boldsymbol{p})$, the treatment for $\boldsymbol{\delta_q}(\boldsymbol{q})$ being similar). PC$\hat{\imath}$_$\hat{\jmath}$ will not be used and hence we do not distinguish between $\hat{\imath}$ and $\hat{\jmath}$ subscripts on the FITS keywords.

CRPIX$\hat{\jmath}$ defines the elements, $\hat{r}_{\hat{\jmath}}$, of the reference pixel coordinate in the distortion array which correspond to image pixel coordinate elements $\hat{w}_{\hat{\jmath}}$ defined by CRVAL$\hat{\jmath}$.

Elements along axis $\hat{\jmath}$ in the distortion array are spaced by $\hat{s}_{\hat{\jmath}}$ pixels along axis $j_{\hat{\jmath}}$ of the image array, where $\hat{s}_{\hat{\jmath}}$, specified by CDELT$\hat{\jmath}$, need not be integral and may be negative, but must be non-zero.

From the foregoing, the relationship between distortion array pixel coordinate element $\hat{p}_{\hat{\jmath}}$ and image pixel coordinate element $p_{j_{\hat{\jmath}}}$ is

$$p_{j_{\hat{\jmath}}} = \hat{s}_{\hat{\jmath}}\,(\hat{p}_{\hat{\jmath}} - \hat{r}_{\hat{\jmath}}) + \hat{w}_{\hat{\jmath}}. \qquad (13)$$

A distortion array may be associated with a subimage extracted from an image by suitably changing CRPIX$\hat{\jmath}$, CRVAL$\hat{\jmath}$, and/or CDELT$\hat{\jmath}$. If storage space is a consideration, this could also be accompanied by a reduction in the number of columns and rows.

However, as discussed in Sect. 2.1, in all cases the distortion array must cover the whole of the image array. This "no-extrapolation" requirement, in conjunction with the differential of Eq. (13),

$$\Delta p_{j_{\hat{\jmath}}} = \hat{s}_{\hat{\jmath}}\,\Delta\hat{p}_{\hat{\jmath}}, \qquad (14)$$

has repercussions for the geometry of the distortion array:

- $\hat{N}_{\hat{\jmath}} > 1$ for all axes in the image array with $N_{j_{\hat{\jmath}}} > 1$. This arises because if there are two pixels on axis $j_{\hat{\jmath}}$ of the image array so that $\Delta p_{j_{\hat{\jmath}}} > 0$, then there must also be two pixels in the distortion array so that $\Delta\hat{p}_{\hat{\jmath}} > 0$.
- Considering that it may not always be possible to determine a sensible distortion correction beyond the domain of the image array, it will often be required that the edges of the distortion array coincide with the edges of the image array (i.e. so that $p_{j_{\hat{\jmath}}} = 1$ for $\hat{p}_{\hat{\jmath}} = 1$ and $p_{j_{\hat{\jmath}}} = N_{j_{\hat{\jmath}}}$ for $\hat{p}_{\hat{\jmath}} = \hat{N}_{\hat{\jmath}}$). In that case Eq. (14) gives

$$\hat{s}_{\hat{\jmath}} = (N_{j_{\hat{\jmath}}} - 1)/(\hat{N}_{\hat{\jmath}} - 1). \qquad (15)$$

While non-integral values of $\hat{s}_{\hat{\jmath}}$ are allowable, in constructing the distortion array they necessitate computation of the distortion correction at fractional pixel coordinates in the image array. On the other hand, if $\hat{s}_{\hat{\jmath}}$ is to be integral, Eq. (15) restricts the possible values of $N_{j_{\hat{\jmath}}}$ and $\hat{N}_{\hat{\jmath}}$. A good strategy might be to make each of the form $2^n + 1$.

Table 1 illustrates this last point for a distortion array with $\hat{N}_1 = \hat{N}_2 = 129$ constructed for an image array of dimensions $N_{j_1} = 1025$ and $N_{j_2} = 1024$. However, in this case $\hat{s}_2$ might have been made integral by constructing the distortion array with $\hat{N}_2 = 93$ whence $\hat{s}_2 = 11$.

In constructing the distortion array, $\hat{s}_{\hat{\jmath}}$ must be chosen small enough so that the correction at any given pixel may be determined with sufficient accuracy by linear interpolation within the distortion array. Higher-order interpolation is proscribed since it may lead to anomalies in the presence of discontinuities in the coordinate value or its derivatives, and also because having a fixed interpretation will eliminate potential confusion as to what exactly the correction is meant to be.

Linear interpolation in the $\hat{N}$-dimensional distortion array may be done conveniently via the following prescription:

1. Given the image pixel coordinates of point $\mathbf{P}$ for which the distortion correction is required, determine the corresponding pixel coordinate in the distortion array via the inverse of Eq. (13):

$$\hat{p}_{\hat{\jmath}} = \hat{r}_{\hat{\jmath}} + (p_{j_{\hat{\jmath}}} - \hat{w}_{\hat{\jmath}})/\hat{s}_{\hat{\jmath}}. \qquad (16)$$

These must be such that $1 \le \hat{p}_{\hat{\jmath}} \le \hat{N}_{\hat{\jmath}}$ otherwise the point is outside the distortion array and the result undefined. If $\hat{p}_{\hat{\jmath}} = \hat{N}_{\hat{\jmath}}$ then decrement it by 1.
2. Determine the coordinates of point $\mathbf{P_0}$ in the distortion array with integral pixel coordinate elements $\lfloor\hat{p}_{\hat{\jmath}}\rfloor$, where the *floor* function, $\lfloor x\rfloor$, gives the largest integer less than or equal to x.
3. Form the coordinates of the $2^{\hat{N}}$ data points in the distortion array surrounding $\mathbf{P}$ by incrementing the pixel coordinate elements of $\mathbf{P_0}$ in all binary combinations, i.e. the vector sum

$$\mathbf{P_k} = \mathbf{P_0} + \mathbf{B}(k), \qquad k = 0, \ldots, 2^{\hat{N}} - 1 \qquad (17)$$

where $\mathbf{B}(k)$ is the pixel coordinate whose elements correspond to the digits of k expressed as a binary number with least significant digit first. For example, $\mathbf{B}(5) = (1, 0, 1, 0, \ldots, 0)$. In fact, the ordering of the binary combinations is arbitrary as long as each is included exactly once; this particular ordering is suggested because it conforms to the storage order of the distortion array.
4. Calculate the weight, W_k, for point $\mathbf{P_k}$ as the absolute value of the product of the elements of $\bar{\mathbf{B}}(k) - (\mathbf{P} - \mathbf{P_0})$, where $\bar{\mathbf{B}}(k)$ is the complement of $\mathbf{B}(k)$, e.g. $\bar{\mathbf{B}}(5) = (0, 1, 0, 1, \ldots, 1)$.
5. Look up the distortion correction, δ_k, at each of the $2^{\hat{N}}$ data points, $\mathbf{P_k}$. The interpolated correction is then

$$\delta_{p_j} = \sum_{k=0}^{2^{\hat{N}}-1} W_k \delta_k. \qquad (18)$$

In a practical implementation steps (3) to (5) would be rolled into a loop over index k.

Use of polynomial distortion functions will be signalled by setting CPDIS*ja* or CQDIS*ia* to 'Lookup'.

4. Applications

4.1. Astrometry

Photographic plates or CCD images taken with optical or other telescopes are susceptible to distortions. Astrometrists have traditionally dealt with the problem by means of a *plate solution*.

Section 2.3 of Eichhorn (1974) discusses two schools of thought in the construction of a plate solution. Each is based on analysis of the Cartesian (x, y) *plate coordinates* of reference stars that have accurately known celestial coordinates. Plate coordinates are measured in the image plane (e.g. photographic

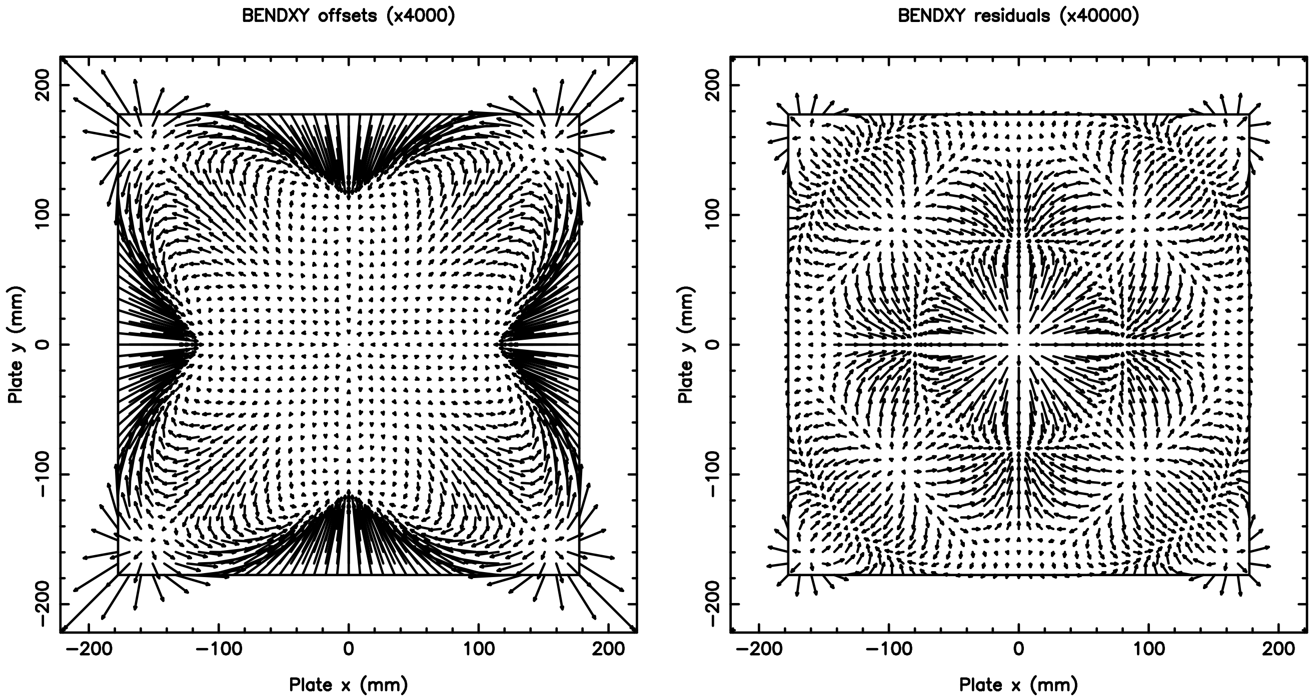

Fig. 3. (Left) Offsets (×4, 000) computed by BENDXY on a 40 × 40 grid over the 350mm × 350mm plate area. The rms deviation of the offsets is 3.7 μm and the maximum deviation is 15.4 μm. (Right) Residuals (× 40,000) of the BENDXY offsets after fitting by a 7th-degree polynomial in x, y, and r. The rms deviation of the residuals is 0.10 μm and the maximum deviation is 0.47 μm. Note that the residuals are scaled by an additional factor of ×10 over the offsets.

plate) to high-precision with a measuring engine (usually in μm or mm rather than degrees). They may deviate in a complicated way from the *standard coordinates* (ξ, η) expected of a simple gnomonic projection (see also Murray 1983).

The *model approach* seeks to identify what it is that causes the plate coordinates to deviate from standard coordinates. Mathematical formulæ are derived for geometric effects such as the translation, rotation, and tilt of the photographic plate; non-linearities in the coordinate measuring machine; physical effects such as refraction, aberration, precession and nutation; and optical defects such as radial and decentering distortion. These effects are parameterized by a number of unknowns that must be determined by comparing the plate coordinates of the reference stars with their computed standard coordinates.

The *empirical approach*, on the other hand, simply writes

$$\begin{aligned} \xi &= \sum a_{ijkl} x^i y^j m^k c^l, \\ \eta &= \sum b_{ijkl} x^i y^j m^k c^l. \end{aligned} \quad (19)$$

where m and c are magnitude and colour index, and determines the polynomial coefficients by least squares analysis of the plate coordinates of the reference stars. It is left to empirical investigation to decide which terms are required in each of the polynomials.

In terms of efficiency the model approach has a definite advantage in that it generally requires fewer free parameters to be determined than the empirical approach and they tend to be much less correlated thereby making the fit more reliable. However, the empirical approach is arguably simpler and more flexible and could unwittingly account for effects inadvertently omitted from a plate model. It was the method used for the Guide Star Catalogue (GSC, Lasker et al., 1990) of the Hubble Space Telescope (HST) and consequently for the FITS headers of the Digitized Sky Survey (DSS, 1992) and is the method adopted here for *image transport* purposes. Of course, a model fit can still be used initially to determine a plate solution, then a least squares polynomial fit to this model used solely for FITS image transport. This will satisfy FITS readers who simply want accurate image coordinates and are not particularly interested in the details of the plate solution (such as was the case with DSS).

Magnitude and colour terms in Eq. (19) cannot be handled because any dependence of coordinates on image pixel *values* is outside the scope of this paper. However, the provision of secondary coordinate descriptions in Paper I does alleviate this to some extent since FITS writers could provide alternate descriptions for particular magnitude or colour ranges if that level of accuracy was warranted.

The main question is the degree of the polynomial that might be needed. A first-degree polynomial, comprising three terms for each of ξ and η is sufficient to account for an affine transformation - translation, scale, rotation and skew. We note that this so-called *six-constant model* can be handled by Eq. (1) alone. Second degree terms are required to account for "plate skewness" which results from assuming the incorrect tangent point of the projection. König (1962) shows that 3rd-degree terms are usually sufficient to account for refraction effects even in large fields (though higher degrees may sometimes be necessary), and Eichhorn (1974) provides terms up to 3rd-degree for radial and decentring distortion of the optical system.

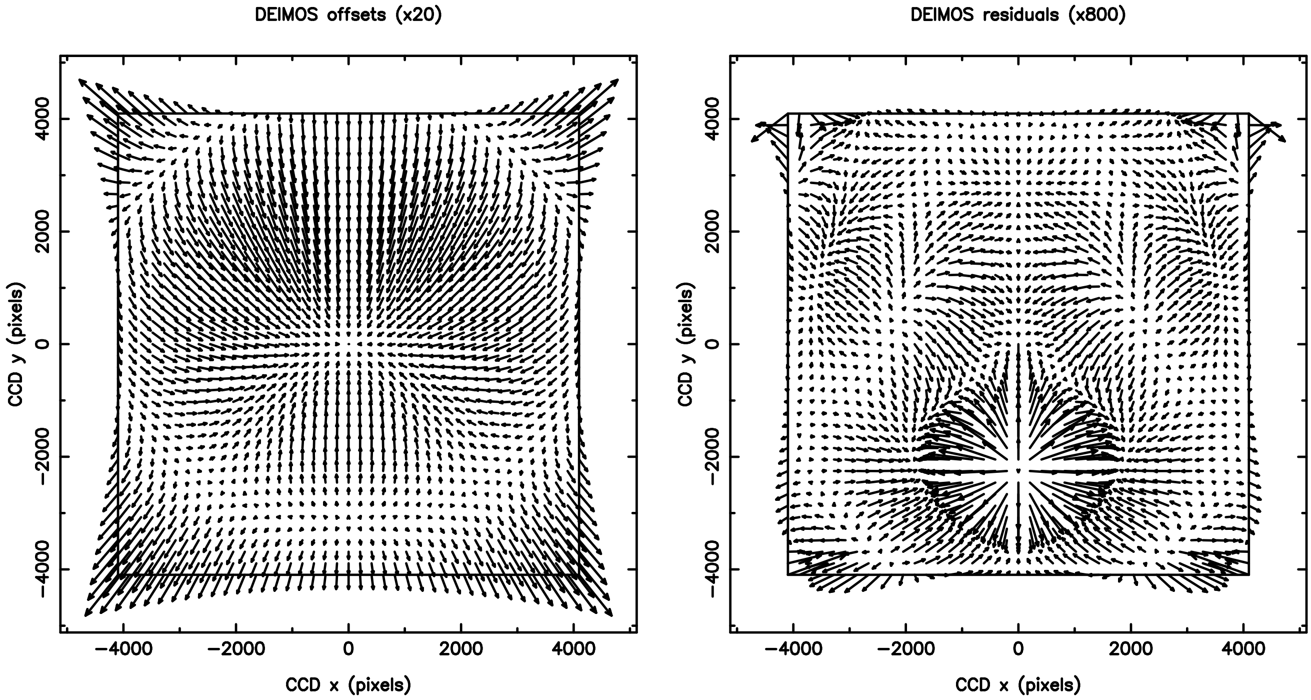

Fig. 4. (Left) DEIMOS offsets ($\times$20) computed on a 40 $\times$ 40 grid over the 8192 $\times$ 8192 pixel CCD array. The rms deviation of the offsets is 7.3 pixel with maximum deviation 46.3 pixel. (Right) Residuals ($\times$ 800) of the DEIMOS offsets after fitting by a 7th-degree polynomial in x, y, and r. The residuals, which are scaled by an additional factor of $\times$40 over the offsets, have an rms deviation of 0.19 pixel and a maximum deviation 1.38 pixel. This corresponds to 23 and 164 milliarcsec at the stated scale of 0.119 arcsec/pixel.

In practice, Russell et al. (1990) used a 3rd-degree polynomial for the GSC, though the DSS did provide for a 5th-degree term of the form $(\xi, \eta) \propto (xr^4, yr^4)$ but it was never used. However, they report that 3rd-degree polynomials may leave systematic uncorrected errors towards the edge of the Schmidt plate. Murray (1984) notes that the Schmidt plate geometry is more naturally described by a zenithal equidistant (ARC) projection rather than gnomonic, but a 3rd-degree radial distortion term accounts for the difference. The solution presented by Holtzman et al. (1995) for the Wide Field Planetary Camera 2 (WFPC2) on the HST also only contains terms up to 3rd-degree, as does the solution by Aussel et al. (1999) for ISOCAM observations of the Hubble Deep Field.

However, it has been found that terms up to 7th-degree may be required to model the complex distortions found in some optical systems. For example the three coefficient trigonometric function that corrects for the distortions found in the corners and towards the perimeter of UKST Schmidt plates, known as BENDXY, gives the azimuthal and radial distortion as

$$\Delta\zeta = ar^4 \cos 4\zeta, \qquad (20)$$

$$\Delta r = br^4 \sin 4\zeta + cr^4 \cos(4r/f), \qquad (21)$$

where $\zeta = \arg(x, y)$ and[4] f is the plate half-width, and this was well approximated by a pair of 7th-degree polynomials with 10 terms for each of ξ and η. The results are shown in Fig. 3.

Another example considered was the so-called dual point-of-symmetry distortion model for the DEIMOS Deep Extragalactic Imaging Multi-Object Spectrograph developed by the UCO/Lick Observatory for installation on the Keck II telescope. This nine-coefficient model consists of a 3rd-degree polynomial that defines a radial correction for the telescope distortion combined with a 5th-degree polynomial that defines a radial correction for the camera distortion. A displacement between the camera and telescope optical axes results in a complicated non-radial distortion pattern in the image plane as shown in Fig. 4. The pattern has reflection symmetry - "dual point-of-symmetry" referring to the component distortion polynomials. The DEIMOS distortion field was closely approximated by a 7th-degree polynomial with 24 terms for ξ and 16 terms for η. In fact, within the imaging region of the CCD array a fifth-, or perhaps a 4th-degree polynomial with 15 + 9 or 11 + 6 terms would almost certainly suffice.

However, it was found in the DEIMOS analysis that inclusion of terms in odd powers of r^n significantly improved the fit. We note that these r^n terms *do not* define a radial distortion, this being handled by terms of the form $(\xi, \eta) = (xr^n, yr^n)$.

Nevertheless, some instruments require polynomials of degree much higher than seven to model their distortions. The Faint Object Camera (FOC) of the HST is one such and Perry Greenfield (1995) used B-spline functions in this case.

From the discussion in Sect. 5.1 and 5.1.3 of Paper II we may write the equations for the gnomonic projection in standard coordinates as

$$\xi = \frac{180^\circ}{\pi} \sin\phi \cot\theta, \qquad (22)$$

$$\eta = -\frac{180^\circ}{\pi} \cos\phi \cot\theta, \qquad (23)$$

[4] arg() is the inverse tangent function returning angles in the correct quadrant.

(24)

with inverse

$$\phi = \arg(-\eta, \xi), \tag{25}$$

$$\theta = \tan^{-1}\left(\frac{180^\circ}{\pi\sqrt{\xi^2+\eta^2}}\right). \tag{26}$$

It remains only to connect (x, y) to (ξ, η) via a suitable distortion function.

In this case (ξ, η) are elements of the intermediate world coordinate vector $\boldsymbol{x}$ that must be scaled to "degree" units from the corrected intermediate pixel coordinates $\boldsymbol{q'}$ by applying $\boldsymbol{s}$. As noted in Sect. 2.1, the units of $\boldsymbol{q'}$ and $\boldsymbol{q}$ must match. It is therefore invalid, for example, for a CD*i*_*ja* matrix to scale q_i to mm, and then for $\delta_{q_i}(\boldsymbol{q})$ to scale q'_i to degrees. Thus when CD*i*_*ja* is used, q_i must be in degrees.

As noted above, the six-constant plate model, which accounts for translation, rotation, skewness and scale, can be handled solely by Eq. (1). Ideally the plate solution should be constructed in such a way that this affine transformation is handled by the CRPIX*ja*, PC*i*_*ja*, and CDELT*ia* (or CD*i*_*ja*) header cards. The first-degree terms of the distortion polynomial would then define an identity transformation with the remaining terms providing second-, and higher-degree corrections.

When CD*i*_*ja* is used q_i must be in degrees so the distortion polynomial must be expressed in degrees. When PC*i*_*ja* and CDELT*ia* are used there is greater freedom, q_i might be in pixels, mm, or degrees, though ideally scaling to degrees should be left for CDELT*ia*. In either case, the renormalization factors described in Sect. 2.5.1 provide an opportunity to rescale the independent variables of the distortion polynomial to convenient units, and this may facilitate translation of existing plate solutions. Section 5.2 provides an example of translating a Digitized Sky Survey (DSS) FITS header into a TAN projection with sequent polynomial distortion function.

In practice the distorted TAN projection differs little in appearance from the gnomonic projection shown in Fig. 8 of Paper II.

4.2. Spectroscopy

Wide field Doppler correction...

5. Example headers

5.1. Header interpretation example 1

Example of interpreting a one-dimensional wavelength spectrum with a simple polynomial distortion correction...

5.2. Header construction example 1

In creating the Guide Star Catalogue for the Hubble Space Telescope (HST), the Space Telescope Science Institute (STScI) digitized the optical plates of the SERC Southern Sky Survey obtained by the UK Schmidt Telescope together with the Palomar Observatory Sky Survey (POSS) of the northern sky obtained by the Oschin Schmidt Telescope.

The resulting Digitized Sky Survey (DSS) which covers the entire sky is generally available as a set of 102 CDROM disks and constitutes an extremely valuable astronomical resource.

The coordinate system associated with the DSS is described in the booklets provided with the CDROM set. It is established by a set of coefficients that define a polynomial plate solution. The coefficients are encoded in an ad hoc way in a set of FITS header cards and special purpose software is provided to interpret the coordinate system.

Equations (27) to (32) reproduce the coordinate transformation described in the DSS booklet, except that some variable names have been changed to avoid conflict with usage in this paper, and conversion from arcsec to radians is shown explicitly in Eqs. (31) and (32).

The first step in computing celestial coordinates from DSS pixel coordinates (P_1, P_2) is to compute linear offsets (X, Y) in a *left-handed* Cartesian coordinate system measured in mm from the plate centre

$$X = (x_c - \rho_x P_1)/1000, \tag{27}$$

$$Y = (\rho_y P_2 - y_c)/1000, \tag{28}$$

where (x_c, y_c) are the plate centre coordinates in μm, and (ρ_x, ρ_y) are the dimensions of a pixel, in μm. DSS pixel coordinates are measured with respect to the *bottom left-hand corner* of a pixel, thus the first sample in the image has $(P_1, P_2) = (1.5, 1.5)$, which is at variance with the basic FITS standard. Subimaging is done by defining (P_1, P_2) for the bottom left-hand corner of the first pixel in the subimage via the CNPIX1 and CNPIX2 header cards. Standard coordinates are computed via

$$\begin{aligned}\Xi = {} & A_1X + A_2Y + A_3 + A_4X^2 + A_5XY + A_6Y^2 + \\ & A_7\left(X^2+Y^2\right) + A_8X^3 + A_9X^2Y + A_{10}XY^2 + \\ & A_{11}Y^3 + A_{12}X\left(X^2+Y^2\right) + A_{13}X\left(X^2+Y^2\right)^2,\end{aligned} \tag{29}$$

$$\begin{aligned}H = {} & B_1Y + B_2X + B_3 + B_4Y^2 + B_5XY + B_6X^2 + \\ & B_7\left(X^2+Y^2\right) + B_8Y^3 + B_9XY^2 + B_{10}X^2Y + \\ & B_{11}X^3 + B_{12}Y\left(X^2+Y^2\right) + B_{13}Y\left(X^2+Y^2\right)^2,\end{aligned} \tag{30}$$

where (Ξ, H) are in arcsec. Note that $(\Xi, H) = (A_3, B_3)$ at $(X, Y) = (0, 0)$ so the nominal plate centre is offset from the reference point. In practice this offset frequently exceeds 5 arcmin.

The J2000.0 right ascension and declination are then given by

$$\alpha = \alpha_c + \tan^{-1}\left(\frac{\gamma\Xi/\cos\delta_c}{1-\gamma H\tan\delta_c}\right), \tag{31}$$

$$\delta = \tan^{-1}\left(\frac{(\gamma H + \tan\delta_c)\cos(\alpha-\alpha_c)}{1-\gamma H\tan\delta_c}\right), \tag{32}$$

where (α_c, δ_c) are the plate centre J2000.0 right ascension and declination, and $\gamma = \pi/(180\times3600)$ is a factor to convert from arcsec to radians. These coefficients are encoded in the DSS FITS header via the following keywords

x_c	PP03
y_c	PP06
ρ_x	XPIXELSZ
ρ_y	YPIXELSZ
A_1	AMDX1
$\vdots$	$\vdots$
A_{13}	AMDX13
B_1	AMDY1
$\vdots$	$\vdots$
B_{13}	AMDY13
α_c	PLTRAH, PLTRAM, PLTRAS
δ_c	PLTDECSN, PLTDECD, PLTDECM, PLTDECS

The basic FITS header cards, CRPIXj, CDELTi, CTYPEi, CRVALi do not appear in the DSS header.

In this example we will translate a DSS header into a gnomonic (TAN) projection as defined in Sect. 4.1. We note that, like the TAN projection, the DSS transformation is defined as a deprojection. That is, the prescription given is that for computing celestial coordinates from Cartesian coordinates in the plane of projection. Moreover, Eqs. (31) and (32) follow standard astrometric practice in describing a gnomonic projection in celestial coordinates (Russell et al., 1990) and this makes the translation relatively straightforward. The derivation is given in astrometry texts such as Section 5.4 of van de Kamp (1967) or Section 161 of Smart (1965). To reproduce it we use the equations for computing celestial coordinates (α, δ) from native coordinates (ϕ, θ) from Paper II,

$$\begin{aligned} \alpha &= \alpha_p + \arg(\sin\theta\cos\delta_p - \cos\theta\sin\delta_p\cos(\phi-\phi_p), \\ &\qquad -\cos\theta\sin(\phi-\phi_p)) \\ \delta &= \sin^{-1}(\sin\theta\sin\delta_p + \cos\theta\cos\delta_p\cos(\phi-\phi_p)). \end{aligned} \tag{33}$$

Setting $\phi_p = 180°$ for a zenithal projection in the first of these and, noting that $\sin\theta\cos\delta_p > 0$ in the region of the native pole, we have

$$\alpha = \alpha_p + \arg(1 + \cos\phi\cot\theta\tan\delta_p, \sin\phi\cot\theta/\cos\delta_p).$$

Substituting Eqs. (22) and (23) and noting that the x-term of the arg() function is always positive so that it reduces to a simple arctangent we obtain

$$\alpha = \alpha_p + \tan^{-1}\left(\frac{\kappa\xi/\cos\delta_p}{1 - \kappa\eta\tan\delta_p}\right), \tag{34}$$

where $\kappa = \pi/180°$. Likewise, using the second of Eqs. (33) and the following relation from Paper II

$$\cos\delta\cos(\alpha-\alpha_p) = \sin\theta\cos\delta_p - \cos\theta\sin\delta_p\cos(\phi-\phi_p) \tag{35}$$

together with Eq. (23) we have

$$\delta = \tan^{-1}\left(\frac{(\kappa\eta + \tan\delta_p)\cos(\alpha-\alpha_p)}{1-\kappa\eta\tan\delta_p}\right). \tag{36}$$

Comparing Eq. (31) with (34) and Eq. (32) with (36) it is clear that we must make the associations

$$\begin{aligned} \Xi &= 3600\xi, \\ H &= 3600\eta, \\ \alpha_c &= \alpha_p, \\ \delta_c &= \delta_p. \end{aligned}$$

In other words, (Ξ, H) are simply (ξ, η) measured in arcsec. In the first instance we may therefore write

$$\begin{aligned} \texttt{CTYPE1} &= \texttt{`RA---TAN'}, \\ \texttt{CTYPE2} &= \texttt{`DEC--TAN'}, \\ \texttt{CRVAL1} &= \alpha_c, \\ \texttt{CRVAL2} &= \delta_c, \\ \texttt{LONPOLE} &= 180°. \end{aligned}$$

It now remains to translate the steps of the algorithm chain that comprise the linear transformation and distortion function. As mentioned in Sect. 4.1, use of PCi_ja and CDELTia allows a degree of freedom in the scaling of (q_1, q_2). Since the DSS polynomial is expressed in terms of independent variables (X, Y) measured in mm, it is convenient to use a sequent distortion correction following a PCi_ja matrix that scales $\boldsymbol{p}$ to $\boldsymbol{q}$ in mm, so that (q_1, q_2) are analogous to (X, Y). Even so, the DSS polynomial coefficients cannot be copied directly since (Ξ, H) are in arcsec, contrary to the caveat that $\boldsymbol{q}'$ must be measured in the same units as $\boldsymbol{q}$. Nevertheless, the translation may be effected with a straightforward rescaling of the DSS coefficients to produce $\boldsymbol{q}'$ in mm. Then a rescaling applied by CDELTia converts $\boldsymbol{q}'$ to $\boldsymbol{x}$ in degrees, as required for a TAN projection, so (x_1, x_2) may be identified with $(\xi, \eta) = (\Xi, H)/3600$.

It has often been stated in this paper that the distortion function should be used to correct for small residual errors that cannot otherwise be accounted for. Accordingly, the translation will proceed in two steps. Firstly we will deduce CRPIXja, PCi_ja and CDELTia for the first-order approximation of Eqs. (29) and (30)

$$\Xi \approx A_1X + A_2Y + A_3, \tag{37}$$

$$H \approx B_1Y + B_2X + B_3. \tag{38}$$

Then we will deduce the distortion function that corrects for the higher-order terms. It will simplify matters to work consistently in mm, rewriting Eqs. (27) and (28) as

$$X = X_c - R_xP_1, \tag{39}$$

$$Y = R_yP_2 - Y_c, \tag{40}$$

where

$$\begin{aligned} (X_c, Y_c) &= (x_c, y_c)/1000, \\ (R_x, R_y) &= (\rho_x, \rho_y)/1000. \end{aligned}$$

The first task is to translate non-standard DSS pixel coordinates (P_1, P_2) to standard FITS (p_1, p_2) pixel coordinates. The bottom left-hand corner of the first pixel in a DSS subimage has coordinates defined by the CNPIX1 and CNPIX2 header cards whereas the (p_1, p_2) pixel coordinates of this point are always $(0.5, 0.5)$. Hence

$$P_1 = p_1 + (\texttt{CNPIX1} - 0.5), \tag{41}$$

$$P_2 = p_2 + (\texttt{CNPIX2} - 0.5). \tag{42}$$

It was noted above that the nominal plate centre does not coincide with the reference point of the projection to within tolerable accuracy. Consequently Eqs. (29) and (30) introduce a significant offset that otherwise can only be accounted for by CRPIXja. Referring to Eqs. (37) and (38), let (X_0, Y_0) be such that

$$A_1X_0 + A_2Y_0 + A_3 = 0, \tag{43}$$
$$B_1Y_0 + B_2X_0 + B_3 = 0. \tag{44}$$

Substituting Eq. (41) into (39) and Eq. (42) into (40) we have

$$X = X_c - R_x(p_1 + (\mathtt{CNPIX1} - 0.5)), \tag{45}$$
$$Y = R_y(p_2 + (\mathtt{CNPIX2} - 0.5)) - Y_c. \tag{46}$$

Substituting (X_0, Y_0) and rearranging we obtain the reference pixel coordinates CRPIXja

$$\begin{aligned}\mathtt{CRPIX1} &= r_1 = (X_c - X_0)/R_x - (\mathtt{CNPIX1} - 0.5),\\ \mathtt{CRPIX2} &= r_2 = (Y_c + Y_0)/R_y - (\mathtt{CNPIX2} - 0.5),\end{aligned}$$

and also

$$X = X_0 - R_x(p_1 - r_1), \tag{47}$$
$$Y = Y_0 + R_y(p_2 - r_2). \tag{48}$$

It is convenient first to deduce the elements of the CDi_ja matrix and then consider how to split this into PCi_ja and CDELTia. Substituting Eqs. (47) and (48) into Eqs. (37) and (38), and using Eqs. (43) and (44) to simplify we have

$$\begin{aligned}\Xi &\approx -A_1R_x(p_1 - r_1) + A_2R_y(p_2 - r_2),\\ H &\approx -B_2R_x(p_1 - r_1) + B_1R_y(p_2 - r_2).\end{aligned}$$

Referring to Eqs.(1) and (2) and noting that $(x_1, x_2) = (\xi, \eta) = (\Xi, H)/3600$ we obtain the elements of the CDi_ja matrix

$$\begin{aligned}\mathtt{CD1\_1} &= s_1m_{11} = -A_1R_x/3600,\\ \mathtt{CD1\_2} &= s_1m_{12} = A_2R_y/3600,\\ \mathtt{CD2\_1} &= s_2m_{21} = -B_2R_x/3600,\\ \mathtt{CD2\_2} &= s_2m_{22} = B_1R_y/3600.\end{aligned}$$

We require that the PCi_ja matrix scale $\boldsymbol{p}$ to $\boldsymbol{q}$, and hence $\boldsymbol{q}'$, in mm, and that CDELTia scale $\boldsymbol{q}'$ from mm to degrees. Now A_1, A_2, B_1, and B_2 mainly describe a small rotation followed by a scaling from mm to degrees that is very nearly isotropic. Since we are completely free in the choice of CDELTia it is convenient to force these to scale isotropically with any small non-isotropic residuals remaining in the PCi_ja matrix. Thus

$$\begin{aligned}\mathtt{CDELT1} &= s_1 = -S/3600,\\ \mathtt{CDELT2} &= s_2 = S/3600,\end{aligned}$$

where

$$S = \sqrt{A_1A_2 - B_1B_2}.$$

Note that s_1 is negative in accord with the general practice of setting CDELTia negative for the RA axis. The elements of the PCi_ja matrix follow directly:

$$\begin{aligned}\mathtt{PC1\_1} &= m_{11} = -A'_1R_x,\\ \mathtt{PC1\_2} &= m_{12} = A'_2R_y,\\ \mathtt{PC2\_1} &= m_{21} = -B'_2R_x,\\ \mathtt{PC2\_2} &= m_{22} = B'_1R_y,\end{aligned}$$

where

$$(A'_m, B'_m) = (-A_m, B_m)/S.$$

The reason for this definition will become apparent later.

At this point we have approximated the DSS coordinate description by a classical six-constant plate model solely using the methods of Papers I and II. To complete a faithful translation it remains to define a distortion function using the methods of this paper.

Now the intermediate pixel coordinates (q_1, q_2) obtained from the PCi_ja matrix that we have constructed do not match the independent variables (X, Y) of the DSS polynomial given by Eqs. (29) and (30). We have

$$\begin{aligned}q_1 &= -A'_1R_x(p_1 - r_1) + A'_2R_y(p_2 - r_2),\\ q_2 &= -B'_2R_x(p_1 - r_1) + B'_1R_y(p_2 - r_2),\end{aligned}$$

but we want (X, Y) given by Eqs. (47) and (48). Solving these we obtain

$$\begin{aligned}X &= X_0 - B'_1q_1 + A'_2q_2,\\ Y &= Y_0 + B'_2q_1 - A'_1q_2.\end{aligned}$$

These may be handled as auxiliary variables of a sequent polynomial distortion function as defined by Eq. (11). Thus

```
CQDIS1 = 'Polynomial',
CQDIS2 = 'Polynomial'.
```

We can now begin to write the DQia parameters for each axis. For $\hat{N} = 2$, the axis coupling parameters (Sect. 2.5.1) are:

```
DQ1 = 'NAXES: 2',        DQ2 = 'NAXES: 2',
DQ1 = 'AXIS.1: 1',       DQ2 = 'AXIS.1: 1',
DQ1 = 'AXIS.2: 2',       DQ2 = 'AXIS.2: 2'.
```

Renormalization of the independent variables is not required so the relevant parameters may all be omitted.

The foregoing parameters are the ones with a fixed interpretation for all distortion functions, the remainder are specific to the polynomial distortion function of Sect. 3.1. These will be written in abbreviated form to save space. Firstly, the two auxiliary variables, X and Y, are defined as above:

DQ1·NAUX = 2,	DQ2·NAUX = 2,
DQ1·AUX.1.COEFF.0 = X_0,	DQ2·AUX.1.COEFF.0 = Y_0,
DQ1·AUX.1.COEFF.1 = $-B'_1$,	DQ2·AUX.1.COEFF.1 = B'_2,
DQ1·AUX.1.COEFF.2 = A'_2,	DQ2·AUX.1.COEFF.2 = $-A'_1$,
DQ1·AUX.2.COEFF.0 = Y_0,	DQ2·AUX.2.COEFF.0 = X_0,
DQ1·AUX.2.COEFF.1 = B'_2,	DQ2·AUX.2.COEFF.1 = $-B'_1$,
DQ1·AUX.2.COEFF.2 = $-A'_1$,	DQ2·AUX.2.COEFF.2 = A'_2.

All of the DQi·AUX.k.POWER.$\hat{\jmath}$ parameters have been omitted since they default to 1.0 as required here. Note that the auxiliary variables have been defined as $(\rho_1, \rho_2) = (X, Y)$ for $i = 1$, and $(\rho_1, \rho_2) = (Y, X)$ for $i = 2$ since this allows a more direct translation of Eqs. (29) and (30). Equation (29) may now be encoded in 13 terms, though not the same 13 as given:

```
DQ1.NTERMS = 13

DQ1.TERM.1.COEFF = A'_1
DQ1.TERM.1.AUX.1 = 1

DQ1.TERM.2.COEFF = A'_2
DQ1.TERM.2.AUX.2 = 1

DQ1.TERM.3.COEFF = A'_3

DQ1.TERM.4.COEFF = A'_4 + A'_7
DQ1.TERM.4.AUX.1 = 2

DQ1.TERM.5.COEFF = A'_5
DQ1.TERM.5.AUX.1 = 1
DQ1.TERM.5.AUX.2 = 1

DQ1.TERM.6.COEFF = A'_6 + A'_7
DQ1.TERM.6.AUX.2 = 2

DQ1.TERM.7.COEFF = A'_8 + A'_12
DQ1.TERM.7.AUX.1 = 3

DQ1.TERM.8.COEFF = A'_9
DQ1.TERM.8.AUX.1 = 2
DQ1.TERM.8.AUX.2 = 1

DQ1.TERM.9.COEFF = A'_10 + A'_12
DQ1.TERM.9.AUX.1 = 1
DQ1.TERM.9.AUX.2 = 2

DQ1.TERM.10.COEFF = A'_11
DQ1.TERM.10.AUX.2 = 3

DQ1.TERM.11.COEFF = A'_13
DQ1.TERM.11.AUX.1 = 5

DQ1.TERM.12.COEFF = 2A'_13
DQ1.TERM.12.AUX.1 = 3
DQ1.TERM.12.AUX.2 = 2

DQ1.TERM.13.COEFF = A'_13
DQ1.TERM.13.AUX.1 = 1
DQ1.TERM.13.AUX.2 = 4
```

Distortion parameters for Eq. (30) are identical except that DQ1 is replaced by DQ2 and A' by B'.

Parameters that have been omitted all default to zero as required. The A_m coefficients of Eq. (29) have all been rescaled by $-1/S$ here to take account of the scaling that will be applied to q'_1 by CDELT1 in computing $x_1 = \Xi/3600$. Likewise the B_m coefficients are scaled by $1/S$.

To complete the translation the polynomial distortion function should be evaluated at all points across the image to determine values of CQERR1, CQERR2, and DVERR. The header should also include

```
RADESYS = 'FK5',
EQUINOX = 2000.0.
```

5.3. Header construction example 2

Curved slit. Minkowski (1942)

5.4. Header construction example 3

Wide field Doppler correction.

5.5. Header construction example 4

Correction for oblateness in terrestrial coordinates.

Mars Global Surveyor.

6. Summary

Table 2 summarizes all new FITS header keywords defined in this paper, and Table 3 lists all field-specifiers defined for the record-valued keywords DP*ja* and DQ*ia*.

We have developed a ...

Acknowledgements. The authors wish to acknowledge valuable comments and suggestions from the following: Herve Aussel, Bob Hanson, Pat Wallace, ...

The Australia Telescope is funded by the Commonwealth of Australia for operation as a National Facility managed by CSIRO.

The National Optical Astronomy Observatory is a facility of the (U.S.) National Science Foundation operated under cooperative agreement by Associated Universities for Research in Astronomy, Inc.

The National Radio Astronomy Observatory is a facility of the (U.S.) National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

UCO/Lick Observatory is operated by the University of California.

References

Aussel, H., Cesarsky, C.J., Elbaz, D., & Starck, J. L. 1999, A&A, 342, 313

Calabretta, M. R., & Greisen, E. W. 2002, A&A., 395, 1077 (Paper II)

Cotton, W. D., Tody, D., & Pence, W. D. 1995, A&AS, 113, 159

Digitized Sky Survey (DSS). Located at `http://stdatu.stsci.edu/dss/`

Eichhorn, H. K. 1974, Astronomy of Star Positions, Fredrick Ungar Publishing Co., New York, U.S.A.

Greenfield, P. 1995, Instrument Science Report FOC-087. Available from `http://www.stsci.edu/ftp/instrument_news/FOC/foc_bib.html`

Greisen, E. W. 1983, Non-linear Coordinate Systems in AIPS, AIPS Memo No. 27, National Radio Astronomy Observatory, Charlottesville, Virginia. Revised January 1993. Located at `http://www.cv.nrao.edu/fits/wcs/` as `aips27.ps.Z`

Greisen, E. W., Calabretta M. R., Valdes, F. G., & Allen, S. L. 2004, Representations of spectral coordinates in FITS, A&A, in preparation (Paper III)

Greisen, E. W., & Calabretta, M. R. 2002, A&A, 395, 1061 (Paper I)

Hanisch, R. J., Farris, A., Greisen, E. W., Pence, W. D., Schlesinger, B. M., Teuben, P. J., Thompson, R. W., & Warnock III, A. 2001, A&A, 376, 359

Holtzman, J., et al. 1995, PASP., 107, 156

König, A. 1962, Astronomy with Astrographs, Chap. 20 of Astronomical Techniques, Vol. II of Stars and Stellar Systems, University of Chicago Press, Chicago and London

Lasker, B. M., Sturch, C. R., McLean, B. J., Russell, J. L., Jenkner, H., & Shara, M. 1990, AJ., 99, 2019

Minkowski, R. 1942, ApJ., 96, 306

Murray, C. A. 1983, Vectorial Astrometry, Adam Hilger Ltd., Bristol

Table 2. Summary of new FITS coordinate keywords introduced in this paper. None of these keywords have default values.

Keyword	Type	Sect.	Use	Status	Comments
CPDIS*ja*	string	2.4.1	distortion code	new	Prior distortion function type.
CQDIS*ia*	string	2.4.1	distortion code	new	Sequent distortion function type.
DP*ja*	record	2.4.2	distortion parameter	new	Parameter for a prior distortion function, for use in an image header.
DQ*ia*	record	2.4.2	distortion parameter	new	Parameter for a sequent distortion function, for use in an image header.
*j*DP*na*	record	2.4.2	distortion parameter	new	Form of DP*ja* for use in binary table image arrays.
*i*DQ*na*	record	2.4.2	distortion parameter	new	Form of DQ*ia* for use in binary table image arrays.
TDP*na*	record	2.4.2	distortion parameter	new	Form of DP*ja* for use for use with pixel lists.
TDQ*na*	record	2.4.2	distortion parameter	new	Form of DQ*ia* for use for use with pixel lists.
CPERR*ja*	floating	2.5.1	error measure	new	Maximum value of prior distortion correction for axis j.
CQERR*ia*	floating	2.5.1	error measure	new	Maximum value of sequent distortion correction for axis i.
DVERR*a*	floating	2.5.1	error measure	new	Maximum value of the combination of all distortion corrections.

Table 3. Summary of field-specifiers defined for DP*ja* parameter values. Those for DQ*ia* are essentially identical though formally the indexing variable $\hat{\jmath}$ is replaced by $\hat{\imath}$, and $\hat{p}_{\hat{\jmath}}$ by $\hat{q}_{\hat{\imath}}$.

Field-specifier	Default	Sect.	Comments
NAXES	0	2.5.1	Number of independent variables, $\hat{N}$, in a distortion function.
AXIS.$\hat{\jmath}$	$\hat{\jmath}$	2.5.1	Axis number, $j_{\hat{\jmath}}$ ($1 \leq j_{\hat{\jmath}} \leq N$), of the $\hat{\jmath}$th ($1 \leq \hat{\jmath} \leq \hat{N}$) independent variable, $\hat{p}_{\hat{\jmath}}$, in a distortion function.
OFFSET.$\hat{\jmath}$	0.0	2.5.1	Offset to be subtracted from $\hat{p}_{\hat{\jmath}}$ before use.
SCALE.$\hat{\jmath}$	1.0	2.5.1	Scale to multiply $\hat{p}_{\hat{\jmath}}$ by before use.
NAUX	0	3.1	Number of auxiliary variables, K, used by a polynomial distortion function.
AUX.k.COEFF.$\hat{\jmath}$	0.0	3.1	Coefficient of $\hat{p}_{\hat{\jmath}}$ used in computing the kth auxiliary variable for a polynomial distortion function.
AUX.k.POWER.$\hat{\jmath}$	1.0	3.1	Power of $\hat{p}_{\hat{\jmath}}$ used in computing the kth auxiliary variable for a polynomial distortion function.
NTERMS	0	3.1	Number of terms in a polynomial distortion function.
TERM.m.COEFF	1.0	3.1	Coefficient of the mth term of a polynomial distortion function.
TERM.m.VAR.$\hat{\jmath}$	0.0	3.1	Power of the $\hat{\jmath}$th independent variable, $\hat{p}_{\hat{\jmath}}$, in the mth term of a polynomial distortion function.
TERM.m.AUX.k	0.0	3.1	Power of the kth auxiliary variable, ρ_k, in the mth term of a polynomial distortion function.
EXTVER	1	3.4	Extension version number (EXTVER keyword) of the image extension with name EXTNAME = 'WCSDVARR' containing a table lookup distortion function.

Murray, C. A. 1984, in Astrometry with Schmidt-Type Telescopes, Vol. 110, Astrophysics and Space Science Library, D. Reidel Publishing Company, Dordrecht
Ponz, J. D., Thompson, R. W., & Muñoz, J. R. 1994, A&AS, 105, 53
Russell, J. L., Lasker, B. M., McLean, B. J., Sturch, C. R., & Jenkner, H. 1990, AJ., 99, 2059
Smart, W. M. 1965, Spherical Astronomy (5th ed.), Cambridge University Press
van de Kamp, P. 1967, Principles of Astrometry, W. H. Freeman and Co., San Francisco and London
Wells, D. C., Greisen, E. W., & Harten, R. H. 1981, A&AS, 44, 363