A User’s Guide to IRIS Data Retrieval, Reduction & Analysis

Release 1.0

Tiago M. D. Pereira, Scott W. McIntosh, Bart De Pontieu, Viggo Hansteen, Mats Carlsson, Paul Boerner, Wei Liu, Milan Gošić, Scott Green

Dec 03, 2023
CONTENTS

1 Introduction .............................. 1
  1.1 About this Guide ........................................... 1
  1.2 IRIS instrumentation ........................................ 1
  1.3 IRIS Data Level Definitions ................................. 4
  1.4 Sample IRIS data ............................................ 5
  1.5 IRIS IDL routines in SSW ..................................... 8
  1.6 IRIS Operations ............................................. 8
  1.7 IRIS Documentation and Links ............................... 10

2 Quickstart .............................. 13
  2.1 IRIS overview ............................................... 13
  2.2 Getting the data ............................................. 14
  2.3 Reading the data in IDL ..................................... 14
  2.4 Data calibration ............................................. 15
  2.5 Data analysis and visualisation ............................. 15

3 IRIS Level 2 Data ...................... 17
  3.1 Structure of IRIS level 2 FITS files ......................... 17
  3.2 Searching and Downloading .................................. 19
  3.3 Browsing Level 2 Data with iris_xfiles ...................... 23
  3.4 Reading Level 2 Data in IDL ................................ 31
  3.5 NUV Data Analysis ........................................... 39
  3.6 FUV Data Analysis ........................................... 41

4 IRIS Level 3 Data ...................... 43
  4.1 Level 3 Data Structure .................................... 43
  4.2 Creating Level 3 Data in IDL ................................. 43
  4.3 Reading Level 3 Data in IDL ................................ 46
  4.4 Browsing Level 3 Data with crispex ......................... 46

5 Calibration of IRIS Observations ...... 61
  5.1 Wavelength Calibration ...................................... 61
  5.2 Radiometric Calibration ..................................... 61
  5.3 Background in FUV data ..................................... 62
  5.4 Coalignment between channels and SJI/spectra ................ 63

6 IRIS data notes ......................... 65
  6.1 Cosmic rays ................................................. 65
  6.2 Particles on slitjaw images .................................. 65
  6.3 CCD camera readout noise ................................... 65
  6.4 Flagging of saturated data ................................... 66

i
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5</td>
<td>Cosmetic finishing in quicklook</td>
<td>66</td>
</tr>
<tr>
<td>7</td>
<td>Tutorials</td>
<td>69</td>
</tr>
<tr>
<td>7.1</td>
<td>IRIS xfiles and CRISPEX</td>
<td>69</td>
</tr>
<tr>
<td>7.2</td>
<td>Mg II Dopplergrams</td>
<td>72</td>
</tr>
<tr>
<td>7.3</td>
<td>Mg II spectral feature identification</td>
<td>78</td>
</tr>
<tr>
<td>7.4</td>
<td>Time series analysis</td>
<td>83</td>
</tr>
<tr>
<td>7.5</td>
<td>Other tutorials</td>
<td>85</td>
</tr>
<tr>
<td>8</td>
<td>Useful codes</td>
<td>89</td>
</tr>
<tr>
<td>8.1</td>
<td>IRIS_getAIAdata</td>
<td>89</td>
</tr>
<tr>
<td>8.2</td>
<td>IRIS file handling routines</td>
<td>91</td>
</tr>
<tr>
<td>8.3</td>
<td>IRIS dustbuster</td>
<td>92</td>
</tr>
<tr>
<td>8.4</td>
<td>IDL Routines for Level 2 Analysis</td>
<td>92</td>
</tr>
</tbody>
</table>
CHAPTER ONE

INTRODUCTION

1.1 About this Guide

This guide is written as a “cookbook” to help novice IRIS users through the steps of how to acquire data, reduce it, and perform some data analysis. It provides an overview of the instrumental capabilities, how the data is organised and calibrated, and how to read and interact with it.

If you are in a hurry or pressed for time, we suggest you go straight to Quickstart. You can then go through the other chapters to get more detail of a particular topic.

This introductory chapter provides an overview of the instrument, data products, operations, and where to find technical documentation. After familiarising themselves with the instrument, we encourage all new users of IRIS data to start with the Level 2 chapter.

This guide is evolving and is frequently updated. It is a collaborative effort and is currently maintained by Tiago Pereira. Much of this guide comes from the tireless work of Scott McIntosh, who steered the guide from the early days of IRIS when the tools and data calibration were changing quickly. Please email astro.uio.no@tiago.pereira (reverse user/domain) with corrections or questions, and we will respond as soon as possible.

This cookbook is a guide for browsing IRIS data before further analysis - a “quicklook” guide. The codes/tools discussed for data analysis will be black boxes to the non-expert user, but those codes are simple in structure and we encourage the user to study them and use them to analyze the IRIS data that they have downloaded. Please keep in mind that there are always assumptions and simplifications made in applying a priori physics to line spectra analysis, e.g., is an optically thin emission line profile best represented by a Gaussian?

1.2 IRIS instrumentation

IRIS is the Interface Region Imaging Spectrograph small explorer (NASA Small Explorer- SMEX). The IRIS investigation combines advanced numerical modeling with a high resolution, high throughput multi-channel UV imaging spectrograph fed by a 20 cm UV telescope. The main science goal of IRIS is to understand how the solar atmosphere is energized. IRIS obtains UV spectra and images in two main passbands around 1400Å and 2800Å at high resolution in space (0.33-0.4”), time (1s) and spectrally (~26 and ~53 mÅ respectively) that are focused on the chromosphere and transition region including some coverage in the corona.

The IRIS telescope feeds light from three passbands into the spectrograph box:

- Far Ultraviolet (FUV1): 1331.56–1358.40 Å
- Far Ultraviolet (FUV2): 1390.00–1406.79 Å
- Near Ultraviolet (NUV): 2782.56–2833.89 Å

In the spectrograph, the light follows several paths (see spectrograph schematic), either:
Fig. 1: Schematic view of IRIS showing the 20 cm UV telescope, with and without solar panels (for clarity). Light from the Cassegrain telescope (green) is fed into the spectrograph box (light blue).
Fig. 2: Schematic diagram of path taken by light in the FUV spectrograph (dark blue), NUV spectrograph (orange), FUV slit-jaw (light blue) and NUV slit-jaw (purple) path.

- Spectrograph (SG): passing through a slit that is 0.33 arcsec wide and 175 arcsec long, onto a grating that is sensitive in both FUV and NUV passbands, then onto 3 CCDs to produce spectra in three passbands (FUV1, FUV2, NUV; Table 1)

- Slit-Jaw Imager (SJI): reflected off the reflective area around the slit ("slit-jaw"), passing through or reflected off broadband filters on a filterwheel, then onto 1 CCD to produce an image of the scene around the slit (slit-jaw = SJI) in 6 different filters (2 for calibration, 4 for solar images, Table 2)

Exposure times are controlled by 3 different shutters (FUV, NUV and SJI). Light is collected onto 4 CCDs which are read out by 2 cameras (see Section 3 for details) and which cover 3 different spectral bands and the slit-jaw images (Table 1, 2). The IRIS spectral lines cover temperatures from 4,500 K to 10 MK, with the images covering temperatures from 4,500 K to 65,000 K (and possibly 10 MK under flaring conditions). See IRIS Technical Note 1 for more details on IRIS.

Table 1. Overview of spectrograph (SG) channels. These are imaged onto 3 identical 1096x2072 pixel2 CCDs and can all be simultaneously read using two different camera electronics boards (CEB). Ranges, dispersion and effective area are current best estimates based on pre-launch measurements. Spatial pixel size is 0.166", and the maximum spatial extent is 175".

<table>
<thead>
<tr>
<th>Band</th>
<th>Wavelength (Å)</th>
<th>Dispersion (mA/pix)</th>
<th>Effective area (cm²)</th>
<th>Temperature (log T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUV 1</td>
<td>1331.7–1358.4</td>
<td>12.98</td>
<td>1.6</td>
<td>3.7–7.0</td>
</tr>
<tr>
<td>FUV 2</td>
<td>1389.0–1407.0</td>
<td>12.72</td>
<td>2.2</td>
<td>3.7–5.2</td>
</tr>
<tr>
<td>NUV</td>
<td>2782.7–2851.1</td>
<td>25.46</td>
<td>0.2</td>
<td>3.7–4.2</td>
</tr>
</tbody>
</table>

Table 2. Overview of slit-jaw (SJI) channels. Slit-jaw passbands are chosen using a filterwheel. The light is imaged onto one 2072x1096 pixel CCD with only one passband exposed/read-out at one time. Read-out is done with the same CEB as NUV SG. Ranges, full width half max (FWHM), and effective areas of the passbands are best estimates based
on pre-launch measurements. SJI passband types are either mirrors (M) or transmission filter (T). Spatial pixel size is 0.166", and the spatial range is 175” x 175”.

<table>
<thead>
<tr>
<th>SJI Passband</th>
<th>Type</th>
<th>Wavelength (Å)</th>
<th>FWHM (Å)</th>
<th>Effective area (cm²)</th>
<th>log T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>T</td>
<td>5000</td>
<td>2000</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>C II</td>
<td>M</td>
<td>1330</td>
<td>40</td>
<td>0.5</td>
<td>3.7–7.0</td>
</tr>
<tr>
<td>Si IV</td>
<td>M</td>
<td>1400</td>
<td>40</td>
<td>0.6</td>
<td>3.7–5.2</td>
</tr>
<tr>
<td>Mg II h/k</td>
<td>T</td>
<td>2796</td>
<td>4</td>
<td>0.005</td>
<td>3.7–4.2</td>
</tr>
<tr>
<td>Mg II wing</td>
<td>T</td>
<td>2832</td>
<td>4</td>
<td>0.004</td>
<td>3.7–3.8</td>
</tr>
<tr>
<td>Broad-band</td>
<td>M</td>
<td>1600</td>
<td>400</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 3. IRIS data level descriptions.

<table>
<thead>
<tr>
<th>Level</th>
<th>Processing</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLM</td>
<td>Capture</td>
<td>Raw telemetry</td>
</tr>
<tr>
<td>0</td>
<td>Depacketized</td>
<td>Raw images with housekeeping and overscans.</td>
</tr>
</tbody>
</table>
| 1     | Reorient images to common axes:  
• North up (0° roll),  
• increasing wavelength to right | Lowest distributed level |
| 1.5   | • Dark current and offsets removed  
• Flag bad pixels and spikes pixels  
• Flat-field correction  
• Geometric and wavelength calibration | Transitory data product for level 2 production. Not distributed, for internal use only. |
| 1.6   | Physical units (exposure and photon conversion) | Not distributed. |
| 2     | Recast as rasters and SJI time series | Standard science product. Scaled and stored as 16-bit images |
| 3     | Recast as 4D cubes for NUV/FUV spectra | CRISPEx format |
| HCR   | Description of observing sequences | Ingested by HCR at LMSAL. To be searched by VSO, etc. |

1.3 IRIS Data Level Definitions

The convention on IRIS Data Levels is shown in the table above and at length in IRIS Technical Note 11. Raw spacecraft telemetry is converted into Level 0 image files. Level 1 images are reoriented so that wavelength increases left to right. This constitutes the lowest level of scientifically-useful data, however since it is uncalibrated, Level 2 is the correct data product for most analyses. However, if an end user desires to download level 1 data (despite the fact that it is NOT recommended!), it can be done using the following IDL commands (more details about IRIS SolarSoft package is provided in Section 1.5 and Section 2):
The type of processing for data Levels beyond 1 is dependent on whether the data is from the slit-jaw imager or spectrographs. Darks and pedestal offsets are removed, and flat-fielding corrections for telescope and CCD properties are applied to generate Level 1.5 data. The data at Level 1.5 has had the geometric and wavelength corrections applied and the images are mapped to a common spatial plate scale. Spectral images are remapped to align with an equal-sized array where wavelength and spatial coordinates align with the grid. An array mapping the wavelength axis to physical wavelength is created in this process. As with AIA, equivalent procedures to those used internally to transform level 1 to level 1.5 are distributed via SolarSoft as iris_prep.pro [http://sohowww.nascom.nasa.gov/solarsoft/iris/idl/lmsal/calibration/iris_prep.pro].

Levels 2 and 3 are generated from Level 1 or Level 1.5 data and are reorganized so that they can be analyzed using tools adapted from Hinode/EIS and SST/CRISP. Level 2 data consists of sets of 3D image extensions of each wavelength band stored as \((\lambda, x, y)\) assembled from rasters of NUV and FUV Level 1.5 data. Level 3 data exist only for spectral rasters, and are 4D datacubes stored as \((x, y, \lambda, t)\). We will describe some of those tools below.

**Note:** Level 1 vs. level 2 data: This guide is written with the general solar physics community in mind. In the following sections we will discuss IRIS data retrieval and analysis. The spectral data of IRIS is distinct from many contemporary observatories like SDO. IRIS Level 2 data is equivalent to Level 1 data products of those other observatories. The Level 2 data are fully reduced, calibrated, etc. and packaged such that they are “shovel ready” for further analysis. On the other hand IRIS Level 1 data **MUST** be passed through the calibration routines iris_prep.pro by the expert user to reach only level 1.5. The transition from level 1.5 to level 2 is a non-trivial exercise in packaging the data and while the code is available, it is currently not being supported for general use. **Therefore, we strongly recommend that the non-expert or casual IRIS user use the Level 2 data products.**

### 1.4 Sample IRIS data

#### 1.4.1 Sample Spectra and the NUV/FUV Lines

These sample spectra taken by IRIS show the number of counts per pixel per second in a 15 second exposure in several solar regions (plage, sunspot, and network). You can see the strong lines in each spectral range and their relative strength in regions with different degrees of activity. Using the very narrow photospheric lines in each channel we estimated that the spectral resolution (2 x the Nyquist sampling of the spectrograph) of the FUV spectra is 25mÅ and 60mÅ for the NUV. Indeed, those narrow photospheric lines, because they typically display very small intrinsic velocities and broadening (~1km/s in each), are used for wavelength calibration and the geometric correction steps in the spectrographic data. IRIS Technical Note 20 and IRIS Technical Note 19 cover these processes in detail.
Fig. 3: Quiet Sun “FUV” Sample Spectra

Fig. 4: Quiet Sun “NUV” Sample Spectra
1.4.2 Sample Slit-jaw Images

These sample images taken by IRIS on August 20, 2013 show four of the wavelengths available with the SJI filter selection. Clockwise from the top left is the continuum image in the far red wing of the Mg II k line (“SJI_2832”) and it provides high-contrast photospheric imaging, the Mg II k line (“SJI_2796”) which images the upper chromosphere, the S IV transition region filter (“SJI_1400”), and C II transition region filter (“SJI_1330”). On each of the images note the position of the IRIS SG slit (the dark vertical line) this helps us know that, at that particular time, where the SG slit was placed on the Sun. A preliminary study of the IRIS SJI point spread function (PSF) in the transition region filters indicates that IRIS records the highest resolution images ever taken in the transition region. The SJI images should be flat-fielded to remove small residual CCD artifacts and in the future it will also become possible to deconvolve the point spread function to make the images sharper still (see IRIS Technical Note 29 for further details).
1.5 IRIS IDL routines in SSW

The bulk of the data calibration and analysis routines is written in IDL. Therefore, we recommend that users have a solar soft IDL installation (SSW; http://www.lmsal.com/solarsoft/) to follow this guide. The IRIS branch of the IDL solar soft tree is supported by the mission science team and contains most of the tools you need to handle the data from the instrument FITS files through to manipulating the reduced spectra.

If you have IDL SSW already installed then type the following command upon entering your SSW session to obtain the IRIS package:

```
IDL> ssw_upgrade, /spawn, /passive, /verb, /iris
```

To also update the gen folder add , /gen to the IDL command above.

Then, to load the IRIS routines into your path you’ll need to modify the SSW_INSTR environment variable to include them, on UNIX/Mac systems this can usually be found in your .cshrc or .login file:

```
setenv SSW_INSTR 'iris hessi xrt aia eit mdi secchi sot eis'
```

On a windows OS this can be modified in your IDL-DE settings/preferences pane.

**Warning:** The IRIS SSW routines were updated on March 6, 2020 to make them compatible with the latest upgrades of the IRIS data archive. It is therefore essential to update SSW to its latest version.

1.6 IRIS Operations

The observations of IRIS are planned typically for one day (weekdays) or a few days (weekends/holidays). In coordination with the science team, a planner decides on the targets and observing sequences. The resulting work is called a timeline, a list of commands and observing sequences that are run onboard the observatory.

The timeline allows you to see a brief description of each observation along with its “OBS ID” or observing program, the time over which it ran, how many repeats of the sequence were taken, etc. The archive of IRIS timelines can be found online here:


The timelines are available in 3 formats: TIM, SCI, and GIF. If you choose to download the TIM/SCI file for August 20 2013 and wish to read it, then you can go to the folder http://iris.lmsal.com/health-safety/timeline/iris_tim_archive/2013/08/20/, download the timeline file IRIS_science_timeline_20130820.V00.txt and use the IRIS/SSW routine iris_timeline2struct:

```
IDL> tl = iris_timeline2struct('IRIS_science_timeline_20130820.V00.txt')
```

The output of this routine is an array of structures. Each element of the array is a structure describing a single IRIS observing sequence that was run in that time interval:

```
IDL> help, tl
TL STRUCT = -> <Anonymous> Array[23]
IDL> help, tl[0], /str
** Structure <381a378>, 7 tags, length=64, data length=62, refs=2:
DATE_OBS STRING '2013-08-20T04:10:21.000'
DATE_END STRING '2013-08-20T04:11:21.000'
OBSID ULONG 3800
REPEATS INT 10
```

(continues on next page)
The GIF version of the timeline (below) shows graphically how the instrument activities are laid out during the day, the telemetry load, station passes, orbital anomalies (South Atlantic Anomaly – SSA, eclipses, etc). This document provides a detailed legend for the timeline gifs. The timeline GIF for August 20, 2013 is shown below where we can see the characteristics of our sample observation.

Alternately, if you want to query the timeline from your IRIS/SSW command line you can use iris_time2timeline, for example:

```
IDL> t0 = '2013-08-20 00:00:00' & t1 = '2013-08-21 00:00:00'
IDL> tl = iris_time2timeline(t0,t1)
IDL> print, n_elements(tl)
```

Fig. 5: Sample IRIS timeline for August 20, 2013
IDL> info = get_infox(tl, tag_names(tl), /more)
; Gives a complete report of the observations with the header given by the
; tags of the timeline structure.
IDL> help, tl[8]
** Structure <3954f18>, 7 tags, length=64, data length=62, refs=2:
DATE_OBS STRING '2013-08-20T10:59:49.000'
DATE_END STRING '2013-08-20T11:33:12.000'
OBSID ULONG 4180256145
REPEATS INT 1
DURATION FLOAT 2003.70
SIZE FLOAT 13679.0
DESCRIPTION STRING 'Very large dense raster 132"x175" 400s C II Si IV Mg II h/k Mg II w s....'

The SSW command struct_where will allow you to search for strings in the description tag which can be useful for finding particular observations:

IDL> ss=struct_where(tl, search=["DESCRIPTION=*coarse*"], program_count)
IDL> help, program_count
PROGRAM_COUNT LONG = 5
IDL> print, tl[ss[0]].description
Large coarse raster 126"x120" 64s C II Si IV Mg II h/k Mg II w s Deep x 15 SJI cadence 0.5x

If you are interested in a particular IRIS sequence run over an extended period you can search the timeline structures by OBSID. If we want to identify the number of times the IRIS throughput test sequence (OBSID = 4182010156) was run in August of 2013 then:

IDL> t0 = '2013-08-01 00:00:00' & t1 = '2013-08-30 00:00:00'
IDL> tl = iris_time2timeline(t0,t1)
IDL> through = where(tl.obsid eq 4182010156, count)
IDL> help, count
COUNT LONG = 27

1.7 IRIS Documentation and Links

For an in-depth view of the many aspects of the mission, a repository of technical notes built by the science and engineering teams is made available at http://iris.lmsal.com/documents.html. These technical notes (of which the current guide is a part) encompass the areas of Operations, Data Flow, Calibration, Data Analysis, and Numerical Modelling. A list of the different notes can be found below.

<table>
<thead>
<tr>
<th>Operations/Planning</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ITN 1</td>
<td>IRIS Operations Overview</td>
</tr>
<tr>
<td>ITN 2</td>
<td>Manual for Table Creator</td>
</tr>
<tr>
<td>ITN 3</td>
<td>Manual for Timeline Tool (superseded by ITN 8)</td>
</tr>
<tr>
<td>ITN 4</td>
<td>Manual for Synthetic Observations Tool</td>
</tr>
<tr>
<td>ITN 5</td>
<td>Operations Under Roll Conditions</td>
</tr>
<tr>
<td>ITN 6</td>
<td>AEC Operations</td>
</tr>
<tr>
<td>ITN 7</td>
<td>Compression Approach</td>
</tr>
<tr>
<td>ITN 8</td>
<td>Checklist for IRIS planner</td>
</tr>
</tbody>
</table>

continues on next page
### Table 1 – continued from previous page

<table>
<thead>
<tr>
<th>ITN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITN 9</td>
<td>Periodic Calibration Activities</td>
</tr>
<tr>
<td>ITN 50</td>
<td>How to request IRIS coordinated observations [NEW]</td>
</tr>
<tr>
<td>ITN 51</td>
<td>Introduction to IRIS operations</td>
</tr>
</tbody>
</table>

#### Data Flow

<table>
<thead>
<tr>
<th>ITN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITN 10</td>
<td>General Approach to Data Flow and Archiving</td>
</tr>
<tr>
<td>ITN 11</td>
<td>Definition of Data Levels</td>
</tr>
<tr>
<td>ITN 12</td>
<td>Definition of Keywords</td>
</tr>
<tr>
<td>ITN 13</td>
<td>VSO and IRIS</td>
</tr>
<tr>
<td></td>
<td>Level 2 keywords</td>
</tr>
</tbody>
</table>

#### Calibration

<table>
<thead>
<tr>
<th>ITN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITN 14</td>
<td>Dark Current/Offset</td>
</tr>
<tr>
<td>ITN 15</td>
<td>Despiking</td>
</tr>
<tr>
<td>ITN 16</td>
<td>Flat-field</td>
</tr>
<tr>
<td>ITN 16b</td>
<td>FUV background</td>
</tr>
<tr>
<td>ITN 19</td>
<td>Geometric Calibration</td>
</tr>
<tr>
<td>ITN 20</td>
<td>Wavelength Calibration</td>
</tr>
<tr>
<td>ITN 21</td>
<td>Recasting into Level 2/3 Data</td>
</tr>
<tr>
<td>ITN 22</td>
<td>Co-alignment, Plate Scale Analysis</td>
</tr>
<tr>
<td>ITN 23</td>
<td>MTF/PSF Determination</td>
</tr>
<tr>
<td>ITN 24</td>
<td>Stellar Calibration</td>
</tr>
<tr>
<td>ITN 25</td>
<td>Gain Determination</td>
</tr>
</tbody>
</table>

#### Data Analysis

<table>
<thead>
<tr>
<th>ITN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITN 26</td>
<td>User Guide To Data Analysis (this document)</td>
</tr>
<tr>
<td>ITN 27</td>
<td>Quicklook Tools Manual</td>
</tr>
<tr>
<td>ITN 28</td>
<td>IRIS IDL Data Structure</td>
</tr>
<tr>
<td>ITN 29</td>
<td>Deconvolution Approach</td>
</tr>
<tr>
<td>ITN 30</td>
<td>60 Day Observing Plan</td>
</tr>
<tr>
<td>ITN 31</td>
<td>IRIS science planning: tables, linelists, targets</td>
</tr>
<tr>
<td>ITN 32</td>
<td>Co-aligned IRIS, SDO and Hinode observations</td>
</tr>
<tr>
<td>ITN 38</td>
<td>Analysis of optically thin lines observed by IRIS</td>
</tr>
<tr>
<td>ITN 39</td>
<td>Optically thick diagnostics</td>
</tr>
<tr>
<td>ITN 40</td>
<td>IRIS Data Idiosyncrasies</td>
</tr>
<tr>
<td>ITN 41</td>
<td>A low-level guide to IRIS with Python</td>
</tr>
</tbody>
</table>

#### Numerical Modelling

<table>
<thead>
<tr>
<th>ITN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITN 33</td>
<td>General Overview of Numerical Simulations</td>
</tr>
<tr>
<td>ITN 34</td>
<td>Numerical Simulations Quicklook Tools</td>
</tr>
<tr>
<td>ITN 35</td>
<td>Numerical Simulations Synthetic Observables</td>
</tr>
<tr>
<td>ITN 36</td>
<td>RH 1.5 D Manual</td>
</tr>
<tr>
<td>ITN 37</td>
<td>How to Derive Physical Information from Mg II h/k</td>
</tr>
<tr>
<td>ITN 42</td>
<td>IRIS² Inversions</td>
</tr>
</tbody>
</table>

In addition to the documentation, below are a few more useful links related to IRIS.
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IRIS Recent Observations</td>
<td><a href="http://www.lmsal.com/hek/hcr?cmd=view-recent-events&amp;instrument=iris">http://www.lmsal.com/hek/hcr?cmd=view-recent-events&amp;instrument=iris</a></td>
</tr>
<tr>
<td>UiO IRIS/Hinode Scientific Data Center</td>
<td><a href="http://sdc.uio.no/sdc/">http://sdc.uio.no/sdc/</a></td>
</tr>
<tr>
<td>SUMER UV Spectral Atlas</td>
<td><a href="https://umbra.nascom.nasa.gov/spectral_atlases.html">https://umbra.nascom.nasa.gov/spectral_atlases.html</a></td>
</tr>
<tr>
<td>SDO Context Information</td>
<td><a href="http://sdowww.lmsal.com/">http://sdowww.lmsal.com/</a></td>
</tr>
<tr>
<td>Hinode Operations</td>
<td><a href="http://www.isas.jaxa.jp/home/solar/hinode_op/">http://www.isas.jaxa.jp/home/solar/hinode_op/</a></td>
</tr>
</tbody>
</table>
CHAPTER TWO

QUICKSTART

This chapter provides a quick guide to let you jump straight into action using IDL. Be sure to read the rest of this guide for a more in-depth view of IRIS data analysis. This chapter assumes the user has a SSW IDL installation with a recent IRIS package. This can be installed by:

```
IDL> ssw_upgrade, /spawn, /passive, /verb, /iris
```

To also update the gen folder add , /gen to the IDL command above.

Then, to load the IRIS routines into your path you will need to modify the SSW_INSTR environment variable to include them, on UNIX/Mac systems this can usually be found in your .cshrc or .login file:

```
setenv SSW_INSTR 'iris hessi xrt aia eit mdi secchi sot eis'
```

Please note that the IRIS tools work best when using IDL 8.4 or above.

**Warning:** The IRIS SSW routines were updated on March 6, 2020 to make them compatible with the latest upgrades of the IRIS data archive. It is therefore essential to update SSW to its latest version.

2.1 IRIS overview

IRIS is a UV slit-spectrograph that also takes slit-jaw images. It obtains spectra in two FUV bands (1331.56–1358.40 Å, 1390.00–1406.79 Å) and one NUV band (2782.56–2833.89 Å), all exposed simultaneously. In addition, slit-jaw images in the bands of 1330, 1400, 2796, and 2832 Å can also be taken (only one exposure a time). The spatial resolution is 0.33” in the FUV and 0.4” in the NUV. The spectral resolution is 26 mÅ in the FUV and 56 mÅ in the NUV. The slit is 0.33” wide and 175” long. The observatory can operate in a fixed target mode (“sit-and-stare”) or by scanning a region by moving the spacecraft (“rasters”) with various numbers of steps possible (2-400) and different step increments (0.3”, 1”, 2”). There are about 50 basic observing modes, which are encoded in a unique identifier called OBSID.
2.2 Getting the data

IRIS data are available in different degrees of calibration. However, Level 2 represents fully calibrated data and is the recommended data product. The IRIS data search webpage provides a powerful search engine and can be used to download the data and browse quicklook movies and plots. One can also search for and download data inside SSW IDL:

```idl
IDL> t0 = '18:50:00 10-nov-2014'
IDL> t1 = '19:00:00 10-nov-2014'
IDL> files = iris_time2files(t0, t1, level=2, drms, /urls, /compressed)
```

This list of files can then be downloaded to the current directory and unzipped with:

```idl
IDL> sock_copy, files, dir='./'
IDL> $gunzip *fits.gz
IDL> $ls *.gz |xargs -n1 tar -xzf
```

Level 2 data have two types of files: slit-jaw images (“SJI” in filename) or spectral rasters (“raster” in filename). Each SJI file contains all the frames for a given filter for the total duration of an observation. Each raster file contains all the spectra for a given sequence; when an observation consists of multiple raster sequences there is one file per raster. In the special case of sit-and-stare observations, only one raster file exists.

2.3 Reading the data in IDL

The most convenient way to load the data in IDL is to use the object interface:

```idl
IDL> myfile=iris_files('*.SJI*.fits')
IDL> d = iris_load(myfile[0])
IDL> header = d->gethdr(/struct)
IDL> print, header.DATE_OBS
2014-11-10T18:55:49.920
IDL> data = d->getvar() ; for a slit-jaw image
```

Most observations do not expose the full detector. Instead, spectral windows around the lines of interest are used, and those windows are saved in the raster files. When loading spectral data one must select which window to load, and there are helper functions for that:

```idl
IDL> my_spec_file=iris_files('/*.raster*.fits')
IDL> d = iris_load(my_spec_file[0])
IDL> d->show_lines
Spectral regions(windows)
  0  1335.71  C II  1336
  1  1343.26  1343
  2  1349.43  Fe XII 1349
  3  1355.60  O I  1356
  4  1402.77  Si IV 1403
  5  2832.79  2832
  6  2826.68  2826
  7  2814.51  2814
  8  2796.20  Mg II k 2796
Loaded Slit Jaw images
  0 SJI_1330
  1 SJI_1400
  2 SJI_2796
```

(continues on next page)
Other values such as the wavelength scale and observation times can also be obtained:

```
IDL> wavelength = d->getlam(8)
IDL> times = d->ti2utc()
IDL> print, times[0]
2014-11-10T18:55:49
```

### 2.4 Data calibration

The level 2 data are dark subtracted, flat fielded, corrected for geometrical distortion (spectra) and wavelength calibrated.

There is no absolute wavelength calibration. Instead, the positions of known spectral lines are measured to calibrate for wavelength. As of May 2014 the wavelength calibration corrects for the orbital velocity and the thermal drifts of the spectrograph. However, in some cases it may be necessary to apply further corrections. The O I 1355.5977 Å line is recommended reference for the FUV, and the Ni I 2799.474 Å for the NUV. A full explanation can be found in ITN 20.

The level 2 data are spatially coaligned both within channels and between slit-jaws and spectra. This procedure is automatic and should be checked by verifying the position of fiducial marks (they show up as dark bands on spectrograms and as bright spots on the slit in slit-jaw images). To coalign IRIS data with SDO we suggest cross-correlating the IRIS 1400 SJI with AIA 1700, and the IRIS 2832 with the HMI continuum. A full explanation can be found in ITN 22.

### 2.5 Data analysis and visualisation

Several graphical tools are available for quick look and detailed analysis. `iris_xfiles` is a tool to search for, quickly visualise and calculate several quantities of level 2 data. `CRISPEX` is a powerful tool for visualising multi-dimensional spectral data (up to 4D: x, y, wavelength, time), and it has been adapted to work with IRIS level 3 files.

IRIS level 3 files are a reorganisation of level 2 raster files. Multiple rasters are combined in a single file (plus a transposed version). They are not distributed, but can be produced by the user, e.g.:

```
IDL> f = iris_files('./*raster*.fits') ; get all raster files
IDL> windows = [0, 6] ; which wavelength windows to include
IDL> iris_make_fits_level3, f, windows, /sp, wdir='my_output_dir'
```
IRIS Level 2 data can be downloaded from the mission web page or through the European Hinode/IRIS Science Data Center. The IRIS Level 2 files are the calibrated, “science-ready” FITS files distributed to the end-user. The FITS files are designed to allow easy access to the data and metadata; this section describes their structure and typical use cases.

### 3.1 Structure of IRIS level 2 FITS files

The level 2 data are a combination of individual frames for the duration of a given observing sequence (defined by an OBSID number). There are two types of IRIS level 2 files: spectrograph and slit-jaw. The internal structure is different for spectrograph and slit-jaw files, and file naming convention is the following:

- **Spectrograph**: `iris_l2_<day>_<time>_<OBSID>_raster_t000_r<raster number>.fits`, where `<day>` is YYYMMDD, `<time>` is the starting time in HHMMSS, and the raster number starts at zero and up to the total number of raster scans (or repeats) minus one.

- **Slit-jaw**: `iris_l2_<day>_<time>_<OBSID>_SJI_<filter>_t000.fits`, where `<filter>` is the filter central wavelength in Å (1330, 1400, 2796, or 2832).

The number of spectrograph files depends on the type of observing sequence. For sit and stare or single-raster programs (e.g. 400-step raster) there is only one file. For multiple-repeat rasters, there is one file per raster scan. Data from both FUV and NUV detectors are combined in each file. The FITS structure comprises one primary Header-Data Unit (HDU) and several extensions. **The primary HDU contains no data, only a header.** The science data are saved in the subsequent extensions (one extension per spectral window), and additional metadata are saved in the last two extensions.

For slit-jaw files there is only one file per filter per observing sequence. The FITS structure comprises one primary Header-Data Unit (HDU) and two extensions. The science data are saved in the primary HDU, while additional metadata are saved in the first and second extension.

The tables below illustrate the HDU structure for slit-jaw and spectrograph files.
Table 1: Structure of level 2 spectrograph files. \( n_{\text{wave}_n} \) denotes number of wavelength points in spectral window \( n \). \( n_y \) denotes number of spatial Y points. \( n_{\text{rt}} \) denotes number of raster positions (multiple raster programs) or number of time positions (sit and stare programs).

<table>
<thead>
<tr>
<th>HDU #</th>
<th>HDU type</th>
<th>Contents</th>
<th>Data dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Primary</td>
<td>Main header</td>
<td>No data</td>
</tr>
<tr>
<td>1</td>
<td>Image Extension</td>
<td>Data for wavelength window 1</td>
<td>([n_{\text{wave}<em>1}, n_y, n</em>{\text{rt}}])</td>
</tr>
<tr>
<td>2</td>
<td>Image Extension</td>
<td>Data for wavelength window 2</td>
<td>([n_{\text{wave}<em>2}, n_y, n</em>{\text{rt}}])</td>
</tr>
<tr>
<td>(\ldots)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n)</td>
<td>Image Extension</td>
<td>Data for wavelength window ( n )</td>
<td>([n_{\text{wave}<em>n}, n_y, n</em>{\text{rt}}])</td>
</tr>
<tr>
<td>(n+1)</td>
<td>Image Extension</td>
<td>Auxiliary metadata</td>
<td>([47, n_{\text{rt}}])</td>
</tr>
<tr>
<td>(n+2)</td>
<td>Table Extension</td>
<td>Technical metadata</td>
<td>([n_{\text{rt}}, 7])</td>
</tr>
</tbody>
</table>

Table 2: Structure of level 2 slit-jaw files. \( n_x \) denotes number of spatial X points. \( n_y \) denotes number of spatial Y points. \( n_t \) denotes number of time positions.

<table>
<thead>
<tr>
<th>HDU #</th>
<th>HDU type</th>
<th>Contents</th>
<th>Data dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Primary</td>
<td>Main header and data</td>
<td>([n_x, n_y, n_t])</td>
</tr>
<tr>
<td>1</td>
<td>Image Extension</td>
<td>Auxiliary metadata</td>
<td>([30, n_t])</td>
</tr>
<tr>
<td>2</td>
<td>Table Extension</td>
<td>Technical metadata</td>
<td>([n_t, 5])</td>
</tr>
</tbody>
</table>

A description of the level 2 keywords, for primary and extension headers, is available for download.

The auxiliary metadata contain several quantities for each exposure. These are parameters that may vary for different exposures and are therefore stored here. The header of this HDU gives the different keywords and their index in the table. For example, \( \text{TIME} \) has the value 0, meaning that \( \text{data}[0, \ast] \) will give an array with the times of each exposure, for all timesteps or raster steps.

**Note:** In case of missing exposures, most of the parameters in the auxiliary metadata are set to 0. There are some exceptions: \( \text{DSRCFIX}, \text{DSRCNIX}, \) and \( \text{DSRCSIX} \) are set to -1. Some auxiliary metadata values are not given for both NUV and FUV detectors. In such cases, the values are taken from the FUV source file, and if the FUV file is missing they are taken from the NUV file. If both NUV and FUV source files are missing, the parameters are set to 0. Exceptions are \( \text{TIME} \), which is set to the planned time of the respective exposure, and \( \text{PZTX} \) and \( \text{PZTY} \), which are set to the minimum value of all \( \text{PZTX} \) and \( \text{PZTY} \) values, respectively, within the particular fits file. The following parameters are not separate for FUV and NUV: \( \text{TIME}, \text{PZTX}, \text{PZTY}, \text{XCENIX}, \text{YCENIX}, \text{OBS_VRIX}, \text{OPHASEIX}, \text{PC1}_1\text{IX}, \text{PC1}_2\text{IX}, \text{PC2}_1\text{IX}, \text{PC2}_2\text{IX}, \text{PC3}_1\text{IX}, \text{PC3}_2\text{IX}, \text{PC3}_3\text{IX}, \text{PC2}_3\text{IX} \).

The technical metadata in the last extension are usually not useful for the end-user. Their content is only useful for reproducing the exact steps of the data calibration, and contain details such as FRM, FDB, and CRS IDs, names of level 1 files used. The header of the last extension contains some information about the format.
3.2 Searching and Downloading

3.2.1 Using the IRIS Data Search Webpage

The IRIS data search webpage (http://iris.lmsal.com/search) is designed to quickly guide researchers to IRIS datasets appropriate for their research. It consists of five graphical elements and three steps to the data:

1. IRIS Banner
2. Selection widgets
3. Graphical display of search results on a solar image
4. Tabular display of search results
5. Dataset browser/inspector with links to download the data sets

The IRIS data search tool is optimized for use on landscaped displays of at least 1280x768 pixels. The banner and the solar image can be hidden (displayed) by clicking on the red (green) buttons in the upper left corners to accommodate smaller screens. The tool has been tested with recent versions of Firefox, Safari and Chrome browsers. If you have difficulty with the tool, you might first try one of these browsers to ensure compatibility.

Fig. 1: IRIS search sample screenshot

Selection widgets: There are six widgets available for customized, dynamic, data searches. At the most basic this search consists of specifying the start and end of a time range of interest. When first loaded these default to select the week surrounding the current date. The start and end times can be moved forward and back a day or a week by using the single and double arrow buttons. Specific dates can be entered directly into the text boxes or by using the calendars that popup when one clicks on them. The total count of datasets available within the time range appears at the bottom left of this selection area. By default, only datasets that are completely processed are displayed. If you wish to include ones that are still processing, uncheck the only OBS with data box below count (not shown in figure).
The remaining widgets are used to filter the selections within the specified time range. The count of available data sets updates dynamically to reflect the effects of your selections.

**Fig. 2: IRIS search selection widgets**

**Raster:** Limit results to datasets with rasters within a (min, max) range of: fields of view in arcseconds; number of repeats (count); and of the cadence in seconds and with raster steps within a range of number (count); size in arcseconds and cadence in seconds.

**Slit Jaw Imager (SJI):** Limit results to datasets with slitjaw images within a range of fields of view and cadences for each wavelength band.

**Exposure time:** Limit results to datasets within a range of minimum exposure and mean exposure times based on all
images within the dataset.

**Target:** Limit results to a range of target positions relative to disk center in arcseconds either as a bounding box (\(\text{xcen, ycen}\)) or an annulus between radii. Limit sets to specific IRIS Observation IDs or target. The colors of these last two change to indicate the presence (green) or absence (red) of matching datasets based upon other selections.

When all selections are made, clicking the search button refreshes the results in the display area. **Note that the display does not update while you are constructing a search.** A range of background SDO/AIA images of the sun corresponding to the start time of query can be selected for the display. All filters (other than dates) and displays are cleared by clicking the reset button.

---

**Fig. 3:** IRIS search display widget.

**Display Widget:** The results of a search are displayed on a co-temporal AIA image that is selectable from the search widget. The default setting displays the bounding boxes for the slit jaw (raster) image as green (red) rectangles on an 193 Å AIA image. A sortable list of IRIS observations on the right presents details of the dataset including the time interval, short descriptions, pointing, fields of view, cadences and observation IDs. Clicking on an entry in either widget, highlights the selection in the table along with a detailed description in the inspection widget.

---

**Fig. 4:** IRIS search inspection widget

The latest version of the search engine includes the option to find all IRIS observations for which co-aligned SDO and/or Hinode data cubes are available. This can be done by checking the appropriate option buttons located under the “More” button.

**Inspection Widget:** The inspection widget shows more details of the dataset, including a thumbnail slitjaw image, pointing information and links to and sizes of the data products (when they become available). Clicking on the image or title will bring up a separate details page with summary movies, paths to the data and links to the AIA cutout service. Clicking on the data links will immediately download the corresponding gzipped dataset.

---

**3.2. Searching and Downloading**
3.2.2 Using SSW IDL

IRIS Level 2 data can be accessed via the SSW command line using a simple extension of the IDL/SSW timeline queries used above. So, our earlier example:

```
IDL> t0 = '2013-08-20 00:00:00' & t1 = '2013-08-21 00:00:00'
IDL> tl = iris_time2timeline(t0,t1)
IDL> info = get_infox(tl, tag_names(tl), /more)
```

returns the details of all IRIS observations taken on August 20, 2013. If we want to browse those sequences with OBSID = 4182010156 then we can use the following command to return the location of the IRIS Level 2 data for each relevant sequence:

```
IDL> hcr = iris_time2hcr(t0,t1,/expand_eventid,limit=2000,/struct)
IDL> fg=where(hcr.obsid eq '4182010156')
IDL> l2_folder = str_replace(hcr[fg].url,'/level2/','/level2_compressed/')
IDL> print, l2_folder
http://www.lmsal.com/solarsoft/irisa/data/level2_compressed/2013/08/20/20130820_150507_4182010156/
http://www.lmsal.com/solarsoft/irisa/data/level2_compressed/2013/08/20/20130820_185222_4182010156/
http://www.lmsal.com/solarsoft/irisa/data/level2_compressed/2013/08/20/20130820_194022_4182010156/
http://www.lmsal.com/solarsoft/irisa/data/level2_compressed/2013/08/20/20130820_201022_4182010156/
http://www.lmsal.com/solarsoft/irisa/data/level2_compressed/2013/08/20/20130820_211522_4182010156/
```

The www subfolder in each example contains a collection of browsable movies of the slit-jaw and spectrograph image sequences taken during the observation like that shown below. These movies can be used to view the data before downloading the (large) Level 2 FITS files.

```
(IDL> hcr = iris_time2hcr(t0,t1,/expand_eventid,limit=2000,/struct)
IDL> fg=where(hcr.obsid eq '4182010156')
IDL> l2_folder = str_replace(hcr[fg].url,'/level2/','/level2_compressed/')
IDL> print, l2_folder
http://www.lmsal.com/solarsoft/irisa/data/level2_compressed/2013/08/20/20130820_150507_4182010156/
http://www.lmsal.com/solarsoft/irisa/data/level2_compressed/2013/08/20/20130820_185222_4182010156/
http://www.lmsal.com/solarsoft/irisa/data/level2_compressed/2013/08/20/20130820_194022_4182010156/
http://www.lmsal.com/solarsoft/irisa/data/level2_compressed/2013/08/20/20130820_201022_4182010156/
http://www.lmsal.com/solarsoft/irisa/data/level2_compressed/2013/08/20/20130820_211522_4182010156/
```

Extending this example to a more specific case let’s pick the first of these OBSID = 4182010156 observations and recover the URLs for the IRIS Level 2 FITS files from the command line:

```
IDL> tmp = where(tl.obsid eq 4182010156)
IDL> tl = tl[tmp[0]]
IDL> 12_rasters=iris_time2files(t0, t1, obsid = 4182010156, /raster, level=2, /urls, /compressed)
IDL> help, 12_rasters
```

Fig. 5: Browsable movies from data webpage.
Naturally these files could have been viewed by opening the web folder found earlier. These L2 FITS files can be downloaded to your local folder using a web browser or by using the SSW command:

`; raster files
IDL> sock_copy, l2_rasters, out_dir='./'
`; SJI files
IDL> sock_copy, l2_sji, out_dir='./'
IDL> $gunzip *fits.gz
IDL> $ls *.gz |xargs -n1 tar -xzf

Once the files are finished downloading you are ready for the next step - read them or use our tools to dig a little deeper.

### 3.3 Browsing Level 2 Data with **iris_xfiles**

One way to browse and manipulate Level 2 IRIS data is to use the widget routine **iris_xfiles**. This routine is run from the IDL command line as follows:

`IDL> iris_xfiles`

The **iris_xfiles** interface appears as below. The search directory window will let you browse your IRIS data directory tree. But in this case it is better to remove the file search filter so that you can see where you are navigating. When navigating double click on a directory name in order to enter the directory.

If the user is downloading Level 2 FITS files on a locally mounted drive (like the example we show here) then the user
Fig. 6: IRIS_xfiles main interface.
should edit the “search pattern” tab (below) to the folder in which the IRIS Level 2 FITS files are contained. Click on the edit button to change the configuration.

Level 2 FITS files of two types can be picked from the file picker: `iris_l2*SJI*.fits` & `iris_l2*_raster_*.fits` which, as we have discussed above, contain the slit jaw images in a given filter taken during an observing sequence, or the spectral images of an observing sequence, respectively.

Selecting one of the slit-jaw raster FITS files - by clicking on Confirm Selection, or just double-clicking the file - will bring up the widget called `iris_ximovie`. `iris_ximovie` (which can also be used individually on FITS data loaded at the command line) allows the user to view the slit jaw sequence as a movie. It contains a number of options for playback speed, change of magnification, zooming, blinking, and the generation of postscript, jpeg or gif output as well as MPEG movies through the “file/save_as menu”.

`iris_xfiles` can multi-task so you can have multiple analysis/movie/widget windows open simultaneously while you study your data. So, selecting the raster FITS file you will see the following X11 window pop up, the `iris_xcontrol` widget. In this case the requested raster is read, as are the available slit jaw images that were taken during this particular raster.

`iris_xcontrol` (above) is the main control widget for the IRIS L2 quicklook software. It launches an array of other QL widget programs.

An overview of the raster is given in the middle top window that includes the OBSID, the number of raster positions, the number of spectral (line) windows, their wavelength and pixel ranges on the IRIS CCDs as well as their name - the names are usually associated with the principal spectral line in the window. Many of the other quicklook widgets driven by `iris_xcontrol` require a line list given by this selectable list. The “Generate Level 3 files” button of `iris_xcontrol` will generate a set of Level 3 files for analysis like CRISPEX.

The left of the `iris_xcontrol` widget is an SDO/AIA 171Å image of the Sun taken closest in time to the start of the observation. The location of the IRIS scan or sit-and-stare observation on the Sun is shown as a box or a vertical line respectively. The image, if found, should be current in the sense that it is taken on the same day as the raster. Right clicking on the solar image will toggle between various AIA images. Left clicking will bring up a widget containing a magnified copy of the image including the raster pointing.

The lower middle window of `iris_xcontrol` show the layouts of the spectral windows (or regions) on the NUV
Fig. 8: *iris_ximovie* widget.
Fig. 9: \texttt{iris\_xcontrol} main interface.

3.3. Browsing Level 2 Data with \texttt{iris\_xfiles}
Fig. 10: `iris_xdetector` widget.

(top) and FUV (bottom) CCDs. The lower window shows the combined FUV1 and FUV2 CCDs (see the introduction for further information). Clicking on these windows will start the “Detector” quicklook widget (below) which shows the layout of the spectral windows on the CCDs with options to cycle through the exposures of the raster, plot/print pixel values, change the color table, etc. This widget can also be started by clicking on the “Detector” button, which can be found just above the AIA image window. The “Create Animation” button in the Detector widget often provides a very interesting movie of the spectral evolution during the observation - particularly for sit-n-stare observations.

The right column of `iris_xcontrol` shows up to four slit jaw images taken during the raster. Clicking on any of these will bring up a widget (`iris_sji_image`; see below) containing a magnified image of the slit jaw, sliders to cycle through the slit jaws taken during the raster, and the option to plot the location of the raster exposures taken (compare with `iris_ximovie`). An instance of `ximovie` can be started for the SJI sequence by clicking on the button.

Below the “Detector” button on `iris_xcontrol` - in the center of left column - are the buttons for starting the “Browser”, the “Spectroheliogram”, “Whisker” and “Intensity Map” widgets.

**Browser:** The “Browser” is similar to the Hinode/EIS quicklook browser tool (above), and has recently been modified for IRIS. We note that the browser routine (`iris_raster_browser`) can run independently of `iris_xfiles` by calling:

```idl
IDL> iris_raster_browser, l2f
; where l2f can be an IRIS data object or a Level 2 FITS file
```

**Spectroheliogram:** Selecting one or more line windows in the upper left panel of `iris_xcontrol` and clicking the “Spectroheliogram” button will bring up a widget that contains images of the spectral windows taken during the raster (see below). The spectral movie strips are arranged vertically according to line window and horizontally according to exposure number. Options in the spectroheliogram widget include setting the spectra on pixel or Å wavelength scales and/or pixel or arcsec.

**Whisker:** Select one (or more) spectral windows in order to view the windows arranged according to raster position (or exposure number) at a given slit position. This is a widget that presumably is best used for “sit and stare” type observations where one can follow the time evolution of a given location on the sun in a specific spectral line.
Fig. 11: *iris_sji_image* widget.

3.3. Browsing Level 2 Data with *iris_xfiles*
Fig. 12: \texttt{iris\_xfiles} browser.

Fig. 13: \texttt{iris\_xfiles} spectroheliogram.
**Intensity**: The “Intensity Map” function of *iris_xfiles* is very powerful and will integrate the selected spectral windows over a given range of wavelengths and display the result as an image (see below). This is an excellent tool for examining the wing behavior, or the properties of the complex Mg II h &k (below and left) lines or the C II 1330Å lines.

By default “Intensity” will integrate over the entire spectral window chosen. However, by clicking on the “Define Line” button parts of the spectral line (or the continuum) can be chosen for integration and presentation in the image window. When “Define Line” is pressed a small widget is brought up (see below) where one can define properties such as the “line start” and “line stop” locations. The integration of the line intensity is done between these locations. Furthermore, when the “Continuum Start” and “Continuum Stop” sliders are used “Intensity” will compute an average intensity of the continuum that is then subtracted from the line intensity integral. The example shown above (for the Mg II h line) shows an image of the integrated core reversal wing minus a small continuum patch in the red wing.

As with the other tools there are options to zoom in on the images, plot pixel values, change color tables and gamma factors in the images, swap between pixel and arcsec spatial scales, etc.

**The “Line Fit” gadget**: The Line Fit gadget on the *iris_xcontrol* interface can perform rudimentary spectral analysis of the optically thin FUV lines. It can also be used to inspect the optically thick Mg II and C II lines, but the analysis required for these lines is significantly more complex than this tool permits (see chapter on *IRIS Level 3 Data*).

The “Profile moments” pull down menu will do simple calculations of line moments (see below), either directly or via Gaussian fits to the selected lines. The results of these calculations will be displayed with the Intensity map tool, now with the possibility of choosing whether to view intensities, velocities, widths, or continuum intensities.

Running with “moments” is relatively straightforward. This will compute the zeroth (intensity), first (doppler velocity), and second (line width) moments of the intensity profile. First, the “define line” window will pop up, once for each line checked off in the “select line” pane. Thereafter, the “intensity map” tool will pop up with some extra options for displaying the intensity, velocity, or line width of the various lines chosen. The image can be resized, with or without the aspect ratio being retained. The height of this image is the same as that produced by the slit jaw viewer (“xsji_image”) so the images should be directly comparable.

### 3.4 Reading Level 2 Data in IDL

#### 3.4.1 Using The IRIS Level 2 Data Object

The IRIS level 2 software is designed to allow the user to easily read and access the data and keywords contained in IRIS level 2 fits files. It is also designed to be used by the IRIS QL software, i.e. those widgets called by *iris_xfiles*. The software is made up of several objects; *iris_data, iris_aux, iris_sji, iris_cal, iris_moment*, etc, the most important of which is by far the *iris_data* object. The casual user does not have to worry too much about this, at least not initially. The following are examples that show how to read the fits file header, load an IRIS raster window (region) into memory, as well as locate important auxiliary information.

To construct an *iris_data* object one first needs to find a set of iris files. Go to a directory that contains iris files, or make a text variable “path” that contains the path to iris files. The function *iris_files* will load in selected iris files:

```idl
IDL> f = iris_files(path=path)
```

returns the list of fits files in the directory path (default ‘./’) and prints this list on the screen. Then, assuming that *f[X]* is a level 2 iris raster file:

```idl
IDL> d = iris_load(f[X])
```

This populates the data object with the fits header, auxiliary information, and (pointers to) the data itself. You can get help on the data object and its methods with:
Fig. 14: *iris_xfiles* intensity map.
Fig. 15: Define line dialong for intensity map or profile moments.
Fig. 16: Profile moments dialog.
A User’s Guide to IRIS Data Retrieval, Reduction & Analysis, Release 1.0

IDL> d.help
Class: IRIS_DATA
file: /opt/solarsoft/iris/idl/uio/objects/iris_data__define.pro
version: $Id: iris_data__define.pro,v 1.114 2017/05/22 09:23:23 mawiesma Exp$
FUNC init, file, verbose=verbose : initializes
→ object, calls read if "file" parameter given
PRO close : frees pointer to
→ main data array "w" and closes all associated files
PRO cleanup : called by obj_
→ destroy, frees all pointers and closes all associated files
PRO help, description=description : prints out this_
→ help, setting the 'description' keyword will also print the header info
FUNC getfilename : returns the_
→ raster filename
FUNC getfilename_sji, lwin : returns the sji_
→ filename of lwin (lwin=0 through 5)
FUNC getcomment : returns comment_
→ for object
PRO setcomment, comment : sets comment for_
→ object
FUNC getaux : returns the aux_
→ data as an iris_aux object
FUNC getcal : returns IRIS_
→ specific parameters as an iris_cal object
FUNC missing : returns the_
→ "missing" value
FUNC getnwin : returns the_
→ number of line windows
(...)

Here we detail some of the available methods. For example, to retrieve the fits header:

IDL> hdr = d->gethdr(iext, struct=struct)

The function gethdr takes a parameter iext (default 0) which gives the extension to be displayed (remember that the level 2 fits files have a main header “0” and one header for each line window or region). There is one keyword, /struct which when set, return the header as a IDL structure instead of a string array. Or if one wants to look at a specific keyword tag:

IDL> print, d->getinfo('tag')

will produce it. In addition to tag this function takes another parameter iext (default 0) and a keyword sji such that /sji will return the value of the keyword tag in slit jaw header iext.

A very useful procedure is:

IDL> d->show_lines
Spectral regions (windows)
0 1335.71 C II 1336
1 1349.43 Fe XII 1349
2 1351.17 1351
3 1355.60 O I 1356
4 1393.78 Si IV 1394
5 1402.77 Si IV 1403
6 2786.52 2786
7 2796.20 Mg II k 2796
8 2831.33 2831

(continues on next page)
Loaded Slit Jaw images
0  SJI_1330
1  SJI_1400
2  SJI_2796

which gives an overview of the line(s) and SJI windows loaded into the object. This function only works on raster FITS files, not SJI FITS files.

To actually look at the data, use:

```idl```
```
IDL> win = d->getvar(iwin, load=load)
```
```
```

```idl```
```
IDL> dum = win[*,*],12
```
```
```

or:

```idl```
```
IDL> dum = (d->getvar(iwin))[*,*],12
```
```
```

will contain the 12th exposure (raster position) of window iwin.

If one requires the entire window to be read into memory instead of looking at one exposure at a time, a /load option should be passed to getvar:

```idl```
```
IDL> win = d->getvar(iwin, /load)
```
```
```

Note that the /load option will also descale the data (using the descale_array method). At this point you may decide that you have had enough of objects. “Just give me the data”:

```idl```
```
IDL> s = d->getdata()
```
```
```

Will return a structure that contains the “entire” object, along with various auxiliary information. Note that IRIS level 2 files can be quite large, so do not use this method uncritically. Note that reading in data to the structure may take some time.

For those sticking with objects, the wavelength lambda for window iwin is given by:

```idl```
```
IDL> lambda = d->getlam(iwin)
```
```
```

where iwin can be either the window number or the approximate wavelength of the window (the software will find the window if the wavelength given lies inside the wavelength range of the spectral window).

The slit position (y) is given by:

```idl```
```
IDL> y = d->getypos()
```
```
```

The method getypos takes an iwin argument, but all windows share the same y-scale so it is not necessary to specify it. The raster (x) and/or time (t) coordinate are found via:

```idl```
```
IDL> x = d->getxpos(iwin=iwin)
IDL> t = d->gettime()
```
```
```

The time returned is relative to STARTOBS, note that you can also get the absolute time via:
Here is a simple IDL script that shows how this can be done avoiding objects all together (the script can be found in the “utils” directory of the SSW distribution):

```idl
iris_readheader,f,struct=struct,extension=extension
   if n_elements(extension) eq 0 then extension=0
   d = iris_obj(f)
   hdr = d->gethdr(extension,struct=struct)
   obj_destroy, d
   return, hdr
end
```

The `gethdr` method by default will return the main (extension=0) fits header, but since the various IRIS line windows (regions) are stored as extensions 1, ..., NWIN, there is a small header associated with each which may be useful. Using the `struct` keyword will return the header as an idl structure instead of a string array. In the latter case header tags (keywords) can be accessed with the usual SSW `fxpar(hdr, tag)` routines. Note that since the main header is contained in extension=0, the window headers are accessed as extension=window nr + 1.

In general, this should be the recipe for writing small “one-liners”:

- open and load the object, viz `d = iris_obj()`
- call the methods needed to do what you want to do
- manipulate and make the output available
- destroy the object

Other examples (of many, see below) are:

```idl
IDL> exp = d->getexp(iexp, iwin=iwin)
```

or:

```idl
IDL> xpos = d->getxpos(indx, iwin=iwin)
```

to get the exposure time for exposure number `iexp` or spatial index `indx` in window number `iwin`. These functions return an array of exposure times or spatial positions if no parameter `iexp` or `indx` is given, and default to the default window (the first one read) if no `iwin` keyword is given.

To check the data integrity and returned structures:

```idl
IDL> s = d->aux_info() ; extension nwin+1
IDL> s = d->obs_info() ; extension nwin+2
```

The help command can be used to view their content:

```idl
IDL> help, s ; will show their content.
```

There are a number of objects that are detailed in IRIS Technical Note 28 as well as a list of other methods that can be applied to the data.
3.4.2 Using read_iris_l2.pro

IRIS Level 2 FITS files can be read into memory using the read_iris_l2 procedure:

```idl
read_iris_l2, l2files, index, data, _extra=_extra, keep_null=keep_null, $  
append=append, silent=silent, wave=wave, remove_bad=remove_bad
```

where `l2files` can be an array of Level 2 FITS files. The `wave` keyword can be used to select a specific wavelength window (e.g., `wave = 'Si IV 14'`) for a raster FITS file. The option has no impact on SJI FITS files. An example call to `read_iris_l2` to read an SJI Level 2 FITS file:

```idl
IDL> sjifile = 'iris_l2_20131025_050530_3880013447_SJI_1400_t000.fits'
IDL> read_iris_l2,sjifile, index, data
IDL> help, index, data
INDEX STRUCT = -> <Anonymous> Array[48]
DATA FLOAT = Array[1214, 1092, 48]
```

where, naturally `index` and `data` are arrays that contain the header information for each raster step and the corresponding spectrograph data.

Using IDL procedure `iris_prep_version_check.pro`, we can check whether the fits files are for the most recent processing, i.e., calibration:

```idl
IDL> calcheck=iris_prep_version_check(index, /loud)
```

where `index` is a fits header and `l2file` is an IRIS level2 fits file (with full path). For this particular example we can therefore run the following command:

```idl
IDL> calcheck=iris_prep_version_check(index, /loud)
```

here `calcheck` is a boolean function, which returns 1 if current data is good, while 0 implies that the fits files should be downloaded again since calibration procedure was improved in the meantime and updated fits files are available. The keyword `/loud` prints extra info when set.
3.5 NUV Data Analysis

3.5.1 Mg II Diagnostics

In the following sub-sections we’ll document a couple of methods to extract physical information from the IRS NUV spectra. These optically thick lines are typically tough to interpret but the IRIS team has done some exploratory work to help the community get as much from the data as possible. The singly ionized Mg II h&k lines ([http://adsabs.harvard.edu/abs/1997SoPh..172..109U](http://adsabs.harvard.edu/abs/1997SoPh..172..109U)) provide information that spans from the photosphere to the upper chromosphere (and possibly as high as the transition region).

The image below shows a comparison synthetic and observed Mg II spectra adapter from the paper by Pereira et al. (2013) The h and k emission cores are typically double-peaked - and can be characterized on the violet ‘V’ or red ‘R’ side of the rest wavelength - see the inset.

IRIS Technical Note 37, as well as the three IRIS diagnostic papers from the ITA/UiO team:

- [http://adsabs.harvard.edu/abs/2013ApJ...772...89L](http://adsabs.harvard.edu/abs/2013ApJ...772...89L)
- [http://adsabs.harvard.edu/abs/2013ApJ...772...90L](http://adsabs.harvard.edu/abs/2013ApJ...772...90L)

provide a comprehensive review of how these parameters can be interpreted in terms of the Bifrost simulations (see IRIS Technical Note 33). The interested IRIS user should consult these papers before studying FUV data in detail. The table below gives a summary of the basic physical properties that can be extracted from the Mg II h&k lines, the bonus being that having two lines that there is some level of comfort in getting consistent measures.

<table>
<thead>
<tr>
<th>Spectral observable</th>
<th>Atmospheric property</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta v_{k3} ) or ( \Delta v_{h3} )</td>
<td>upper chromospheric velocity</td>
</tr>
<tr>
<td>( \Delta v_{k2} ) or ( \Delta v_{h2} )</td>
<td>mid chromospheric velocity</td>
</tr>
<tr>
<td>( \Delta v_{k3} - \Delta v_{h3} )</td>
<td>upper chromospheric velocity gradient</td>
</tr>
<tr>
<td>( k ) or ( k ) peak separation</td>
<td>mid chromospheric velocity gradient</td>
</tr>
<tr>
<td>( k_2 ) or ( h_2 ) peak intensities</td>
<td>chromospheric temperature</td>
</tr>
<tr>
<td>( (I_{k2v} - I_{k2r})/(I_{k2v} + I_{k2r}) )</td>
<td>sign of velocity above ( z(\tau = 1) ) of ( k_2 )</td>
</tr>
</tbody>
</table>

**Note:** The above table shows a simplified view, and all the correlations have scatter.

The codes discussed below provide measures of these properties and a few others.
3.5.2 Mg II Line Peak Information Extraction

Tiago Pereira (ITA/UiO) has developed a piece of IDL software which will permit IRIS users to extract properties of the Mg II h&k lines in the NUV spectra. The code, when given an IRIS Level 2 data file, will return the properties of the red peak, blue peak and central reversals of the Mg II h&k line spectra based on a relatively straightforward peak finding algorithm.

The code, *iris_get_mg_features_lev2* is executed in the following way:

```
IDL> myfile = 'iris_l2_20131013_090250_3821104045_raster_t000_r00000.fits'
IDL> d = iris_obj(myfile)
IDL> d->show_lines
Spectral regions (windows)
0 1335.71 C II 1336
1 1349.43 Fe XII 1349
2 1355.60 O I 1356
3 1393.78 Si IV 1394
4 1402.77 Si IV 1403
5 2832.75 2832
6 2814.47 2814
7 2796.20 Mg II k 2796
IDL> vr = [-40, 40] ; Velocity Range about line center to search for features
IDL> iris_get_mg_features_lev2, myfile, 7, vr, lc, rp, bp
```

Fig. 17: Sample Mg II h/k velocities obtained with *iris_get_mg_features_lev2*.

In this example the results are stored in the lc, rp, bp arrays corresponding to the central reversal, red and blue peaks respectively. Each of these arrays is organized [line, feature, slit position, raster position]. The line index corresponds to Mg II k [0] and Mg II h [1]. The feature index corresponds to Doppler shift [0] and intensity [1]. Bad values are marked with NaN. There are also keyword options for calculating these properties for the Mg II h line (/onlyh) or Mg II k line (/onlyk) only. The images below show the h3 and k3 shift from *iris_get_mg_features_lev2* and are largely although there are differences which, as indicated in the table, provide information about the line-of-sight
component of velocity gradient in the upper chromosphere.

Some of the current limitations of iris_get_mg_features_lev2:

- Single-peaked profiles off-limb don’t work well - the algorithm was designed for double peaked or strongly shifted single peak (i.e., not for the optically thin regime). See the following sub-section for a possible alternative method.

- There are many instances where noisy line profiles can represent many peaks in the spectra. In short exposure observations, or complex regions, this presents the biggest problem to the approach. The IRIS team strongly suggests that the user explore different noise filtering to approaches to avoid these issues and identify robust features in the spectra.

- The line centre properties (k3, h3) are set to NaN when the result is believed to be unreliable. The same setting is used for the peak properties, but it is considerably harder to verify when the peak properties are not reliable, so more ‘dark noise’ will appear in the peak properties.

The ITA/UiO team welcome users to explore, modify and re-share the code given their experiences with it.

### 3.5.3 Mg II Line Variable Component Fitting

Coming Soon: Document application of the Mg II line fitting approach.

### 3.6 FUV Data Analysis

Several IDL codes exist in the IRIS SSW data tree that are dedicated for analysis of FUV spectra. Some of them are described under *IDL Routines for Level 2 Analysis.*
IRIS Level 3 data permits the user to explore the connection between the slit-jaw imagers and the spectral data in one (time-)sequenced FITS file. The primary tool to navigate the Level 3 FITS files is called CRISPEx (the CRISP Spectral EXplorer - developed for the CRISP instrument on the Swedish 1m Solar Telescope on La Palma by Gregal Vissers from the University of Oslo; http://folk.uio.no/gregal/crispex/). The current version is included in the IRIS IDL/SSW distribution so please take time to ensure that all of the Level 3 tools in your path are up to date. This can be done by updating the IRIS IDL SSW package.

4.1 Level 3 Data Structure

Level 3 data can exist in a variety of configurations. For a given observation, it combines multiple level 2 raster files into one or two level 3 files. The user can decide which spectral line windows to include in the level 3 file (e.g. include all the lines, only the NUV lines, or a selection), and so multiple level 3 files can be created from the same level 2 files.

There are two types of level 3 files: im and sp. They contain the same data, but one is the transpose of the other (this is to speed up access for visualization). The files are standard FITS files and the data is written in the primary Header Data Unit (HDU). The im files have the dimensions of (nx, ny, nwave, ntime), while the sp files have the dimensions of (nwave, ntime, nx, ny). Here ny is the number of pixels along the slit, and nx the number of steps in the raster; when rotation is used these are not aligned with solar (x, y) coordinates. For rasters with only one repeat (ntime = 1) the sp files are unnecessary and therefore not created. The different spectral windows of level 2 files are merged into the nwave dimension of the level 3 files.

Besides the primary HDU, level 3 files have three extensions, see below:

<table>
<thead>
<tr>
<th>Ext. No.</th>
<th>Contents</th>
<th>Units</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>Main data</td>
<td>DN</td>
<td>(nx, ny, nwave, ntime)</td>
</tr>
<tr>
<td>1</td>
<td>Wavelength scale (vacuum units)</td>
<td>Å</td>
<td>(nwave)</td>
</tr>
<tr>
<td>2</td>
<td>Time of each exp. since DATE_OBS</td>
<td>Seconds</td>
<td>(nx, ntime)</td>
</tr>
<tr>
<td>3</td>
<td>Location of slit in SJI image</td>
<td>Pixels</td>
<td>(2, ntime, nx)</td>
</tr>
</tbody>
</table>

4.2 Creating Level 3 Data in IDL

IRIS Level 3 FITS files (documented in IRIS Technical Note 21) can be created in two ways (plus through two wrappers):

- Via iris_xfiles
- Via iris_make_fits_level3.pro - power user option. Can also be called through the wrapper iris_obs1223.pro - create level3 files for a given obs
Let's look at an example of each.

Choose a raster file from `iris_xfiles`. You then get a window like this:

![Image of IRIS xfiles main interface](image)

Fig. 1: IRIS xfiles main interface.

In the upper left corner you can choose which lines to include in the level3 file and which SJI cube to use for reference. From the Options pull down menu you can choose the directory for the level3 files. Once the Generate level3 files button is selected, the progress is shown in the SSW terminal window where `iris_xfiles` was started. Using `iris_make_fits_level3`:

```
IDL> cd, getenv('IRIS_DATA') + '/level2/2013/10/07/20131007_075037_3800257135/iris_l2_20131007_07503...
IDL> f = iris_files('*raster*.fits')
IDL> s = iris_files('*SJI*.fits')
```

(continues on next page)
The second argument is the spectral window list that you want to study, in this example we’ve chosen \([0, 3, 6]\), or C II, Si IV, Mg II h&k. The user should employ the `tmp_size` keyword, which sets the max temporary memory size, if a lot more memory than the default `tmp_size` of 12 GB is available. The unit is GB. The raster files go in the first argument, the desired spectral window(s) in the second - use the `/all` keyword instead of the second argument to get all windows, and the reference slit jaw image in `sjifile` (note that currently only one channel at a time is allowed). The `/sp` option produces a (lambda, time, x, y) cube in addition to the default (x, y, lambda, time) cube. This is only done if there is more than one raster file or if it is a sit-and-stare series. The routine will write the Level 3 data to directory `wdir`, default is the current working directory.

There is another optional argument to `iris_make_fits_level3`, called `yshift`. This can be used to correct for situations when the spectra and slit-jaws are not correctly aligned (e.g. issues with the automatic alignment). For more details on this calibration, see *Coalignment between channels and SJI/spectra*.

Looking in the working directory you now have:

```
IDL> f3=iris_files('*{im,sp}*fits')
0 iris_l3_20131007_054001_3800259115_t000_CII1336_SiIV1403_MgIIk2796_im.fits 1 GB
1 iris_l3_20131007_054001_3800259115_t000_CII1336_SiIV1403_MgIIk2796_sp.fits 1 GB
```

**Note:** These two Level 3 files are arranged differently but contain the same information. The `im` file is arranged by (X, Y, lambda, t) while the `sp` file, that is used by CRISPEX in the next section, is ordered (lambda, t, X, Y).

If one is working with many datasets, it may be advantageous to organize the level 3 files in a tree-structure similar to the level 2 files. This is easy to accomplish with the wrapper `iris_obs1223`. The example from above is then achieved with the call:

```
IDL> iris_obs1223, '20131007_054001_3800259115', iwin=[0, 3, 6], $
    /sp, root12=getenv('IRIS_DATA')+'/level2'
```

By default `iris_obs1223` uses the 1400 SJI as reference slit-jaw cube. The level 3 root can be specified with the `root13` keyword. The default is the same as `root12` with the first string `level2` replaced by `level3`. `iris_obs1223` will create the directories necessary and will create symbolic links for the SJI images (with the `iris_xfiles` and `iris_make_fits_level3` methods this has to be done manually).
4.3 Reading Level 3 Data in IDL

The level 3 files can be read in IDL with a regular FITS reader. For example, using `readfits`:

```idl
IDL> data = readfits('iris_l3_20140305_110951_3830113696_t000_all_im.fits', header)
% READFITS: Now reading 400 by 548 by 1410 by 1 array
IDL> help, data
DATA FLOAT = Array[400, 548, 1409]
IDL> wave = readfits('iris_l3_20140305_110951_3830113696_t000_all_im.fits', ext=1)
% READFITS: Reading FITS extension of type IMAGE
% READFITS: Now reading 1409 element vector
IDL> times = readfits('iris_l3_20140305_110951_3830113696_t000_all_im.fits', ext=2)
% READFITS: Reading FITS extension of type IMAGE
% READFITS: Now reading 400 element vector
IDL> help, times
TIMES FLOAT = Array[400]
```

In this example the level 3 file is from a single repeat raster, so the time dimension is collapsed when reading the data. Note that the main headers were read into the variable `header`. While one can also read the extension headers, most of the relevant information is in the main header.

4.4 Browsing Level 3 Data with crispex

4.4.1 Overview

crispex (http://folk.uio.no/gregal/crispex/) is called with an `imcube`, `spcube` (if there are more than one raster or a sit-and-stare series, it is always possible to call crispex with only `imcube`) and (optionally) a slit-jaw cube:

```idl
CRISPEX, imcube, spcube, sjicube=sjicube
```

If you are in the above example level 3 directory and have linked in the slit-jaw cubes (either manually or through running `iris_obsl223`):

```idl
; Exclude raster files if same directory contains any of those:
IDL> f=iris_files('*{im,sp,SJI}*fits')
IDL> f=iris_files() ; enough if only level3 files in directory
0 iris_l2_20131007_054001_3800259115_SJI_1330_t000.fits 57 MB
1 iris_l2_20131007_054001_3800259115_SJI_1400_t000.fits 57 MB
2 iris_l2_20131007_054001_3800259115_SJI_2796_t000.fits 57 MB
3 iris_l3_20131007_054001_3800259115_t000_CI11336_SiIV1403_MgIIk2796_im.fits 1.2 GB
4 iris_l3_20131007_054001_3800259115_t000_CI11336_SiIV1403_MgIIk2796_sp.fits 1.2 GB
;It is then possible to start crispex using the f array
IDL> crispex, f[3], f[4], sji=f[1]
```

You will be greeted by a drop screen giving messages about the progress of the initialization. Then the crispex control panel, main detailed spectrum plot, spectral-T slice and the slit-jaw image will appear (window names written in blue here). The raster FOV for a given timestep and a given spectral position is shown in the control panel. This example dataset is a 4-step raster so it is very narrow and tall. There are eight tabs for the control of the behavior.

The lower part in the control panel stays the same for all tabs. The timestep can be changed with the Frame number slider, the spectral position with the Main spectral position slider. Change the Main spectral position to 866 to get the spectroheliogram in the core of the MgII k-line. Note that a vertical line shows the spectral position in both the detailed spectrum window and in the Spectral T-slice window. When the Frame number is
Fig. 2: crispex main view with a 400-step IRIS raster.
changed, a horizontal line in the Spectral T-slice window shows the position in time. When using the play buttons, the frame number is stepped through. When the cursor is moved around in the raster in the control panel, the Detailed spectrum window shows the spectrum at that position at the time given by Frame number. The Spectral T-slice window shows the temporal behavior of the spectrum at that position. It is possible to lock and unlock the position with the buttons in the control panel. The field below the lock and unlock buttons give values at the cursor position in different units.

Fig. 3: crispex main view with a 4-step IRIS raster.

4.4.2 Tabs

The top part of the side panel is divided into several tabs. Here are short descriptions of their functions:

Temporal-tab. The slit-jaw image is updated to be the closest in time to the time of the raster-step given with the slider Raster timing offset. It is possible to change the master time from Main (described so far) to SJI with the buttons above the Raster timing offset slider. In that case it is the slit-jaw frame number that is stepped by the play buttons and the raster closest in time to the slit-jaw is shown. To increase the speed of the movie in cases where there are many frames, one may change the speed with the Animation speed slider or increase the Frame increment with its slider. It is also possible to restrict the time-range with the boxes at the top of the temporal tab.

Overlays-tab. The slit-jaw image has an overlay showing the position of the raster. This may be switched on and off (and color set etc) from the Overlays tab.

Displays-tab. Here one can switch on and off the various display windows and change the appearance of the Detailed spectrum plot. The lower and upper y-value can be set under the “Plots” tab. Switching on the Spectral-Phi display will show the spectrum along a slit shown in the raster in the control panel. The length of the slit and its orientation can be set in the spectral tab.

Spectral-tab. Here it is possible to set the slit angle, slit length, slowly step the slit and set up a blink between two spectral positions (set the “Position to blink against” first and then the “Reference spectral position”. If there is only
Fig. 4: CRISPEX temporal tab.
Fig. 5: CRISPEX overlays tab.
4.4. Browsing Level 3 Data with **crispex**

Fig. 6: CRISPEX displays tab.
Fig. 7: CRISPEX plots tab.
Fig. 8: CRISPEX spectral tab.
one line selected when producing the level3 file, it is possible to restrict the spectral indices in the top boxes.

**Fig. 9: CRISPEX diagnostics tab.**

**Diagnostics-tab** is used to select which of the lines to show in the various windows (default is all of them). Can also be used to control which detailed spectra of the available lines is shown on main window.

Under **Stokes** tab it is possible to select which element of the Stokes vector is show on main window. In case of IRIS, only intensity $I$ is available.

**Scaling-tab** is used to set the scaling of all the image windows (min, max, gamma) and relative scaling of the various detailed spectrum plots. All line-windows are set separately so only the window where the main spectral position is is affected. The window to control is set in the top drop-down box.

**Spatial-tab** can be used instead of the cursor to set the (x,y) position within the raster.

**Analysis-tab** is used for some limited analysis like calculating space-time diagrams along a path. However, this has not yet been optimized for IRIS use.

This section will be expanded upon to provide the user more detail on:

- movie making,
- space-time plot construction,
Fig. 10: CRISPEX Stokes tab.
Fig. 11: CRISPEX scaling tab.
Fig. 12: CRISPEX spatial tab.
Fig. 13: CRISPEX analysis tab.
By far the best way to learn about crispex is by exploring, so try it. You can find detailed documentation for crispex at the IRIS documentation page and http://folk.uio.no/gregal/crispex/documentation.html
5.1 Wavelength Calibration

The wavelength calibration is automatically performed to the best of current knowledge. This is accurate to only a few pixels, and should be manually checked. There are several photospheric spectral lines that can be used for accurate wavelength calibration, most notably the Ni I 279.9474 nm line in the NUV and the O I 135.560 nm line in the FUV.

Note: The IRIS automatic wavelength calibration is based on these lines averaged over the entire slit. The slit may cover regions of significant line-of-sight flows, such as flux emergence in active regions, and thus these photospheric lines may not necessarily be at their nominal rest wavelengths as assumed by the automatic calibration. In such cases, the user is advised, again, to perform manual wavelength calibration by avoiding such regions under the slit coverage.

A detailed discussion of the wavelength calibration steps for IRIS and how to use them on data can be found in IRIS Technical Note 20.

5.2 Radiometric Calibration

Note: The throughput calibration as a function of wavelength in the FUV passband is subject to significant uncertainty. This uncertainty can significantly impact studies of how the line ratio between Si IV 1394 and Si IV 1402 varies with time throughout the mission.

Note: Due to significant uncertainties in the reference calibration data (from SORCE/SOLSTICE and TIMED/SEE) for the Si IV part of the FUV window, the wavelength-dependent correction for the effective area (for the range between 1388Å to 1408Å) is replaced by a wavelength-independent correction. This decision helps to address the Si IV 1393.8Å/1401.8Å line ratio issue (seen to progressively change with time when using previous versions of the radiometric calibration) preliminarily but with a caveat, i.e., the accuracy of this calibration is likely not sufficient to allow studies of the line ratio of Si IV on long timescales. Differences seen in Si IV ratio of network and internetwork regions might suggest a bias in the FUV background.

The IRIS data are given in counts or Data Number units (DN). To convert these to a flux in physical units (e.g. erg s⁻¹ sr⁻¹ cm⁻² Å⁻¹) one must perform a radiometric calibration. The calibration data is included in the IRIS solarsoft branch, and can be read into an IDL structure with the iris_get_response routine:

IDL> iresp = iris_get_response(time)
where `time` is a string with the time of the observations (compatible with `anytime`). The output has the following structure:

```
IDL> help, iresp, /str
DATE_OBS STRING ''
LAMBDA FLOAT Array[3601]
AREA_SG FLOAT Array[3601, 2]
NAME_SG STRING Array[2]
DN2PHOT_SG FLOAT Array[2]
AREA_SJI FLOAT Array[3601, 4]
NAME_SJI STRING Array[4]
DN2PHOT_SJI FLOAT Array[4]
COMMENT STRING ''
VERSION STRING '003'
VERSION_DATE STRING '20150331'
```

where `AREA_SG` and `AREA_SJI` are the effective areas (in cm\(^{-2}\)) as a function of wavelength (LAMBDA) respectively for the spectrograph and slit-jaw camera. The `DN2PHOT` tags give the conversion from DN counts to photons.

**Note:** By default, `iris_get_response` will apply the most up to date calibration, which is recommended. It is also possible to specify a specific calibration version through the `version` keyword. With early versions the `DN2PHOT` tags are not present.

To convert the spectral units from DN to flux one must do the following:

\[
\text{Flux(erg s}^{-1}\text{cm}^{-2}\text{Å}^{-1}\text{sr}^{-1}) = \text{Flux(DN)} \frac{E_\lambda \cdot \text{DN2PHOT}_\text{SG}}{A_{\text{eff}} \cdot \text{Pix}_{xy} \cdot \text{Pix}_\lambda \cdot t_{\text{exp}} \cdot W_{\text{slit}}},
\]

where \(E_\lambda \equiv h \cdot c/\lambda\) is the photon energy (in erg), `DN2PHOT_SG` is the number of photons per DN (get from `iris_get_response`), \(A_{\text{eff}}\) is the effective area (in cm\(^2\)), \(\text{Pix}_{xy}\) is the size of the spatial pixels in radians (e.g. multiply the spatial binning factor by \(\pi/(180 \cdot 3600 \cdot 6)\)), \(\text{Pix}_\lambda\) is the size of the spectral pixels in Å, \(t_{\text{exp}}\) is the exposure time in seconds and \(W_{\text{slit}}\) is the slit width in radians (\(W_{\text{slit}} \equiv \pi/(180 \cdot 3600 \cdot 3)\)).

A detailed discussion of the radiometric calibration steps for IRIS and how to use them on data can be found in IRIS Technical Note 24.

**Note:** The exposure time \(t_{\text{exp}}\) can be different for each exposure in the same sequence, when Automatic Exposure Control (AEC) is switched on. This is the default for most active region observations, although different exposures are only used in extreme cases when very bright phenomena can lead to CCD saturation (e.g. a flare). The level 2 FITS headers have only one value for the exposure time (the value without AEC). The sequence-dependent exposure times are available in the auxiliary metadata in the FITS files (see Level 2 chapter), with table index given by `EXPTIMEF`, `EXPTIMEN`, and `EXPTIME` for FUV, NUV, and slit-jaw, respectively.

### 5.3 Background in FUV data

FUV spectra with longer exposure times show a faint background most likely caused by a light leak from wavelengths significantly longer than the FUV. This means that the light leak is absorbed at a different CCD depth than the FUV light and thus does not show the same CCD flat-field (which for the FUV is quite prominent and dominated by the CCD annealing pattern). The light leak effectively acts as an extra “dark current” although it appears to have varying intensity levels for different pointings on the Sun. This background has been characterized and is automatically removed by `iris_prep`, and therefore subtracted in level 1.5 and level 2 data.
5.4 Coalignment between channels and SJI/spectra

In level 2 data the slit-jaw images from different filters and detectors are automatically co-aligned. This automatic approach is not failsafe, especially for observations with very short exposure times. Therefore, for precise analysis one should always check if the images match. There are two spectral marks on the slit that are called fiducials and block the light from entering. They are used for calibration, and their position should match between slit-jaw images. With smaller fields of view only one of the fiducials is visible.

**Note:** The position of the slit in different slit-jaw channels is not necessarily the same. Depending on the observing program, different slit-jaw filters may be exposed at different parts of a raster. This is particularly true for two or four step rasters. In such cases the alignment should have in mind the header coordinates from CRPIX and CRVAL.

As in the slit-jaw images, so too the NUV and FUV spectrograms are co-aligned in level 2 data. These too should be checked for the alignment, both between FUV, NUV and slit-jaws. In spectrograms the fiducial marks appear as solid black lines along the wavelength direction, and they should appear in the same exact spatial position for the NUV and FUV channels.

Any misalignment can be corrected for when using option `yshift` in `iris_make_fits_level3`. This option can be set to a 3-element array with the shift to be applied in the y direction to the raster files, where `yshift=[fuv1, fuv2, nuv]` e.g.:

```
IDL> iris_make_fits_level3, f, [0, 3, 6], yshift=[2, 2, 1]
```
Fig. 2: Position of fiducial marks on an NUV spectrogram.
CHAPTER
SIX

IRIS DATA NOTES

6.1 Cosmic rays

IRIS passes through the South Atlantic Anomaly (SAA) on a regular basis. The impact of energetic particles on the CCD camera causes bright hits/pixels. These can be removed with any of the multitude of cosmic ray removal procedures available in solarsoft (e.g., array_despike.pro, tracedespike.pro, nospike.pro, despike.pro, eis_despike.pro, etc...). Cosmic rays are not removed from the IRIS data during normal calibration/pipeline processing to avoid introducing artifacts.

6.2 Particles on slitjaw images

The slit-jaw CCD contains some particles that cause dark regions of order up to a few arcseconds in size in the slit. These features are marked as bad pixels and set to zero values (0) in iris_prep so they can be easily recognized during data analysis. The particles are stable in position and do not let any light through - they are completely dark. They are most prominent in the FUV images (1400Å and 1330Å) and much less visible in the NUV images (2796Å and 2830Å). For SJI timeseries in which the SJI were taken at various positions in a raster a solarsoft routine is being developed to use data values from a previous image to fill in the dark spots.

6.3 CCD camera readout noise

When both spectrograph cameras are read out simultaneously, a read interference noise pattern is superimposed on the resulting data, which can impact the weakest lines in the FUV. The readout noise is only present when the last two digits of the OBSID are less than 50 (for the OBSID generations starting with 38, 40 or 41 numbers). This can be avoided altogether by reading the cameras sequentially, and most of the data is now observed using the sequential read (last two digits in OBSID larger than 50 for OBSID). For OBSIDs starting with 36, those whose 4th digit are 0-4 are sequential read and those with 5-9 are simultaneous readout (3624103603 is sequential and 3629103603 is simultaneous). Also the high-cadence flare OBSIDS 4204700126-4204700143 are simultaneous readout.
6.4 Flagging of saturated data

Some observations show strong solar activity and resulting saturation either on the CCD or (especially in OBS sequences where data is summed) in the A/D converter. Saturated pixels are flagged as Inf so they can be clearly identified.

6.5 Cosmetic finishing in quicklook

The quicklook movies on the IRIS website use a standard set of color tables and intensity scales that have been designed to give a consistent, recognizable and generally pleasing appearance to IRIS observations of a range of solar features. For users who wish to replicate the appearance of off-the-shelf IRIS movies with their own processed data, the IDL code used to apply the scaling and color tables is available through solarsoft. In addition, the scaling routines have some optional parameters that allow the user to tweak the scaling so it is better optimized for a particular observation.

**Note:** The following assumes that IDL is run in indexed color mode (DEVICE, decomposed=0)

6.5.1 Color Tables

The color tables can be loaded as follows:

```
IDL> IRIS_LCT, hdr
```

where `hdr` is a structure containing the FITS header for the image to be displayed (read in using `read_iris_l2`, for example). Alternately, you can provide a string specifying the image path whose color table should be loaded, for example:

```
IDL> IRIS_LCT, 'SJI_1330'
```

Reddish color tables are defined for the IRIS FUV slit-jaw images and spectra (the 1330 SJI channel uses a more yellowish-red than the 1400); yellowish tables are defined for the NUV images and spectra. Byte arrays giving the red, green and blue channels of the color table can be returned by the routine as well:

```
IDL> IRIS_LCT, 'NUV', r, g, b
```

6.5.2 Intensity Scaling

The IRIS data consists of 14-bit pixel values (DN in level 0 data can range from 0-16383; corrections applied during processing to level 2 can move the data values somewhat outside this range); in general, it must be scaled to 8 bits (0-255) for display. The algorithm used for conversion determines the brightness and contrast of the displayed image or movie – the raw data usually has more dynamic range than can be easily displayed, but with appropriate scaling it is possible to maximize the information content of the resulting image. Scaled (8-bit) versions of IRIS data arrays can be generated as follows:

```
IDL> scaledat = IRIS_INTSCALE(dat, hdr)
```

Here `dat` and `hdr` are the results of, e.g., `read_iris_l2`. `dat` is a 3D float or int array of level-2 data (slit-jaw images or spectra), and `hdr` is a scalar or vector structure of FITS headers. The output (`scaledat`) is a byte array with the same dimensions as `dat`. The routine checks the image path (channel) and exposure time from the headers; it is possible to specify those quantities directly rather than passing in the header array.
IDL> scaledat = IRIS_INTSCALE(dat, img_path = 'SJI_1330')

The exposure time is used to apply a relative normalization to the image sequence (e.g. to adjust for AEC triggering during the movie), so if all the frames have the same exposure time it is not necessary to specify it. If you do not want exposure time normalization (you want to see the image darken when the AEC goes off), you can set the /nonorm keyword.

The routine applies logarithmic scaling to the FUV channels, and power-law (gamma) scaling to the NUV channels. The maximum and minimum thresholds are set by examining the image histogram (the maximum is set so that 0.1\% of the pixels are saturated white, and the minimum is set so that 0.2\% of the pixels are saturated black). These are reasonable settings in general, but for certain observations, the thresholds may not be optimal (e.g. perhaps during a particle storm, it may be best to allow a higher fraction of the pixels to be saturated white). They can be adjusted by the user:

IDL> scaledat = IRIS_INTSCALE(dat, hdr, maxfrac=0.995, minfrac=0.003)

### 6.5.3 Cleaning Up

The slit-jaw CCD contains some particles that cause dark regions of order up to a few arcseconds in size in the slit. These features are marked as bad pixels and set to zero values (0) in iris_prep so they can be easily recognized during data analysis. The particles are stable in position and do not let any light through - they are completely dark. They are most prominent in the FUV images (1400Å and 1330Å) and much less visible in the NUV images (2796Å and 2830Å).

These dust spots can be cosmically corrected in slitjaw images and movies using IRIS_DUSTBUSTER. The routine is faster and more effective than general purpose image correction routines because it is able to take advantage of two characteristics of the dust spots:

1. Their location in the CCD frame is well-known, and
2. The IRIS level 2 movies often consist of rasters such that there are usually valid measurements of the solar features obscured by the dust available from previous or subsequent frames.

It can be run as follows:

IDL> cleandat = IRIS_DUSTBUSTER(hdr, dat)

where, again, dat and hdr are the results of, e.g., read_iris_l2; the output cleandat has the same data type and dimensions as dat, with the dust pixels replaced by the average value of the same solar coordinates in nearby frames (for a raster), or by a blurred value of the nearby neighborhood (for a sit-and-stare). Note that the dustbuster should be run prior to intensity scaling, so a quick series of commands to read, set the color table, clean and scale, and display a slit-jaw movie would be as follows:

IDL> read_iris_l2, lev2_sji_file, hdr, dat
IDL> iris_lct, hdr
IDL> ploop, iris_intscale(iris_dustbuster(hdr, dat), hdr)

The dustbuster generally does not try to remove the slit from the SJI movies; if you want to remove it (for example, if you are trying to cross-correlate image sequences without locking on to the slit), set the /slit keyword in the call to iris_dustbuster. Note that the slit contrast has been decreasing in the FUV channels, especially the 1330 SJI, over time; this is not a result of the action of the dustbuster, but an intrinsic property of the data.

---

6.5. Cosmetic finishing in quicklook

[67]
7.1 IRIS xfiles and CRISPEX

This tutorial will guide you step-by-step into some features of `iris_xfiles` and CRISPEX.

Start by downloading an IRIS dataset from 2013 December 26: follow this link and download the three slit-jaw files and the raster file (about 900 Mb in total). Download them to a directory of your choice (here we’ll call it `/~/iris_data/`). Unzip all the files, e.g. on a UNIX system:

```
% gunzip *.fits.gz
% tar zxvf *.tar.gz
```

Start a solarsoft IDL session, and then launch `iris_xfiles`:

```
% sswidl
(...)
IDL> iris_xfiles
```

In the middle panel, next to **Search Directory**: press **Change**. Then navigate to the directory `/~/data_temp/iris/20131226_171752_3840007146`, press **OK** when this directory is selected. Notice that **Search Pattern** changed to **free search**. Make sure the time of these observations (2013 December 26, 17:17 UT) is contained in between the **Start Time** and **Stop Time** range, adjust if necessary.

Now press **Start Search**, and on the middle panel you will see a row with a summary of the observations, and a list of files on the bottom panel.

If you double click on the slit-jaw files, you can see a movie with `Ximovie`. In this case the observations are a 400-step raster of an active region. Because the slit-jaw images have been aligned to fixed coordinates on the Sun, the images are moving from left to right and black bars appear in the regions which are not exposed (the images are enlarged to fit the maximum extent of the observing sequence). With `Ximovie` you can adjust the display limits and also apply a gamma to the images.

If you double click on the raster file, an `iris_xcontrol` window will appear. This has a lot of information about the particular observing sequence, and displays some sample spectra and slit-jaw images. The middle spectral panel is split in the NUV (green-white) and FUV (red-white) spectrograms. You can also identify which spectral regions were observed. Clicking on either spectrogram will let you inspect individual spectrograms for all the steps in the sequence. You can also do the same with the SJI images.

Now we’ll use the profile moments tool. On the top left panel select the Mg II k 2796 line and under **Line fit** select **Profile Moments**. On the **Moments Prep Tool** window adjust the reference wavelength so that it matches the k3 core, and set the line start and stop so that it is about 5 pixels wide around k3. Set the continuum to be about 5 pixels wide on the right side of the plot, in a region with no absorption line, like the figure below.

Press **Finished**. It will take a while to calculate the moments statistics in the region defined, and a new window opens. The result for intensity should look like the figure below.
Fig. 1: How the wavelength selection  \texttt{iris\_xfiles} moment tool should look like.
Fig. 2: How the intensity from the k3 first moment should look like.
This looks very much like the Mg II k3 intensity, because we selected a very narrow window around it. On the bottom left side of the iris_xmap window you can also change Select data to show the first moment (centroid) velocity, but in this case this is not so reliable because we chose a small wavelength window.

Now to the same in Profile moments but select the k2v feature instead of k3. The result for intensity should look like the image below.

You can also experiment with the many other options of iris_xfiles as a quick way to explore the data.

Now we are going to create level 3 files from this set to use with CRISPEX. Go back to the IRIS_Xcontrol window of the raster file. Select the Si IV 1403 and Mg II k 2796 lines and press Generate level3 files. In the next dialog to not select anything and just press OK. We will use this level 3 file next with CRISPEX. Note that only one file will be created, as the time domain is missing.

Now start CRISPEX and use the name of the file you just created as an argument. Assuming you are in the same directory as the file, do:

```
IDL> crispex, 'iris_l3_20131226_171752_3840007146_t000_SiIV1403_MgIIk2796_im.fits'
```

It will take a while to start as CRISPEX calculates mean spectra and scaling factors. If everything went well you can start exploring the data. The main window will look black. This is because CRISPEX takes the first wavelength in the continuum region of the Si IV 1403 line. The intensity here is mostly noise, so it appears as black. You can see the spectral windows in the other two windows: detailed spectrum and Spectral phi-slice.

Change the Main spectral position slider to around position 205. You can see on the spectra phi-slice that a vertical gray line moves. This indicates the current position in the spectrogram. Now you see more structure in the main image, but it is still a little dark. You can change the scaling in the Scaling tab: set the Histogram optimisation to 0.01 (this means that the scaling will be set from the 1% to 99% percentiles), or change the gamma to a lower value. On the Detailed spectrum window you still see only a vague outline of spectral lines. You can change this scaling in the Displays tab: set Lower y-value to -0.01 and the Upper y-value to 0.1. Now you should see the Mg II k line well, but the Si IV line is only visible for the brightest points.

When you move the mouse the windows update with the spectrum at the mouse pointer location. The Spectral phi-slice window shows a spectrogram with y being the position along a slit (the vertical white line that moves with the mouse cursor). If you click with the left button on a point it locks it. To unlock it you need to select the option Unlock from position on the middle of the main panel. Below it you can also see some properties of the current position: coordinates, wavelength being show, time, data values, etc.

You can change the parameters of the slit made to construct the Spectral phi-slice window. On the Spectral tab, middle part, there are controls named Slit controls. There you can adjust the angle and length of the slit, and see how the window changes.

### 7.2 Mg II Dopplergrams

In this tutorial we are going to produce a Dopplergram for the Mg II k line from an IRIS 400-step raster. The Dopplergram is obtained by subtracting the intensities at symmetrical velocity shifts from the line core (e.g. ±50 km/s). For this kind of analysis we need a consistent wavelength calibration for each step of the raster.

Start by downloading an IRIS dataset from 2014 July 8: follow this link and download the raster file (726 Mb). Download it to a directory of your choice. Untar it, e.g. on a UNIX system:

```
% tar zxfv iris_l2_20140708_114109_3824262996_raster.tar.gz
```

Feel free to examine these data in iris_xfiles. This very large dense raster took more than three hours to complete the 400 scans (30 s exposures), which means that the orbital velocity and thermal drifts were changed during the observations. This means that any precise wavelength calibration will need to correct for those shifts.
Fig. 3: How the intensity from the k2v first moment should look like.
First let's start sswidl and load the data using the IDL object interface:

```idl
% sswidl
(...)
IDL> filename = 'iris_l2_20140708_114109_3824262996_raster_t000_r00000.fits'
IDL> d = iris_obj(filename)
```

Let us see the spectral windows that are saved in this raster:

```idl
IDL> d->show_lines
Spectral regions(windows)
0 1335.71 C II 1336
1 1349.43 Fe XII 1349
2 1355.60 O I 1356
3 1393.77 Si IV 1394
4 1402.77 Si IV 1403
5 2832.70 2832
6 2814.43 2814
7 2796.20 Mg II k 2796
```

Let us load the Mg II lines into memory:

```idl
IDL> wave = d->getlam(7)
IDL> data = d->getvar(7, /load)
```

We can see how the spatially averaged spectrum looks like:

```idl
IDL> mspec = total(total(data, 2), 2)
IDL> plot, wave, mspec
IDL> plot, wave, mspec, xrange=[2794, 2799], /xst
```

To better understand the orbital velocity problem let us look at how the line intensity varies for a strong Mn I line at around 280.2 nm, in between the Mg II k and h lines. For this dataset, the line core of this line falls around index 350.

To plot it in the correct orientation we will make use of IDL's `transpose`, and the procedure `pih` (available in the IRIS tree of solarsoft) to make the plot:

```idl
IDL> pih, transpose(reform(data[350,*,*])), min=0, max=200, scale=[0.35,0.1667]
```

The result should look like this:

You can see a regular bright-dark pattern along the x axis, and indication that its intensities are not taken at the same position in the line because of wavelength shifts.

To calculate the wavelength shifts from the orbital velocity and thermal drifts we do the following:

```idl
IDL> wavecorr = iris_prep_wavecorr_l2(filename)
```

This routine measures the wavelength position of 5 neutral lines (3 NUV, 2 FUV) whose rest wavelengths are reasonably well known, and saves the shifts (in Ångström) into the structure variable called `wavecorr`. The structure format is as follows:

```idl
** Structure <484f608>, 14 tags, length=3552144, data length=3552140, refs=1:
NOTES STRING 'corrs[file, rasterstep, line] gives target_wave - measured_wave in angstroms'
FILES STRING 'iris_l2_20140708_114109_3824262996_raster_t000_r00000.fits'
LNAME STRING Array[5]
WAVERDS FLOAT Array[5]
```
Fig. 4: Intensity at Mn I 280.2 nm line when orbital velocity and thermal drifts are not accounted for.
TIMES STRING Array[1, 400]
TAIS DOUBLE Array[1, 400]
NX LONG 400
NY LONG 1094
COORD FLOAT Array[1, 400, 1094, 2]
CORRS DOUBLE Array[1, 400, 5]
CHISQ DOUBLE Array[1, 400, 5]
CORR_TAI DOUBLE Array[400]
CORR_NUV DOUBLE Array[400]
CORR_FUV DOUBLE Array[400]

Note: A vector of level 2 fits raster files can also be passed to `iris_prep_wavecorr_l2` instead of a single file; this example uses a single raster, so it has a single file, and as a result the leading dimension is 1 for the `corrs`, `chisq`, `times` and `tais` fields in the `wavecorr` structure.

The five spectral lines that have been measured are as follows:

```idl```
for i = 0, 4 do print, wavecorr.lname[i], wavecorr.wave0s[i]
Ni I 2799.47
Mn I 2801.90
Fe I 2805.35
O I 1355.60
Fe II 1392.82
```

The wavelength shift in the i-th line in the j-th frame of the k-th input file is stored in `wavecorr.corrs[k, j, i]`. Averaged and smoothed corrections for the NUV and FUV are stored in `corr_nuv` and `corr_fuv`. To plot the per-frame measured wavelength shift and the smoothed correction in the NUV and FUV one would do:

```idl```
!p.multi = [0,1,2]
utplot, wavecorr.times, wavecorr.corrs[0,*,0], psym = 4, chars = 1.5, title = 'NUV', ytitle = 'Wavelength shift [Å]', /xst
outplot, wavecorr.times, wavecorr.corr_nuv, col = 2, thick = 2
utplot, wavecorr.times, wavecorr.corrs[0,*,3], psym = 4, chars = 1.5, title = 'FUV', ytitle = 'Wavelength shift [Å]', /xst, yrange = [-0.1, 0]
outplot, wavecorr.times, wavecorr.corr_fuv, col = 2, thick = 2
!p.multi = 0
```

Note: The smoothing of the wavelength variation assumes the IRIS orbital period. Shorter-timescale variations are sometimes evident; you can apply the shift of the individual measurements, or apply your own smoothing to the per-image measurements rather than using the smoothed `corr_nuv` and `corr_fuv` arrays if necessary.

To look at intensities at any given scan we only need to subtract this shift from the wavelength scale, but to look at the whole image at a given wavelength we must interpolate the original data to take this shift into account. Here is a way to do it (note that array dimensions apply to this specific set only!):

```idl```
new_data = fltarr(536, 1092, 400, /n)
for i=0, 399 do begin
  for j=0, 1091 do begin
    new_data[*, j, i] = interpol(data[*, j, i], wave + wavecorr.corr_nuv[i], wave)
  endfor
endfor
```

76 Chapter 7. Tutorials
Fig. 5: Fit to the orbital velocity/thermal shifts from iris_prep_wavecorr_l2.
A User’s Guide to IRIS Data Retrieval, Reduction & Analysis, Release 1.0

(This double loop may take a while to complete.)

Once you have the calibrated data, we can compare again how it looks at the Mn I line wavelength:

```idl
IDL> pih, transpose(reform(new_data[350,*,*])), min=0, max=200, scale=[0.35,0.1667]
```

And now we can see that the intensity map is uniform along the solar disk:

We can use this calibrated data for example to calculate dopplergrams. A dopplergram is the difference between the intensities at two wavelength positions at the same (and opposite) distance from the line core. For example, at +/- 50 km/s from the Mg II k3 core. To do this, let us first calculate a velocity scale for the h line and find the indices of the -50 and +50 km/s velocity positions (here using the convention of negative velocities for up flows):

```idl
IDL> k_centre = 2796.32 ; mean position of k3
IDL> vel = (wave - k_centre) * 3e5 / k_centre
IDL> tmp = min(abs(vel - 50), i50p) ; find index of -50 and 50 km/s
IDL> tmp = min(abs(vel + 50), i50m)
```

Now get the Dopplergram and plot it:

```idl
IDL> doppgr = transpose(reform(new_data[i50m, *, *] - new_data[i50p, *, *]))
IDL> pih, doppgr, min=-200, max=200, scale=[0.35, 0.1667]
```

### 7.3 Mg II spectral feature identification

In this tutorial we will measure the intensities and velocity shifts of the Mg II k3 and k2 features. We will make use of the iris_get_mg_features_lev2 procedure, which is included in the IRIS SSW package.

Here we will use the same dataset as for the tutorial IRIS xfiles and CRISPEX above. If you haven’t done so, start by downloading an IRIS dataset from 2013 December 26: follow this link and download the raster file (491 Mb). Download it to a directory of your choice. Unzip it, e.g. on a UNIX system:

```
% tar xzvf iris_l2_20131226_171752_3840007146_raster.tar.gz
```

We can calculate the properties of the Mg II k line in the following manner:

```idl
IDL> filename = 'iris_l2_20131226_171752_3840007146_raster_t000_r00000.fits'
IDL> iris_get_mg_features_lev2, filename, 3, [-40, 40], lc, rp, bp, /onlyk
```

(This will take a while.)

The output is saved in the arrays lc (line centre), rp (red peak), and bp (blue peak). To save time, we calculated only for the k line. We can then visualise both the derived velocities and intensities. For the intensities:

```idl
IDL> pih, transpose(reform(lc[0,1,*,*])), min=0, max=500, scale=[0.35,0.1667]
IDL> pih, transpose(reform(bp[0,1,*,*])), min=0, max=750, scale=[0.35,0.1667]
IDL> pih, transpose(reform(rp[0,0,*,*])), min=0, max=750, scale=[0.35,0.1667]
```

and for the velocities:

```idl
IDL> pih, transpose(reform(lc[0,0,*,*])), min=-15, max=15, scale=[0.35,0.1667]
IDL> pih, transpose(reform(rp[0,0,*,*])), min=0, max=30, scale=[0.35,0.1667]
IDL> pih, transpose(reform(bp[0,0,*,*])), min=-30, max=0, scale=[0.35,0.1667]
```

As you can see, the code is not perfect at finding the positions of the spectral features in all pixels (see obvious black and white isolated pixels). Instead, it provides a reasonable estimate when the line profiles are well behaved, and a starting point to further analysis.
Fig. 6: Intensity at Mn I 280.2 nm line when orbital velocity and thermal drifts are accounted for.

7.3. Mg II spectral feature identification
Fig. 7: Dopplergram for Mg II k at +/- 50 km/s.
Fig. 8: Intensity for the k3 peak from `iris_get_mg_features`.

7.3. Mg II spectral feature identification
Fig. 9: Velocity shifts for the k3 peak from `iris_get_mg_features`.
7.4 Time series analysis

In this tutorial we are going to work with spectra and slit-jaw images to study dynamical phenomena. The subject of this example is umbral flashes.

Start by downloading this IRIS dataset from 2013 September 2. Download the three slit-jaw files and the raster file (about 1 Gb in total) to a directory of your choice. Unzip all the files, e.g. on a UNIX system:

```
% gunzip *.fits.gz
% tar zxvf *.tar.gz
```

Start `sswidl` and use the `iris_files` function to get a list of all the IRIS FITS files:

```
IDL> f = iris_files('./*')
0 iris_l2_20130902_163935_4000255147_SJI_1330_t000.fits 123 MB
1 iris_l2_20130902_163935_4000255147_SJI_1400_t000.fits 123 MB
2 iris_l2_20130902_163935_4000255147_SJI_2796_t000.fits 123 MB
3 iris_l2_20130902_163935_4000255147_raster_t000_r00000.fits 1 GB
```

You can now start an `iris_obj` with this list of files, but it must be reversed because `iris_obj` expects the spectral raster first:

```
IDL> d = iris_obj(reverse(f))
IDL> d->show_lines
Spectral regions(windows)
0 1335.71 C II 1336
1 1349.43 Fe XII 1349
2 1351.66 Cl I 1352
3 1355.60 O I 1356
4 1393.78 Si IV 1394
5 1402.77 Si IV 1403
6 2786.14 2786
7 2796.20 Mg II k 2796
8 2830.93 2830
```

We can now get the data and wavelength for the Mg II and C II windows, and the array of times (since the start of the observations):

```
IDL> data_c = d->getvar(0, /load)
IDL> wave_c = d->getlam(0)
IDL> data_mg = d->getvar(7, /load)
IDL> wave_mg = d->getlam(7)
IDL> times = d->gettime()
```

For this dataset the spectral cadence is about 3 seconds. The Mg II k3 core is located around wavelength pixel 103. We can use this information to make a space-time image of the Mg II k3 wavelength:

```
IDL> pih, transpose(reform(data[103,*,*])), min=0, max=200, scale=[3./60.,0.1667]
```

The result can be seen below, with the x axis in minutes and the y axis in arcsec.

The middle section between 30"-40" is on the umbra of a sunspot, even though it is not obvious from this image. One can see very clearly the umbral oscillations, with a clear regular pattern of dark/bright streaks. Let us now load the 1400 slit-jaw and plot it for context:

```
7.4. Time series analysis 83
```
Fig. 10: Space-time diagram of the intensity at around the Mg II k3 core.
The slit pixel 220 is a location on the sunspot’s umbra. We will use it to get some plots. For example, let’s plot the k3 intensity (spectral pixel 103 of data_mg) and the core of the brightest C II line (spectral pixel 90 of data_c) vs. time in minutes (showing first 10 minutes only):

```
IDL> plot, times/60., data_mg[103, 220, *], xrange=[0, 10], yrange=[0, 80], $ 
   /xst, /yst
IDL> oplot, times/60., data_c[90, 220, *] * 2, linestyle=1
```

In the above we are multiplying the C II data by two to get the two lines closer. Image now you wanted to compare these oscillations with the intensity on the slit-jaw. How to do it? The slit-jaws are typically taken at a different cadence, so you will need to load the corresponding time array for the 1400 slit-jaw:

```
IDL> times_sji = d->get_time_sji(1)
IDL> help, times_sji
TIMES_SJI DOUBLE = Array[400]
```

This is the time in seconds since the start of the observations, so comparable to the array times, which holds the same quantity for the spectra. Armed with this, we can now plot the SJI intensity for a pixel close to the slit at the same y position (index 220):

```
IDL> oplot, times_sji/60., sji[220, 190, *], linestyle=2
```

The resulting plot can be found below. You are now ready to explore all the correlations, anti-correlations, and phase differences.

### 7.5 Other tutorials

Notes and videos are also available from tutorials held at IRIS workshops. Some of these are based on the above tutorials and focused on IRIS data, while others cover related topics such as radiative transfer and analysing output from Bifrost simulations:

#### 7.5.1 Tutorials from IRIS-4 workshop

- Introduction to IRIS data analysis: [video](#) and [materials](#)
- Bifrost simulations: [video](#) and [materials](#)
- Optically Thick Line Formation and Interpretation of IRIS Observables: [video](#) and [materials](#)

#### 7.5.2 Tutorials from IRIS-7 workshop

- Introduction to IRIS data analysis: [materials](#)
- Bifrost simulations: [materials](#) and [materials](#)
- Operation of IRIS: [materials](#)
- Flare simulations: [materials](#)
- UV Spectroscopy: [materials](#)
Fig. 11: First SJI 1400 image of the observing sequence.
Fig. 12: Intensity vs. time in minutes for an umbral flash oscillation, for Mg II k3 (solid), C II 1335 core (dotted) and 1400 slit-jaw (dashed).
CHAPTER
EIGHT

USEFUL CODES

In this chapter we list a series of utility codes developed by the IRIS team that may serve as a basis for further investigations. The user is encouraged to use and study them, being aware of any assumptions and simplifications.

8.1 IRIS_getAIAdata

The routine iris_getaiadata is a graphical tool to download the AIA data that corresponds to a IRIS observation. The routine creates SJI-like level 2 files with the FOV fitted to the IRIS observation.

It is called in the following way:

```
IDL> iris_getaiadata, id
```

where id is a string that identifies a specific IRIS observation from the level 2 files. It can have three different forms:

- id = 'iris_l2_20131022_042024_3882010144_SJI_1330_t000.fits' (IRIS level 2 file)
- id = '/mn/xsan/d2/iris/data/level2/2014/02/01/20140201_090500_3880012095/' (directory name where a level 2 file exists)
- id = '20140410_051024_3882010194' (string with date and IRIS OBSID, for users that have a directory structure that mimics the IRIS data path, or a local data mirror)

Calling the routine will open a window with an overview of the parameters. One should choose an output directory where the AIA data will be saved, which AIA channels to save, and then confirm the values for XCEN/YCEN and FOVX/FOVY (a bug can sometimes result in them being incorrectly read). One can also extend the FOV and the time window. Finally, one can change the maximum number of files that will be downloaded per wavelength; the approximate temporal resolution is calculated whenever this number is changed. The highest cadence of AIA data is 12 sec, but this time interval can be larger.

There are three options for downloading the AIA data:

1. **SSW_cutout**: This method downloads the AIA data via ssw_service, it first sends a request for a cutout of the data according to a IRIS OBS. This request is processed at LMSAL and can take from 5 min to a few hours. The download itself can also relatively slow. This method is recommended for large requests, when the full disk images will take long to download.

2. **Full disk images**: The full disk AIA data is first downloaded and then prepared locally. The files to download are very large, but the download speed can be higher when compared to ssw_cutout, and there is no waiting time for the data. This method is recommended for small and medium requests, for users with a good internet connection and plenty of disk space.

3. **Archive**: Does not download any data, assumes that the user has local access to /archive/sdo/AIA/lev1/. The preparation is done directly on the archived files. Max number frames is ignored, all available files are used. This mode is useful only for users at LMSAL.
Fig. 1: iris_getAIAdata dialog.
After clicking on “Request Data” the window disappears, and the request is sent. Every few seconds the routine checks if the data are ready for download. At this point the tool can be interrupted, and restarted at a later time with:

```
IDL> IRIS_processAIArequest, outdir=outdir
```

The output of this routine will be one or more files similar in format to IRIS SJI level 2 FITS files, one file for each AIA channel. The header is the same as in the SJI FITS files, except that not all keywords are populated. The extensions of the FITS files are also in the same format as the SJI files, again, not all values are populated, because not all values are available in the AIA data. The data in the resulting fits files have the original AIA resolution, but are rotated to the same rotation as the IRIS data.

### 8.2 IRIS file handling routines

Several routines have been developed by Peter Young at NRL to handle IRIS files for users who don’t have a local mirror of the IRIS data. They are available in the IRIS Solarsoft directory `idl/nrl`. See below for more details.

#### 8.2.1 The `$IRIS_DATA` environment variable

This variable should point to where the user’s data files are stored, and it should be set in the `.cshrc` (or equivalent) or `idl_startup.pro` file. The variable can contain multiple paths. Example for `idl_startup.pro` file:

```
setenv, 'IRIS_DATA=$HOME/iris_data:/Volumes/DATA_DISK'
```

In this example the user has a data directory in his home directory, and another one on an external data disk. The software will search for files first in the home directory and then on the data disk.

#### 8.2.2 Ingesting files

With `$IRIS_DATA` one can automatically transfer an IRIS file into the correct location within the IRIS directory tree using the routine `iris_ingest.pro`:

```
IDL> iris_ingest, local_file [, index=index]
```

By default the routine will put the file in the first directory path listed in `$IRIS_DATA`. To put it in another path, one can use the optional input `index=index`. For example, `index=1` will put it in the second path name (i.e., on the data disk in the example above). If you are not sure what the index numbers are, then do:

```
IDL> iris_ingest, /help
```

#### 8.2.3 Finding files

With files correctly placed in the directory tree, one can find them from IDL using `iris_find_file.pro`. For example:

```
IDL> file = iris_find_file('29-Mar-2014 17:00')
```

By default this routine returns only the raster files, not the SJI files. To return the SJI files, use the keyword `/sji`. If the observation consists of multiple raster repeats, then the names of all of the individual files will be returned.
8.2.4 Finding a matching SJI file(s)

To find the SJI files corresponding to a given raster file, one can do:

```idl
idl> sji_files = iris_sji_match(file)
```

8.3 IRIS dustbuster

The function `iris_dustbuster` cleans up the dust specs on the level 2 FUV SJI movies by picking good pixels from frames that are adjacent in time (so it works better on rasters, not sit-and-stares, though it’s safe to run on anything).

```idl
definition

functioanl iris_dustbuster, l2index, l2data, bpaddress, clean_values

Parameters

• l2index – Index structure from a L2 SJI FITS file (read in using read_iris_l2)
• l2data – Data cube from a L2 SJI FITS file (read in using read_iris_l2)
• bpaddress – An N-element LONG array specifying the (1-D) address of the bad pixels in the L2 data cube
• clean_values – An N-element FLOAT array giving the pixel values to poke into the bad pixels

Returns: clean_data, A data cube with the same dimensions as l2data, with the dust busted (note that if the /list_only keyword is set, then only a scalar integer is returned (the user can do the replacement separately)

Usage:

cd, '/irisa/data/level2/2014/04/26/20140426_010036_3820257468/'
read_iris_l2, 'iris_l2_20140426_010036_3820257468_SJI_1400_t000.fits', l2index, l2data
ctime = IRIS_DUSTBUSTER(l2index, l2data, bpaddress, clean_values, /run)
```

8.4 IDL Routines for Level 2 Analysis

These codes are available in the IRIS SSW tree. They work with level 2 data.

```idl
pro iris_ne

Purpose: Derive the electron density from line pair intensity ratio and plot the result.

Required: The theoretical density-ratio files can be generated using CHIANTI:

density_ratios, 'o_4', 1398., 1402., 7., 13., den, rat, desc
d = interpol(alog10(den), 200, /spline)
r = interpol(rat,200,/spline)
save, filename='o4_1399to1401_den.sav', d, r
```

Usage:

```idl
int1 = [116., 56., 40.] ; three intensities of O IV 1399Å
err1 = [17., 4., 5.] ; the corresponding measurement error
int2 = [482., 155., 200.] ; three intensities of O IV 1401Å
err2 = [25., 12., 10.] ; the corresponding measurement error
```
pro \texttt{iris\_ne\_oiv}

\textbf{Purpose}: Derive the electron density from the O IV 1401Å and 1399Å line pair.

\textbf{Required}: The theoretical density-ratio data is included in the code. So users do not need to retrieve this data from CHIANTI.

\textbf{Usage}:

\begin{verbatim}
int1 = [116., 56., 40.] ; three intensities of O IV 1399Å
err1 = [17., 4., 5.] ; the corresponding measurement error
int2 = [482., 155., 200.] ; three intensities of O IV 1401Å
err2 = [25., 12., 10.] ; the corresponding measurement error
\end{verbatim}

\pro \texttt{iris\_te}

\textbf{Purpose}: Derive the electron temperature from line pair intensity ratio and plot the result. Joint observation between IRIS and EIS can be used to diagnose the electron temperature. Note that O IV 279.93Å/1401.16Å has little density sensitivity. Fe XII 195.12Å/1349.40Å has some density sensitivity but the density can be determined from EIS Fe XII line pairs.

\textbf{Required}: The theoretical density-ratio files can be generated using CHIANTI. For O IV:

\begin{verbatim}
temperature\_ratios, 'o\_4', 279, 1402, 4.5, 6.0, temp, rat, desc
t = interpol(alog10(temp), 250, /spline)
r = interpol(rat, 250, /spline)
save, filename='o4_279to1401_temp.sav', t, r
\end{verbatim}

For Fe XII (need to specify the density):

\begin{verbatim}
temperature\_ratios, 'fe\_12', 195, 1350, 5.5, 7.0, temp, rat, desc, density=10^8.5
t = interpol(alog10(temp), 250, /spline)
r = interpol(rat, 250, /spline)
save, filename='fe12_195to1349_temp_n8.5.sav', t, r
\end{verbatim}

\textbf{Usage}:

\begin{verbatim}
int1 = [100., 200.] ; two intensities of Fe XII 195Å
err1 = [12., 7.] ; the corresponding measurement error
int2 = [1., 2.] ; two intensities of Fe XII 1349Å
err2 = [0.05, 0.12] ; the corresponding measurement error
temp\_file = './NeTe/fe12_195to1349_temp_n8.5.sav'
\end{verbatim}

\pro \texttt{sgf\_rbp\_1lp}, \texttt{wvl}, \texttt{lp}, \texttt{ee}

\textbf{Parameters}:

\begin{itemize}
  \item \texttt{wvl} – wavelength vector
  \item \texttt{lp} – line profile
  \item \texttt{ee} – error vector
\end{itemize}

\textbf{Purpose}: Perform a single Gaussian fit to an optically thin emission line profile and derive an “RBp” profile as a function of velocity (this is a modified version of the “RB” line profile asymmetry analysis originally developed by De Pontieu et al. (2009, ApJ, 701, L1.) See the definition in Section 2 of Tian et al. (2011, ApJ, 738, 18.).
pro dgf_1lp, wvl, lp, ee, ini, range0, range1

Parameters
• wvl – wavelength vector
• lp – line profile
• ee – error vector
• ini – initial guess
• range0 – allowed ranges of first component
• range1 – allowed ranges of second component

Purpose: Perform a double Gaussian fit to an optically thin emission line profile by supplying the initial guess of intensity ratio, 2nd component speed and width as well as the allowed ranges of the parameters for the two components.

Usage:
ini = [0.2, -50, 30]
range0 = [0.5, 2, 1]
range1 = [0.8, 2, 1]
dgf_1lp, wvl, lp, ee, ini, range0, range1

pro dgf_rb_1lp, wvl, lp, ee

Parameters
• wvl – wavelength vector
• lp – line profile
• ee – error vector


function gen_rb_profile_err, v, profile, err, steps, dv

Parameters
• v – vector of velocity from line centroid
• profile – line profile
• err – vector of measurement error at different spectral positions
• steps – velocity steps, e.g., [10, 20, 30, 40, ...]
• dv – size of velocity bin, e.g., 20 km/s

Purpose: Get red wing and blue wing intensities of an optically thin emission profile as a function of velocity (spectral distance from line centroid). The result will be used to build an RB asymmetry profile.

pro tgf_1lp, wvl, lp, ee, ini, range0, range1, range2

Parameters
• wvl – wavelength vector
• lp – line profile
• ee – error vector
**Purpose**: Do triple Gaussian fit to an optically thin emission line profile by supplying the initial guess of 2nd/core intensity ratio, 2nd component speed and width, 3rd/core intensity ratio, 3rd component speed and width, as well as the allowed ranges of the parameters for the 3 components.

**Usage**:

```idl
ini = [0.2, -50, 30, 0.05, 80, 20]
range0 = [0.5, 2, 1]
range1 = [0.8, 2, 1]
range2 = [0.8, 2, 1]
tgf_1lp, wvl, lp, ee, ini, range0, range1, range2
```

**pro iris_nonthermalwidth**

**Purpose**: Compute thermal and nonthermal widths in the unit of km/s.

**Usage**:

```idl
Wobs_v = 28.0 ; observed (1/e) line width in km/s
instr_fwhm = 0.026 ; instrumental FWHM in Å
Wnt_v = iris_nonthermalwidth('Si', 'IV', 1402.77, Wobs_v, instr_fwhm, ti_max=ti_˓→max, Wt_v=Wt_v)
```

**Notes**: According to the IRIS paper, the spectral resolution (FWHM) is 26 mÅ in the FUV and 53 mÅ in the NUV.

**pro iris_orbitvar_corr_l2, file, dw_orb_fuv, dw_orb_nuv, date_obs, (...)**

**Parameters**

- **file** – Level 2 file name
- **dw_orb_fuv** – the correction vector for orbital variation in FUV. Both the thermal and S/C velocity components are included. The unit is Ångström. (output)
- **dw_orb_nuv** – the correction vector for orbital variation in NUV. Both the thermal and S/C velocity components are included. The unit is Ångström. (output)
- **date_obs** – the vector of observation times
- **dw_th** – the thermal component of the orbital variation derived by using the Ni I 2799.474 Å (vacuum wavelength) line. The unit is unsummed wavelength pixel (about 0.0256 Ångström for NUV, 0.013 Ångström for FUV) (output)
- **dw_sc** – the spacecraft velocity (along the Sun-IRIS line) component of the orbital variation, positive value means the Sun is moving away from IRIS. The unit is km/s. (output)
- **abswvl_nuv** – the amount (unit Ångström) that has to be subtracted from the wavelengths if you want to do absolute wavelength calibration for NUV. (output)

**Purpose**: Purpose: Make corrections to the FUV/NUV spectral images for the spacecraft orbital variation of the spectral line positions using the Ni I 2799.474Å line for IRIS Level 2 FITS files. The input is a Level 2 spectrograph FITS file name. The *iris_orbitvar_corr_l2* deals with a list of files.

**Notes**: **dw_orb_fuv, dw_orb_nuv & date_obs** saved into the file of *datetime'*_orbitvar. genx', can be loaded using *restgen, dw_orb_fuv, dw_orb_nuv, date_obs, file=datatime'*_orbitvar.genx'.
**Note:** The orbital variation (both the thermal component and the spacecraft velocity component) has been subtracted in the level 2 data generated from the May 2014 pipeline. The thermal component was evaluated using an empirical relationship between roll, temperature and line positions. In principle no further orbital variation correction is needed and the users can just use the level 2 data downloaded from the IRIS data website. The routine `iris_orbitvar_corr_l2` can be used to correct for any data that has eventually not been properly calibrated.

**pro iris_obs2hcr**

**Purpose:** This function maps from IRIS OBSIDs, timelines and/or time range to create IRIS Heliophysics Coverage Registry (HCR) records [http://www.lmsal.com/solarsoft/ssw/idl/lmsal/util/iris_obs2hcr.pro](http://www.lmsal.com/solarsoft/ssw/idl/lmsal/util/iris_obs2hcr.pro) (e.g., this is a wrapper for the general purpose `ssw_hcr_query.pro` suite of SSW codes). See the above URL for the doc-header for calling options and the `ssw_hcr_query` context example.

**pro iris_obs2ssw_cutout**

**Purpose:** A procedure to call the above - pulls IRIS-coordinated SDO images. Using IRIS HCR info, sets up and calls the SSW cutout service [http://www.lmsal.com/solarsoft/ssw/idl/lmsal/util/iris_obs2ssw_cutout.pro](http://www.lmsal.com/solarsoft/ssw/idl/lmsal/util/iris_obs2ssw_cutout.pro)

**Usage:**

```idl
IDL> iris_obs2ssw_cutout, '20140610_072252_3820011653', fov_expand=60, $ 
    minute_window=5, blend_waves='304,193,171', $ 
    description='IRIS_Vortex_SST', max_frames=500, $ 
    max_movie_frames=500, email='your@email.here'
```

This calls `iris_obs2hcr.pro` to get IRIS time/pointing specifics, then calls the aia/sdo ssw cutout service (`ssw_cutout_service.pro`) using those windows, optionally expanded (+/-60” & +/- 5 minutes in above example) - cutout service inherits any keywords (max_frames, blend_waves, etc) as described in [http://www.lmsal.com/solarsoft/ssw_servic/ssw_service_track_fov_api.html](http://www.lmsal.com/solarsoft/ssw_servic/ssw_service_track_fov_api.html)

**Output of the above:**

```
Job Summary (service parameters per call, job/size summary)
http://www.lmsal.com/solarsoft/ssw_service/queue/finished/ssw_service_140618_
→113133_38871.html

→ WWW/movie summary (via link in above summary)
http://sdowww.lmsal.com/sdowww/ssw/mmedia/ssw/ssw_client/data/ssw_service_140618_
→113133_38871/www/
```

Retrieve the AIA/HMI FITS cutout data from above -> local (see get data cut & paste line in Job Summary) - optionally, specify output directory and WAVES list (default=Current and All, resp.) - here I ask for SDO/AIA 193Å and SDO/HMI Blos only - cut & paste this into IDL session:

```idl
IDL> ssw_service_get_data, "ssw_service_140618_113133_38871", /loud, waves='193, 
    blos'
```