The POL-2 Data Reduction Cookbook 1.0
Abstract

This cookbook provides an introduction to POL-2 data reduction, using the Starlink facilities SMURF (the Sub-Millimetre User Reduction Facility) and in particular its command pol2map. This cookbook illustrates the various steps required to reduce the data, including an overview of the method. It also describes how to calibrate and display the data as images or vector maps.
# Contents

## Acronyms

<table>
<thead>
<tr>
<th>1 Introduction</th>
<th>1 Introduction</th>
<th>1 Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 This cookbook</td>
<td>1.2 Before you start: computing resources</td>
<td>1.3 Before you start: software</td>
</tr>
<tr>
<td>1.1.1 Data formats</td>
<td>1.1.2 Initialising Starlink</td>
<td>1.1.3 KAPPA and SMURF for data processing</td>
</tr>
<tr>
<td>1.1.4 GAIA for viewing your images and vector maps</td>
<td>1.1.5 How to get help</td>
<td></td>
</tr>
</tbody>
</table>

## 2 POL-2 Overview

<table>
<thead>
<tr>
<th>2 POL-2 Overview</th>
<th>2 POL-2 Overview</th>
<th>2 POL-2 Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 The instrument</td>
<td>2.2 Instrumental Polarisation</td>
<td>2.3 Observing mode</td>
</tr>
<tr>
<td>2.4 The raw data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## 3 POL-2 Data Reduction – The Theory

<table>
<thead>
<tr>
<th>3 POL-2 Data Reduction – The Theory</th>
<th>3 POL-2 Data Reduction – The Theory</th>
<th>3 POL-2 Data Reduction – The Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 The Data Flow</td>
<td>3.2 MAKEMAP</td>
<td>3.3 CALCQU</td>
</tr>
<tr>
<td>3.4 PCA</td>
<td>3.5 Masking</td>
<td>3.6 Tailoring a reduction</td>
</tr>
</tbody>
</table>

## 4 POL-2 Data Reduction – Running pol2map

<table>
<thead>
<tr>
<th>4 POL-2 Data Reduction – Running pol2map</th>
<th>4 POL-2 Data Reduction – Running pol2map</th>
<th>4 POL-2 Data Reduction – Running pol2map</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 How to use pol2map</td>
<td>4.2 pol2map – producing the initial I map</td>
<td>4.3 pol2map – producing the I, Q, U maps and catalogue</td>
</tr>
<tr>
<td>4.4 Output vectors from pol2map</td>
<td>4.5 POL-2 FCFs</td>
<td>4.6 Changing pixel size in pol2map</td>
</tr>
</tbody>
</table>

## 5 POL-2 Image Display

<table>
<thead>
<tr>
<th>5 POL-2 Image Display</th>
<th>5 POL-2 Image Display</th>
<th>5 POL-2 Image Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 GAIA</td>
<td>5.2 KAPPA and polpack</td>
<td>5.3 TOPCAT</td>
</tr>
<tr>
<td>5.2.1 Example 1 – a vector map with no background</td>
<td>5.2.2 Example 2 – a vector map over a contour map</td>
<td>5.2.3 Example 3 – a vector map over an image</td>
</tr>
</tbody>
</table>

## 6 POL-2 – Advanced Data Reduction

<table>
<thead>
<tr>
<th>6 POL-2 – Advanced Data Reduction</th>
<th>6 POL-2 – Advanced Data Reduction</th>
<th>6 POL-2 – Advanced Data Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Adding new observations</td>
<td>6.2 Experimenting with pixel sizes</td>
<td></td>
</tr>
</tbody>
</table>
6.3 Investigating systematic error in IP ................................. 38
6.4 Adding WCS information back into a vector catalogue .................. 38
References .............................................................................. 40
A PCA on SCUBA-2 data ......................................................... 41
List of Figures

2.1 POL-2 mounted on SCUBA-2 .................................................. 5
2.2 POL-2 optical components .................................................. 7
2.3 Modulation by the HWP - basic description ......................... 7
2.4 Attenuation of signal by HWP .............................................. 8
2.5 POL-2 components .......................................................... 9
2.6 Scan Pattern Comparison .................................................. 10
2.7 Detail of POL-2 Scan Pattern ............................................. 11
3.1 POL-2 Data Flow .......................................................... 14
4.1 I map in GAIA ............................................................. 23
4.2 Final I map in GAIA .......................................................... 26
4.3 Q and U maps in GAIA ..................................................... 26
5.1 Over Plotting Vectors in GAIA ........................................... 28
5.2 Selecting Vectors in GAIA .................................................. 29
5.3 Over Plotting Vectors in GAIA ........................................... 29
5.4 Vector map with polplot .................................................... 32
5.5 Vector map with contour map in polplot .............................. 33
5.6 Vector map with negative image in polplot ......................... 34
5.7 A TOPCAT scatter plot ................................................... 35
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CADC</td>
<td>Canadian Astronomy Data Centre</td>
</tr>
<tr>
<td>FCF</td>
<td>Flux Conversion Factor</td>
</tr>
<tr>
<td>FITS</td>
<td>Flexible Image Transport System</td>
</tr>
<tr>
<td>GAIA</td>
<td>Graphical Astronomy and Image Analysis tool</td>
</tr>
<tr>
<td>HWP</td>
<td>Half-Wave Plate</td>
</tr>
<tr>
<td>ITC</td>
<td>Integration Time Calculator</td>
</tr>
<tr>
<td>I</td>
<td>Total intensity</td>
</tr>
<tr>
<td>IP</td>
<td>Instrumental Polarisation</td>
</tr>
<tr>
<td>JCMT</td>
<td>James Clerk Maxwell Telescope</td>
</tr>
<tr>
<td>NDF</td>
<td>Extensible N-Dimensional Data Format</td>
</tr>
<tr>
<td>P</td>
<td>Percentage polarisation</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>$I_p$</td>
<td>Polarised intensity</td>
</tr>
<tr>
<td>SCUBA-2</td>
<td>Submillimetre Common User Bolometer Array-2</td>
</tr>
<tr>
<td>SMURF</td>
<td>Sub-Millimetre User Reduction Facility</td>
</tr>
<tr>
<td>SUN</td>
<td>Starlink User Note</td>
</tr>
<tr>
<td>WCS</td>
<td>World Coordinate System</td>
</tr>
</tbody>
</table>
Chapter 1
Introduction

1.1 This cookbook

This guide is designed to instruct POL-2 users on the best ways to reduce and visualise their data using Starlink packages: SMURF, KAPPA, POLPACK and GAIA. This guide covers the following topics.

- Chapter 1 - Computer resources needed before getting started.
- Chapter 2 - A description of POL-2 and its observing modes.
- Chapter 3 - POL-2 Data Reduction - The Theory
- Chapter 4 - POL-2 Data Reduction - Running pol2map
- Chapter 5 - POL-2 Image Display
- Chapter 6 - POL-2 Advanced Data Reduction

Throughout this document, a percent sign (%) is used to represent the Unix shell prompt. What follows each % will be the text that you should type to initiate the described action.

1.2 Before you start: computing resources

Compared with SCUBA-2 observations, POL-2 observations are far less memory-intensive to reduce. POL-2 time-series data is down-sampled to 2 Hz as a part of the reduction process. Assuming a typical 35-minute POL-2 observation, the reduction requires 35 GB of memory (in comparison with SCUBA-2 maps that may require up to 96 GB of memory).

The main consideration for POL-2 reductions is processing power. PCA calculations in makemap can be lengthy so fast processors with lots of cores are advised.

1.3 Before you start: software

This manual uses software from Starlink packages: SMURF, KAPPA, POLPACK and GAIA. Starlink software must be installed on your system, and Starlink aliases and environment variables must be defined before attempting to reduce any SCUBA-2 data (see Section 1.3.2).
1.3.1 Data formats

Data files for POL-2 are structurally the same as for SCUBA-2, and use the Starlink \( N \)-dimensional Data Format (NDF, see Jenness et al. 2014[16]), a hierarchical format which allows additional data and metadata to be stored within a single file. [KAPPA] contains many commands for examining and manipulating NDF structures. The introductory sections of the [KAPPA] document (SUN/95) contain much useful information on the contents of an NDF structure and how to manipulate them.

A single NDF structure describes a single data array with associated meta-data. NDFs are usually stored within files of type .sdf. In most cases (but not all), a single .sdf file will contain just one top-level NDF structure, and the NDF can be referred to simply by giving the name of the file (with or without the .sdf suffix). In many cases, a top-level NDF containing JCMT data will contain other ‘extension’ NDFs buried inside them at a lower level. For instance, raw files contain a number of NDF components, which store observation-specific data necessary for subsequent processing. The contents of these (and other NDF) files may be listed with [HDSTRACE]. Each file holding raw JCMT data on disk is also known as a ‘sub-scan’.

The main components of any NDF structure are:

- an array of numerical data (which may have up to seven dimensions—usually three for JCMT data);
- an array of variance values corresponding to the numerical data values;
- an array holding up to eight Boolean flags (known as ‘quality flags’) for each pixel;
- World Co-ordinate System information;
- history;
- data units; and
- other extensions items. These are defined by particular packages, but usually include a list of FITS-like headers together with provenance information that indicates how the NDF was created. Raw JCMT files also include extensions that define the state of the telescope and instrument at each time slice within the observation.

The Starlink [CONVERT] package contains commands fits2ndf and ndf2fits that allow interchange between FITS and NDF format.

1.3.2 Initialising Starlink

The commands and environment variables needed to start up the required Starlink packages ([SMURF],[KAPPA] etc.) must first be defined. For C shells (csh, tcsh), the commands are:

```bash
% setenv STARLINK_DIR <path to the starlink installation>
% source $STARLINK_DIR/etc/login
% source $STARLINK_DIR/etc/cshrc
```

before using any Starlink commands. For Bourne shells (sh, bash, zsh), the commands are as follows.

```bash
% export STARLINK_DIR=<path to the starlink installation>
% source $STARLINK_DIR/etc/profile
```
1.3.3 KAPPA and SMURF for data processing

The Starlink Sub-Millimetre User Reduction Facility package, or SMURF, contains the Dynamic Iterative Map-Maker, which will process SCUBA-2 time-series data into images (see SUN/258). KAPPA, meanwhile, is an application package comprising general-purpose commands mostly for manipulating and visualising NDF data (see SUN/95). Before starting any data reduction it is necessary to initiate both SMURF and KAPPA.

% smurf
% kappa

After entering the above commands, the help information for the two packages can be accessed by typing smurfhelp or kaphelp respectively in a terminal, or by using the showme facility to access the hypertext documentation. See Section 1.3.5 for more information.

Tip

The .sdf extension on file names need not be specified when running most Starlink commands (the exception is PICARD).

1.3.4 GAIA for viewing your images and vector maps

Images and vector maps can be displayed and analysed using GAIA (see SUN/214) – an interactive GUI-driven tool that incorporates facilities such as vector selection, vector binning, source detection, photometry and the ability to query and overlay on-line or local catalogue data.

% gaia map.sdf

Alternatively, the KAPPA package includes many visualisation commands that can be run from the shell command-line or incorporated easily into your own scripts—see Appendix “Classified KAPPA commands” in SUN/95.

1.3.5 How to get help
<table>
<thead>
<tr>
<th>Help command</th>
<th>Description</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>showme</td>
<td>If you know the name of the Starlink document you want to view, use showme. When run, it launches a new web page or tab displaying the hypertext version of the document.</td>
<td>% showme sun95</td>
</tr>
<tr>
<td>findme</td>
<td>findme searches Starlink documents for a keyword. When run, it launches a new web page or tab listing the results.</td>
<td>% findme kappa</td>
</tr>
<tr>
<td>docfind</td>
<td>docfind searches the internal list files for keywords. It then searches the document titles. The result is displayed using the Unix more command.</td>
<td>% docfind kappa</td>
</tr>
<tr>
<td>Run routines with prompts</td>
<td>You can run any routine with the option prompt after the command. This will prompt for every parameter available. If you then want a further description of any parameter, type ? at the relevant prompt.</td>
<td>% makemap prompt&lt;br&gt; ① REF - Ref. NDF /?/&gt; ?</td>
</tr>
<tr>
<td>Google</td>
<td>A simple Google search such as “starlink kappa fitslist” will usually return links to the appropriate documents. However, the results may include links to out-of-date versions of the document hosted at non-Starlink sites. You should always look for results in &quot;www.starlink.ac.uk/docs&quot; (or &quot;www.starlink.ac.uk/devdocs&quot; for the current development version of the document).</td>
<td></td>
</tr>
</tbody>
</table>
Figure 2.1: POL-2 mounted on the front of SCUBA-2. The left image shows the SCUBA-2 window. The right image shows the components of POL-2 inserted in front of the SCUBA-2 window: the calibrator grid, rotating half-wave-plate (HWP) and the analyser grid. The calibrator grid is only inserted for test purposes.

Chapter 2
POL-2 Overview

2.1 The instrument

The POL-2 instrument is a linear polarimetry module for the Submillimetre Common User Bolometer Array-2 (SCUBA-2), a 10,000 bolometer camera on the JCMT [17] [2]. POL-2 in itself is not a detector - thus requiring SCUBA-2 and its detectors for operation. SCUBA-2 operates simultaneously at both 850 and 450 µm. The POL-2 instrument is currently commissioned at 850 µm only.

Polarisation

In polarimetric terms light is conventionally described by the four Stokes parameters: $I$, $Q$, $U$ and $V$.

1450 µm data can in fact be processed using the same commands as 850 µm data, but the noise levels and other possible artefacts have not yet been fully characterised. Note, the default pixel size at 450 is 4 arc-seconds
$I$ is the total intensity; $Q$ is the radiation linearly polarised in the direction parallel or perpendicular to the reference plane. $U$ is the radiation linearly polarised in the directions $45^\circ$ to the reference plane; and $V$ is the circularly polarised radiation.

POL-2 is designed to characterise linear polarisation. The $V$ parameter, consequently, is not discussed further with the focus on $I$, $Q$ and $U$.

The linear Polarised Intensity ($I_p$) and polarisation angle ($\theta$) can be described as:

\[ I_p = \sqrt{Q^2 + U^2} \]  
\[ \theta = 0.5 \arctan(U/Q) \]  

with $Q$ and $U$ related to the polarisation angle and the polarised intensity by:

\[ Q = I_p \cos(2\theta) \]  
\[ U = I_p \sin(2\theta) \]  

where

\[ Q = Q_m - I.p_q \]  
\[ U = U_m - I.p_u \]  

where $Q_m$ and $U_m$ are the measured values of $Q$ and $U$. $I$ is the astronomical total intensity. $IP$ is the instrumental polarisation. The IP affects both $Q$ (via $ip_q$) and $U$ (via $ip_u$).

**How POL-2 works**

POL-2 is located in front of the window to the SCUBA-2 instrument (as is seen in Figure 2.1), and covers the full field of view of SCUBA-2. The POL-2 polarimeter uses three optical components that cover the full field of SCUBA-2:

1. a wire-grid polariser used as a calibrator (only included in the beam for test purposes),
2. a Half-Wave Plate (HWP), and
3. a second wire-grid polariser used as an analyser.

These components can be seen in Figure 2.1. A schematic of POL-2 is given in Figure 2.2.

Rotating the HWP rotates any linearly polarised component of incoming radiation. The HWP rotates this incoming linear polarisation with twice the speed of the HWP angle ($\delta$) producing the effective analyser position ($\phi$ - as defined in the POLPACK documentation), such that:

\[ \phi = 2\delta \]  

The rotating linearly polarised component is transmitted or reflected by the grid, causing a modulation in the transmitted intensity. The amplitude of the polarised component transmitted by the polariser is $\sim \cos(\phi)$ while the power is $\sim \cos^2(\phi)$.

The radiation passing through the polarimeter is detected by SCUBA-2. The detected intensity ($I_{\text{detector}}$) is a combination of both the unpolarised intensity ($I_{\text{unpolarised}}$) and the linearly polarised intensity ($I_p$). This detected intensity can be described by:

\[ I = I_{\text{unpolarised}} + I_p \]  

---

\(^2\)The total intensity of the source, $I$, is $I_{\text{unpolarised}} + I_p$. 
Figure 2.2: The main optical components in a typical single-beam imaging polarimeter such as POL-2 (taken from SUN/223).

Figure 2.3: Left: If there was a single rotating analyser this would be the resulting curve of the power transmitted of the linearly polarised component. Right: With the HWP the linearly polarised component is rotated at twice the speed. It may be useful to remind the reader of the trigonometric identity: 
\[ \cos^2 x = 0.5(1+\cos(2x)) \]

with the above equation being in terms of the effective analyser angle, \( \phi \) and the angle of the polarisation (\( \theta \)). This can also be expressed in terms of the the HWP angle (\( \delta \)).

\[
I_{\text{detected}} = \frac{I_{\text{unpolarised}}}{2} + I_p \cdot \left( \frac{1 + \cos(2\phi - 2\delta)}{2} \right)
\] (2.8)
Figure 2.4: The incoming polarised radiation (with a polarised angle, $\theta$, of zero) is attenuated by the HWP. The HWP rotates at 2Hz (through $2\pi$) so we see the signal is modulated at 8Hz as the instrument scans at $8''/s$.

$$I_{\text{detected}} = \frac{I_{\text{unpolarised}}}{2} + I_p \cdot \left(\frac{1 + \cos(4\delta - 2\theta)}{2}\right)$$  \hspace{1cm} (2.9)

The Half-Wave Plate

As described in the POL-2 commissioning document the HWP is constructed from five individual synthetic sapphire layers approximately 0.9 mm thick and 200 mm in diameter. The transmission properties of sapphire are generally good at the SCUBA-2 wavelengths but are dependent on the thickness and ambient temperature. The total effective transmission of the HWP integrated across the 850 and 450 µm filter bands are about 86% and 57% respectively (Savini et al. 2009 - insert full reference).

The HWP rotates the incoming linear polarisation with twice the speed of the wave plate angle. The HWP is typically rotated at 2 Hz, providing a fast modulation of any linear polarisation by 8 Hz (see Equation 2.9). The data acquisition rate is $\sim$175 Hz, yielding 20 samples per cycle. The atmosphere is stable on the order of 2 Hz and can be removed.

2.2 Instrumental Polarisation

At the angular resolution of JCMT, planets such as Uranus should appear as unpolarised point sources. In practice, however, POL-2 observations of such sources exhibit a measurable level of polarisation – albeit typically less than 1.5% at 850 µm. This is evidence that some part of the incoming astronomical radiation
is being partially polarised by one or more of the components of the telescope/POL-2/SCUBA-2 that are in the light path. This polarisation is referred to as "Instrumental Polarisation" (IP).

In order to establish the true $Q$ and $U$ from an astronomical source, it is necessary to correct for this effect. For the case of a low degree of polarisation in the incoming radiation and a low degree of IP, the following is a good approximation for correcting the measurement for the effects of the IP:

$$Q = Q_m - I.ip_q$$  \hspace{1cm} (2.10)

$$U = U_m - I.ip_u$$  \hspace{1cm} (2.11)

where $Q_m$ and $U_m$ are the measured values for a single bolometer sample at some point on the sky. $Q$ and $U$ are the true (corrected) values, $I$ is the astronomical total intensity at the same point on the sky (i.e. the total intensity after removal of the sky and electronic backgrounds) and $ip_q$ and $ip_u$ are factors that may vary slowly with focal plane position and/or azimuth and elevation.

IP correction of a POL-2 map therefore requires a total intensity map of the same area of the sky to be available. This total intensity map is referred to as the IP reference map.

Whilst flat mirrors or surfaces will produce a small, constant polarisation across the beam, curved mirrors and other structures (for example the secondary mirror supports) will produce more complex polarisation effects, and these may distort the beam shape. Side-lobes can often show up with strong (typically 10-20%) polarisation but these effects are usually far from the main-beam. Calculations of typical antenna patterns for symmetrical Cassegrain antennas have not predicted strong polarisation in the main beam.

The JCMT IP footprint is stronger than might be expected from the above considerations above (though typically less than 1.5% of the total intensity), and has the following distinctive features:
(1) the polarisation intensity is elevation dependent,
(2) there is ellipticity of the beam and it is elevation dependent, and
(3) the beam is elongated in the horizontal direction.

The dominant source of IP at the JCMT is the woven Goretex membrane, used as a wind blind. This membrane introduces both losses and polarisation. This effect is elevation dependent.

2.3 Observing mode

The standard POL-2 observing mode, POLCV_DAISY, is a “scan and spin” mode, in which the telescope is moving continuously in a Daisy-type pattern while the HWP spins.

The POLCV_DAISY scan mode is similar to the established Daisy scan mode routinely used for non-polarimetric SCUBA-2 observations of point-like or compact sources. However, it is slightly altered to allow for a slower telescope scanning speed.

The telescope must scan slowly enough to obtain sufficient data at each point on the sky to allow good Q and U values to be determined. The current commissioned scan pattern has a size of 200″ and a scan speed of 8″/s. The data reduction splits the data stream into short segments and determines a pair of Q and U values from each segment.

The length of each such data segment is the time it takes the telescope to traverse a pixel in the generated map. With the current scanning parameters this is 0.5 and 0.25 seconds for 850 and 450 µm, respectively. The modulation generated by any polarisation is 8 Hz at the current HWP rotation speed (2 Hz).

The standard POLCV_DAISY scan parameters are given in Table 2.1 and shown in Figure 2.7.

2.4 The raw data

SCUBA-2 is the detector for POL-2, and as such, the raw data format of POL-2 data is the same as a typical SCUBA-2 observation. The sequence for both observations is:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half-wave plate rotation frequency</td>
<td>2 Hz</td>
</tr>
<tr>
<td>Antenna scanning speed</td>
<td>8''/s</td>
</tr>
<tr>
<td>R₀ (map pattern radius)†</td>
<td>133''</td>
</tr>
<tr>
<td>Rₜ (turn radius)</td>
<td>99''</td>
</tr>
<tr>
<td>Rₐ (nominal avoidance radius)</td>
<td>77''</td>
</tr>
</tbody>
</table>

Table 2.1: The scan parameters used in the POLCV_DAISY mode. †This radius is not the size of the resulting map.

Figure 2.7: Detail of POLCV_DAISY. R₀ is the map pattern radius, Rₜ the turn radius, and Rₐ is the nominal avoidance radius. For more details see Table 2.1.

- (1) dark noise,
- (2) flat-field,
- (3) science scans, and
- (4) flat-field.

The SEQ_TYPE keyword in the FITS header may be used to identify the nature of each scan. When you access raw data from the CADC archive [http://www3.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/jcmt/](http://www3.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/jcmt/) you will get all of the files listed above.

Critically the INBEAM keyword in the FITS header may be used to identify if POL-2 is in the beam, and hence differentiate between SCUBA-2 and POL-2 observations.

Shown below is an incomplete list of the raw files for a single sub-array (in this case s8a) for a short POL-2 observation. The first and last scans are the flat-field observations, which occur after the shutter opens to the sky at the start of the observation and closes at the end (note the identical file size); all of the scans in between are science scans.

```
% ls -lh /jcmtdata/raw/scuba2/s8a/20160112/00056
```
Tip

Use the KAPPA command fitslist to see all FITS headers in a particular NDF. To obtain a specific header simply use the command fitsval:

```
% fitsval s8a20160112_00056_0001.sdf INBEAM pol
```

The FITS header information may also be viewed via the GAIA View / FITS header drop-down menu option.

The SCUBA-2 data acquisition (DA) system writes out a data file every 30 seconds; each of which contains 22 MB of data. The only exception is the final science scan which will usually be smaller (7.9 MB in the example above), typically requiring less than 30 seconds of data to complete the observation.

Note: All of these files are written out eight times, once for each of the eight sub-arrays. It should also be noted that the POL-2 instrument has not yet been fully released from commissioning at 450 \( \mu m \) (a “shared risk” approach is currently in use).

The main data array in each NDF is a cube, with the first two dimensions corresponding to bolometer columns and rows within a sub-array, and the third dimension corresponding to time slice index (sampled at roughly 200 Hz).

A standardised file naming scheme is used in which each file name starts with the sub-array name, followed by the UT date of the observation in the format yyyyMMdd, followed by a five-digit observation number, followed by the sub-scan number. The name ends with the standard suffix .sdf used by all Starlink NDF data files. For instance, the files listed above hold data from the s8a sub-array for Observation 34 taken on 12\(^{th}\) January 2016.

Units/Calibration

Raw POL-2 data come in uncalibrated units. The first calibration step is to scale the raw data to units of pico Watts (pW) by applying the flat-field solution. This step is performed internally by the SMURF command calcqu – used to calculate I, Q and U time-streams from the raw data – but can be done manually when examining the raw data.

If the purpose of a given POL-2 observation is to determine the percentage polarisations or vector angles within a source/region of interest then the data may remain in pW. On the other hand, if the purpose is to establish the absolute polarised intensities then a value for the Flux Conversion Factor (FCF) is required. The resulting map may have the FCF applied to convert it into units of janskys. As is recommended with SCUBA-2 observing, it is advisable to check that the FCF value applied to the data is sensible (and must be done manually). For more details see Chapter 4.5.
Chapter 3
POL-2 Data Reduction – The Theory

3.1 The Data Flow

POL-2 data reduction is an involved process for which a broad overview is first presented here before the specific details are discussed. It is noted that this same procedure is used irrespective of whether a single or multiple observations are to be reduced.

The data reduction process can be broken down into three main stages – referred to as “Run 1”, “Run 2” and “Run 3” in Figure 3.1.

Step 1

The initial step of the process (see Run 1 in Figure 3.1) creates a preliminary co-added total intensity (I) map from the raw data files for all observations provided to the reduction routine (see Chapter 4).

The analysed intensity values in the raw data time-streams are first converted into Q, U and I time-streams using the SMURF:calcqu command (these are stored for future use in the directory qudata, specified by the QUDIR parameter in the example command below).

The SMURF:makemap command is then used to create a separate map from the I time-stream for each observation, using SNR-based “auto-masking” to define the background regions that are to be set to zero at the end of each iteration. This step uses a PCA threshold of 50 (see Section 3.4 for more details).

\[ pca\_pcathresh = -50 \]

These maps are stored for future use in the directory maps, specified by the MAPDIR parameter. Each map has a name of the form:

\[ \text{<UT\_DATE>_<OBS\_NUM>_<CHUNK\_NUM>\_imap.sdf} \]

where \( \text{<CHUNK\_NUM>} \) indicates the raw data file at the start of the contiguous chunk of data used to create the map, and is usually 0003.

A co-add is then formed by adding all these maps together. Each individual map is then compared to the co-add in order to determine a pointing correction to be applied to the observation in future. These corrections are stored in the FITS header of the individual maps.

Step 2

In the second step of the process (see Run 2 in Figure 3.1) an improved I map is produced. These improvements come from

1. applying the pointing corrections determined in Step 1
2. the use of an increased number of PCA components (\( pca\_pcathresh=-150 \))
3. using a single fixed mask for all observations. The mask is determined from the preliminary co-add I map and thus includes fainter structure than would be used if the mask was based on only one observation.
Figure 3.1: The data flow of the POL-2 data reduction method is presented. In this example, three POL-2 observations are reduced and combined in various stages and combination to produce I, Q and U maps and a vector catalogue.

Dashed lines are time series
Solid lines are maps
Yellow lines refer to Q
Green lines refer to U
Black lines refer to I
Heavy blue lines refer to AST & PCA masks
Heavy black line is the IP reference map
Step 3

In the third step of the reduction process (see Run 3 in Figure 3.1), both the Q and U maps are produced. The production of the Q and U maps requires the Q and U time-series data (produced in Step 1), the final I map (produced in Step 2) and the output masks (also produced in Step 2). Once the Q and U maps are produced a final vector catalogue is created.

3.2 MAKEMAP

The POL-2 data reduction builds upon the existing SCUBA-2 Dynamic Iterative Map-Maker, hereafter just referred to as the map-maker. This is the tool used to produce SCUBA-2 maps, and is invoked by the SMURF makemap command. It performs some pre-processing steps to clean the data, solves for multiple signal components using an iterative algorithm, and bins the resulting time-series data to produce a final science map.

In pol2map the map-maker is used in conjunction with calcqu (see Section 3.3) to produce maps of Q and U, as well as I.

3.3 CALCQU

In addition to the POL-2 data reduction building on the existing SCUBA-2 map-maker, pol2map also relies on the SMURF command CALQU. This calcqu tool creates time series holding Q and U values from a set of POL-2 time series holding raw data values. The supplied time-series data files are first flat-fielded, cleaned and concatenated, before being used to create the Q and U values. The Q and U time-series are down-sampled to 2Hz (i.e. they contain two Q or U samples per second), and are chosen to minimise the sum of the squared residuals between the measured raw data values and the expected values given by Equation 2.9.

3.4 PCA

One difference between the reduction of SCUBA-2 data and POL-2 data is the method used to remove the sky background. The sky background is usually very large compared with the astronomical signal, and both are subject to the same form of instrumental polarisation (IP – see Section 2.2). This IP acting on the high sky background values causes high background values in the Q and U maps. However, there is evidence that the IP is not constant across the focal plane, resulting in spatial variations in the background of the Q and U maps.

For non-POL-2 data, the background is removing using a simple common-mode model, in which the mean of the bolometer values is found at each time slice and is then removed from the individual bolometer values. This ignores any spatial variations in the background and so fails to remove the background properly in POL-2 Q and U maps.

To fix this, a second stage of background removal is used when processing POL-2 data, following the initial common-mode removal. This second stage is based upon a Principal Component Analysis (PCA) of the 1280 time-streams in each sub-array (the Q and U data are processed separately). The PCA process identifies the strongest time-dependent components that are present within multiple bolometers. These components are assumed to represent the spatially varying background signal and are removed, leaving
just the astronomical signal. You may specify the number of components to remove, via a makemap configuration parameter called `pca.pcthresh` although `pol2map`, the reduction command for POL-2 data, provides suitable defaults for this parameter.

- first stage uses `pca.pcthresh = -50`
- second stage uses `pca.pcthresh = -150`

On each `makemap` iteration, the PCA process removes the background (thus reducing the noise in the map) but also removes some of the astronomical signal. The amount of astronomical signal removed will be greater for larger values of `pca.pcthresh`. However, this astronomical signal is still present in the original time-series data and so can be recovered if sufficient `makemap` iterations are performed. In other words, using larger values of `pca.pcthresh` slows down the rate at which astronomical signal is transferred from the time-series data to the map, thus increasing the number of iterations required to recover the full astronomical signal in the map.

Spatial variations in the sky background may also be present in non-POL-2 data, but at a lower level. For a discussion of why PCA is not routinely run on non-polarimetric SCUBA-2 data, see Appendix A.

### 3.5 Masking

A mask is a two-dimensional array which has the same shape and size as the final map, and which is used to indicate where the source is expected to fall within the map. ‘Bad’ pixel values within a mask indicate background pixels, and ‘good’ pixel values indicate source pixels. Masks are used for two main purposes.

1. They prevent the growth of gradients and other artificial large scale structures within the map. For this purpose, the astronomical signal at all background pixels defined by the mask is forced to zero at the end of each iteration within `makemap` (except for the final iteration).
2. They prevent bright sources polluting the evaluation of the various noise models (PCA, COM, FLT) used within `makemap`. Source pixels are excluded from the calculation of these models.

The `pol2map` script uses different masks for these two purposes – the “AST” mask and the “PCA” mask. The PCA mask is in general less extensive than the AST mask, with the source areas being restricted to the brighter inner regions. Each of these two masks can either be generated automatically within `pol2map`, or be specified by a fixed external NDF.

### 3.6 Tailoring a reduction

**Variances between POL-2 maps**

`MAPVAR` is a `pol2map` parameter that controls how the variances in the co-added I, Q and U maps are formed.

If `MAPVAR` is set `TRUE`, the variances in the co-added I, Q and U maps are formed from the spread of pixel data values between the individual observation maps. If `MAPVAR` is `FALSE` (the default), the variances in the co-added maps are formed by propagating the pixel variance values created by `makemap` from the individual observation maps (these are based on the spread of I, Q or U values that fall in each pixel).
Use MAPVAR=TRUE only if enough observations are available to make the variances between them meaningful. A general lower limit on its value is difficult to define, but is advised a minimum of 10 observations.

If a test of the effect of this option is required on a field for which the I, Q and U maps from a set of individual observations are already available, the following may be done:

```bash
% pol2map in=maps/\* iout=imapvar qout=qmapvar uout=umapvar mapvar=yes \
ipcor=no cat=cat_mapvar debias=yes
```

assuming that the I, Q and U maps are in directory maps. The variances in imapvar.sdf, qmapvar.sdf and umapvar.sdf will be calculated using the new method, and these variances will then be used to form the errors in the cat_mapvar.FIT catalogue.

In general, within the source regions the variances created using MAPVAR=TRUE will be larger than those created using MAPVAR=FALSE (within background regions there should be little difference). This is partly caused by residual uncorrected pointing errors, which have a particularly large effect near bright point sources if MAPVAR=TRUE.

It is also partly caused by intrinsic instabilities within the iterative map-making algorithm, which allow low-level artificial extended structures to develop within the source regions defined by the AST mask. Such artificial structures will vary from observation to observation and so will contribute to the variances calculated using MAPVAR=TRUE.

Two options are provided by pol2map that may be useful in reducing the larger than expected dispersion between maps made from different observations:

1. Setting the parameter OBSWEIGHT=TRUE when running pol2map will cause each observation to be assigned a separate weight, which will be used when forming the coadd of all observations. This will affect both the data values and the variances in the resulting coadd. The purpose of these weights is to down-weight observations that produce maps that vary dissimilar to maps made from the other observations.

   Without this parameter setting, the coadd is formed using weights equal to the reciprocal of the pixel variance values in each individual observation’s map. As mentioned above, these pixel variance values can sometimes seriously under-estimate the dispersion between observations. For instance, observations that are clearly bad (out of focus for instance) can have relatively low pixel variance values, and thus get included with high weight in the coadd.

   If the OBSWEIGHT parameter is set TRUE, each observation is given an additional weight that is used to factor the per-pixel weights derived from the pixel variance values, in order to down-weight observations that are clearly bad. To form these weights, an initial coadd is formed using equal weights for all observations. The maps made from the individual observations are then compared to this initial coadd, and each observation is assigned a weight equal to the reciprocal of the mean squared residual between the individual observation’s map and the initial coadd (any required pointing correction is applied to the individual observation map before forming these residuals). The calculation of the mean squared residual is limited to those pixels inside the AST mask (i.e. source pixels). The weights derived in this manner are normalised to have a median value of 1.0, and any normalised weights larger than 1.0 are reduced to 1.0. An improved coadd is then formed using these observation weights.

   Another iteration is then performed in which individual maps are compared with this improved coadd and new weights are derived. This iterative process continues until the typical error in the middle of the coadd stops falling significantly.

2. Setting the parameter SKYLOOP=TRUE when running pol2map will cause maps to be made using the SMURF:skyloop command instead of makemap. In the context of the skyloop documentation, one “chunk” of data usually corresponds to a single observation.

   A single invocation of skyloop creates an I, Q or U map from all supplied observations, using a method that attempt to minimise the intrinsic instabilities of the map-making algorithm within the
AST mask. Note, convergence can require a significantly greater number of iterations when using skyloop than when using makemap. Also, skyloop requires much more disk space than makemap.

The skyloop command combines all observations together at each iteration of the map-making algorithm. Since the spurious large-scale structures created at each iteration are independent of each other, taking the mean of the maps after each iteration reduces the level of such structures, and prevents them growing in amplitude on successive iterations due to the instability in the map-making algorithm.

The above two methods can be used together by supplying TRUE values for both OBSWEIGHT and SKYLOOP. Using skyloop to produce the co-add usually requires more time and disk space. Therefore it is usually advisable to restrict its usage within pol2map to the external-masking phase that produces the final required maps - that is, Steps 2 and 3 as described above. The OBSWEIGHT parameter can generally be used on all steps as it adds little to the time taken to run pol2map.

The following panels show the effects of using SKYLOOP and OBSWEIGHT on a total intensity mosaic of 21 observations. All data value maps are shown with a single scaling, and all standard deviation maps are shown with a single scaling (different to the scaling for the data value maps):
For comparison, below are the equivalent auto-masked maps made by Step 1:
Chapter 4
POL-2 Data Reduction – Running pol2map

The previous chapter, Chapter 3, described how pol2map produces I, Q and U maps from raw POL-2 data. It showed that this reduction process – which uses pol2map – comprises three steps.

As with the other Python scripts in SMURF, you can get more information about the available parameters by doing either:

```bash
% pol2map --help
```
or

```bash
% smurfhelp pol2map
```

4.1 How to use pol2map

Before running pol2map directly, it is necessary to ensure that the Starlink environment has been initialised and the SMURF package started (see Section 1.3.2 and Section 1.3.3).

This chapter describes how to run pol2map firstly to produce an initial I map and then again to produce the final I, Q, U maps and vector catalogue as described in Section 4.2.

To run pol2map, values should normally be supplied for the following command-line parameters:

1. **IN**
   - A list of input NDFs containing raw POL-2 data. There are many ways in which the list of files can be supplied, as described in Section ‘Specifying Groups of Objects’ in SUN/95. The easiest is to create a simple text file containing the names of the raw data files – one per line – and then supply the name of the text file, preceded by an up-caret character (^), as the value for parameter IN. Note, the names of the raw data files can contain wildcards such as “*” and “?”.

2. **IOUT**
   - The name of the NDF in which to store the total intensity (I) map (in pW) incorporating all supplied observations. The supplied file name should either have a file type of .sdf, or no file type at all (in which case .sdf will be appended to the supplied value). Any existing file with the same name will be overwritten.

3. **QOUT**
   - The output NDF in which to return the Q map including all supplied observations. This will be in units of pW. Null (!) should be supplied if no Q map is required.

4. **UOUT**
   - The output NDF in which to return the U map including all supplied observations. This will be in units of pW. Null (!) should be supplied if no U map is required.

Note the distinction between “command-line parameters” that are supplied on the pol2map command line, and “configuration parameters” that are specified within a configuration file. Values for all configuration parameters are obtained using a single command-line parameter called CONFIG.
MAPDIR The name of a directory in which to put the Q, U and I maps made from each individual observation supplied via IN, before co-adding them. If null (!) is supplied, the new maps are placed in the same temporary directory (chosen automatically) as all the other intermediate files and so will be deleted when the script exists (unless Parameter RETAIN is set TRUE). Note, these maps are always in units of pW. Each one will contain FITS headers specifying the pointing corrections needed to align the map with the reference map. \[1\]

QUDIR The name of a directory in which to put the Q, U and I time series generated by SMURF calcqu, prior to generating maps from them. If null (!) is supplied, they are placed in the same temporary directory as all the other intermediate files and so will be deleted when the script exists (unless Parameter RETAIN is set TRUE). \[1\]

Some additional command-line parameters are required when pol2map is used for the second time – as discussed in Section 4.3 – to produce the final I, Q, U maps and vector catalogue\[2\]:

CAT The output FITS vector catalogue. No catalogue is created if null (!) is supplied. Note – by default the Q, U and Ip values in this catalogue will be in units of mJy/beam. \[!\]

MASK Specifies the type of masking to be used within makemap (the same type of masking is used to create all three maps – I, Q and U).

MASKOUT1 If a non-null value is supplied for MASKOUT1, it specifies the NDF in which to store the AST mask created from the NDF specified by Parameter MASK. Only used if an NDF is supplied for Parameter MASK. \[!\]

MASKOUT2 If a non-null value is supplied for MASKOUT2, it specifies the NDF in which to store the PCA mask created from the NDF specified by Parameter MASK. Only used if an NDF is supplied for Parameter MASK. \[!\]

IPREF The total intensity map to be used for IP correction. The map must be in units of pW. If the same value is supplied for both IOUT and IPREF, the output I map will be used for IP correction. \[!\]

DEBIAS TRUE if a correction for statistical bias is to be made to percentage polarisation and polarised intensity in the output vector catalogue specified by Parameter CAT. \[FALSE\]

The pol2map command provides many other parameters that can be used to modify its behaviour in various ways. To see a full list, do this.

\% pol2map --help

4.2 pol2map – producing the initial I map.

As discussed in Chapter \[3\] pol2map must first be run on the raw data to produce an initial I map. In this first step:

\% pol2map in="myfiles.list" iout=iauto qout=! uout=! mapdir=maps qudir=qudata

\[2\]This second usage of pol2map includes both "Run 2" and "Run 3" in Figure 3.1
the file `myfiles.lis` contains a list of the raw data files to be included in the map, and could (for instance) look like this.

```bash
% cat myfiles.lis
/jcmtdata/raw/scuba2/s8a/20160125/00043/*
/jcmtdata/raw/scuba2/s8b/20160125/00043/*
/jcmtdata/raw/scuba2/s8c/20160125/00043/*
/jcmtdata/raw/scuba2/s8d/20160125/00043/*
```

This uses all available data for all four 850 \( \mu \)m sub-arrays, for Observation 43 taken on 25th January 2016. In addition, the data used in this example also comes from Observations 56 and 59 taken on January 11th 2016 (UT).

Tip

An up-caret (^) is required any time you are reading in a group text file in Starlink. For the map-maker this includes the configuration file (a group of configuration parameters) and the list of input files (a group of NDFs e.g. `in=` ^ `myfiles.lis`).

To check if the files are POL-2 files, run the `pol2check` command.

```bash
% pol2check ^myfiles.list
```

Note that `qout` and `uout` are set to null values as no Q or U maps are required to be produced during this initial step 1 reduction stage.

The following shows the output from running this initial `pol2map` command.

Logging to file `pol2map.log`
Calculating Q, U and I time streams from raw analysed intensity data...
1/3: Processing 116 raw data files from observation 20160125_00043 ...
2/3: Processing 116 raw data files from observation 20160112_00059 ...
3/3: Processing 116 raw data files from observation 20160112_00056 ...

>>>> Making I map from 20160125_00043_0003...

>>>> Making I map from 20160112_00056_0003...

>>>> Making I map from 20160112_00059_0003...

Co-adding I maps from all observations:

20160125_00043_0003: Storing pointing corrections of (0.0,0.0) arc-seconds for future use
20160112_00056_0003: Storing pointing corrections of (1.9,2.8) arc-seconds for future use
20160112_00059_0003: Storing pointing corrections of (2.1,2.4) arc-seconds for future use

---

3The input files should all be for a single waveband from one or more POL-2 observations – do not mix files from different wavebands and/or astronomical regions
Figure 4.1: The I map, iauto.sdf, as viewed with GAIA.

The files and folders produced in this reduction are described below.

- **pol2map.log** A log file containing the output from the various SMURF, KAPPA and POLPACK commands run as part of the pol2map command (pol2map is a Python script, which runs various other Starlink tasks behind the scenes to perform the bulk of the work).

- **qudata/** A folder containing the I, Q and U time-series data for each sub array for each observation (these are produced by calcqu (see Section Chapter 3.3).

- **maps/** A folder containing the individual I maps from each separate observation. These will have names that end with _imap.sdf.

- **iauto.sdf** Output total intensity map (the term “auto” is used to indicate that it was created using an automatically generated AST mask).

The output I map, iauto.sdf, can be opened and viewed with GAIA.

The maps folder contains the individual I maps from each separate observation:

```
20160112_00056_0003_imap.sdf  20160112_00059_0003_imap.sdf  20160125_00043_0003_imap.sdf
```

and the qudata folder contains these files.
4.3 pol2map – producing the I, Q, U maps and catalogue

As discussed in Chapter 3, the I map output from the initial run of pol2map is used to derive the final I, Q and U maps. If requested, a vector catalogue is also produced.

The second and third steps of the POL-2 data reduction process can be run via a single command.

```
% pol2map in=qudata/* iout=iext qout=qext uout=uext mapdir=maps mask=iauto 
   maskout1=astmask maskout2=pcamask ipref=iext cat=mycat debias=yes
```

The following shows the output from running this second pol2map command. First, pol2map produces new I maps for each map, correcting the position using the correction stored in the old I map, and then coadds all the observations.

```
Logging to file pol2map.log
(existing file pol2map.log moved to pol2map.log.1)

Masking will be based on SNR values in 'iauto'.

>>>> Making I map from 20160112_00056_0003...

Using pre-calculated pointing corrections of (1.9,2.8) arc-seconds

>>>> Making I map from 20160125_00043_0003...

Using pre-calculated pointing corrections of (0.0,0.0) arc-seconds

>>>> Making I map from 20160112_00059_0003...

Using pre-calculated pointing corrections of (2.1,2.4) arc-seconds

Coadding I maps from all observations:

As pol2map continues, the Q and U maps are produced, again with pointing corrections. This is followed by the creation of the output vector catalogue.

```
>>>> Making Q map from 20160112_00056_0003...

Using pre-calculated pointing corrections of (1.9,2.8) arc-seconds
```

4This correction is found by aligning the old I map with the iauto.sdf map.
Making Q map from 20160125_00043_0003...
Using pre-calculated pointing corrections of (0.0,0.0) arc-seconds

Making Q map from 20160112_00059_0003...
Using pre-calculated pointing corrections of (2.1,2.4) arc-seconds

Coadding Q maps from all observations:

Making U map from 20160112_00056_0003...
Using pre-calculated pointing corrections of (1.9,2.8) arc-seconds

Making U map from 20160125_00043_0003...
Using pre-calculated pointing corrections of (0.0,0.0) arc-seconds

Making U map from 20160112_00059_0003...
Using pre-calculated pointing corrections of (2.1,2.4) arc-seconds

Coadding U maps from all observations:
Creating the output catalogue: 'mycat'...

45604 vectors written to the output catalogue.

The output of this final run of pol2map is as follows.

pol2map.log  A log file containing the output from the pol2map command. Note previous log files are moved to a new name such as pol2map.log.1.
astmask.sdf  The AST mask used in the creation of the final I, Q and U maps.
pcamask.sdf  The PCA mask used in the creation of the final I, Q and U maps.
iext.sdf     The total intensity image, created using the external AST and PCA masks described above.
qext.sdf     The Q map (i.e. the intensity of the radiation linearly polarised in the direction parallel or perpendicular to the reference plane), created using an external AST and PCA mask.

maps/        A folder containing the individual I, Q and U maps from each separate observation. These will have names that end with _Imap.sdf, _Qmap.sdf or _Umap.sdf.
uext.sdf     The U map (i.e. the intensity of the radiation linearly polarised in the direction ±45° to the reference plane).

mycat.FIT    The output vector catalogue containing a range of values derived by pol2map for each pixel contained within the I map.

The maps folder now contains individual Q and U maps alongside the existing I maps listed below.

20160112_00056_0003_Imap.sdf  20160112_00059_0003_Imap.sdf  20160125_00043_0003_Imap.sdf
20160112_00056_0003_Qmap.sdf  20160112_00059_0003_Qmap.sdf  20160125_00043_0003_Qmap.sdf
20160112_00056_0003_Umap.sdf  20160112_00059_0003_Umap.sdf  20160125_00043_0003_Umap.sdf
20160112_00056_0003_imap.sdf  20160112_00059_0003_imap.sdf  20160125_00043_0003_imap.sdf
4.4 Output vectors from pol2map

The output vector catalogue contains a range of values derived by pol2map for each pixel contained within the I map. Intensity values and errors in the catalogue are expressed in units of mJy/beam. If desired it is possible to switch the catalogue to units of pW by using Jy=no on the pol2map command line. The columns are listed below.

- **X**: Pixel coordinate at the centre of the pixel
- **Y**: Pixel coordinate at the centre of the pixel
- **RA**: RA coordinate at the centre of the pixel
\[
\begin{array}{ll}
\text{Dec} & \text{Dec coordinate at the centre of the pixel} \\
I & \text{Total intensity} \\
DI & \text{Error in I} \\
Q & \text{Stokes Q parameter} \\
DQ & \text{Error in Q} \\
U & \text{Stokes U parameter} \\
DU & \text{Error in U} \\
P & \text{Percentage polarisation} \\
DP & \text{Error in P} \\
\text{ANG} & \text{Angle of polarisation} \\
D\text{ANG} & \text{Error in ANG} \\
\text{PI} & \text{Polarised intensity} (I_p) \\
\text{DPI} & \text{Error in polarised intensity}
\end{array}
\]

4.5 **POL-2 FCFs**

Inserting POL-2 in front of SCUBA-2 reduces the throughput to SCUBA-2. POL-2 is not a perfect polarimeter. Its wire grid absorbs and scatters incoming signal so the modulation amplitude is lower than for a perfect polarimeter. In addition cross polarization and depolarization decreases the modulation amplitude without decreasing the power in the transmitted signal. The first type of inefficiencies can be measured by comparing normal SCUBA-2 maps with and without the polarimeter inserted. Such observations have been done on Uranus, Mars and Jupiter. The second type of losses can be measured with a source of know polarization.

To convert POL-2 data to astronomical units such as mJy/beam a Flux Conversion Factor, FCF, must be applied to the data. For POL-2 the FCFs are quoted in terms of the SCUBA-2 FCF.

At 850 µm and 450 µm the FCFs for POL-2 are found to be a factor of 1.35 and 1.96 times higher than the standard SCUBA-2 FCF for 850 µm and 450 µm, respectively.

4.6 **Changing pixel size in pol2map**

Inevitably, as with unpolarised SCUBA-2 data reduction, it will probably be necessary for you to tweak the pol2map reduction for specific situations.

The bin size within the final vector catalogue is controlled by the BINSIZE parameter in the SMURF pol2map command.

\[
\% \text{pol2map binsize=12}
\]

Changing the catalogue bin size in this way does not change the pixel size of the maps created pol2map. Instead, the maps are binned up to the requested bin size before the catalogue is created. There is another parameter, called PIXSIZE, which controls the map pixel size, but it is usually advisable to leave this at its default value as the map pixel size can affect the behaviour of the iterative algorithm used to create maps.
Chapter 5

POL-2 Image Display

5.1 GAIA

The Starlink package GAIA can be used to inspect the results of the data reduction. To plot the output vector catalogue onto the final total intensity map first open up the I map in GAIA.

% gaia iext.sdf

In the main GAIA window, select the drop-down menu option Image Analysis / Polarimetry toolbox.... This should launch a new toolbox window entitled GAIA: Polarimetry. From this window, use the drop-down menu option File / Open to load the file mycat.FIT. This should then populate the lower part of the window with the contents of this polarimetry catalogue file. Each of the vectors in this file will be automatically overlaid on the main image window (see Figure 5.1).

In order to filter the number of overlaid vectors down to a more useful number and size, you can use various options in the GAIA: Polarimetry toolbox. First, select the Rendering tab on the left hand side. This will reveal a panel that will indicate which quantities are currently being used for the vector overlays. In this case, the Vector length is taken from the P column of the table, and the Vector angles are taken from the ANG column.

Currently the figure has too many vectors to be scientifically meaningful. To filter out most of the extraneous vectors, click on the Selecting tab, and set the Expression field to be the following:

$I/\Delta I < 10$
Figure 5.2: Left: specifying vectors to display via the expression $I/\text{DI}>10$. This will only plot vectors with an intensity signal-to-noise ratio greater than 10 in GAIA. To ensure that this is specified, ensure you press the carriage return after entering the expression.

Figure 5.3: Left: Selected vectors are marked in blue in this example, Right: after removal of selected vectors all that remains are the vectors on the (zoomed) regions where $I/\text{DI}>10$.

Ensure you press return after entering in the above expression.

The above expression selects the data points in the polarimetry table which have an associated total intensity (Column I) less than 10 times the associated error value for that intensity (Column DI). To remove all of these extraneous vectors, either press control-X or use the drop-down menu option Edit / Cut. This should leave just a small number of vectors clustering around the target object (see Figure 5.2).

Zooming in on the central region of the map, it can already be seen that the level of vector ordering (and hence polarisation) is quite low (see Figure 5.3). If needed it is possible to change the scaling by selecting the Rendering tab in the GAIA: Polarimetry window, and increasing the vector scale.

Finally it is useful for future use (as in the examples in the following sections) to save the final selection of vectors. To save the displayed vectors to a new catalogue, use the drop-down menu File / Save in
the polarimetry toolbox.

5.2 KAPPA and polpack

It is possible to use KAPPA and POLPACK to create POL-2 plots.

```
% kappa
% polpack
```

Note that in the following examples, it will be necessary to ensure that only the vectors to be plotted are included in the file mycat.FIT.

There are two main ways to do this – either by saving the output catalogue from GAIA or using the Starlink package CURSA to manipulate the catalogue you produced. To use CURSA simply run:

```
% cursa
```

then to select the vectors of interest:

```
% catselect catin=mycat.FIT catout=selcat.FIT norejcat seltyp=e "expr='i>10*di'"
```

it is also possible to crop images using catselect, by using the expression command. In this example we use only pixels above $-10$ on the $y$ axis.

```
% catselect catin=mycat.FIT catout=selcat.FIT norejcat seltyp=e "expr='i>10*di and y>-10'"
```

**Tip**

For a better font on PGQLOT PostScript devices, set the following environment variable.

```
setenv PGPLOT_PS_FONT Times
```

For more info, see [http://pipelinesandarchives.blogspot.com/2015/02/better-fonts-in-postscript-output-from.html](http://pipelinesandarchives.blogspot.com/2015/02/better-fonts-in-postscript-output-from.html)

Graphics-related attributes that can be set are described in [SUN95: Descriptions of Plotting Attributes](http://pipelinesandarchives.blogspot.com/2015/02/better-fonts-in-postscript-output-from.html), and the coordinate system attributes that can be set are described in [SUN/95: Descriptions of Frame Attributes](http://pipelinesandarchives.blogspot.com/2015/02/better-fonts-in-postscript-output-from.html).

5.2.1 Example 1 – a vector map with no background

In this section an output file: plot1.pdf is created from the input catalogue mycat.FIT.
Select a higher quality PostScript font (Times New Roman in this case).

```
setenv PGPLOT_PS_FONT Times
```
Select the PostScript graphics device, writing to file `plot1.ps`.

```
gdset plot1.ps/acps
```

For convenience, create a text file holding the main plotting style for `polplot`.

```
% cat sty
colour=black
drawtitle=0
format(1)=hms
format(2)=dms
```

 Likewise, create a text file holding the style for the vector length key.

```
% cat ksty
colour=black
drawtitle=0
```

Plot the vector map (the `vscale` parameter controls the vector scale, and the `keyvec` parameter controls the length of the vector used as the key. There are many other parameters that can be used to control the behaviour of `polplot`—see the \textit{POLPACK} manual (SUN/223).

```
% polplot selcat.FIT style=\textasciitilde sty keystyle=\textasciitilde ksty vscale=20 keyvec=20
```

Convert the map into a PDF file and remove blank margins (if required).

```
% ps2pdf plot1.ps temp.pdf
% pdfcrop temp.pdf plot1.pdf
```

5.2.2 Example 2 – a vector map over a contour map

In this section we create an output file: `plot2.pdf` from the input catalogue `mycat.FIT`.

Select the PostScript graphics device, writing to file `plot2.ps`. Note, in this example we do not assign a value to the `PLOT_PS_FONT` environment variable. This means the resulting plot uses the default PGPLOT fonts rather than the higher quality PostScript fonts used in the previous example.

```
% gdset plot2.ps/acps
```

Set up the main plotting style for `contour` and `polplot`.

```
% cat sty
colour=black
colour(curves)=red
width(curves)=3
drawtitle=0
format(1)=hms
format(2)=dms
```

Produce the contour map as follows.

```
% contour iext\(0^\circ,0^\circ\)\ mode=perc percentiles=\[88,90,92,94,96,98\]\ style=\textasciitilde sty key=no
```
Figure 5.4: Result from Example 1: Producing a vector map with no background using polplot.

Modify the above style for the vector map to produce black vectors.

```
cat vsty
%-sty
colour(curves)=black
```

Set the style for the vector length key.

```
cat ksty
colour=black
width=3
drawtitle=0
```

Plot the vector map over the contour map. The vectors and contours are aligned automatically in sky coordinates.

```
polplot selcat.FIT axes=no clear=no style=~vsty keystyle=~ksty vscale=20 keyvec=20
```

Convert into a PDF file and remove blank margins (if required).

```
pdfcrop temp.pdf plot2.pdf
```
5.2.3 Example 3 – a vector map over an image

In this section we create an output file: plot3.pdf from the input catalogue mycat.FIT. First select the appropriate PostScript device (we use the default PGPLOT fonts again, as in the last example).

% gdset plot3.ps/acps

To ensure a monochrome colour table is used for the image run lutgrey.

% lutgrey

Set the main plotting style for display and polplot:

% cat sty
colour=black
drawtitle=0
format(1)=hms
format(2)=dms

The following function is used to reduce the dynamic range in the map (so that we can see structure in the faint bits without saturating the brightest regions).

maths "'((ia+0.0003)/0.14)**0.2'" ia=iext out=tmp1

Display the image, using a reduced range of colours (pens) so that the darkest regions are grey rather than black. This means the black vectors can still be seen within the dark regions.
Figure 5.6: Result from Example 3: Producing a vector map over a negative image.

% display tmp1(0^-50,0^-50) mode=perc percentiles=[2,98] style='sty \penrange=[0.4,1.0]'

Modify the above style for the vector map to produce wider vectors.

% cat vsty
~sty
width(curves)=3

Set the style for the vector length key.

% cat ksty
colour=black
width=3
drawtitle=0

Plot the vector map over the contour map. The vectors are aligned automatically with the map.

% polplot selcat.FIT axes=no clear=no style='vsty keystyle='ksty vscale=20 keyvec=20

Convert into a PDF file and remove blank margins (if required).

% ps2pdf plot3.ps temp.pdf
% pdfcrop temp.pdf plot3.pdf
5.3 TOPCAT

Catalogues produced by pol2map can be explored using the popular TOPCAT catalogue browser (see http://www.starlink.ac.uk/topcat/). For instance:

```bash
% topcat -f fits mycat.FIT
```

Unlike the other tools described above, TOPCAT cannot visualise the catalogue as a set of vectors. However, it goes well beyond the facilities of the other tools in allowing you to explore correlations between different quantities in the catalogue via two- and three-dimensional scatter plots – see Figure 5.7. It can also be used to cross-correlate two different catalogues, create subsets of the catalogue, create new columns containing related quantities, etc. It also allows the modified catalogue to be saved to a new catalogue file on disk. However, beware that any new catalogue will not contain the WCS information required by other Starlink applications to perform WCS-related operations such as displaying annotated axes and aligning data-sets. However, this WCS information can be copied back into the new catalogue using the polwcscopy command. See Section 6.4.
Chapter 6
POL-2 – Advanced Data Reduction

The `pol2map` tool for reducing POL-2 data was released to the science community for the start of 17B observing. As with all newly commissioned instrumentation the “ideal” reduction has yet to be finalised. This advanced section of the POL-2 data reduction documentation aims to provide you with tools for expanding and examining the POL-2 reduction process further and in more detail.

For further ideas, see Section 3.6.

6.1 Adding new observations

This section describes the six-step process of combining data for one or more new POL-2 observations into existing I, Q and U maps and vector catalogue created by an earlier run of `pol2map`.

(1) Create a text file listing all the existing auto-masked I maps for individual observations stored in the directory specified by Parameter `MAPDIR`, and then add in the raw data files for the new observations. The auto-masked I maps have names that end in `_imap.sdf`.

```
% ls maps/*imap.sdf > infiles.list
% ls rawdata/*.sdf >> infiles.list
```

(2) Create a new auto-masked, co-added I map including the new observation. The `calcqu` and `makemap` commands will be run on the new data and the resulting maps combined with the existing maps derived from the older observations to create the new map.

```
% pol2map in=^infiles iout=iauto_new qout=! uout=! mapdir=maps 
    qudir=qudata
```

(3) A decision needs to be taken whether to re-create all the externally masked maps using external masks defined by the new auto-masked map. This will be the case if the auto-masked map has been changed significantly by the addition of the new observation. To do this, it is necessary to compare the old and new masks. The old masks should have been created earlier using the `MASKOUT1` and `MASKOUT2` parameters (see Step 3 in Section 3). To create the new masks that would be generated from the new auto-masked map, use this command.

```
% pol2map in=^infiles iout=! qout=! uout=! mapdir=maps mask=iauto_new 
    maskout1=astmask_new maskout2=pcamask_new
```

(4) Decide if the addition of the new data has changed the masks significantly. This involves comparing `astmask.sdf` and `astmask_new.sdf` (and also `pcamask.sdf` and `pcamask_new.sdf`).

(5) If the mask has changed significantly and all observations need to be reprocessed using the new mask, remove the existing externally-masked maps so that they will be re-created by the next invocation of `pol2map`. Note – this will increase the length of time taken by Step 6 enormously. Ensure the new auto-masked co-add is used in place of the old one to define any new masks needed in future.
% rm mapdir/*Qmap.sdf mapdir/*Umap.sdf mapdir/*Imap.sdf
% mv iauto.sdf iauto_old.sdf
% mv iauto_new.sdf iauto.sdf

(6) Re-create the necessary externally masked maps and co-adds, and then create the new vector
catalogue.

% pol2map in=qudata/* out=iext_new qout=! uout=! mapdir=maps
   mask=iauto
% pol2map in=qudata/* out=! qout=qext_new uout=uext_new mapdir=maps
   mask=iauto ipref=iext_new cat=mycat_new debias=yes

6.2 Experimenting with pixel sizes

Currently, the default map pixel size is 4'' at both 450 and 450 µm. The pixel size is controlled by the
PIXSIZE parameter in the SMURF pol2map command:

% pol2map pixsize=12

The following four-step example shows how to investigate the impact of changing pixel size. In this
example, we compare 12'' pixels and 7'' pixels.

(1) Begin with an auto-masked total-intensity map from the raw data. For instance:

% pol2map in="myfiles.list" out=iauto12 pixsize=12 qout=! uout=! mapdir=maps12 qudir=qudata

(2) Create AST and PCA masks with 12'' pixels from the iauto12.sdf file.

% pol2map in=qudata/* out=! qout=! uout=! mapdir=maps12 mask=iauto12
   maskout1=astmask12 maskout2=pcamask12

(3) Create masks with 7'' pixels by resampling the 12'' masks created at Step 2. This is done using the
KMPPA sqorst command:

% sqorst mode=pixelscale pixscale="7,7,7E-05" in=astmask12 out=astmask7
% sqorst mode=pixelscale pixscale="7,7,7E-05" in=pcamask12 out=pcamask7

(4) Create the 7'' externally masked I, Q and U maps using the above 7'' masks (note the mask parameter
value is enclosed in single and double quotes).

% pol2map in=qudata/* out=iext7 qout=qext7 uout=uext7 masktype=mask
   mask=""astmask7,pcamask7" mapdir=maps7 ipref=iext7 cat=cat7 debias=yes
6.3 Investigating systematic error in IP

The error on the IP is reported to be of the order of 0.5%. It is possible to investigate the effects of the systematic error in IP by creating maps using the upper and lower limits on the IP value. The makeemap configuration parameter called ipoffset can be used to do such an investigation. To use it, run pol2map twice as follows:

```bash
% pol2map config="ipoffset=-0.25"
% pol2map config="ipoffset=0.25"
```

to produce maps using the upper and lower IP limits (a range of 0.5%). If pol2map has already been run on POL-2 data then a file will already exist that was created using the mean IP (the mean IP is used if ipoffset is omitted from the configuration value, or the configuration parameter itself is omitted).

6.4 Adding WCS information back into a vector catalogue

Vector catalogues produced by pol2map contain information about World Coordinate Systems (WCS) in two different forms:

1. The catalogue contains “RA” and “Dec” columns that hold the sky position (FK5, J2000) of each vector, in radians.

2. The catalogue header contains a Starlink “WCS FrameSet” which defines (amongst other things) the projection from pixel coordinates within the I, Q and U mosaics, to RA and Dec. This FrameSet is used by Starlink software, together with the pixels coordinates stored in the “X” and “Y” columns, to determine the RA and Dec of each vector. The WCS FrameSet also defines the polarimetric reference direction used by the Q, U and ANG values. See “Using World Co-ordinate Systems” within SUN/95 (the KAPPA manual) for more information on the ways in which Starlink software handles WCS information.

Starlink software such as POLPACK, KAPPA, and GAIA rely on the WCS FrameSet for all WCS-related operations (drawing annotated axes, aligning data sets, etc). Thus problems are likely if the WCS FrameSet
is removed from the vector catalogue. This could happen for instance if you use inappropriate software to process an existing catalogue, creating a new output catalogue – the WCS FrameSet may not be copied to the output catalogue, causing subsequent WCS-related operations to fail. It is safe to use POLPACK, KAPPA, GAIA and CURSA as all these packages copy the WCS FrameSet to any new output catalogues. Unfortunately, the popular TOPCAT catalogue browser (see http://www.starlink.ac.uk/topcat/) and the STILTS package (http://www.starlink.ac.uk/stilts/) upon which it is based, do not copy the WCS FrameSet to any output catalogues.

For this reason, POLPACK contains a command that can be used to copy the WCS FrameSet from one catalogue to another. Say for instance you create catalogue mycat.FIT using pol2map, and then use TOPCAT to remove low signal-to-noise vectors, saving the results to a new catalogue called selcat.FIT. The WCS FrameSet will be missing from selcat.FIT, and so we need to copy it back again from the original catalogue mycat.FIT. To do this we use the `polwcscopy` command:

```
% polwcscopy in=selcat ref=mycat out=selcat2
```

This creates a third catalogue selcat2.FIT, which is a copy of selcat.FIT but with WCS inherited from mycat.FIT.
Bibliography


Appendix A
PCA on SCUBA-2 data

This document has outlined the process of reducing POL-2 data and its reliance on PCA during the makemap process. There are two main reasons why running makemap with PCA is not the default method for all data observed using SCUBA-2:

(1) it is very time-consuming to run because of the faster scanning speed and consequent higher sample rate of non-POL-2 data; and

(2) for non-POL-2 data, it produces similar results to running the default makemap with a decreased filter size.

All the relevant tools are still provided however, and so interested users may wish to try using this method and to compare the results.