HMI & WCS COORDINATES, PROJECTIONS AND ARRAYS FOR DUMMIES

WRITTEN BY: JOHN BECK
beck@Sun.Stanford.EDU

Many decisions regarding solar data processing and analysis start with selecting the proper image coordinate system and projection. When discussing maps, projections, geometry, and coordinate systems, it is important to use consistent terminology. This document is intended to explain vocabulary and some of the properties of typical coordinate systems and projections to make discussing HMI and AIA data easier. Admittedly, the author is a dummy and often doesn't make 'a lick of sense' so suggestions, clarifications and comments are welcome.

There are typically three steps in making a pixel image representing the surface of the Sun: locating the position on the Sun, projecting the location onto a plane, and sampling the projected plane into a pixel array[1]. Before locating a position on the surface of the Sun an appropriate coordinate system must be chosen - the coordinate systems used by HMI are spherical and centered either on the sun or on the observer[2]. Projecting a spherical surface onto a plane is requires a specific map projection and these are like iPhone apps: there are thousands of them but only a few are useful. The map projection is then sampled into pixels in a data to be processed by software or saved to a file.

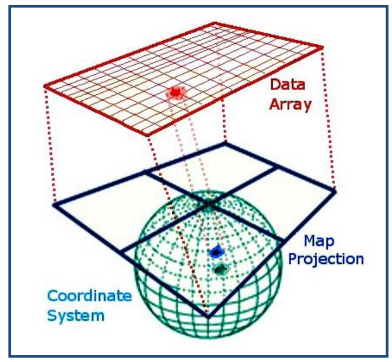


Figure 1 – a graphic depiction of converting between the solar surface using a heliographic coordinate system, projection onto a plane using an azimuthal map and sampling into pixels in a data array.

When thinking of these steps it is useful to think in terms of converting between the steps (solar surface, map projection, pixel array), rather than considering each step independently. For example, pixel size, rather than being a property solely of the pixel array, indicates how the pixel array samples the map projection plane.

The information describing how to convert from coordinates on the solar surface to map projection to pixel array can be expressed in many ways. In fact, the amount of information (meta-data) required to describe these conversions depends on the particulars. Predictably, images from different observatories use different meta-data to describing coordinates and projections and sometimes give different names to the same meta-data. The World Coordinate Systems (WCS) standard was proposed to solve this problem – it provides a set of rules for meta-data labels (i.e. FITS keywords) as well as acceptable values. This makes communication between scientists and observatories easier.

The rest of this document will explain the World Coordinate Systems, solar coordinate systems used by HMI (and AIA) and map projections used by HMI. In the appendix is a table summarizing the keywords used to describe the HMI supported map projections and coordinate systems.

---

[1] Technically, a pixel array could sample the solar surface without the intermediate step of projecting it into a plane, but it would have to be a 3-D array of pixels and have many 'empty' elements which are not on the solar surface.

[2] Although, we often treat the 'plane of the sky' as a literal plane, it is really a small region of the celestial sphere where the small angle approximation allows us to treat it as a plane. This explains why we use arc-seconds as units of measurement on a plane. Occasionally, for extended objects the planar approximation is inadequate – so be careful.

1. THE WORLD COORDINATE SYSTEMS

(This section draws heavily from Greisen & Calabretta, 2002, A&A 395, 1007)

The WCS standard is a general convention for FITS keywords to store the meta-data needed to go from pixel array coordinates to physical (world) coordinates, as shown in Figure 2. These rules are designed to be clear and general, since the possible coordinate systems include not only position, but other physical quantities such as time, frequency, wavelength, or Stokes Parameters (the only limit is our imagination). We can view WCS as a 'Swiss Army Knife' that can handle a lot of different situations but has many tools we won't use.

This paper uses a modified version of Greisen & Calabretta's notation: for WCS keywords, a numerical suffix designates a single axis and a suffix of the letter 'i' denotes the set of that keyword for all axes. For example, the keyword for the coordinate units, CUNIT, is written CUNIT1 for the first axis, CUNIT2 for the second and CUNITi for all axes. When a single axis can have multiple instances of a keyword, the suffix 'm' is used. (For simplicity, this paper does not distinguish between the suffix 'j' and 'i' – used by G&C to denote pixel coordinate axes and world coordinate axes, respectively).[3]

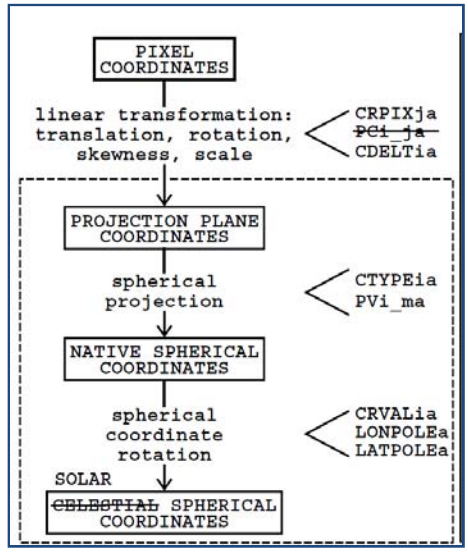


Figure 2 - Conversion of pixels coordinates to solar spherical coordinates, adapted from Calabretta & Greisen 2002.

The HMI project adopts much of the WCS standard, with the notable exception of specifying rotations: we use the (deprecated) CROTA keyword, rather than the more general PCi_m keywords (which refer to elements of a rotation matrix and are more general than the CROTA). This sacrifices some flexibility but matches the traditional conventions of using B_0 and *p-angle*. CROTA specifies *p-angle* (well, the negative *p-angle*). The HMI software does not support PCi_m keywords and their use is strongly discouraged, however, other unsupported WCS keywords are not discouraged (e.g. WCSAXES).

This document uses CTYPEi 'codes' as abbreviations for coordinate systems and projections, these are in the"4-3" or "COOR-PRJ" format. The first four letters specify the coordinate system used for that axis (COOR). Often the first two letters will identify a class of coordinate axes (e.g. Heliographic, HG, or Helioprojective, HP) and the second two letters will identify the specific coordinate axis (e.g. Latitude, LT, or Longitude, LN.[4] After the dash, the three letters represent the map projection (PRJ) such as Plate Carrée, CAR. The table of HMI supported coordinate systems and projections is found in the appendix.

[3]This simplification was made because many details of WCS are of interest only to programmers writing map projection code – where following the rules of how to store the meta-data describing the image geometry is crucial. The gist of WCS should suffice those who wish to call functions which handle the map projections and coordinate transforms. The appendix contains a table specifying the required parameters for each supported coordinate system and projection – along with some suggested parameters when there is a choice.

[4] Coordinate axes often appear in matched pairs (e.g. Heliographic Longitude, HGLN, usually accompanies Heliographic Latitude, HGLT), however there are exceptions (for a solar spectrum image, the axes could be HGLT and WAVE). This also means that the same axis can be described using two different coordinates (for example Carrington Longitude and Carrington Time for a synoptic chart)

Table 1. FITS Keywords Defined in WCS Used in HMI DRMS

FITS KEYWORD	TYPE	DESCRIPTION
CRVALi	**float**	**Coordinate value at reference point (one per axis).** For example, for Heliographic coordinates, this would be the heliographic latitude or longitude of the reference point on the map. Depending on the projection, reference point may be fixed (e.g. at disk center, or at the tangent-point of the map).
CRPIXi	**float**	**Array location of the reference point in pixels (one per axis)** This gives the alignment of the data array with the projected map. It is not necessarily within the range of 1-N (where N is the length of the array along that axis) if the data array is a sub-region of the map.
CDELTi	**float**	**Coordinate increment at reference point (one per axis)** This gives the 'sampling' of the map to the array. Although some maps specify an aspect ratio between the coordinate axes (e.g. Mercator), the CDELTn keywords may have any ratios. Note: this can lead to confusion.
CTYPEi	**string**	**Axis type: 8 characters with a “4-3”format.** The first 4 characters indicate the coordinate type; the last 3 characters identify the map projection. (one per axis)
CUNITi	**string**	**Units for stated coordinate axes (one per axis)** These are fixed according to the CTYPE of the axis. For HPLN/HPLT always use 'arc-sec', for HGLN/HGLT always use 'deg'
CROTA	**Float**	**Rotation from stated coordinate type** note: this is deprecated, but we are using it instead of PC_ij & CD_ij because it is similar to the traditional p-angle representation, and is equal to minus p-angle) It is always specified in degrees according to the FITS standard. (note: for aesthetic reasons we use CROTA2 to specify this angle, never CROTA1)
PVi_m	**Float**	**Special parameters required for some coordinate types (value)** The definition of PVi_m keywords is dependent on the coordinate & projection
PSi_m	**String**	**special parameters required for some coordinate types (name string)** HMI code does not currently take advantage of this, however it will be supported.
WCSNAME	**String**	**Name of coordinate-projection used** This is the official WCS name for the coordinate/projection used to represent the data for this file (or record).

2 SOLAR COORDINATE SYSTEMS

(This section draws heavily from Thompson, 2005, A&A Manuscript no 4262thom)

The WCS was developed largely by night-time astronomers, whose main positional coordinate systems are Equatorial Coordinates or Galactic Coordinates (which are appropriate for locating objects on the Celestial Sphere or in the Milky Way). As a result, much of the WCS support is aimed at that audience (e.g. WCSLIB, a library of functions handling WCS keywords and performing coordinate and projection transformations does not include most of the Sun based coordinate systems we will discuss). Maybe we're partly to blame: As said in Thompson (2005): "Although there is widespread agreement on the coordinate systems to be used for interplanetary space (Russell 1971; Hapgood 1992; Franz & Harper 2002) no formal structure exists for solar image coordinates, except for the well-established heliographic coordinate systems." Three classes of solar coordinate systems were proposed by Thompson: Heliographic, Heliocentric and Helioprojective. Although, all three have their place in representing HMI and AIA data, Heliographic and Helioprojective coordinate systems will be the most commonly used.

2.1 Heliographic Coordinates

There are two heliographic coordinate systems used in HMI data: Stonyhurst (HG) and Carrington (CR). Both use Latitude (LT) and Longitude (LN) coordinates to specify a location on the surface of the Sun. (By combining the two letter codes, we can specify a heliographic coordinate axis.) The Stonyhurst latitude, HGLT, is equal to the Carrington latitude, CRLT, however the longitude measurements differ. The central meridian, or longitude passing through disk center as seen from the Earth (but not necessarily from SDO), is always zero for Stonyhurst but has a time-varying value for Carrington coordinates: $LN_C(t)$. The relation between the Stonyhurst longitude, HGLN, and Carrington longitude, CRLN, is:

$$CRLN = HGLN + LN_C(T)$$

The Carrington coordinate system is fixed to a sphere rotating with a constant period of 25.38 days in the sidereal frame and the Stonyhurst coordinate system is fixed in the synodic frame. Consequently, the Stonyhurst coordinate system does not rotate at a constant rate in the sidereal frame – it matches the Earth's orbital motion and varies according to Kepler's second law. [Note: further complications arise from the fact that the central meridian as seen by the SDO spacecraft will not always match that seen by the Earth - this must be considered when mapping into Heliographic coordinates]. It is recommended to use Carrington coordinates for analyzing precise spatial positions on the Sun's surface over long time scales since it has a constant sidereal rotation.

2.2 Heliocentric Coordinates

The two heliocentric coordinate systems, Cartesian and Radial, locate positions on the visible solar disk and are defined relative to the position of the observer. One axis, *z,* is parallel to the line of sight, the other two axes (x & y in the Cartesian case; ρ and ψ in the Radial case) are in the plane perpendicular to the *z-axis* - distances along the axes are measured in units of length (except for, ψ, which is an angle). Although the alignment of the axes is determined by the observer's position the separation between points is measured in physical units. These coordinate systems are of limited use in the HMI project since "No solar observation from a single perspective can truly be said to be in heliocentric coordinates [...]" (Thompson, 2005). For this reason, the reader is left to her or his own devices for this class of coordinates.

2.3 Helioprojective Coordinates

The two Helioprojective coordinate systems discussed in Thompson (2005) are similar to those described in the Heliocentric class, except distances perpendicular to the line of sight are measured in angles. Helioprojective coordinates are projected onto the 'plane of the sky' or the Celestial Sphere, which are the same, for small regions. The coordinate system is centered on the observer and the separation between two points is the angular separation from the observer's perspective. The Helioprojective-Cartesian coordinate system specifies HPLT, the northward angle from the center of the disk, and HPLN, the westward angle from the center of the disk, which are comparable to Declination and Right Ascension. Since all HP measurements are small angles, the only projection supported by HMI is TAN, and is labeled as 'AERIAL' under the old MDI naming conventions.

2.4 Summary of Solar Coordinate Systems

The two classes of coordinate systems adopted by HMI are Heliographic and Helioprojective, which use spherical coordinate systems centered on the Sun and on the observer, respectively. The Helioprojective -Cartesian coordinate system is natural for the camera, since it 'sees' the projection of the Sun on the sky, and for small angles, when the camera frame is aligned, HPLT is a linear function of CCD array rows and HPLN is a linear function of CCD array columns. The heliographic coordinate systems, Stonyhurst and Carrington are nearly the same, with the difference of the value of the central meridian. For this reason, in discussions of map projections we will refer to Heliographic coordinates and use HG to represent both Stonyhurst and Carrington coordinate systems.

3. Map Projections

One way to think of map projections is "Ways to flatten the surface of a sphere[5]" Since both Heliographic and Helioprojective coordinate systems are spherical coordinate systems, in principle, any map projection which can be applied to one can be applied to the other. In practice, the Helioprojective-Cartesian coordinate system (HPLT, HPLN) is overwhelmingly used to represent data in the same geometry as CCD images, which is the Gnomonic projection (TAN). The only Helioprojective projection which will be discussed is HPLT-TAN, HPLN-TAN. (Since the values on the HPLT and HPLN axes are small angles for the solar disk, the Perspective Azimuthal (AZP) and Orthographic projections (SIN) approximate the TAN projection. However, if either of these projections is desired, it is **strongly** recommended to use Heliographic coordinates to clarify that the image is a projected map. If you choose to do otherwise, you are on your own.)

Throughout this section the WCS codes are used as shorthand for Heliographic coordinates (HGLT, HGLN) and Helioprojective coordinates (HTLT, HPLN) – since, both Stonyhurst and Carrington Heliographic coordinates can be used equally well with these projections, the codes HGLT, HGLN will be used for both coordinate systems within the Heliographic class.

The equations to map from the surface of a sphere to a plane can be found at the Wolfram website (eg. http://mathworld.wolfram.com/MapProjection.html)

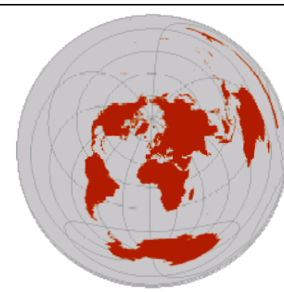
---

[5] Unless you are using data cubes or indulging in one of our famous lunchtime discussions- if you are doing anything fancy like that, you're on your own!

3.1 Azimuthal Projections

This class of projections can be visualized as projecting the surface of a sphere onto a plane by extending a straight line from a point, through the surface of the sphere and onto the plane. (need a figure). Most of these projections are similar to the 'camera view'. The most commonly used of this class are listed in the table below with brief descriptions

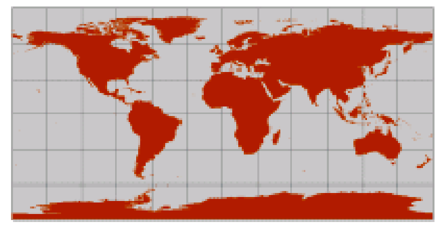
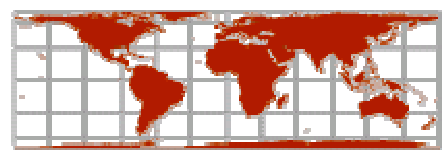
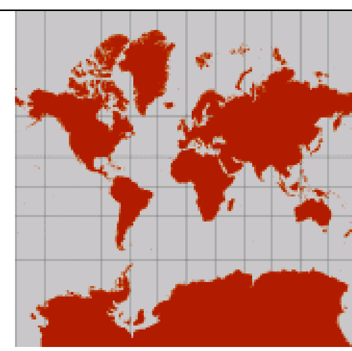
Table 2. Azimuthal Projections

Name	WCS code	Comments	Example
Gnomonic	TAN	This projection is the CCD 'camera view' when used with HPLT,HPLN, often referred to as Aerial since it is the perspective of an observer hovering over a particular location on the surface and looking down upon it.	
Orthographic	SIN	This projection approximates the Aerial as seen from an observer located an infinite distance away. The key difference being how far out to the limb can be seen so near disk center it is almost the same as Gnomonic. Typically used with HGLT, HGLN	
Zenithal Perspective	AZP	This projection approximates the 'camera view' when used with HGLT, HGLN	
Zenithal Equidistant	ARC	Also known as Postel's Projection, does not approximate 'camera view' should only be used with HGLT, HGLN	

3.2 Cylindrical Projections

The cylindrical projections are produced by projecting the surface of the sphere onto a cylinder, since CCD's are not cylinders, these projections are only applicable to HGLT, HGLN coordinates. These projections are suitable for making synoptic charts. The primary difference between them is in the Latitude axis.

Table 3. Cylindrical Projections

Name	**WCS code**	**Comments**	**Examples**
Plate Carrée	CAR	This is an 'equidistant' projection, the axes are proportional to LT and LN	
Cylindrical Equal Area	CEA	This projection have axes which are proportional to sin(LT) and LN	
Mercator's projection	MER	Mapmakers in Greenland love this one! If a solar wants to use this project s/he must have a good reason and already understands this projection.	

3.3 A note about pixels, coordinates and projections.

All map projections discussed in this document are continuous functions which map from the surface of a sphere onto a plane (which may be rolled into a cylinder during the mapping process ;). Some projections have set aspect rations at specific locations (e.g. a Mercator map's LT and LN axes have a 1:1 ratio at the equator). However map projections do not 'know about' pixels, pixel size or pixel dimensions. Conceptually pixelization of the map is a separate process, done after re-projecting (in practice, the conversion from 2-D image array, to 2-D plane is often lumped in along with other projection-related calculations). The WCS keywords, CDELTi specify pixel size (and dimension) and are not part of the projection specification, **so the selection of map projection does not restrict the values of CDELT** (admittedly, some values of CDELT will produce useless maps, but that is another matter). This means, for example, that it is possible to have a square array of pixels which do not produce a square region of a Mercator map.

Table 4. Coordinate-Projection Keywords

	CCD/ Camera	Gnomonic	Orthographic	Zenithal Perspective	Zenithal Equidistant	Lambert	Plate Carrée	Cylindrical Equal Area	Mercator's projection	Cassini	Sanson-Flamsteed
MDI Projname	AERIAL	GNOMONIC	ORTHOGRAPHIC		POSTELS	LAMBERT	RECTANGULAR	CYLEQA	MERCATOR	CASSINI	SINEQA
WCSNAME	TAN	TAN	SIN	AZP	ARC	ZEA	CAR	CEA	MER	CAS**	SFL
CTYPE1	HPLN-TAN	HGLN-TAN	HGLN-SIN	HGLN-AZP	HGLN-ARC	HGLN-ZEA	HGLN-CAR	HGLN-CEA	HGLN-MER	HGLN-CAS	HGLN-SFL
CTYPE2	HPLT-TAN	HGLT-TAN	HGLT-SIN	HGLT-AZP	HGLT-ARC	HGLT-ZEA	HGLT-CAR	HGLT-CEA	HGLT-MER	HGLT-CAS	HGLT-SFL
CUNIT1	arcsec	deg	deg	deg	deg	deg	deg	sin(deg)	???		
CUNIT2	arcsec	deg	deg	deg	deg	deg	deg	deg	???		
CRPIX1	array col of disk center	array col of reference longitude	array col of reference longitude	array col of reference longitude	array col of reference longitude	array col of reference longitude	array col of reference longitude	array col of reference longitude	array col of reference longitude	array col of reference longitude	array col of reference longitude
CRPIX2	array row of disk center	array row of reference latitude	array row of reference latitude	array row of reference latitude	array row of reference latitude	array row of reference latitude	array row of reference latitude	array row of reference latitude	array row of reference latitude	array row of reference latitude	array row of reference latitude
CRVAL1	0.0	reference longitude	reference longitude	reference longitude	reference longitude	reference longitude	reference longitude	reference longitude	reference longitude	reference longitude	reference longitude
CRVAL2	0.0	reference latitude	reference latitude	reference latitude	reference latitude	reference latitude	reference latitude	reference latitude	reference latitude	reference latitude	reference latitude
CRDELT1											
CRDELT2											
CROTA2	-SOLAR_P	-SOLAR_P	-SOLAR_P	-SOLAR_P	-SOLAR_P		-SOLAR_P	-SOLAR_P	-SOLAR_P		
PV2_1	$-D_{sun}/R_{sun}$			$-D_{sun}/R_{sun}$				$\lambda = \cos^2(\theta_{con} f)$			
???											
Class	Azimuthal	Azimuthal	Azimuthal	Azimuthal	Azimuthal	Azimuthal	Cylindrical	Cylindrical	Cylindrical	Cylindrical	Cylindrical

Note: CRVALn, the reference longitude and latitude are special points for particular projections (e.g. for the Helioprojective they are always disk center; for azimuthal maps they specify the tangent point of the map) and the corresponding CRPIXn is free to be anywhere within, or even outside of, the pixel array. Other notable notes (which belong somewhere in the text)

CRPIXn/CRVALn – it may appear that as long as the reference pixel & world coordinate values match, any set will give the same results. This is not true: these keywords specify the 'ground-zero' (or point of secancy) for the map projection where applicable. For example, in an ARC (Postel's) projection the distances measured between the map center and other points are correct, but not necessarily distances between two arbitrary points on the map. If CRPIXn are not centered in the image array the map center will be off-center (in fact, CRPIXn do not even have to be outside the image array). This is permissible but not advised because of its potential for confusion.