Table 21: WCS and celestial coordinates notation.

Variable(s)	Meaning	Related FITS keywords
i	Index variable for world coordinates	
j	Index variable for pixel coordinates	
a	Alternative WCS version code	
p_j	Pixel coordinates	
r_j	Reference pixel coordinates	`CRPIX`*ja*
m_{ij}	Linear transformation matrix	`CD`*i*`_`*ja* or `PC`*i*`_`*ja*
s_i	Coordinate scales	`CDELT`*ia*
(x, y)	Projection plane coordinates	
(ϕ, θ)	Native longitude and latitude	
(α, δ)	Celestial longitude and latitude	
(ϕ_0, θ_0)	Native longitude and latitude of the fiducial point	`PV`*i*`_1`*a*† , `PV`*i*`_2`*a*†
(α_0, δ_0)	Celestial longitude and latitude of the fiducial point	`CRVAL`*ia*
(α_p, δ_p)	Celestial longitude and latitude of the native pole	
(ϕ_p, θ_p)	Native longitude and latitude of the celestial pole	`LONPOLE`*a* (=`PV`*i*`_3`*a*†), `LATPOLE`*a* (=`PV`*i*`_4`*a*†)

Notes. † Associated with *longitude* axis *i*.

in the `CRPIX`*i* keywords, and the world coordinates at the reference point are encoded in the `CRVAL`*i* keywords. For additional details, see Greisen & Calabretta (2002).

The third step of the process, computing the final world coordinates, depends on the type of coordinate system, which is indicated with the value of the `CTYPE`*i* keyword. For some simple, linear cases an appropriate choice of normalization for the scale factors allows the world coordinates to be taken directly (or by applying a constant offset) from the x_i (e.g., some spectra). In other cases it is more complicated, and may require the application of some non-linear algorithm (e.g., a projection, as for celestial coordinates), which may require the specification of additional parameters. Where necessary, numeric parameter values for non-linear algorithms *must* be specified via `PV`*i*`_`*m* keywords and character-valued parameters will be specified via `PS`*i*`_`*m* keywords, where *m* is the parameter number.

The application of these formalisms to coordinate systems of interest is discussed in the following sub-sections: Sect. 8.2 describes general WCS representations (see Greisen & Calabretta 2002), Sect. 8.3 describes celestial coordinate systems (see Calabretta & Greisen 2002)), Sect. 8.4 describes spectral coordinate systems (see Greisen et al. 2006), and Sect. 9 describes the representation of time coordinates (see Rots et al. 2015).

8.2. World coordinate system representations

A variety of keywords have been reserved for computing the coordinate values that are to be associated with any pixel location within an array. The full set is given in Table 22; those in most common usage are defined in detail below for convenience. Coordinate system specifications may appear in HDUs that contain simple images in the primary array or in an image extension. Images may also be stored in a multi-dimensional vector cell of a binary table, or as a tabulated list of pixel locations (and optionally, the pixel value) in a table. In these last two types of image representations, the WCS keywords have a different naming convention which reflects the needs of the tabular data structure and the 8-character limit for keyword lengths, but otherwise follow exactly the same rules for type, usage, and default values. See reference Calabretta & Greisen (2002) for example usage of these keywords. All forms of these reserved keywords *must* be used only as specified in this Standard.

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 `1CRPn` 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. `TTYPEm = '1CRPn'`) 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 "Green Bank Convention"[9] for *FITS* keywords. This usage is illustrated in the example header shown in Table 9 of Calabretta & Greisen (2002),

The keywords given below constitute a complete set of fundamental attributes for a WCS description. Although their inclusion in an HDU is optional, *FITS* writers *should* include a complete set of keywords when describing a WCS. In the event that some keywords are missing, default values *must* be assumed, as specified below.

`WCSAXES` – [integer; default: `NAXIS`, or larger of WCS indexes *i* or *j*]. Number of axes in the WCS description. This keyword, if present, *must* precede all WCS keywords except `NAXIS` in the HDU. The value of `WCSAXES` *may* exceed the number of pixel axes for the HDU.

`CTYPE`*i* – [string; indexed; default: `'␣'` (i.e. a linear, undefined axis)]. Type for the intermediate coordinate axis *i*. Any coordinate type that is not covered by this standard or an officially recognized *FITS* convention *shall* be taken to be linear. All non-linear coordinate system names *must* be expressed in '4–3' form: the first four characters specify the coordinate type, the fifth character is a hyphen ('-'), and the remaining three characters specify an algorithm code for computing the world coordinate value. Coordinate types with names of less than four characters are padded on the right with hyphens, and algorithm codes with less than three charac-

[9] Named after a meeting held in Green Bank, West Virginia, USA in 1989 to develop standards for the interchange of single dish radio astronomy data.

5. The table keywords described in Sect. 7.2.2 and 7.3.2 were originally introduced as a *FITS* convention since 1993, and registered in 2006. The text of the original convention is reported at `http://fits.gsfc.nasa.gov/registry/colminmax.html`. The differences with this standard concern:
   – The exclusion of undefined or IEEE special values when computing maximum and minimum is now *mandatory* while it was *optional*.
   – The original text included the possibility of using the fact `TDMINn` were greater than `TDMAXn` (or `TLMINn` greater than `TLMAXn`) as an indication the values were undefined. This clause has been removed
   – The original text contained usage examples and additional minor explanatory details.
6. The checksum keywords described in Sect. 4.4.2.7 were originally introduced as a *FITS* convention since 1994, and registered in 2007. The text of the original convention is reported at `http://fits.gsfc.nasa.gov/registry/checksum.html`. The differences with this standard concern:
   – The omission of some additional implementation guidelines.
   – The omission of a discussion on alternate algorithms and relevant additional references.
7. The conventions for compressed data described in Sect. 10. were originally introduced as a couple of *FITS* conventions registered in 2007 and 2013. The text of the original conventions is reported at `http://fits.gsfc.nasa.gov/registry/tilecompression.html` for compressed images and at `http://fits.gsfc.nasa.gov/registry/tiletablecompression.html` for compressed binary tables. The differences with this standard concern:
   – In Sect. 10.3.3 the original text for `FZALG`*n* mentioned the possibility that, ‘If the column cannot be compressed with the requested algorithm (e.g., if it has an inappropriate data type), then a default compression algorithm will be used instead.’ But there is no default algorithm. This is irrelevant for the Standard.
   – In Sect. 10.4 the alias `'RICE_ONE'` is *not* adopted in the Standard as a synonym for `'RICE_1'`.
   – In Sect. 10.4.3 a sentence was left out about requiring additional instructions in PLIO to make it work for more then 2^{12} bits, since we aren't allowing this possibility in the Standard.
   – In Sect. 10.4.4 the reference to a ‘smoothing flag’ was dropped.
   – Also in Sect. 10.4.4 the *scale factor* is now floating point, while it was originally integer.
   – In Table 36 (and Sect. 10.3.5) the `NOCOMPRESS` algorithm is explicitly mentioned.
8. The Green Bank convention mentioned in Sect. 8.2 was in use since 1989, and registered in 2010. The text of the registered convention is reported at `http://fits.gsfc.nasa.gov/registry/greenbank/greenbank.pdf`. The original text contains some additional details about the history of the convention.

Appendix I: Random Number Generator

This Appendix is not part of the FITS *standard, but is included for informational purposes.*

The portable random number generator algorithm below is from Park & Miller (1988). This algorithm repeatedly evaluates the function

$$seed = (a * seed) \bmod m$$

where the values of a and m are shown below, but it is implemented in a way to avoid integer overflow problems.

```
int random_generator(void) {

/* initialize an array of random numbers */

   int ii;
   double a = 16807.0;
   double m = 2147483647.0;
   double temp, seed;
   float rand_value[10000];

   /* initialize the random numbers */
   seed = 1;
   for (ii = 0; ii < N_RANDOM; ii++) {
      temp = a * seed;
      seed = temp -m * ((int) (temp / m) );
      /* divide by m for value between 0 and 1 */
      rand_value[ii] = seed / m;
   }
}
```

If implemented correctly, the 10 000th value of seed will equal 1 043 618 065.

Appendix J: `CHECKSUM` Implementation Guidelines

This Appendix is not part of the FITS *standard, but is included for informational purposes.*

J.1. Recommended `CHECKSUM` *Keyword Implementation*

The *recommended* `CHECKSUM` keyword algorithm described here generates a 16-character ASCII string that forces the 32-bit 1's complement checksum accumulated over the entire *FITS* HDU to equal negative 0 (all 32 bits equal to 1). In addition, this string will only contain alphanumeric characters within the ranges 0–9, A–Z, and a–z to promote human readability and transcription. If the present algorithm is used, the `CHECKSUM` keyword value *must* be expressed in fixed format, with the starting single quote character in column 11 and the ending single quote character in column 28 of the *FITS* keyword record, because the relative placement of the value string within the keyword record affects the computed HDU checksum. The steps in the algorithm are as follows:

1. Write the `CHECKSUM` keyword into the HDU header with an initial value consisting of 16 ASCII zeros (`'0000000000000000'`) where the first single quote character is in column 11 of the *FITS* keyword record. This specific initialization string is required by the encoding algorithm described in Sect. J.2 The final comment field of the keyword, if any, must also be written at this time. It is recommended