Memo: UVBeam FITS Format

Bryna Hazelton, and the pyuvdata team

Jan 27, 2018

1 Introduction

This memo introduces a new FITS file format for storing beam models associated with the UVBeam object in pyuvdata[1], a python package that provides an interface to interferometric data. Here, we describe the required and optional elements and the structure of a UVBeam FITS (hereafter *beamfits*) file.

The contents of *beamfits* files are explicitly mapped to attributes of the UVBeam object in pyuvdata. For more details on these parameters, please see http://pyuvdata.readthedocs.io/en/latest/uvbeam_parameters.html. For examples of how to interact with this format using pyuvdata, please see the pyuvdata tutorial: http://pyuvdata.readthedocs.io/en/latest/tutorial.html.

2 Overview

There are two main types of *beamfits* files, depending on whether the beams are pixelized in HEALPix[2] or a regular pixel grid. There are also two subtypes of files for power and E-field beams for each pixelization scheme. So there are four primary flavors of *beamfits* files which have small differences in format. There are also several optional metadata components that may or may not be present in any given file. These variations are described in detail in the following sections.

3 Primary HDU

The primary HDU of *beamfits* files is an Image HDU containing the beam model.

---

[1] https://github.com/HERA-Team/pyuvdata
[2] https://healpix.jpl.nasa.gov/

1

3.1 Primary Header

The following are required keywords in the primary header of a *beamfits* file. For more detailed explanations of what these keywords mean, see the descriptions on pyuvdata's ReadTheDocs uvbeam_parameters page. The UVBeam parameter corresponding to each keyword is noted in parentheses. As with all FITS files, HISTORY and COMMENT cards are allowed.

- **BTYPE**: *string* should be set to either 'power' or 'efield' to designate a power or E-field beam type. (beam_type)

- **NORMSTD**: *string* should be set to one of 'physical', 'peak' or 'solid_angle'. (data_normalization)

- **COORDSYS**: *string* should be set to one of 'az_za' (for an azimuth, zenith angle regular grid), 'orthoslant_zenith' (for an orthoslant projection at zenith) or 'healpix'. (pixel_coordinate_system)

- **TELESCOP**: *string* Telescope name. (telescope_name)

- **FEED**: *string* Feed name (used to distinguish between different physical feed types on the same telescope). (feed_name)

- **FEEDVER**: *string* Feed version (used to distinguish between different versions of the same feed on the same telescope). (feed_version)

- **MODEL**: *string* Model name (used to distinguish between different models of the same hardware). (model_name)

- **MODELVER**: *string* Model version (used to distinguish between different versions of the same models of the same hardware). (model_version)

- **FEEDLIST**: *string list* Only for E-field beams. The list of feeds orientations represented in the data. (feed_array)

- **NSIDE**: *integer* Only for HEALPix beams. The NSIDE of the HEALPix map. (nside)

- **ORDERING**: *string* Only for HEALPix beams. The ordering parameter of the HEALPix map, allowed values are 'nested' and 'ring'. (ordering)

3.2 Beam Data

The axes of the data array in the primary HDU depends on the pixelization scheme.

3.2.1 HEALPix beam axes

For HEALPix beams, the data axes (values of the CTYPEm keywords in order) are:

1. **PIX_IND**: index into the HEALPix pixel array (in the HPX_INDS extension) This is needed because partial sky maps are supported so pixels may be missing (full sky maps are not handled differently, they have the same required elements as partial maps). The corresponding CRVALm, CRPIXm, and CDELTm values should all be 1. (pixel_array)

2. **FREQ**: Frequency axis, the corresponding CRVALm shall have the reference frequency for the data set, both CRVALm and CDELTm are given in units given by the CUNITm keyword, which is usually Hz. (freq_array)

3. **STOKES or FEEDIND (Polarization axis)**: STOKES for power beams give the polarization integer convention from uvfits (see AIPS memo 117), FEEDIND for E-field beams give the index into the FEEDLIST. The corresponding CRPIXm values should be 1 for both and for FEEDIND the CRVALm and CDELTm keywords should 1. (polarization_array or feed_array)

4. **IF**: The IF axis enumerates spectral windows (frequency bands). The corresponding CRVALm, CRPIXm, and CDELTm keywords should all have the value 1. Currently only one spectral window is supported so this can only have length 1. (spw_array)

5. **VECIND**: Basis vector index, the index into the dimensions of the basis vector array (stored in the BASISVEC extension). This is usually length 1 for power beams and length 2 for E-field beams. The corresponding CRVALm, CRPIXm, and CDELTm keywords should all have the value 1.

6. **COMPLEX**: real and imaginary (axis length 2), the first entry on this axis contains the real part of the complex beam and the second contains the corresponding imaginary component. The corresponding CRVALm, CRPIXm, and CDELTm keywords should all have the value 1.

3.2.2 regularly pixelized beam axes

For regularly pixelized beams, the data axes (values of the CTYPEm keywords in order) are:

1. **AZIMUTH or ZENX-SIN (First Image Axis)**: The coordinates for the first image axis, either azimuth angle or the East/West axis of a zenith orthoslant projection, should match the coordinate system given by the COORDSYS keyword. The values of the coordinates are set by the CRVALm, CRPIXm, and CDELTm keywords and the CUNITm keyword gives the units (default is degrees).(axis1_array)

2. **ZENANGLE or ZENY-SIN (Second Image Axis)**: The coordinates for the second image axis, either zenith angle or the North/South axis of a zenith orthoslant projection, should match the coordinate system given by the COORDSYS keyword. The values of the coordinates are set by the CRVALm, CRPIXm, and CDELTm keywords and the CUNITm keyword gives the units (default is degrees).(axis1_array)

3. **FREQ**: Frequency axis, the corresponding CRVALm shall have the reference frequency for the data set, both CRVALm and CDELTm are given in units given by the CUNITm keyword, which is usually Hz. (freq_array)

4. **STOKES or FEEDIND (Polarization axis)**: STOKES for power beams give the polarization integer convention from uvfits (see AIPS memo 117), FEEDIND for E-field beams give the index into the FEEDLIST. The corresponding CRPIXm values should be 1 for both and for FEEDIND the CRVALm and CDELTm keywords should 1. (polarization_array or feed_array)

5. **IF**: The IF axis enumerates spectral windows (frequency bands). The corresponding CRVALm, CRPIXm, and CDELTm keywords should all have the value 1. Currently only one spectral window is supported so this can only have length 1. (spw_array)

6. **VECIND**: Basis vector index, the index into the dimensions of the basis vector array (stored in the BASISVEC extension). This is usually length 1 for power beams and length 2 for E-field beams. The corresponding CRVALm, CRPIXm, and CDELTm keywords should all have the value 1.

7. **COMPLEX**: real and imaginary (axis length 2), the first entry on this axis contains the real part of the complex beam and the second contains the corresponding imaginary component. The corresponding CRVALm, CRPIXm, and CDELTm keywords should all have the value 1.

4 Basis Vector HDU

The basis vector HDU is a required element of E-field *beamfits* files. It is allowed but not required for power beam files. It is an Image HDU containing the basis vector array. The **EXTNAME** keyword in the header must be set to BASISVEC. The only other required keyword in the header is:

- **COORDSYS**: *string* should be set to one of 'az_za' (for an azimuth, zenith angle regular grid), 'orthoslant_zenith' (for an orthoslant projection at zenith) or 'healpix' and must match the corresponding keyword in the primary header (this is used as a consistency check). (pixel_coordinate_system)

4.1 Basis Vector Data

The axes of the data array in the basis vector HDU depends on the pixelization scheme.

4.1.1 HEALPix basis vector axes

For HEALPix beams, the basis vector data axes (values of the CTYPEm keywords in order) are:

1. **PIX_IND**: index into the HEALPix pixel array (in the HPX_INDS extension) This is needed because partial sky maps are supported so pixels may be missing (full sky maps are not handled differently, they have the same required elements as partial maps). The corresponding CRVALm, CRPIXm, and CDELTm values should all be 1. (pixel_array)

2. **AXISIND**: vector coordinate system axis index. For HEALPix beams, the two axes are azimuth and zenith angle in that order. The corresponding CRVALm, CRPIXm, and CDELTm values should all be 1.

3. **VECCOORD**: Basis vector index, the index into the dimensions of the basis vector array (stored in the BASISVEC extension). This is usually length 1 for power beams and length 2 for E-field beams. The corresponding CRVALm, CRPIXm, and CDELTm keywords should all have the value 1.

4.1.2 regularly pixelized basis vector axes

For regularly pixelized beams, the basis vector data axes (values of the CTYPEm keywords in order) are:

1. **AZIMUTH or ZENX-SIN (First Image Axis)**: The coordinates for the first image axis, either azimuth angle or the East/West axis of a zenith orthoslant projection, should match the coordinate system given by the COORDSYS keyword. The values of the coordinates are set by the CRVALm, CRPIXm, and CDELTm keywords and the CUNITm keyword gives the units (default is degrees).(axis1_array)

2. **ZENANGLE or ZENY-SIN (Second Image Axis)**: The coordinates for the second image axis, either zenith angle or the North/South axis of a zenith orthoslant projection, should match the coordinate system given by the COORDSYS keyword. The values of the coordinates are set by the CRVALm, CRPIXm, and CDELTm keywords and the CUNITm keyword gives the units (default is degrees).(axis1_array)

3. **AXISIND**: vector coordinate system axis index. For regularly pixelized beams, the coordinate system matches the pixel coordinate system ([azimuth, zenith angle] or zenith orthoslant [x, y]). The corresponding CRVALm, CRPIXm, and CDELTm values should all be 1.

4. **VECCOORD**: Basis vector index, the index into the dimensions of the basis vector array (stored in the BASISVEC extension). This is usually length 1 for power beams and length 2 for E-field beams. The corresponding CRVALm, CRPIXm, and CDELTm keywords should all have the value 1.

5   HEALPix Index HDU

The HEALPix index HDU is a required element of HEALPix *beamfits* files (regardless of whether the file includes all the pixels in the sky or not). It is a Binary Table HDU containing the HEALPix pixel numbers. The **EXTNAME** keyword in the header must be set to HPX_INDS. There are no other required keywords in the header.

5.1   HEALPix Pixel Table

There is only one required column in the HPX_INDS table:

- **HPX_INDS**: The HEALPix pixel numbers in the order that they are stored in the beam and basis vector arrays. (pixel_array)

6   Band Parameter HDU

The band parameter HDU is an required element of *beamfits* files. It is a Binary Table HDU containing the bandpass and other optional parameters that are functions of frequency only. The **EXTNAME** keyword in the header must be set to BANDPARM. There are no other required keywords in the header, but there are 2 optional header keywords:

- **REFZIN**: *float* Reference input impedance of the receiving chain (sets the reference for the S parameters), units: Ohms. (reference_input_impedance)

- **REFZOUT**: *float* Reference output impedance of the receiving chain (sets the reference for the S parameters), units: Ohms. (reference_output_impedance)

6.1   BANDPARM Table

BANDPASS is the only required column in the BANDPARM table, but there are several optional columns that may be present. The number of rows in the table should equal the number of frequencies in the beam (primary HDU data array).

- **BANDPASS**: *float* Frequency dependence of the beam. Depending on the data_normalization, this may contain only the frequency dependence of the receiving chain or all the frequency dependence (bandpass_array).

- **RX_TEMP**: *float* Receiver temperatures. (receiver_temperature_array)

- **LOSS**: *float* Antenna losses. (loss_array)

- **MISMATCH**: *float* Antenna-amplifier mismatch. (mismatch_array)

- **S11**: *float* S11[3] parameter of the receiving chain. (s_parameters)

- **S12**: *float* S12[3] parameter of the receiving chain. (s_parameters)

- **S21**: *float* S21[3] parameter of the receiving chain. (s_parameters)

- **S22**: *float* S22[3] parameter of the receiving chain. (s_parameters)

---

[3]https://en.wikipedia.org/wiki/Scattering_parameters#Two-Port_S-Parameters