



Project Document

**SPIRE FTS Line Source Analysis for
Cross-Calibration**

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	1 of 31



SUBJECT: **SPIRE FTS Line Source Analysis for Cross-Calibration**

PREPARED BY: Peter Imhof

DOCUMENT No: SPIRE-BSS-REP-003269

ISSUE: Issue 1.0 **Date:** 09 February 2012

APPROVED BY: **Date:**



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	2 of 31

Distribution

Name

Bruce Swinyard
David Naylor
Dominique Benielli
Edward Polehampton
Elena Puga
Ivan Valtchanov
Matt Griffin
Matthijs van der Wiel
Nanyao Lu
Ros Hopwood
Tanya Lim
Trevor Fulton



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	3 of 31

Change Record

ISSUE	DATE	Changes
Draft 0.1	19 December 2011	First draft
Draft 0.2	20 December 2011	Corrections after input from MvdW
Draft 0.3	12 January 2012	Add data on AFGL4106, source velocities Corrections after input from RH Updated scripts
Issue 1.0	09 February 2012	Fix entry in Table 3 – no other change.



Project Document

**SPIRE FTS Line Source Analysis for
Cross-Calibration**

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	4 of 31

TABLE OF CONTENTS

CHANGE RECORD3

TABLE OF CONTENTS4

1. INTRODUCTION.....6

1.1 DOCUMENTS6

1.1.1 *Applicable Documents*.....6

1.1.2 *Reference Documents*6

2. DATA6

3. DATA PROCESSING.....10

4. DATA ANALYSIS10

5. RESULTS.....12

5.1 OVERVIEW12

5.2 LINE FLUX VALUES PER OBSERVATION14

5.3 LINE VELOCITY VALUES PER OBSERVATION19

6. APPENDIX.....24

6.1 BULKOFFAXISUBTRACTION.PY.....24

6.2 SPIREANALYSISLINESOURCES.PY25



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	5 of 31

Glossary

Acronym	Description
CR	Calibration Resolution
CVS	Concurrent Versions System
FTS	Fourier Transform Spectrometer
HIPE	Herschel Interactive Processing Environment
HR	High Resolution
HSA	Herschel Science Archive
LSR	Local Standard of Rest
OD	Operational Day
SPG	Standard Product Generation
SPIRE	Spectral and Photometric Imaging REceiver



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 6 of 31

1. INTRODUCTION

The purpose of this document is to describe how the SPIRE spectrometer data of routine calibration line sources were processed and analyzed.

1.1 Documents

1.1.1 Applicable Documents

Number	Document Title	Document Number	Issue
AD01	SPIRE Spectrometer Pipeline Description	SPIRE-BSS-DOC-002966	3.1
AD02	SPIRE Routine Phase Calibration Plan	SPIRE-RAL-DOC-003131	1.0
AD03			
AD04			

1.1.2 Reference Documents

Number	Document Title	Author(s)	Issue
RD01			
RD02			

2. DATA

The SPIRE Routine Phase Calibration Plan (AD02) defines three point-like sources outside of the Solar System with prominent line features to be observed on a regular basis as routine calibration sources for point source observations. These sources are as per section 4.2.4.5 of AD02:

1. **NGC 7027**
RA = 316.756625 deg, Dec = 42.2361666667 deg
RA = 21:07:01.590, Dec = 42:14:10.200
2. **CRL 618**
RA = 70.7235 deg, Dec = 36.1148333333 deg
RA = 04:42:53.640, Dec = 36:06:53.400
3. **AFGL 2688**
RA = 315.57825 deg, Dec = 36.6947777778 deg
RA = 21:02:18.780, Dec = 36:41:41.200

The SPIRE ICC has also added the following point-like sources to its routine calibration program:

4. **NGC 6302**
RA = 258.434208333 deg, Dec = -37.1044166667 deg
RA = 17:13:44.210, Dec = -37:06:15.900
5. **AFGL 4106**
RA = 155.831125 deg, Dec = -59.53469444444444 deg
RA = 10:23:19.470, Dec = -59:32:04.900
6. **VY CMa**
7. **CW Leo** (also known as IRC+10216)

Dark Sky and Orion Bar have been selected as the main sources for the routine calibration program of jiggle observations.



Project Document

**SPIRE FTS Line Source Analysis for
Cross-Calibration**

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	7 of 31

To date, the following routine observations were made with nominal detector settings of the sources mentioned above under points 1 to 5:

#	OD	Obsid	Number of repetitions	Spectral resolution
1	217	0x50002aa5	10	CR
2	227	0x50002c7e	10	CR
3	240	0x50002e41	17	HR
4	240	0x50002e44	17	HR
5	240	0x50002e45	10	CR
6	250	0x50002fe7	10	CR
7	326	0x50004094	5	CR
8	342	0x50004693	5	CR
9	368	0x50004b86	5	CR
10	383	0x50004eee	5	CR
11	404	0x50005489	5	CR
12	557	0x50007f40	4	CR
13	572	0x5000832a	4	CR
14	601	0x500088e4	4	CR
15	711	0x5000a533	4	CR
16	742	0x5000ad81	4	CR
17	908	0x5000d5b9	4	CR

Table 1: Observations of NGC7027 after OD189;
observations marked in **yellow** were not made at the nominal sky position.

#	OD	Obsid	Number of repetitions	Spectral resolution
1	275	0x50003459	5	CR
2	288	0x5000366e	17	HR
3	288	0x5000366f	5	HR
4	302	0x50003a32	2	CR
5	317	0x50003cc5	2	CR
6	451	0x50006193	4	CR
7	466	0x50006877	4	CR
8	495	0x50006bff	4	CR
9	655	0x500092ca	4	CR
10	682	0x50009ab8	4	CR
11	824	0x5000c3fc	4	CR
12	837	0x5000c548	4	CR
13	837	0x5000c549	4	CR
14	857	0x5000c907	4	CR

Table 2: Observations of CRL618 after OD189;
observations marked in **yellow** were not made at the nominal sky position.

#	OD	Obsid	Number of repetitions	Spectral resolution
1	217	0x50002aa6	5	CR
2	227	0x50002c7d	5	CR



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 8 of 31

#	OD	Obsid	Number of repetitions	Spectral resolution
3	240	0x50002e42	5	CR
4	240	0x50002e43	17	HR
5	354	0x5000483a	5	CR
6	395	0x500051fe	5	CR
7	410	0x500055d3	5	CR
8	543	0x50007983	4	CR
9	572	0x50008328	4	CR
10	601	0x500088e5	4	CR
11	711	0x5000a532	4	CR
12	767	0x5000b218	4	CR

Table 3: Observations of AFGL2688 after OD189;
observations marked in **yellow** were not made at the nominal sky position.

#	OD	Obsid	Number of repetitions	Spectral resolution
1	288	0x50003678	17	HR

Table 4: Observation of NGC6302 after OD189.

#	OD	Obsid	Number of repetitions	Spectral resolution
1	240	0x50002E3B	17	HR
2	543	0x5000797C	4	CR
3	557	0x50007F3E	4	CR
4	602	0x500088F5	4	CR
5	625	0x50008D06	4	CR
6	654	0x5000929E	4	CR
7	742	0x5000AD8E	4	CR
8	767	0x5000B20D	4	CR
9	803	0x5000B96B	4	CR
10	824	0x5000C401	4	CR
11	837	0x5000C541	4	CR
12	837	0x5000C542	4	CR
13	856	0x5000C8DA	4	CR
14	886	0x5000D212	4	CR

Table 5: Observations of AFGL4106 after OD189;
observations marked in **yellow** were not made at the nominal sky position.

The observations of the individual sources were made close to the nominal sky positions. SPIRE's beam steering mirror puts the actual sky position on a circle of about 1.7 arcsec radius around the nominal sky position. The actual sky positions for the various observations are given below for each source:



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 9 of 31

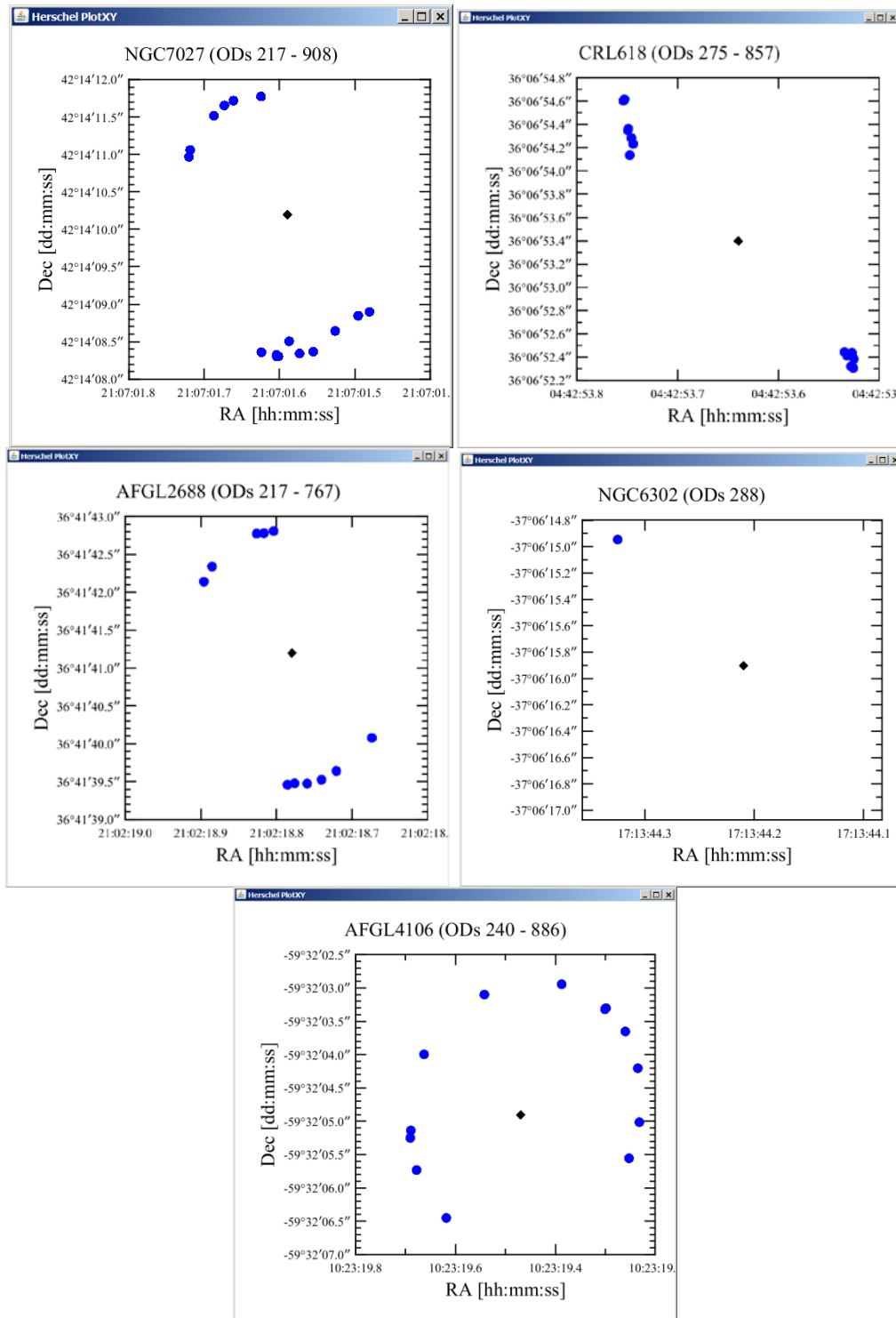


Figure 1: Actual sky positions (blue) for each source in relation to the nominal sky position (black)

3. DATA PROCESSING

All data were processed with the unaltered SOF1 SPG script from HIPE 8.0.3270 and the SPIRE Calibration context `spire_cal_8_1`.

The SPG products from the observation context were then processed further to clean up the continuum shape of the spectrum (particularly for SLW), allowing for an improved low-order, polynomial fit. The SPIRE Useful Script `Spectrometer_BackgroundSubtraction.py` was customized to operate on all relevant observations, more specifically the unapodized spectral products at level 1, and subtract the average sky background as measured by the co-aligned off-axis detectors. This step has the largest impact on NGC6302 where the off-axis detectors measured a signal of up to 10 Jy, see Figure 2.

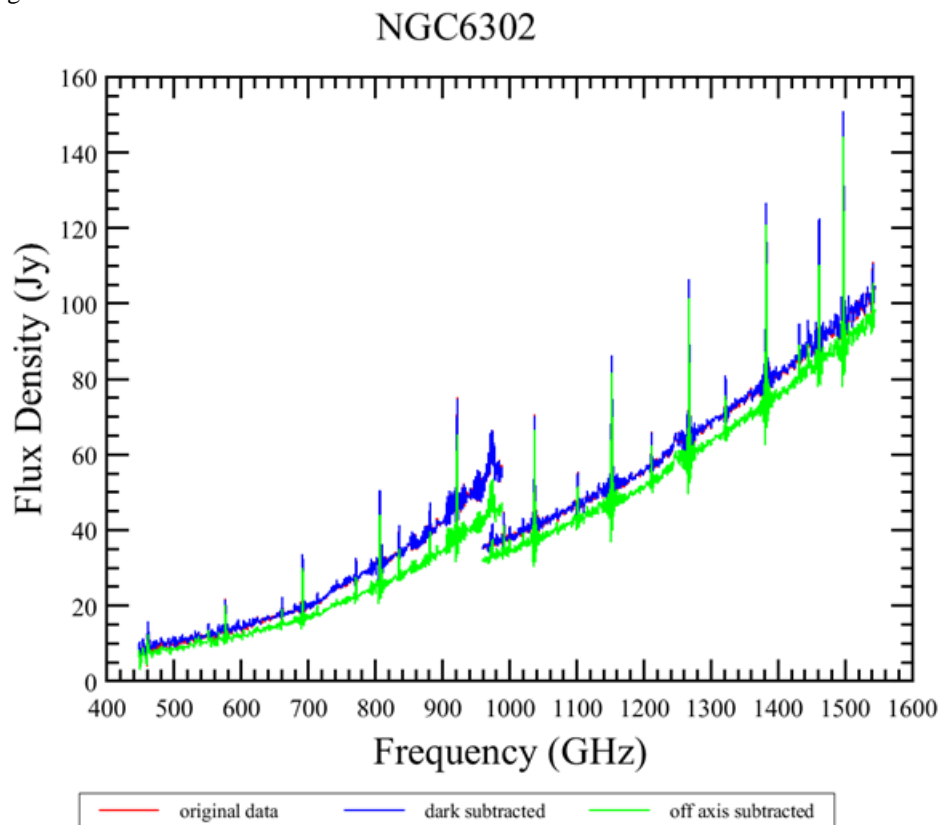


Figure 2: Signal from NGC6302 (obsid: 0x50003678) without (blue) and with background subtraction (green)

The script `BulkOffAxisSubtraction.py` is available via the Herschel Concurrent Versions System (CVS) and the version used for this work is reproduced in section 6.1 `BulkOffAxisSubtraction.py`.

4. DATA ANALYSIS

The SPIRE Useful Script `Spectrometer_LineFitting.py` was customized to analyze all spectra that were processed as described above. The script iterates through a list of sources and, for each source, a specific list of obsid's and significant lines to be fitted. Only the results from the ^{12}CO transitions (see Table 6) are reported even though more lines are used to achieve the best fit.



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 11 of 31

Transition	SPIRE FTS band	Rest Frequency [GHz]
¹² CO 4-3	SLW	461.04063
¹² CO 5-4	SLW	576.26796
¹² CO 6-5	SLW	691.47310
¹² CO 7-6	SLW	806.65187
¹² CO 8-7	SLW	921.79975
¹² CO 9-8	SSW	1036.91226
¹² CO 10-9	SSW	1151.98550
¹² CO 11-10	SSW	1267.01436
¹² CO 12-11	SSW	1381.99496
¹² CO 13-12	SSW	1496.92280

Table 6: Assumed rest frequencies for the ¹²CO transitions within the SPIRE band.

When processing a specific observation, the radial velocity of the spacecraft with respect to the Local Standard of Rest (LSR), as reported in the metadata of the observation context in the Herschel Science Archive (HSA), is used to correct the rest frequencies of the ¹²CO transitions given in Table 6:

$$\text{frequencyCorrected} = \text{frequency} * (1 - \text{radialVelocity}/c)$$

When fitting, the source is assumed to be at rest with respect to the LSR. The derived velocities represent the overall source velocity with respect to the LSR. The script uses the Levenberg-Marquardt fitter to match the spectra to a model that adds a set of sinc functions to a third order polynomial function (for the continuum). Data from the two central detectors of the SPIRE FTS (SSWD4 and SLWC3) are analyzed separately. Each sinc model has three parameters. Two parameters are kept free, the frequency and the amplitude of the peak, whereas the width of the sinc function is set to the value of the metadata "actualResolution" – nominally 1.1854 GHz – which represents the width of the spectral resolution element from the SPIRE FTS, divided by π . Initial guesses for the free parameters are zeros for all polynomial coefficients and, for the sinc functions, the corrected rest frequency and the flux value of the measured spectrum at the corrected rest frequency are the initial guesses for the sinc functions. The script fits to all significant lines in a spectrum and reports results for the ¹²CO transitions. For each sinc model, the integrated line flux and the derived source velocity are calculated as below:

$$\text{Line flux [W m}^{-2}] = 10^{-26} * \text{peakAmplitude [Jy]} * 10^9 * \pi * \text{actualResolution [GHz]}$$

$$\text{Velocity [km s}^{-1}] = c [\text{km s}^{-1}] * (\text{frequencyCorrected [GHz]} - \text{peakFrequency [GHz]}) / \text{frequencyCorrected [GHz]}$$

Error estimates can be calculated based on the standard deviation reported by the fitter. The line flux and velocity errors are calculated as follows:

$$\text{Line flux error [W m}^{-2}] = \text{Line flux [W m}^{-2}] * \text{peakAmplitudeStdDev [Jy]} / \text{peakAmplitude [Jy]}$$

$$\text{Velocity error [km s}^{-1}] = c [\text{km s}^{-1}] * \text{peakFrequencyStdDev [GHz]} / \text{frequencyCorrected [GHz]}$$

Important note on reported standard deviations: Due to a problem when performing a constrained fit in HIPE, the reported standard deviation values from the fitter are not necessarily meaningful, see [HCSS-14964](#). Instead of reporting potentially misleading values from the fitter, the standard deviation of the set of values for all observations of the same source is reported. Divide this number by the square root of the number of observations to get an indication of the uncertainty of the reported average values.

The script also creates one table dataset with line flux and velocity values per observation and stores it as a fits file (SpireResultsLineSources.fits). The fits file contains:

- Column "rest frequencies": Nominal ¹²CO line frequencies in units of GHz.



Project Document

**SPIRE FTS Line Source Analysis for
Cross-Calibration**

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	12 of 31

- Column “Line Flux <obsid>”: integrated line flux for the specified observation for each transition in units of $W m^{-2}$.
- Column “Line Velocity <obsid>”: line velocity for the specified observation for each transition in units of $km s^{-1}$.

The script SpireAnalysisLineSources.py is available via the Herschel CVS and the version used for this work is reproduced in section 6.2 SpireAnalysisLineSources.py.

5. RESULTS

The observations at non-nominal sky positions (see yellow highlights in Table 1 to Table 5) were not used to derive the values reported in the following sections. The average and standard deviation values and plots can be recreated from the data contained in the table dataset stored in SpireResultsLineSources.fits.

5.1 Overview

The average values and the standard deviation of the routine calibration observations are given for each source. The number in brackets next to the source name indicates the number of analyzed observations.

- Colored shading of values indicates that reported values need special consideration:
- **Yellow:** The source is slightly extended in comparison to Uranus, which defines the size of a point source. No attempt was made to account for the source extension.
 - **Blue:** Close-by, non- ^{12}CO lines affect the flux derived for ^{12}CO lines. No attempt was made to fit more than one sinc function to any given, potentially blended line feature. NB: Contamination for AFGL4106 has not been checked.
 - **Red:** The point flux calibration is noisiest in the low frequency range (flux calibrator Uranus has the lowest flux at low frequencies) and the uncertainty in the derived velocity will be high.

	NGC7027 (16)		CRL618 (13)		AFGL2688 (11)		NGC6302 (1)		AFGL4106 (13)	
^{12}CO	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
4-3	601	13.8	188	23.9	310	25.8	62.0		78.6	16.4
5-4	833	22.3	290	17.9	473	35.2	97.3		136.0	15.9
6-5	1140	22.5	469	12.0	819	65.9	153.0		185.0	4.44
7-6	1640	24.1	722	12.7	1270	60.2	229.0		247.0	5.37
8-7	2170	27.3	1080	10.3	1910	35.9	339.0		284.0	8.52
9-8	1950	52.4	1390	47.1	2060	191.0	348.0		280.0	10.2
10-9	2080	72.9	2180	82.4	3080	326.0	425.0		272.0	11.7
11-10	2150	76.9	2240	78.4	2950	339.0	505.0		238.0	9.56
12-11	2110	92.4	2600	124.0	3220	380.0	558.0		195.0	10.7
13-12	2040	74.8	3030	111.0	3440	386.0	636.0		164.0	7.71

Table 7: Average and Standard Deviation of Integrated Line Flux [$1E-18 W m^{-2}$]



Project Document

**SPIRE FTS Line Source Analysis for
Cross-Calibration**

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	13 of 31

¹² CO	NGC7027 (16)		CRL618 (13)		AFGL2688 (11)		NGC6302 (1)		AFGL4106 (13)	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
4-3	29.0	20.7	16.3	50.2	-17.6	27.6	53.4	0.0	-23.7	84.9
5-4	-4.4	3.3	-56.8	13.0	-54.0	9.4	-70.9	0.0	-35.1	24.9
6-5	31.5	3.1	5.6	3.8	7.6	3.5	-17.4	0.0	-11.3	8.5
7-6	25.9	3.6	-28.5	5.7	-41.7	4.6	-45.4	0.0	-18.9	5.1
8-7	32.6	5.1	-17.7	6.2	-34.0	7.7	-31.9	0.0	-9.7	6.5
9-8	24.0	2.7	-25.4	5.0	-38.6	2.5	-36.6	0.0	-21.9	5.3
10-9	29.9	2.9	-0.7	6.3	-8.6	4.0	-34.3	0.0	-15.4	7.1
11-10	25.7	3.9	-31.2	5.9	-39.9	4.1	-37.2	0.0	-15.2	3.3
12-11	26.0	4.5	-26.6	6.7	-39.6	5.0	-39.8	0.0	-14.2	7.9
13-12	21.1	5.4	-29.1	6.8	-40.6	7.9	-43.9	0.0	-21.1	8.8

Table 8: Average and Standard Deviation of Line Velocities [km s⁻¹]

The source velocities and their uncertainty estimates are derived from the non-shaded entries in Table 8:

<i>Source</i>	<i>Average velocity [km s⁻¹]</i>	<i>Average standard deviation [km s⁻¹]</i>
NGC7027	27.1	3.9
CRL618	-26.4	6.1
AFGL2688	-39.1	5.3
NGC6302	-35.8	8.7
AFGL4106	-16.0	6.6

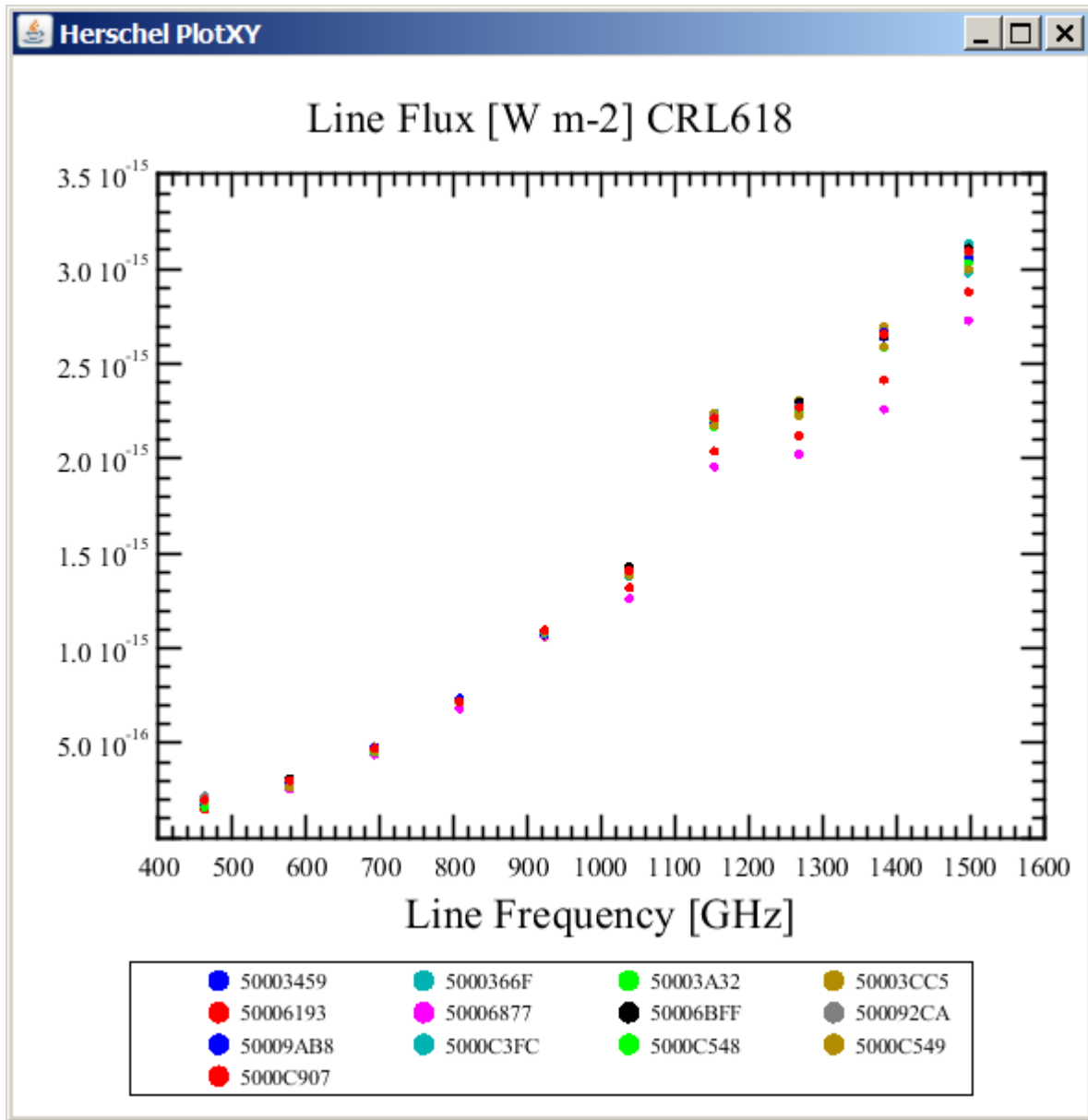
Table 9: Derived source velocities and their uncertainties



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 15 of 31

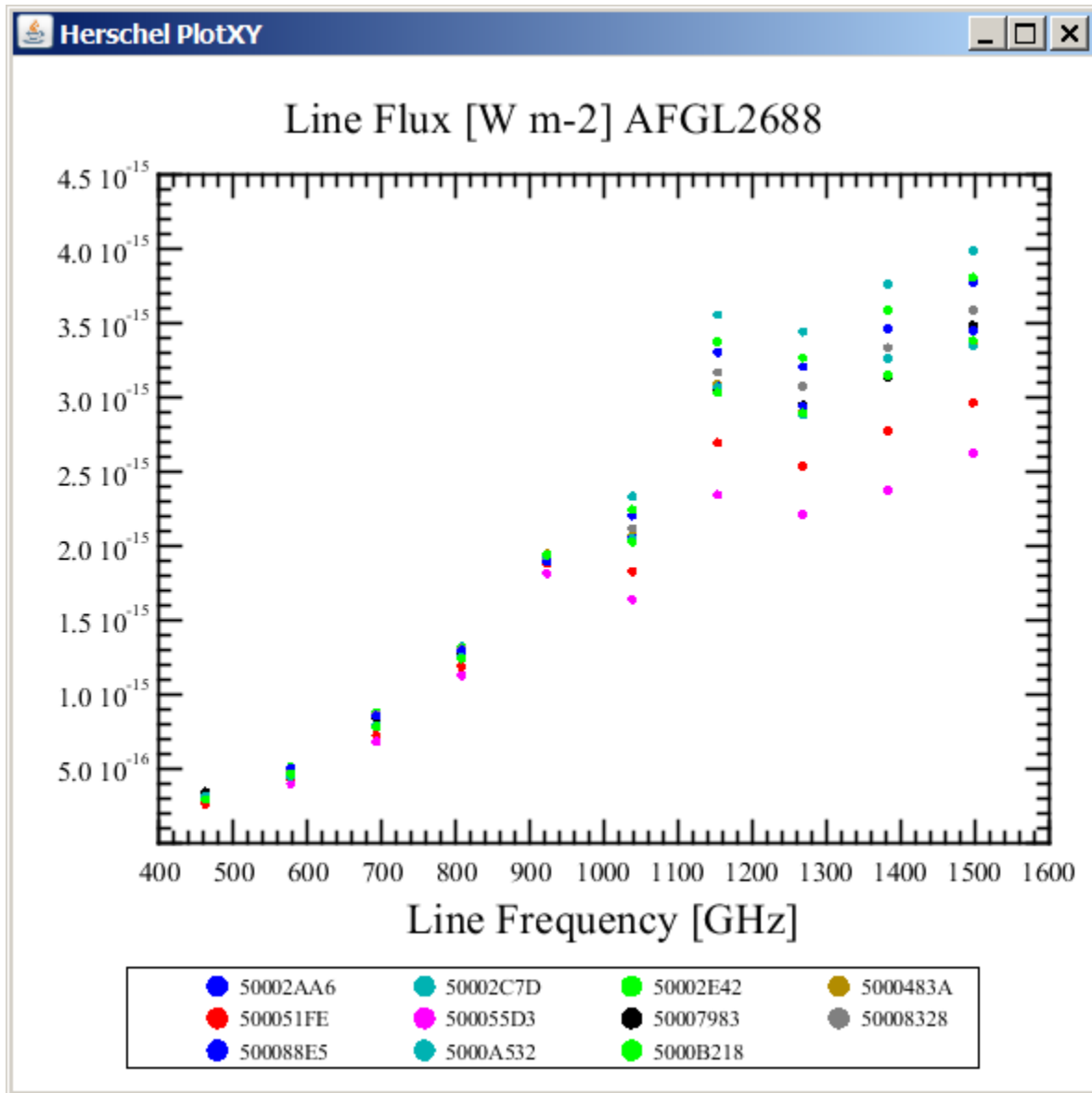




Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 16 of 31

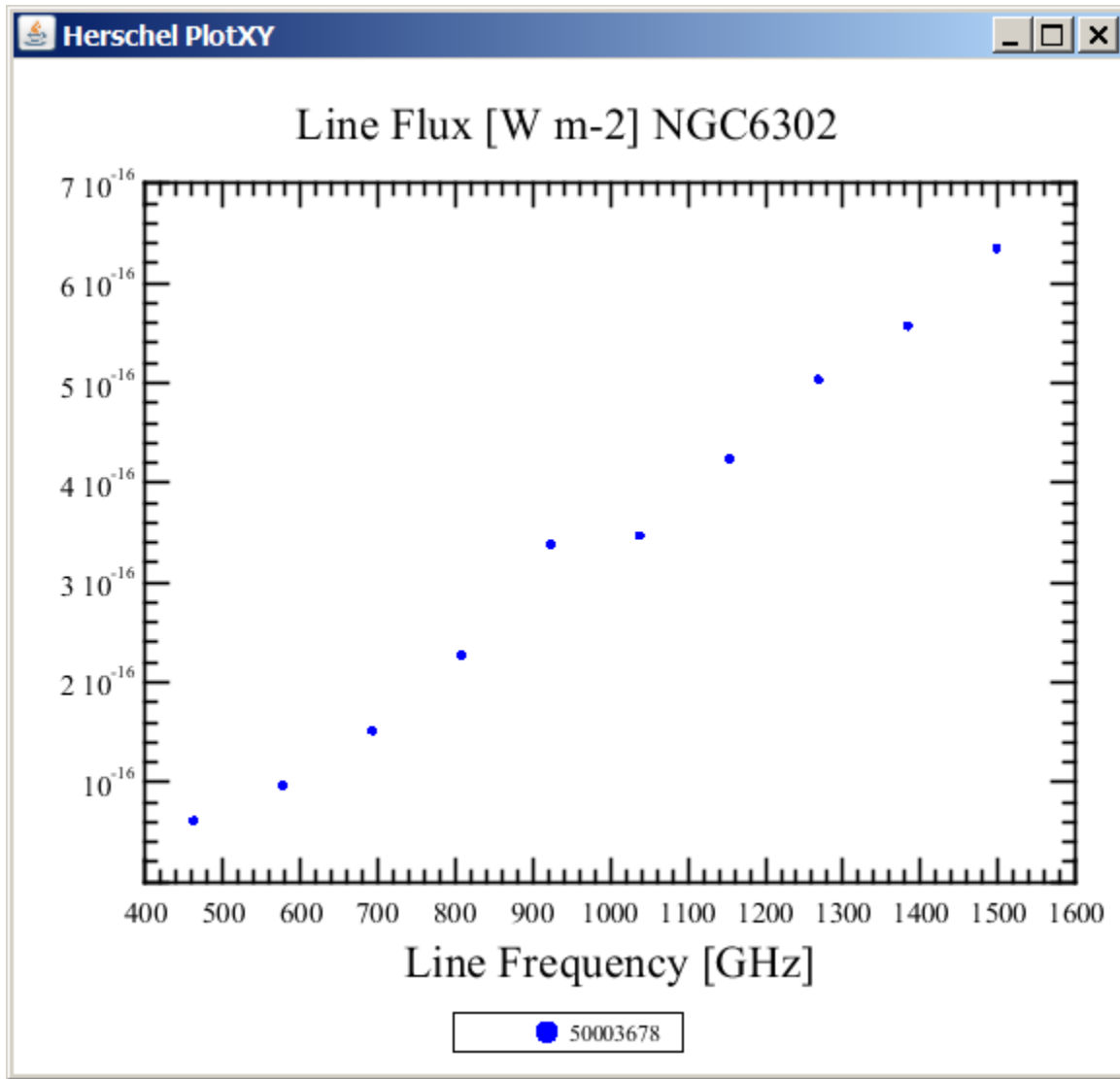




Project Document

**SPIRE FTS Line Source Analysis for
Cross-Calibration**

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 17 of 31

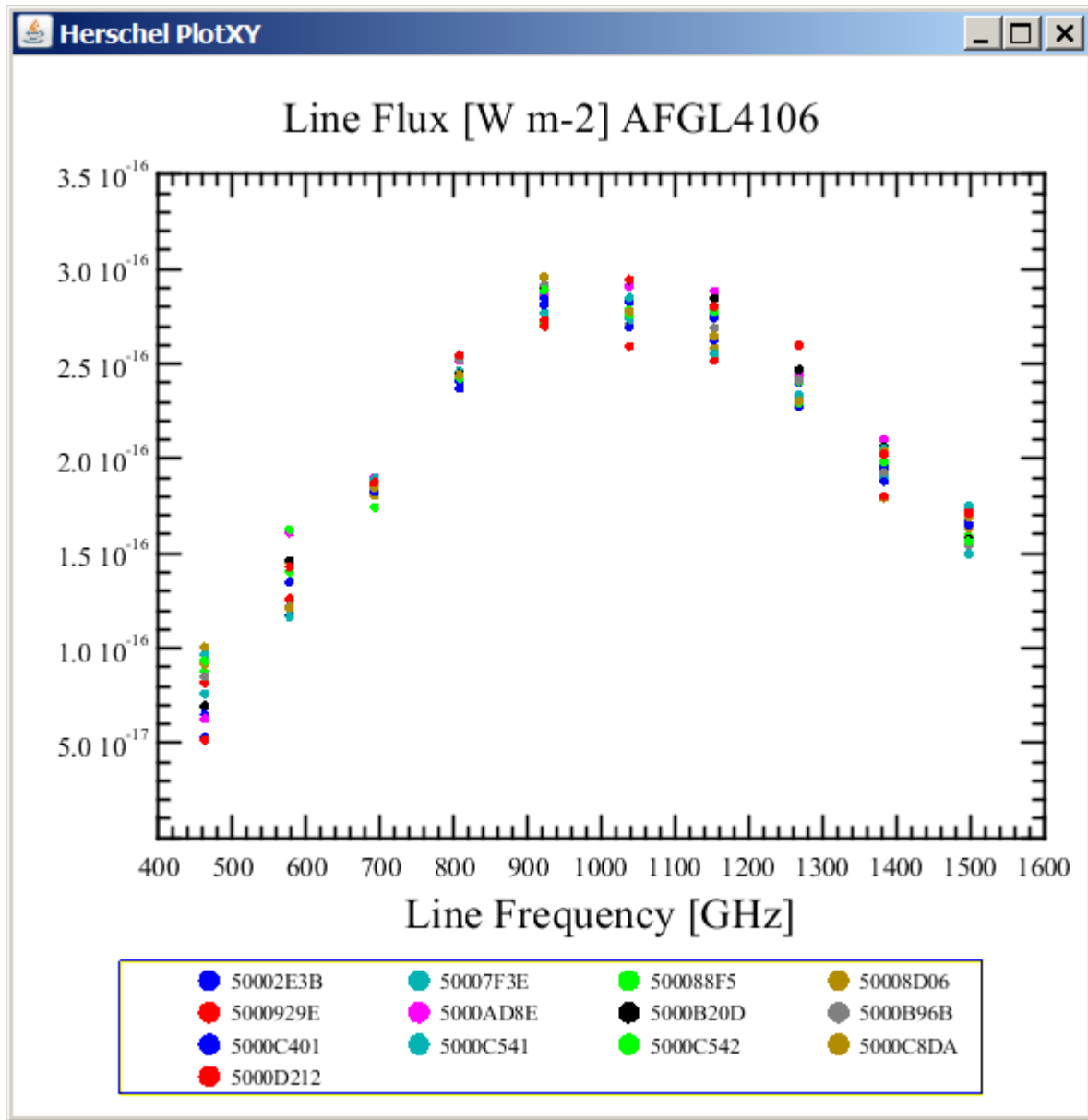




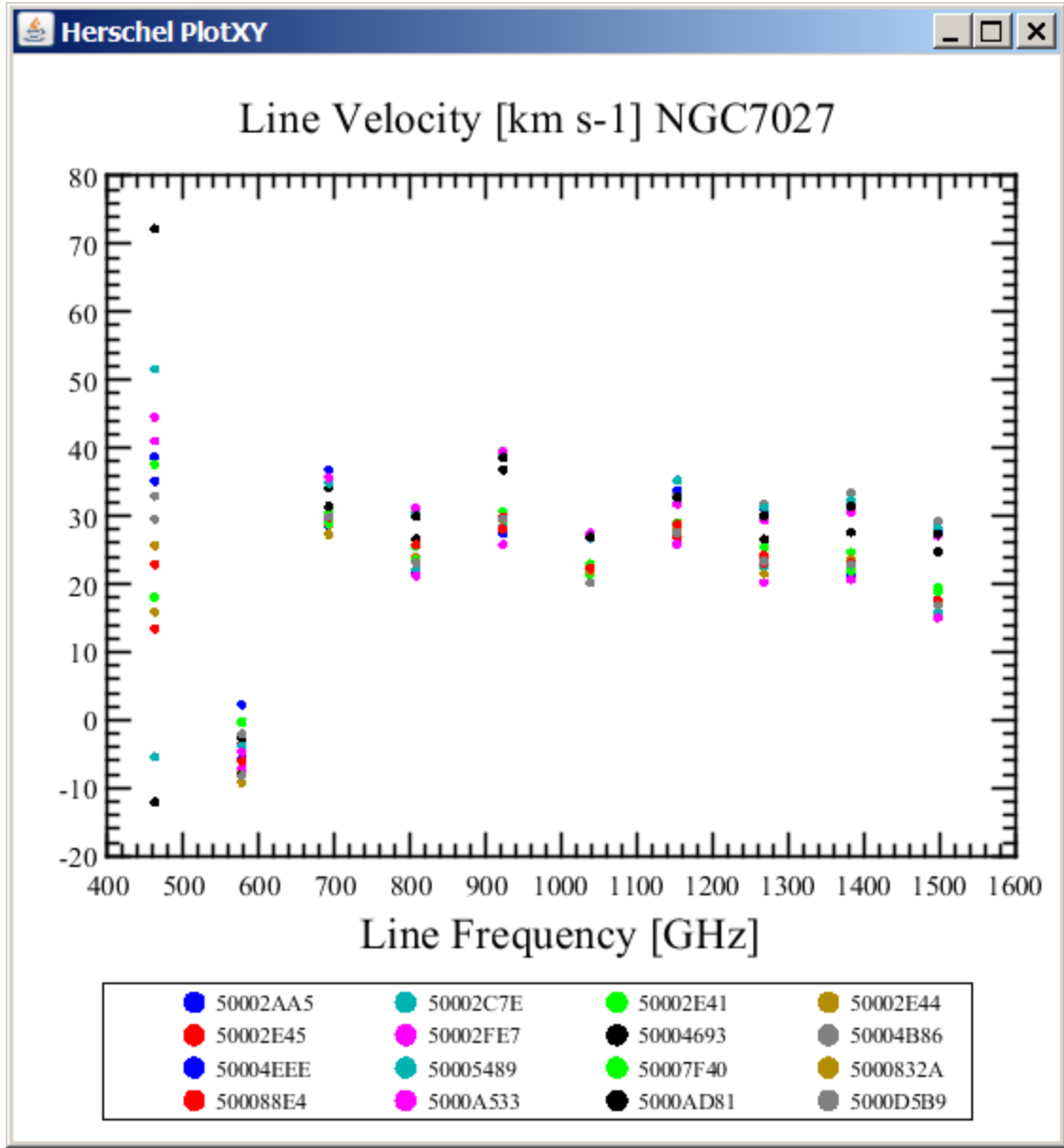
Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 18 of 31



5.3 Line Velocity values per observation

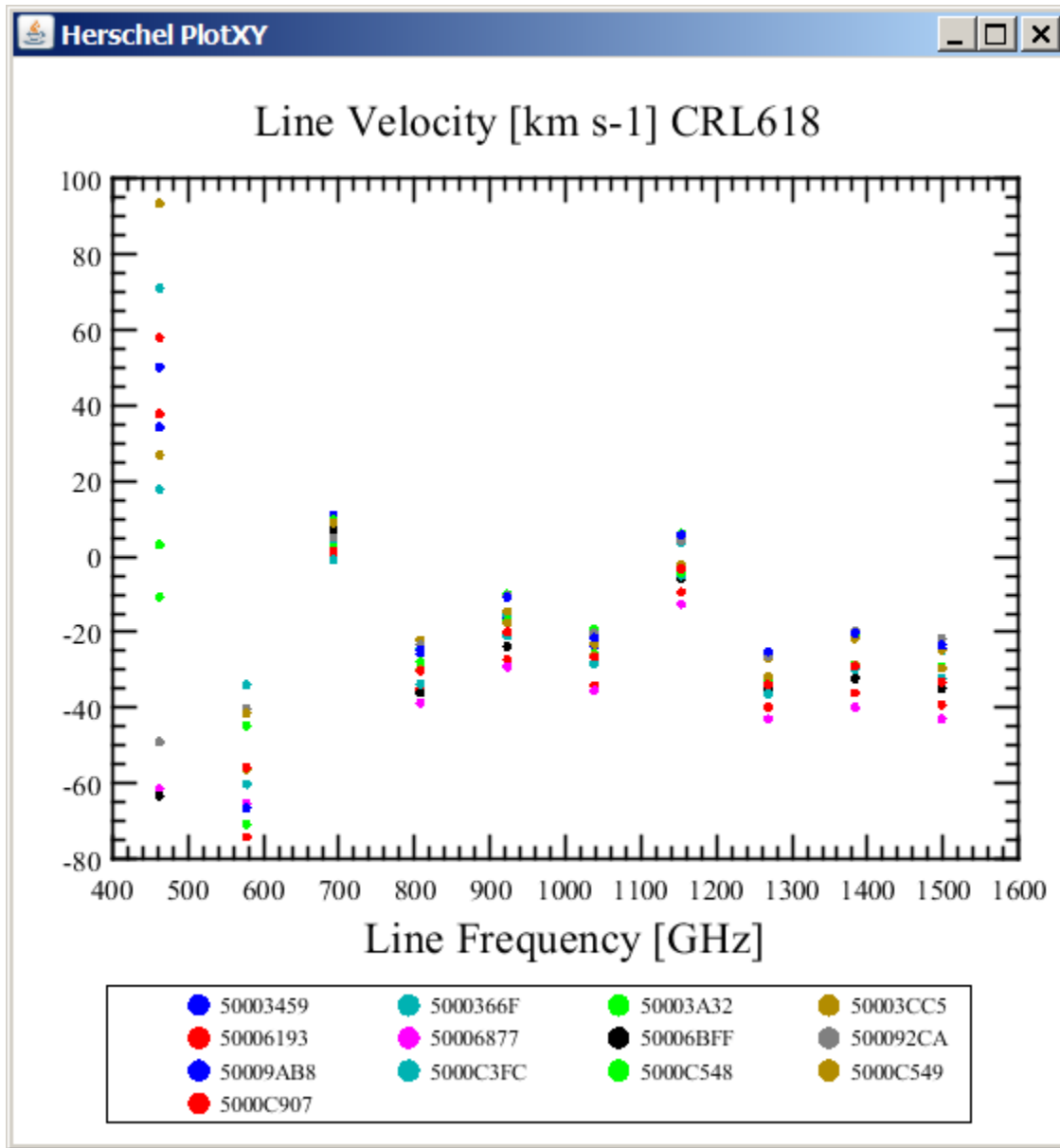




Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 20 of 31

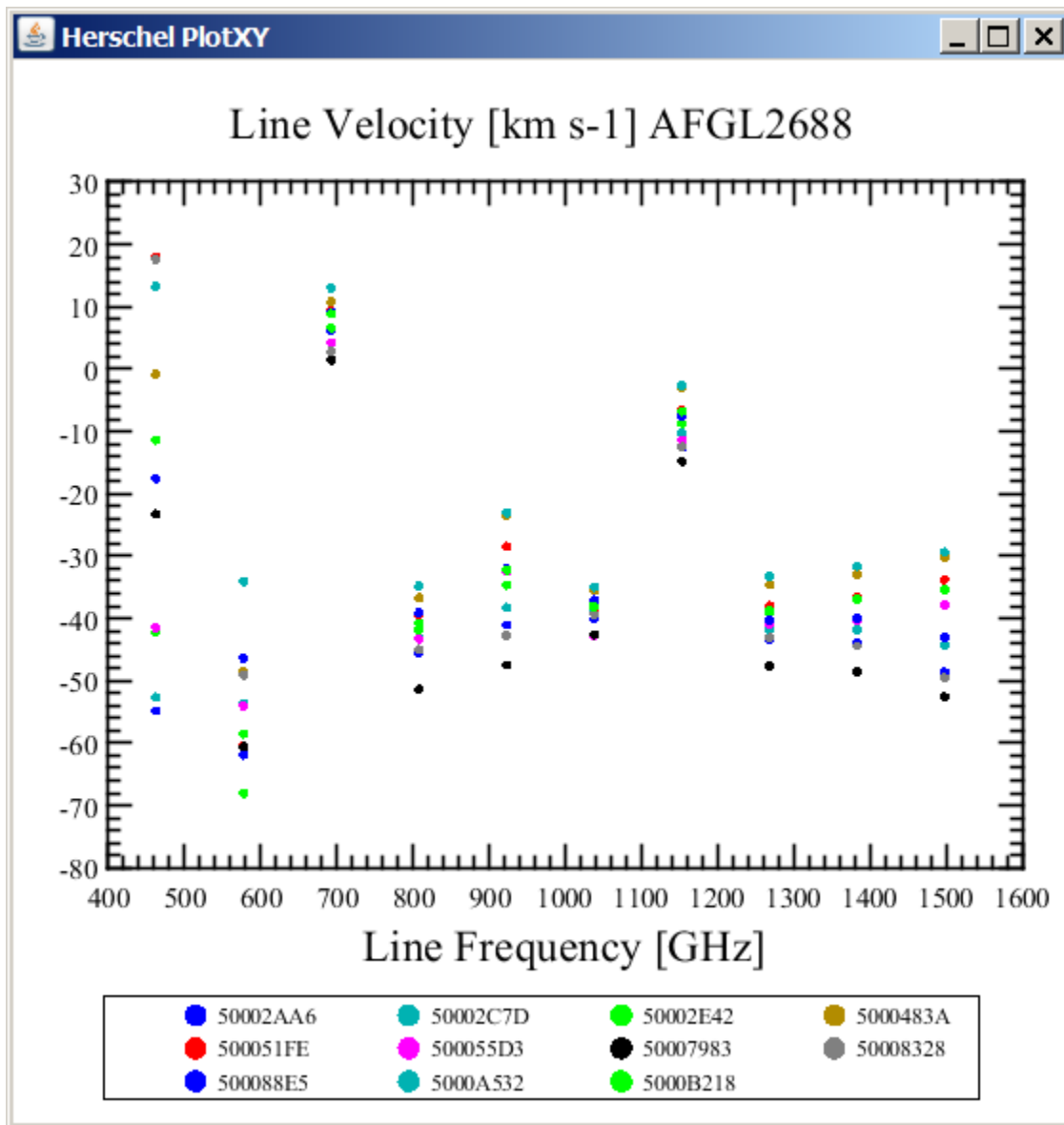




Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 21 of 31

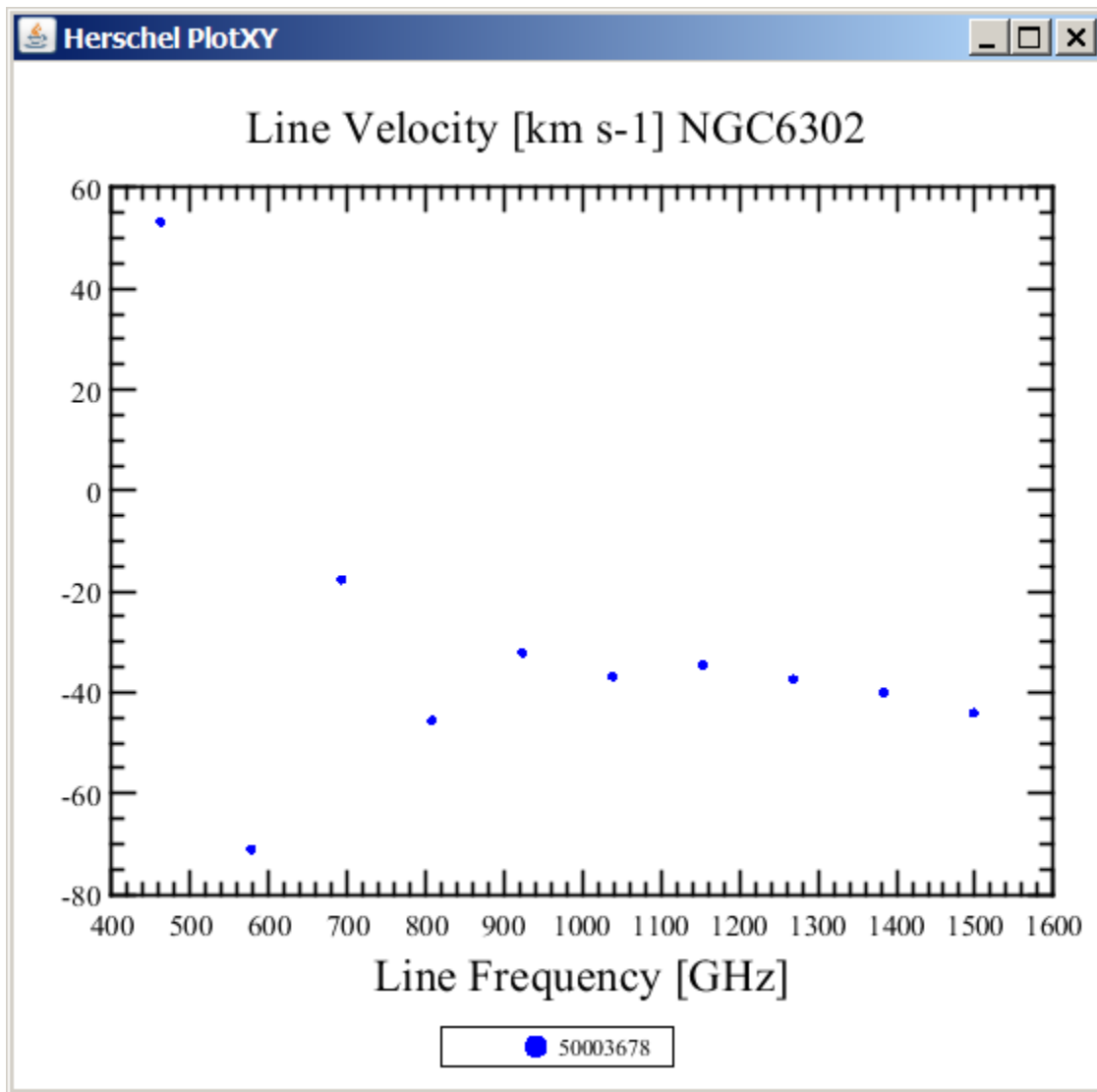




Project Document

**SPIRE FTS Line Source Analysis for
Cross-Calibration**

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 22 of 31

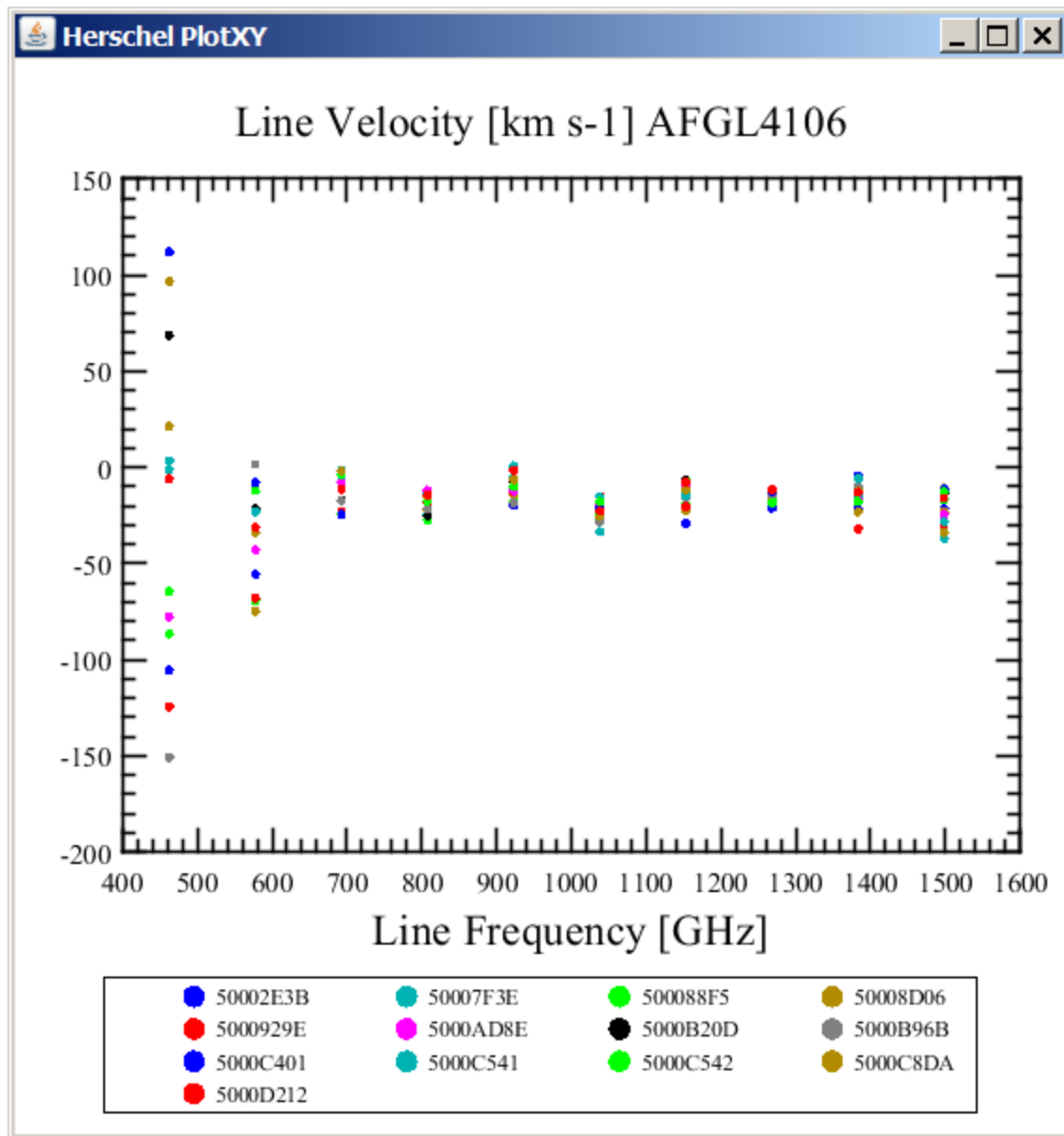




Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 23 of 31





Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	24 of 31

6. APPENDIX

6.1 BulkOffAxisSubtraction.py

BulkOffAxisSubtraction.py from [cvs]/develop/user/ca/lethbridge/pdavis/Scripts/ReProcess/v9/

```
"""
<!--
$Id: BulkOffAxisSubtraction.py,v 1.4 2012/01/12 20:13:33 pdavis Exp $
-->
"""

import os

cal = SpireCal.getInstance()

path = '/SPIRE/TMP/HCSS_8_0_RC5/level_1/calibration_tree/'
foo = StringIO(os.listdir(path))
fileNames = StringIO()
for fo in foo:
    if ('OFF' not in fo): fileNames.append(fo)

centreDetectors = ["SLWC3", "SSWD4"]
offAxisDets = {"SSW":["SSWC2", "SSWB3", "SSWC5", "SSWE5", "SSWF3", "SSWE2"], \
               "SLW":["SLWD3", "SLWC4", "SLWB3", "SLWB2", "SLWC2", "SLWD2"]}

p7027 = PlotXY(titleText='NGC 7027')
p618 = PlotXY(titleText='CRL 618')
p2688 = PlotXY(titleText='AFGL 2688')
p6302 = PlotXY(titleText='NGC 6302')

cnt = 0
for fileName in fileNames:
    cnt += 1
    ssds = FitsArchive().load(path+fileName)
    if ssds.meta['object'].value == 'Mars':
        print 'skipping Mars'
        continue
    ssds = averageSpectra(ssds)
    asds = ssds
    del(ssds)
    if (not asds[0].hasChannel('SSWD4')) or (not asds[0].hasChannel('SLWC3')):
        print 'no center detector for %s - skipping'%asds.meta['object'].value
        continue
    offAxisSubtracted = asds.copy()
    for det in centreDetectors:
        averageOff = 0
        for offDet in offAxisDets[det[:3]]:
            try:
                averageOff = averageOff + asds["0000"][offDet]
            except:
                print '##### WARNING #####'
                print offDet+' does not exist for %s'%asds.meta['object'].value
                continue
        averageOff = averageOff/len(offAxisDets[det[:3]])
        if averageOff == 0:
            print 'skipping %s because there are no valid off-axis
detectors'%asds.meta['object'].value
            continue
        offAxisSubtracted["0000"][det].flux = asds["0000"][det].flux - averageOff.flux
    # Now point source calibrate
    pointOffAxisSub = filterChannels(offAxisSubtracted.copy(), keepChannels=centreDetectors)
    beamParam = cal.spec.beamParamList.getProduct('HR',"unapod","nominal",asds.startDate)
```




Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	25 of 31

```
pointOffAxisSub = specPointFluxConversion(pointOffAxisSub, beamParam=beamParam)
lSlw
LayerXY(pointOffAxisSub[0]['SLWC3'].wave,pointOffAxisSub[0]['SLWC3'].flux,name=hex(pointOffAxisSub
b.meta['obsid'].value))
lSsw
LayerXY(pointOffAxisSub[0]['SSWD4'].wave,pointOffAxisSub[0]['SSWD4'].flux,name=hex(pointOffAxisSub
b.meta['obsid'].value))
if '7027' in pointOffAxisSub.meta['object'].value:
    p7027.addLayer(lSlw)
    p7027.addLayer(lSsw)
elif '618' in pointOffAxisSub.meta['object'].value:
    p618.addLayer(lSlw)
    p618.addLayer(lSsw)
elif '2688' in pointOffAxisSub.meta['object'].value:
    p2688.addLayer(lSlw)
    p2688.addLayer(lSsw)
elif '6302' in pointOffAxisSub.meta['object'].value:
    p6302.addLayer(lSlw)
    p6302.addLayer(lSsw)
else:
    print str(cnt)+'/'+str(len(fileNames)), pointOffAxisSub.meta['object'].value
    newFileName = path+'OFF_AXIS_SUB_POINTSOURCE_FLUX_CONVERTED_A'+fileName[20:]
    print 'OFF_AXIS_SUB_'+fileName
    FitsArchive().save(newFileName, pointOffAxisSub)
pass

p7027.getLegend().setVisible(True)
p618.getLegend().setVisible(True)
p2688.getLegend().setVisible(True)
p6302.getLegend().setVisible(True)
```

6.2 SpireAnalysisLineSources.py

SpireAnalysisLineSources.py from [cvs]/develop/user/ca/lethbridge/pdavis/Scripts/ReProcess/v9/

```
"""
<!--
$Id: SpireAnalysisLineSources.py,v 1.2 2012/01/12 20:18:12 pdavis Exp $
-->
"""

# This script fits the continuum and all point source lines in one go
# It can operate on many sources, many obsid's, many V&V versions and flavors

from herschel.share.unit import Constant
c = Constant.SPEED_OF_LIGHT.value
PI = Math.PI
CO12Lines = Double1d([461.04063,576.26796,691.47310,806.65187,921.79975,\
1036.91226,1151.98550,1267.01436,1381.99496,1496.92280])

SpireResultsLineSources = TableDataset()
SpireResultsLineSources['Rest
Frequencies']=Column(CO12Lines,unit=herschel.share.unit.Frequency.GIGAHERTZ,description='12CO
Rest Frequencies')

# ['CRL618','NGC7027','AFGL2688','NGC6302','AFGL4106','AFGL2591']
sources = ['CRL618','NGC7027','AFGL2688','NGC6302','AFGL4106']
for src in sources:
    print src
    pFlux = PlotXY(titleText='Line Flux [W m-2] %s'%src)
    pVel = PlotXY(titleText='Line Velocity [km s-1] %s'%src)
    if src == 'CRL618':
        calibrations = ['calibration_tree']
        days = '275 - 857'
        obsids = Double1d([0x50003459, \
```



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref: SPIRE-BSS-REP-003269
Issue: Issue 1.0
Date: 09 February 2012
Page: 26 of 31

```
#0x5000366E, \ not at the nominal sky position
0x5000366F, \
0x50003A32, \
0x50003CC5, \
0x50006193, \
0x50006877, \
0x50006BFF, \
0x500092CA, \
0x50009AB8, \
0x5000C3FC, \
0x5000C548, \
0x5000C549, \
0x5000C907)) #
# 0x5000C54A is a fake obsid. It really is the second jiggle of 0x5000C549
# include any lines to be verified
lineFreqs
Doubled([461.04063,532,576.26796,620,691.47310,709,797.5,806.65187,886,921.79975,\
1036.91226,1151.98550,1267.01436,1381.99496,1496.92280])
elif src == 'NGC7027':
  calibrations = ['calibration_tree']
  days = '217 - 908'
  obsids = Doubled([ \
1342188197, \
1342188670, \
1342189121, \
1342189124, \
1342189125, \
1342189543, \
1342195347, \
1342196614, \
1342197486, \
1342198921, \
1342209856, \
1342210858, \
1342212324, \
1342219571, \
1342221697, \
1342231993])
# 1342193812 see http://herschel.esac.esa.int/twiki/bin/view/Spire/SpireSpecObsKnownIssues
#, 1342187089, 1342221698
# include any lines to be verified
lineFreqs = Doubled([461.04063,576.26796,691.47310,806.65187,835.07189,921.79975,\
1036.91226,1151.98550,1267.01436,1381.99496,1496.92280])
elif src == 'AFGL2688':
  calibrations = ['calibration_tree']
  days = '217 - 767'
  obsids = Doubled([0x50002AA6, \
0x50002C7D, \
#0x50002E43, \ Unusual sky position - about 5" from nominal
0x50002E42, \
0x5000483A, \
0x500051FE, \
0x500055D3, \
0x50007983, \
0x50008328, \
0x500088E5, \
0x5000A532, \
0x5000B218])
# include any lines to be verified
lineFreqs
Doubled([461.04063,532,551,576.26796,620,661,691.47310,709,771,797,806.65187,881,886,921.79975,\
1036.91226,1063,1101,1151.98550,1240,1267.01436,1328,1381.99496,1417,1496.92280,1505])
elif src == 'AFGL4106':
  calibrations = ['calibration_tree']
  days = '240 - 886'
```



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	27 of 31

```
obsids = DoubleId([0x50002E3B, \
0x50007F3E, \
0x500088F5, \
0x50008D06, \
0x5000929E, \
0x5000AD8E, \
0x5000B20D, \
0x5000B96B, \
0x5000C401, \
0x5000C541, \
0x5000C542, \
0x5000C8DA, \
0x5000D212])
# 0x5000C543 is a fake obsid. It really is the second jiggle of 0x5000C542
# 0x5000797C, \ see http://herschel.esac.esa.int/twiki/bin/view/Spire/SpireSpecObsKnownIssues
# include any lines to be verified
lineFreqs = DoubleId([461.04063,532,576.26796,661,691.47310,770.5,806.65187,881,921.79975,\
1036.91226,1151.98550,1267.01436,1381.99496,1461,1496.92280])
elif src == 'NGC6302':
    calibrations = ['calibration_tree']
    days = '288'
    obsids = DoubleId([0x50003678])
    # include any lines to be verified
    lineFreqs
DoubleId([461.04063,532,551,576.26796,620,661,691.47310,709,771,797,806.65187,835,881,886,921.799
75,\
1036.91226,1101,1151.98550,1211.5,1267.01436,1321.5,1381.99496,1431,1460.5,1496.92280,1505])
elif src == 'AFGL2591':
    calibrations = ['calibration_tree']
    days = '544 - 682'
    obsids = DoubleId([0x50007CFF, \
0x50009ABC])
    # include any lines to be verified
    lineFreqs
DoubleId([461.04063,532,551,576.26796,620,661,691.47310,709,771,797,806.65187,835,881,886,921.799
75,\
1036.91226,1101,1151.98550,1211.5,1267.01436,1321.5,1381.99496,1431,1460.5,1496.92280,1505])
pass
try:
    foo = simpleAsciiTableReader('C:\\SPIRE\\TMP\\%s_velocities.txt'%src)
    velocities = foo['c0'].data
    if len(velocities) <> len(obsids): stop
    print 'Read s/c velocities from file'
except:
    # archive access to retrieve radialVelocity
    archive = HsaReadPool()
    hsa_storage = ProductStorage()
    archive.setAuthentication("pdavis", "Dexter")
    hsa_storage.register(archive)
    velocities = DoubleId()
    for obsid in obsids:
        # retrieve radial velocity from HSA
        queryStr = "p.obsid == '%i'"%(obsid)
        query = MetaQuery(herschel.ia.obs.ObservationContext, 'p', queryStr)
        product = hsa_storage.select(query)
        for key in range(len(product)):
            obs = product[key].product
            radialVelocity = obs.meta["radialVelocity"].value
            velocities.append(radialVelocity)
    pass
pass
tds = TableDataset(description='radial velocity for %s observations'%src)
tds['velocities'] = Column(data=velocities,
unit=herschel.share.unit.Speed.KILOMETERS_PER_SECOND )
```



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	28 of 31

```
asciiTableWriter(table=tds, file='C:\\SPIRE\\TMP\\%s_velocities.txt'%src)
pass
# Define some constants
path = '/SPIRE/TMP/' # data location
# include any version to be verified
versions = ['HCSS_8_0_RC5']
# Correct the frequencies for the source velocity (for initial guess)
sourceVel = 0.0*1000.0
lineFreqsCorr = lineFreqs*(1.-sourceVel/c)
#
LineFlux = Double4d(len(calibrations),len(versions),len(CO12Lines),len(obsids))
LineFluxError = Double4d(len(calibrations),len(versions),len(CO12Lines),len(obsids))
LineVelocity = Double4d(len(calibrations),len(versions),len(CO12Lines),len(obsids))
LineVelocityError = Double4d(len(calibrations),len(versions),len(CO12Lines),len(obsids))
#
SkyPosition_RA = Double1d()
SkyPosition_DEC = Double1d()
#
j = 0
for calibration in calibrations:
    i = 0
    print calibration
    for version in versions:
        print version
        obsCnt = 0
        for obsid in obsids:
            #print hex(int(obsid))
            lineCnt = 0
            myIndex = obsids.where(obsids == obsid)
            radialVelocity = 1000*velocities[myIndex][0]
            # Get the SDS containing the fully processed spectrum
            try:
                filename = '%s%s/level_1/%s/OFF_AXIS_SUB_POINTSOURCE_FLUX_CONVERTED_ASDS_0x%8.8X_0xA1060001_Point_0_Jiggle_0_unapod_HR.fits'%(path,version,calibration,obsid)
                sds = FitsArchive().load(filename)
            except:
                print 'Could not read %s %s %s'%(filename, version, calibration)
                stop
            #
            SkyPosition_RA.append(sds[0]['SSWD4'].meta['ra'].value)
            SkyPosition_DEC.append(sds[0]['SSWD4'].meta['dec'].value)
            #
            # Loop over the two centre detectors
            for detector in ["SLWC3","SSWD4"]:
                # Read the detector spectrum and convert to GHz
                spc = sds['0000'][detector].copy()
                #
                # Correct for radial velocity of the Satellite (i.e. into the LSR frame)
                freq = spc.wave * (1.0-radialVelocity/c)
                spc.wave = freq
                #
                # Define the fitting model.
                # First, start with a N'th order polynomial for the continuum
                N = 3
                polyInit = [0.0]*(N+1)
                plotMe = False
                sf = SpectrumFitter(spc,0,1,plotMe)
                sf.useFitter('lbn')
                M = sf.addModel('poly', [N], polyInit)
                models = [M]
                #
                # Then, add one sinc model per spectral line
                # Select only lines where there are data
                sel = lineFreqsCorr.where(lineFreqsCorr.gt(MIN(freq)).and(lineFreqsCorr.lt(MAX(freq))))
                #
```



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	29 of 31

```
# Create a Sinc model for each line to be fitted
for line in lineFreqsCorr[sel]:
    d = MIN(ABS(freq - line))
    selLine = freq.where(ABS(freq - line) == d)
    ampInit = spc['flux'].data[selLine][0]
    initSinc = [ampInit, line, sds.meta['actualResolution'].value/Math.PI]
    M = sf.addModel('sinc', initSinc)
    M.setLimits(2, sds.meta['actualResolution'].value/Math.PI-1e-5,
sds.meta['actualResolution'].value/Math.PI+1e-5)
    models.append(M)
# Carry out the global fit
sf.doGlobalFit()
# Calculate the residual
sf.residual()
residual = sf.getResidual()
# Calculate the integrated line flux and print results
for model in models[1:]:
    parameters = model.getFittedParameters()
    parametersStdDev = model.getFittedStdDev()
    # only 12CO lines
    if lineFreqs[lineCnt] not in CO12Lines:
        lineCnt += 1
        continue
    else:
        lineIndex = CO12Lines.where(CO12Lines == lineFreqs[lineCnt]).toIntId()[0]
        # Line flux in W/m2
        lineFlux = 1e-26*parameters[0]*1e9*Math.PI*parameters[2]
        LineFlux.set(j,i,lineIndex,obsCnt,lineFlux)
        # Line flux error in W/m2
        # Original error calculation when sinc width is a free parameter
        #lineFluxError = lineFlux*SQRT((parametersStdDev[0]/parameters[0])**2+\
        # (parametersStdDev[2]/parameters[2])**2)
        # Error calculation when sinc width is NOT a free parameter:
        lineFluxError = lineFlux * parametersStdDev[0] / parameters[0]
        LineFluxError.set(j,i,lineIndex,obsCnt,lineFluxError)
        # Line velocity in km/s
        lineVelocity = 1e-3*c*(lineFreqsCorr[lineCnt]-parameters[1])/lineFreqsCorr[lineCnt]
        LineVelocity.set(j,i,lineIndex,obsCnt,lineVelocity)
        # Line velocity error in km/s
        lineVelocityError = 1e-3*c*parametersStdDev[1]/lineFreqsCorr[lineCnt]
        LineVelocityError.set(j,i,lineIndex,obsCnt,lineVelocityError)
        #if lineFreqs[lineCnt] in CO12Lines:
        # print '%4.3f GHz, %1.3f GHz, %2f km/s, %2f km/s, %1.3e W m-2, %1.3e W m-2\'
        # (%lineFreqsCorr[lineCnt], lineFwhm, lineVelocity, lineVelocityError, lineFlux,
lineFluxError)
        lineCnt += 1
        pass
        pass
        pFlux.addLayer(LayerXY(CO12Lines,
LineFlux[0,0,:,obsCnt],line=0,symbol=14,symbolSize=3,name='%X'%obsid))
        pVel.addLayer(LayerXY(CO12Lines,
LineVelocity[0,0,:,obsCnt],line=0,symbol=14,symbolSize=3,name='%X'%obsid))
        obsCnt += 1
        pass # obsids
        i += 1
        pass # versions
        j += 1
        pass # calibrations
        obsCnt = 0
        for obsid in obsids:
            SpireResultsLineSources['Line Flux %X'%obsid]=Column(data=LineFlux[0,0,:,obsCnt], \
                unit=(herschel.share.unit.Power.WATTS,
herschel.share.unit.Length.METERS**2),description=src)
            SpireResultsLineSources['Line Velocity %X'%obsid]=Column(data=LineVelocity[0,0,:,obsCnt], \
                unit=(herschel.share.unit.Speed.KILOMETERS_PER_SECOND),description=src)
        obsCnt += 1
```



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	30 of 31

```
# MAKE COMPARISON PLOTS
# INTEGRATED LINE FLUX
avgLine = Double2d(len(versions)*len(calibrations), len(CO12Lines))
stdLine = Double2d(len(versions)*len(calibrations), len(CO12Lines))
#
for k in range(len(calibrations)):
    for i in range(len(versions)):
        for j in range(len(CO12Lines)):
            avgLine.set(i,j,MEAN(LineFlux[k,i,j,:]))
            stdLine.set(i,j,STDDEV(LineFlux[k,i,j,:]))
        # Plot the average integrated line flux
        pSngFlux = PlotXY()
        pSngFlux.addLayer(LayerXY(CO12Lines,avgLine[0,:], line=0, symbol=14, symbolSize=4,
color=java.awt.Color.RED))
        pSngFlux.addLayer(LayerXY(CO12Lines,avgLine[0,:]-stdLine[0,:], line=0, symbol=20,
symbolSize=7, color=java.awt.Color.RED))
        pSngFlux.addLayer(LayerXY(CO12Lines,avgLine[0,:]+stdLine[0,:], line=0, symbol=19,
symbolSize=7, color=java.awt.Color.RED))
        pSngFlux.setXtitle('Line Frequency [GHz]')
        pSngFlux.setYtitle('')
        pSngFlux.setTitleText('Average Line Flux [W m-2] %s'%src)
        pSngFlux.getLegend().setVisible(0)
        for oCnt in range(len(CO12Lines)):
            print CO12Lines[oCnt], avgLine[0,oCnt], stdLine[0,oCnt]
#
# LINE VELOCITY
#
avgLine = Double2d(len(versions)*len(calibrations), len(CO12Lines))
stdLine = Double2d(len(versions)*len(calibrations), len(CO12Lines))
#
for k in range(len(calibrations)):
    for i in range(len(versions)):
        for j in range(len(CO12Lines)):
            avgLine.set(i,j,MEAN(LineVelocity[k,i,j,:]))
            stdLine.set(i,j,STDDEV(LineVelocity[k,i,j,:]))
        # Plot the average integrated line flux
        pSngFlux = PlotXY()
        pSngFlux.addLayer(LayerXY(CO12Lines,avgLine[0,:], line=0, symbol=14, symbolSize=4,
color=java.awt.Color.RED))
        pSngFlux.addLayer(LayerXY(CO12Lines,avgLine[0,:]-stdLine[0,:], line=0, symbol=20,
symbolSize=7, color=java.awt.Color.RED))
        pSngFlux.addLayer(LayerXY(CO12Lines,avgLine[0,:]+stdLine[0,:], line=0, symbol=19,
symbolSize=7, color=java.awt.Color.RED))
        pSngFlux.setXtitle('Line Frequency [GHz]')
        pSngFlux.setYtitle('')
        pSngFlux.setTitleText('Average Line Velocity [km s-1] %s'%src)
        pSngFlux.getLegend().setVisible(0)
    pass # calibrations
    pFlux.setXtitle('Line Frequency [GHz]')
    pFlux.setYtitle('')
    pFlux.getLegend().setVisible(1)
    pVel.setXtitle('Line Frequency [GHz]')
    pVel.setYtitle('')
    pVel.getLegend().setVisible(1)
#
# Plot the sky positions
pSky = PlotXY(titleText='%s (Ods %s)'%(src,days))
lSkyNominal = LayerXY(Double1d([sds.meta['raNominal'].value]),
Double1d([sds.meta['decNominal'].value]),symbol=17,line=2,color=java.awt.Color.BLACK)
pSky.addLayer(lSkyNominal)
pSky.addLayer(LayerXY(SkyPosition_RA,
SkyPosition_DEC,symbol=14,line=0,color=java.awt.Color.BLUE))
pSky.setPlotSize(3,3)
pSky.setSize(800,800)
pSky.setXtype(herschel.ia.gui.renderer.AxisConstants.Type.RIGHT_ASCENSION)
pSky.setYtype(herschel.ia.gui.renderer.AxisConstants.Type.DECLINATION)
```



Project Document

SPIRE FTS Line Source Analysis for Cross-Calibration

Ref:	SPIRE-BSS-REP-003269
Issue:	Issue 1.0
Date:	09 February 2012
Page:	31 of 31

```
pSky.setXtitle("RA [hh:mm:ss]")  
pSky.setYtitle("Dec [dd:mm:ss]")  
pass # source
```

```
simpleFitsWriter(file='/spire/tmp/SpireResultsLineSources.fits', product=SpireResultsLineSources)
```