OSKAR Example

Version history:

Revision	Date	Modification
1	2012-04-24	Creation
2	2012-05-15	Updated figures and text for correct polarisation order.
3	2012-06-13	Updated figures and text to reflect changes to the output options for the beam pattern images and the GUI colour settings introduced in OSKAR 2.0.2-beta.
4	2012-06-19	Updated figures for OSKAR 2.0.3-beta.
5	2012-07-27	Updated figures for OSKAR 2.0.4-beta
6	2012-10-22	Updated for revised settings in OSKAR 2.1-beta.
7	2013-03-04	Updated for revised settings in OSKAR 2.2.0.
8	2013-11-13	Updated for revised beam pattern settings in OSKAR 2.3.0
9	2014-07-22	Updated for new element pattern fitting procedure in OSKAR 2.5.0.
10	2014-09-09	Updated figures for revised default settings in OSKAR 2.5.1.

1 Introduction

This document will guide you through an example to perform a couple of simple simulations using OSKAR.

Note that all steps are intended to be followed in order.

2 Getting Started

This example assumes that OSKAR has been built successfully with the CUDA, Qt4, CBLAS, LAPACK, and CFITSIO dependencies satisfied (for details, see the install guide documentation).

2.1 Obtaining the Example Setup Files

The data files used for this example can be downloaded from:

http://www.oerc.ox.ac.uk/~ska/oskar/OSKAR-Example-Data.zip

This archive contains:

Item	Description
setup.ini	An OSKAR configuration file used to set up the example simulation.
sky.osm	An OSKAR sky model file, containing 3 sources.
telescope/	An OSKAR telescope configuration directory structure consisting of an interferometer with 30 aperture array stations, each with 2587 antenna elements.
dipole_CST.txt	A numerically-defined half-wavelength dipole pattern, as an example element pattern.

Decompress the archive once you have downloaded it, and verify that it contains the above files.

2.2 Running the OSKAR GUI

The OSKAR GUI, used to run the simulation examples described in the following sections, should be launched by executing the `oskar` binary. This can be found in the `build/apps` directory after compiling the OSKAR package, and in the `<install prefix>/bin` directory after installing OSKAR using the `make install` command.

2.3 Visualising Simulation Results

The example simulations described in the following sections generate a number of FITS images. These are standard FITS images, and therefore any FITS viewer should be capable of visualising the results. The screenshots shown in this document use SAOImage DS9, an astronomical imaging and data visualization application, which can be downloaded from http://ds9.si.edu

3 Example Simulations

3.1 The OSKAR Simulator GUI

The OSKAR GUI provides an interface for configuring and running beam pattern and interferometry simulations.

Start the OSKAR GUI by running the `oskar` binary and load the `setup.ini` settings file. Settings files can be loaded into the GUI by specifying the settings file path as the first command line argument when starting the OSKAR GUI, or by using the **Open...** action in the **File** menu once the GUI has started.

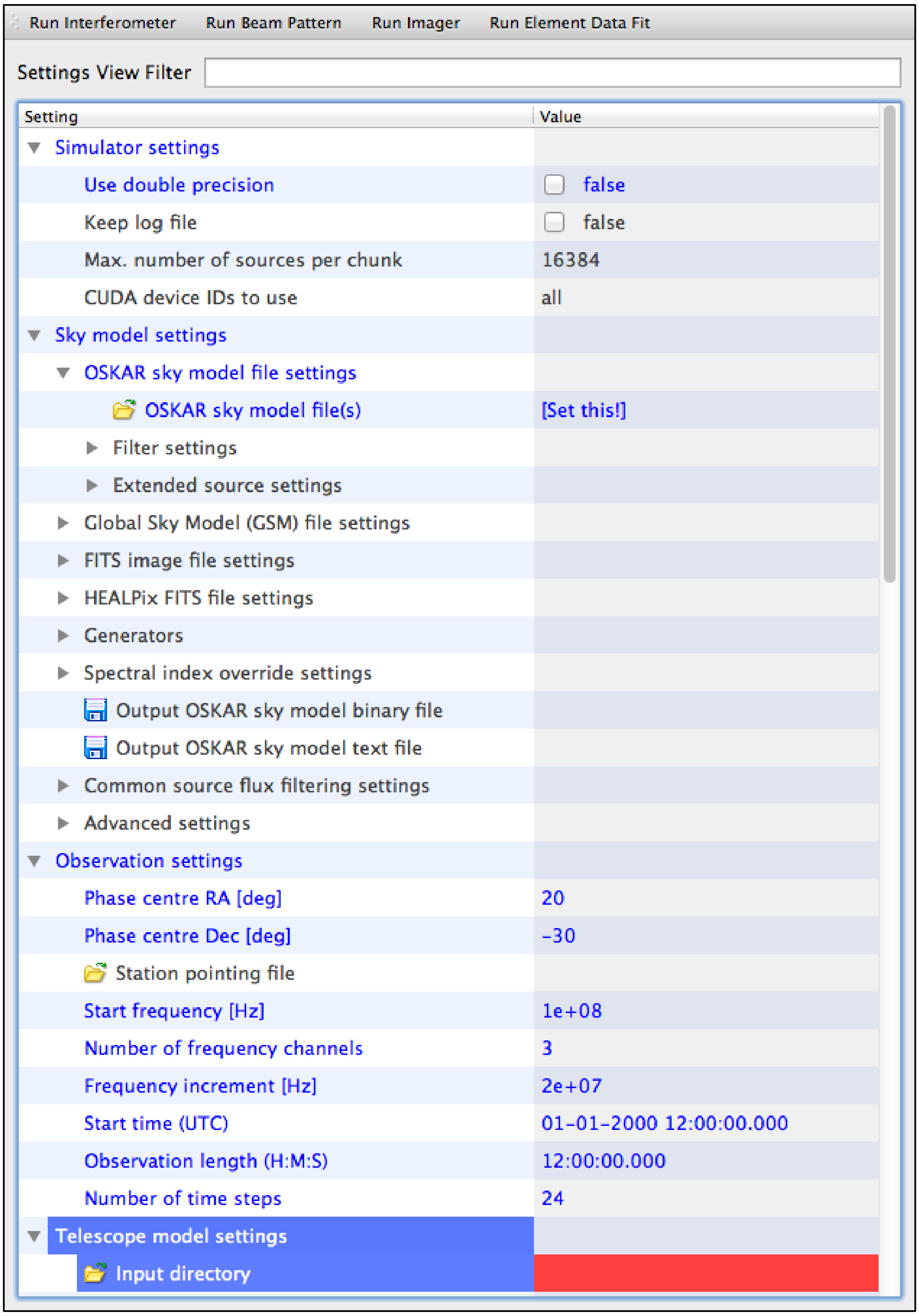

Figure 1: OSKAR GUI with example settings loaded.

The example settings file contains most of the options required to run a simple simulation. The options can be inspected in the settings tree, as shown in Figure 1.

The settings are arranged into a number of groups:

- **Simulator settings**, for parameters which affect the operation of the software.
- **Sky model settings**, for parameters that are used to specify the content of the sky model.
- **Observation settings**, for parameters that are used to specify the observation.
- **Telescope model settings**, for parameters that are used to specify the contents of the telescope model.
- **Interferometer settings**, for parameters used only when running interferometer simulations.
- **Element pattern fitting settings**, for parameters used only when fitting numerical element pattern data.
- **Beam pattern settings**, for parameters used only when running beam pattern simulations.
- **Image settings**, for parameters used only when running the OSKAR imager.

Most settings have a default value, which is indicated by black text in the settings tree. Options set by the user, which have been defined in the settings file, appear as blue text. Options which are required to run the simulator, and have not yet been assigned, have their description and value backgrounds coloured blue and red respectively.

Note that the settings file is updated automatically whenever any setting is modified, so you will never need to save the file explicitly.

This example will run a beam pattern simulation, then an interferometer simulation, and then the OSKAR imager to inspect the results.

Before starting any simulations, the paths to the sky model file and telescope directory found in the example settings data archive (see Section 2.1) must be set as follows:

1. The sky model file (`sky.osm`) should be specified by double clicking the value field to the right of the **OSKAR sky model file(s)** setting found in the **Sky model settings** group. This will open a file dialog from where the path to the `sky.osm` file can be specified.
2. The telescope model should be specified by adding the `telescope` directory path to the **Input directory** setting found in the **Telescope model settings** group, also by double clicking the value field and selecting it from the directory browser dialog.

The settings tree now contains all of the settings required to run a beam pattern and interferometry simulation. Of note:

- The settings define simulations using single precision arithmetic, as this is much faster (the default is double precision).
- The sky model consists of three sources in a field centred at RA 20°, Dec. -30°.
- The observation of this field is made at three frequency channels starting at 100 MHz and increasing in increments of 20 MHz.
- The observation consists of 24 snapshots taken over a 12 hour period.
- The telescope model consists of an interferometer of 30 aperture array stations, each containing 2587 half-wavelength dipole antennas. Antennas within each station are arranged in a circular offset lattice pattern.

3.2 Beam Pattern Simulation

This section describes how to perform a beam pattern simulation using the example observation settings, as configured in Section 3.1.

The beam pattern settings group for this example is shown in Figure 2. The beam pattern simulation will generate an image of the beam pattern for the first station, centred on the observation phase centre, and specified by the field-of-view and image size settings. In this example, we will generate the primary beam response of station 0 of the telescope as a 256 by 256 pixel FITS image covering a field-of-view of 180°.

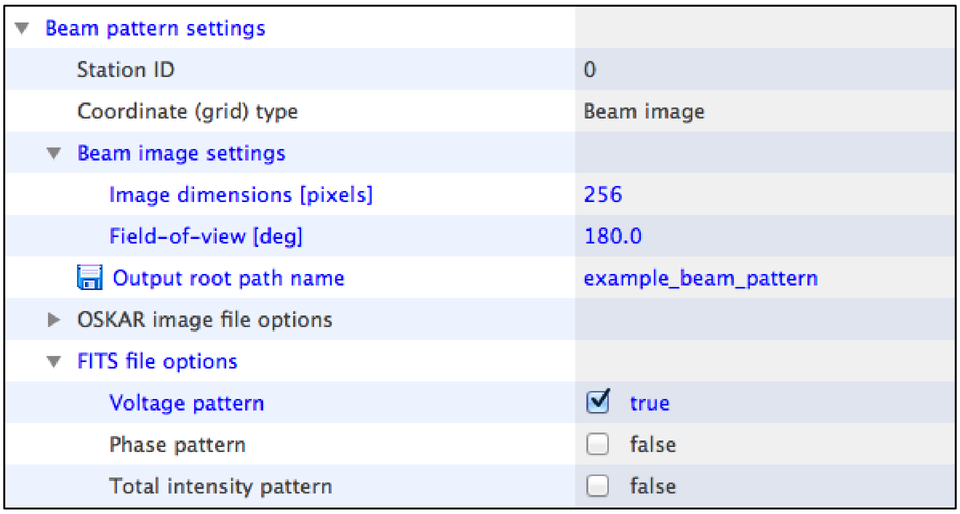

Figure 2: Beam pattern settings.

The beam pattern simulation is run by pressing the **Run Beam Pattern** button on the toolbar of the GUI, or by selecting the **Run Beam Pattern** option in the **Run** menu.

While this is running, OSKAR will display a log indicating the current progress of the simulation. Once the simulation is complete, the results can be found in the FITS image file called `example_beam_pattern_VOLTAGE.fits` (unless you changed it) found in the directory from which the simulator was run.

The output file is a FITS image cube containing the beam pattern for the theta and phi voltage response of the X and Y dipoles, at the three observation frequencies (100, 120, and 140 MHz), and 24 pointing centres corresponding to the 24 observation snapshots.

Figure 3 shows the first plane of this FITS cube, which is the theta pattern voltage response (in Right Ascension and Declination) of the X dipole for the first frequency channel and first time snapshot. Note that the area of zero voltage on the left of the pattern corresponds to parts of the pattern below the horizon (the pointing direction for this observation starts near the horizon).

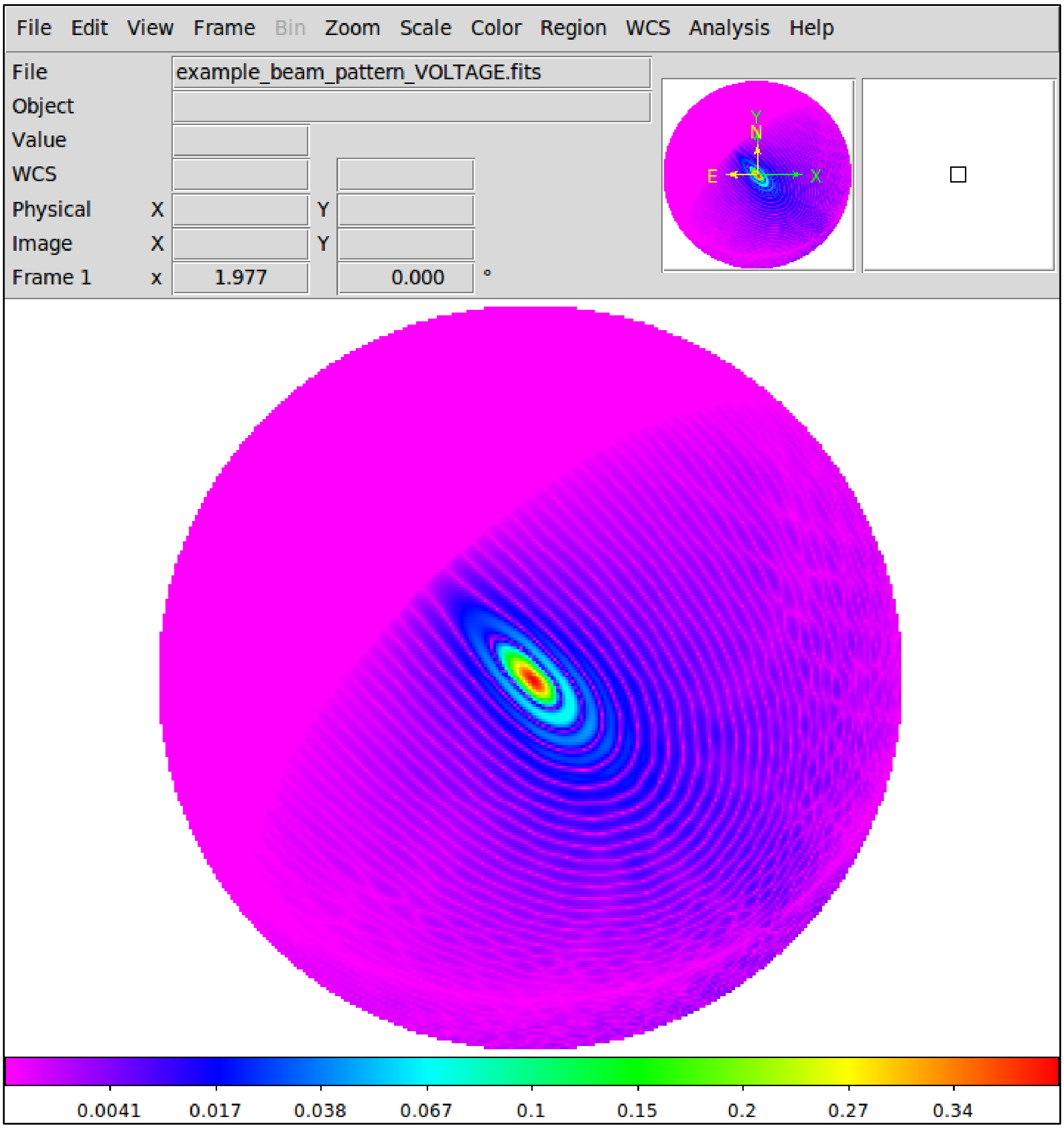

Figure 3: FITS image showing the voltage pattern response in theta for the X-dipoles, and plotted using the 'square root' scale in DS9. The beam pattern is for the first station, the first channel (100 MHz), at the first time snapshot.

3.3 Interferometry Simulation

This section describes how to run an interferometry simulation for the example observation settings, configured in Section 3.1.

To run the simulation, select the **Run Interferometer** action, either by pressing the button on the toolbar, or by selecting the option in the **Run** menu.

While the interferometer simulation is running, a log will be shown.

Once the simulation has successfully completed, the results can be found in an OSKAR visibility binary file called `example.vis` (unless you changed it). This can be found in the directory from which the OSKAR GUI was started. OSKAR can also be configured to output visibilities to a Measurement Set for further post-processing in CASA. However, for this example, we will inspect the results of the simulation by making an image cube of the results with the OSKAR imager, which requires input in the form of an OSKAR visibility data file.

3.4 The OSKAR Imager: Imaging Simulated Visibilities

This section demonstrates how the results of running the interferometry simulation, described in Section 3.3, can be imaged using the OSKAR imager.

The OSKAR imager uses the configuration found in the **Image settings** group, as shown in Figure 4.

For this example the imager is configured to generate a FITS image cube consisting of images of 256 by 256 pixels covering a field-of-view of 4° around the observation phase centre. The polarisation, time and frequency dimensions of the image cube depend on the image type, and whether time and channel snapshots have been selected. The settings in this example will generate an image cube with four linear polarisations, three channels, and (by the application of time synthesis) one time dimension.

In order to run the imager, select the **Run Imager** action, either by pressing the button on the toolbar, or by selecting the option in the **Run** menu.

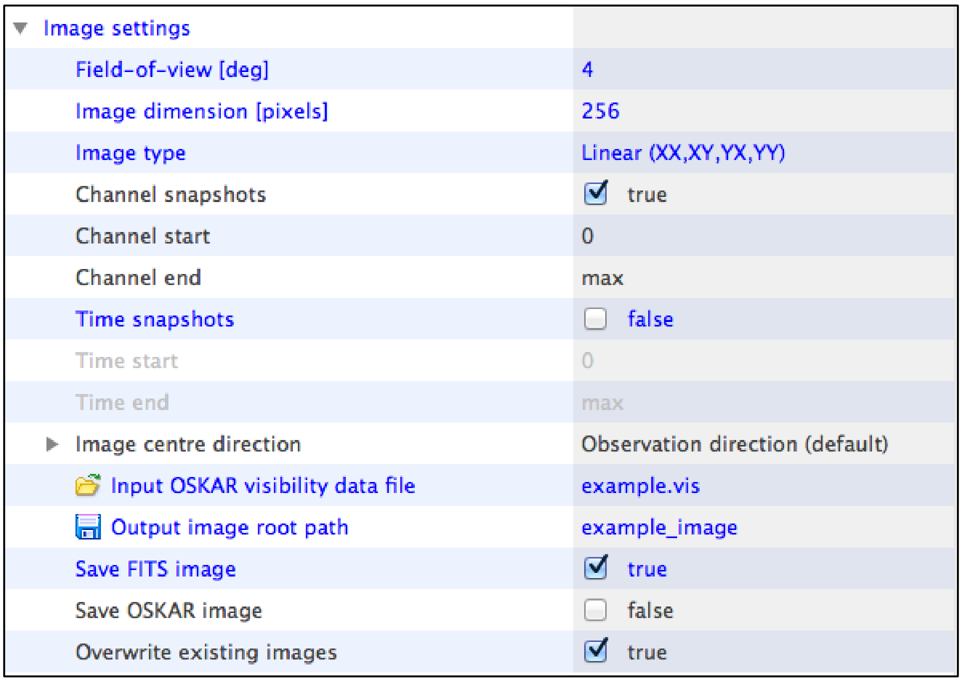

Figure 4: Image settings.

While the imager is running, a progress log will be shown. The file name of the image cube is generated from the **Output image root path** setting and the **Image type**, using the pattern `<root path>_<image type>.fits`. The image generated in this example can be found in the directory from which the OSKAR GUI was started with the name `example_image_LINEAR.fits`.

Figure 5 shows a slice of this generated FITS image cube for the XX polarisation and first frequency channel.

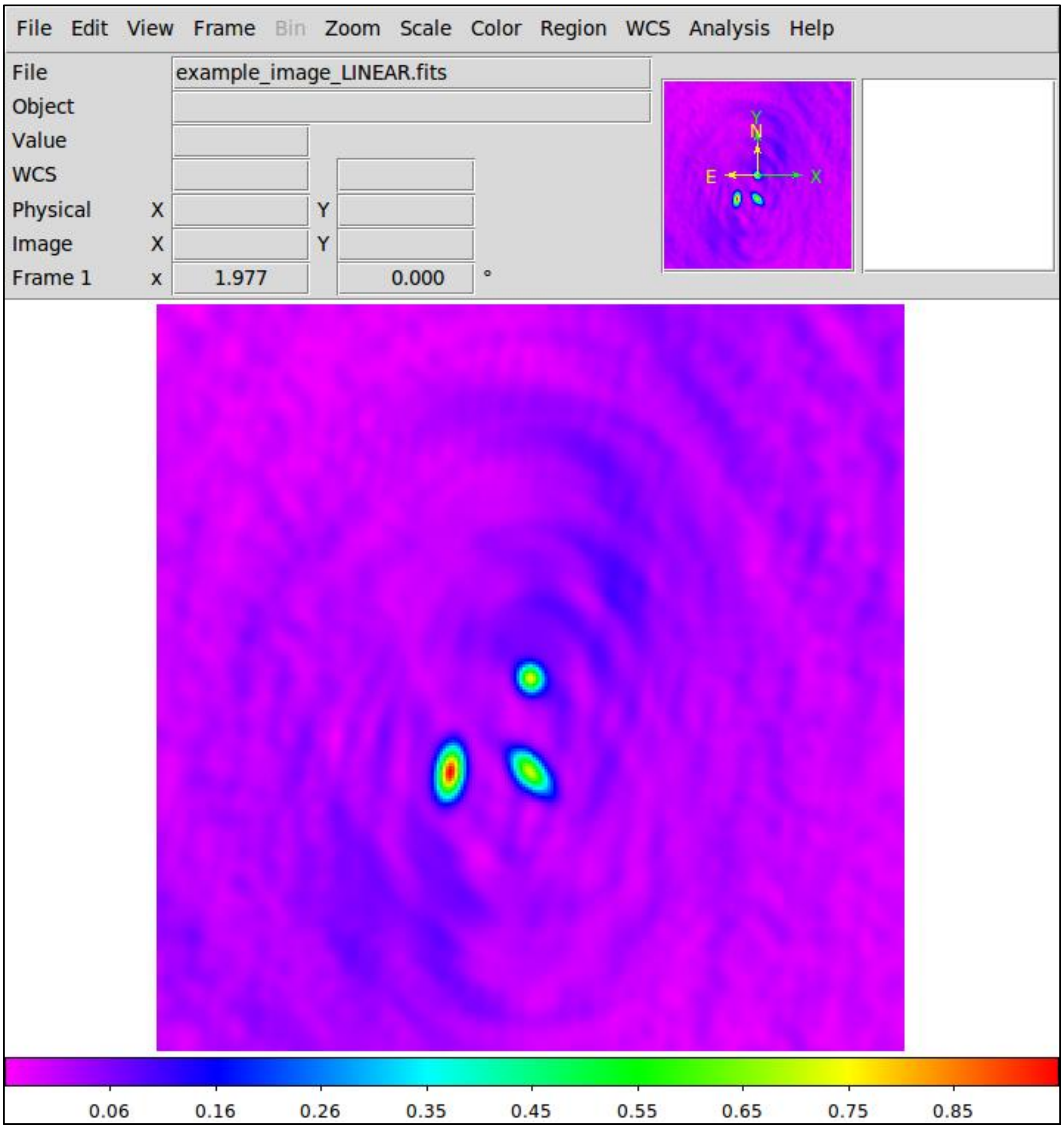

Figure 5: FITS image showing the raw (dirty) image formed by imaging the example simulated visibilities with the OSKAR imager. The image shows the XX polarisation response for channel 0 (100 MHz) for the 12 hour observation.

3.5 Using a Custom Element Pattern

This section describes how to plot a custom element pattern, and run a beam pattern simulation using it. Note that this procedure was different in OSKAR versions prior to 2.5.0.

First, splines must be fitted to the numerical element pattern data, and the fitted coefficients saved out to the telescope model. This can be done using the options in the **Element pattern fitting settings** group. Set these options as shown in Figure 6. We have selected the example input CST-format file `dipole_CST.txt`, the frequency at which it is valid (we have chosen 100 MHz in this example), and the output telescope or station directory that the data files should be saved into. We want to use this data only for the first station, so we have selected the `telescope/station000` directory here. (If we wanted to use the same element pattern data for all stations, then selecting the top-level `telescope` directory would be necessary instead.)

Element pattern fitting settings	
Input CST file	dipole_CST.txt
Frequency [Hz]	1e+08
Polarisation type	XY
Element type index	0
Ignore data at poles	false
Ignore data below horizon	true
Average fractional error	0.005
Average fractional error factor increase	1.1
Telescope or station directory	telescope/station000

Figure 6: Settings for the element pattern fitting procedure.

Select the **Run Element Data Fit** action to run the application to fit the element data with these parameters, and save the fit to the telescope model. *Note that this fitting procedure only needs to be performed if the element pattern itself has been updated*.

In order to make use of this data, the option **Enable numerical patterns if present** (in the **Telescope model settings**) must be set to **true**. To look at the response of the element, perform the following steps in the **Telescope model settings** and then **Aperture array settings** group:

- Enable the option to use custom element patterns by ticking **Enable numerical patterns if present** in the **Element pattern settings** group.
- Disable the option to evaluate the array pattern of the station by un-ticking **Enable array pattern** in the **Array pattern settings** group.

Your **Telescope model settings** should now appear as shown in Figure 7.

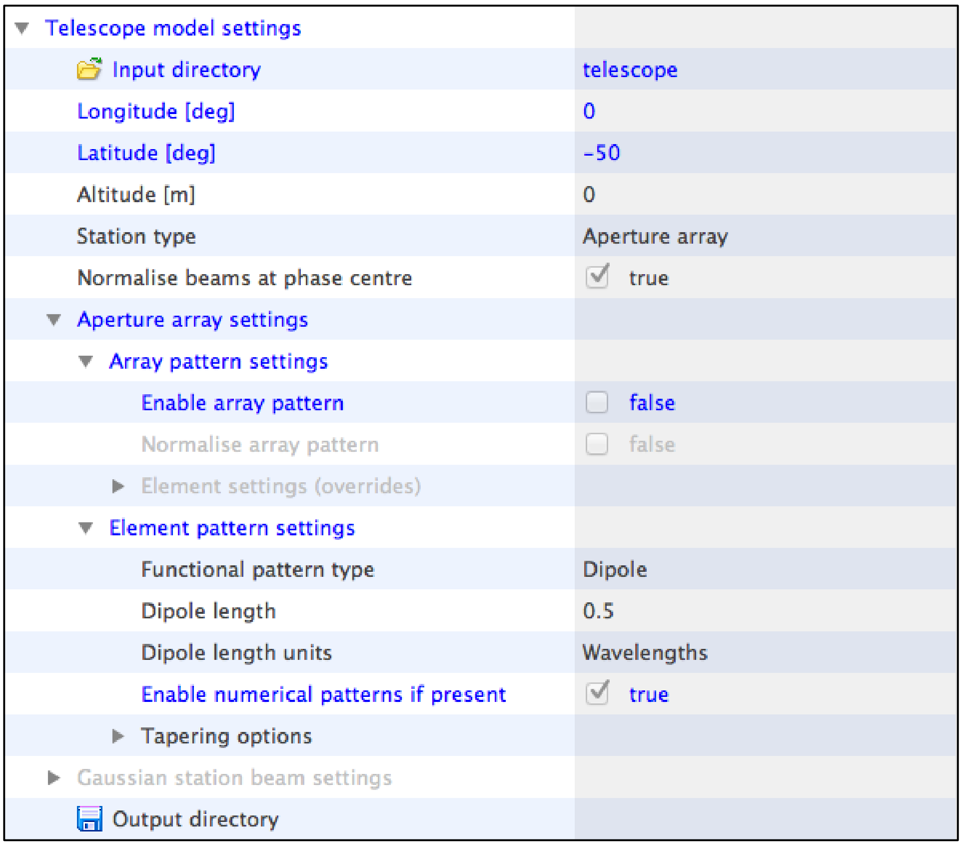

Figure 7: Telescope settings for plotting custom element pattern data.

Run the beam pattern simulation using the **Run Beam Pattern** action as before, and view the output file (`example_beam_pattern_VOLTAGE.fits`, unless you changed it) using DS9. The output of polarisation plane 1 (this is the theta response of the X dipole) and time slice 10 should appear as shown in Figure 8.

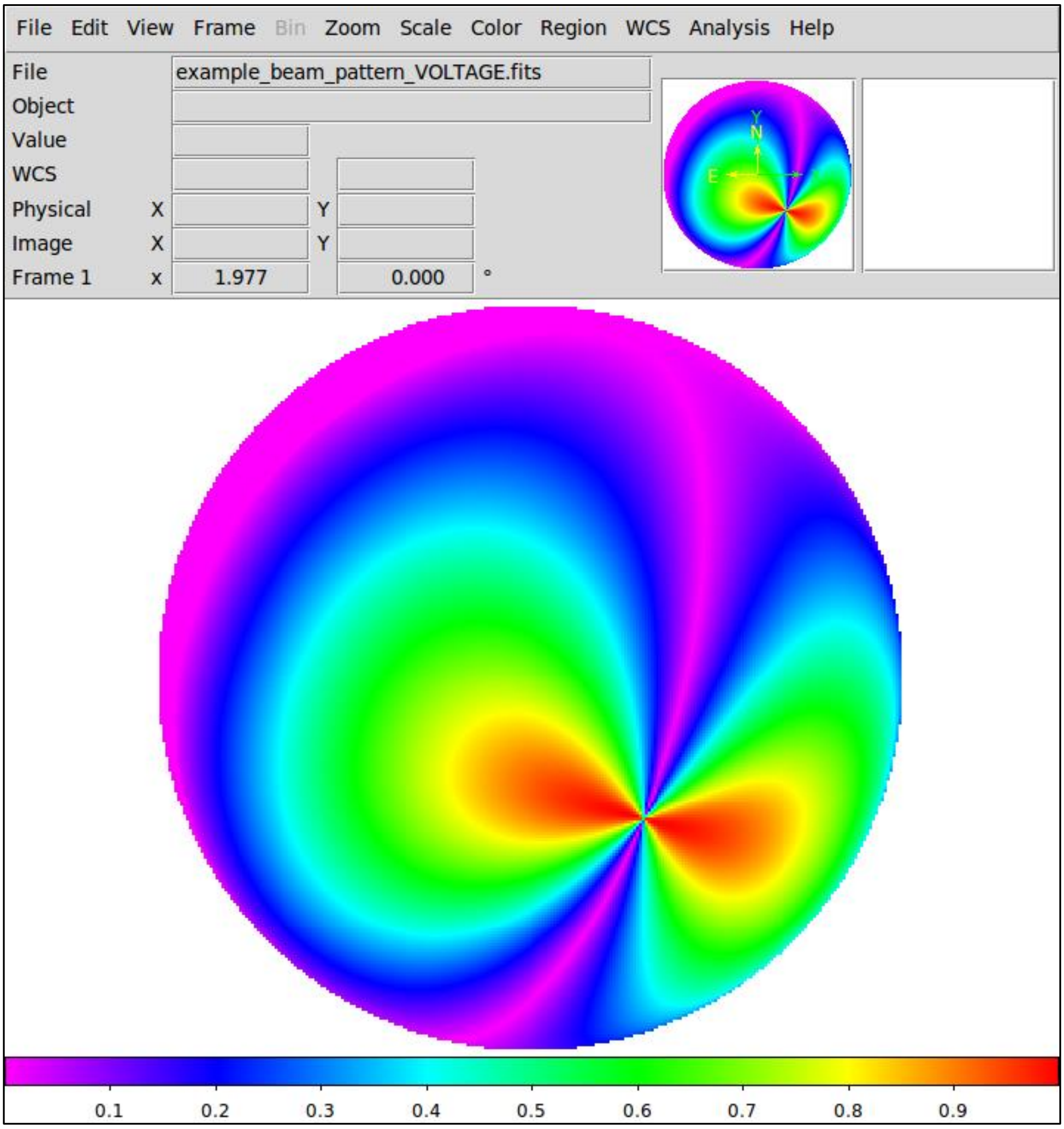

Figure 8: FITS image showing a custom element pattern (which happens to be for an ideal dipole) in polarisation plane 1 and time slice 10 of the data cube.

There is a slight difference between this output and that from an analytical dipole, because the element response in this case was generated by fitting B-splines to the numerical input data. You can try reducing the **Average fractional error** setting in the **Element pattern fitting settings** section if you want to try obtaining a closer fit: Using 0.0005 here will give noticeably better results for a dipole.

Try running a full beam pattern simulation by re-selecting the **Enable array pattern** option and then **Run Beam Pattern** again. The new beam pattern for the same time-step is shown in Figure 9.

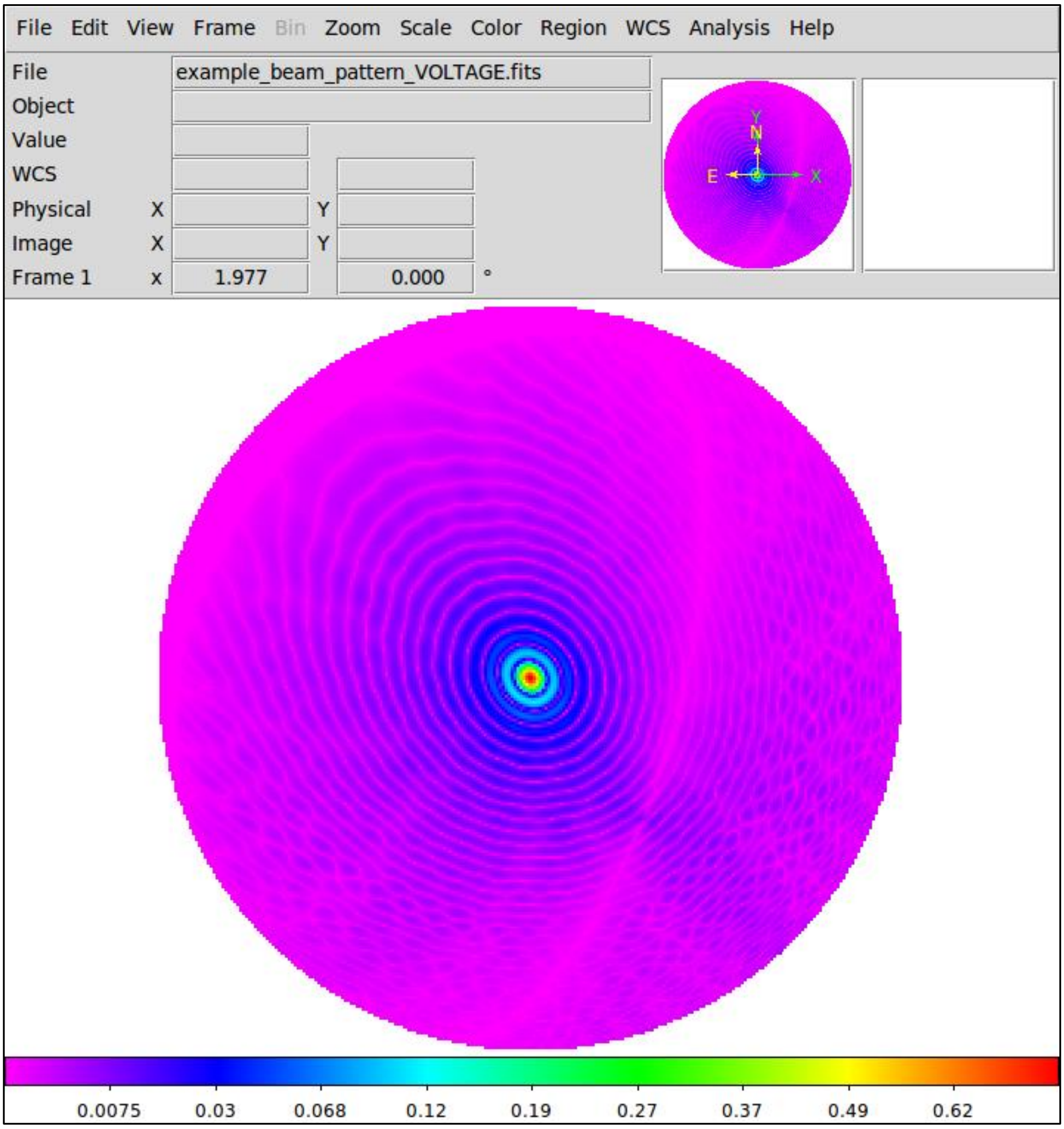

Figure 9: Beam pattern generated using a custom element pattern, displayed at the same time and polarisation index as described in Figure 8.