



ParselTongue

Hans-Rainer Klöckner
&
Stephen Bourke



ParseITongue - appetizer

ParseITongue

- is a Python interface to classic AIPS and Orbit
- has been written by Mark Kettenis
- can be used in an interactive session or via a script
- can directly access the visibilities

Since it is python you have most of the tools available to play around with the data by using

- matplotlib (pylab) plotting library [<http://matplotlib.sourceforge.net/>]
- numpy array library [e.g. statistics] [<http://numpy.scipy.org/>]
- scipy scientific library [e.g. FFT] [<http://www.scipy.org/>]

ParselTongue - part I

- Get familiar running an AIPS 'TASK'
- Get header information of the data files
- Display table entries
- Plot visibilities of the UV - data file
- Modify or add header entries and change visibilities (this is against the AIPS convention!)

- Using AIPSLite - Python interface to ParselTongue that does not require an AIPS installation; written by Stephen Bourke [<http://www.jive.nl/~bourke/aipslite/>]

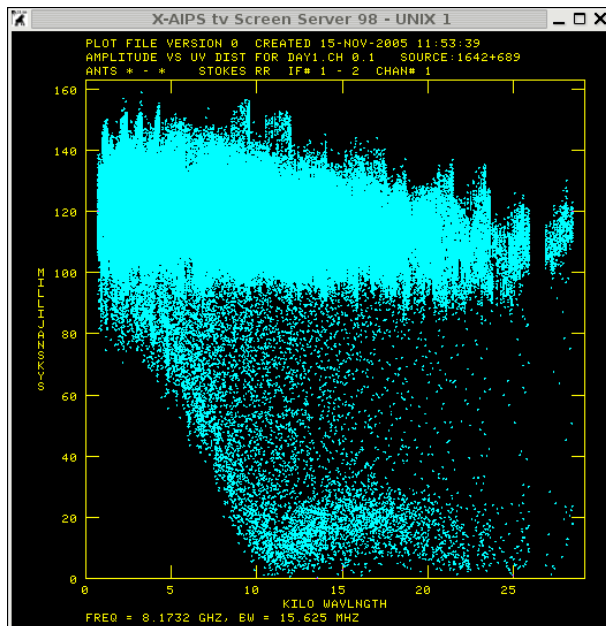
ParselTongue - part II

Setting up a calibration pipeline

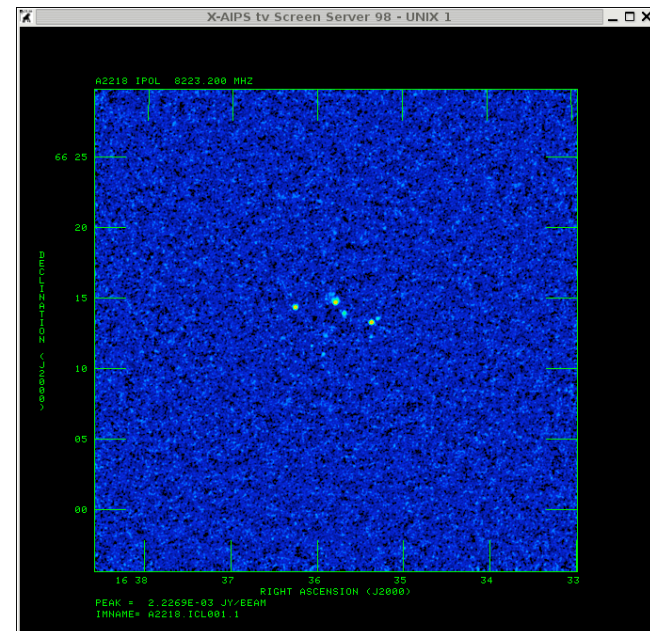
- Using VLA continuum observation at 8 GHz
- Following the calibration guideline in

[http://www-astro.physics.ox.ac.uk/~hrk/AIPS_TUTORIAL/HRK_AIPS_1.html]

raw uv- data



final image



ParselTongue - preliminaries

The best way to get PT installed and running on your system is to have a look into the codex

<http://www.jive.nl/parseltongue/codex.html>

With BIG THANKS TO MIKE SIPIOR

ParselTongue - get started interactive

:-) ParselTongue

Python 2.5.1 (r251:54863, Sep 20 2007, 12:35:05)

[GCC 4.0.1 (Apple Computer, Inc. build 5367)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

Welcome to ParselTongue 1.1.2

Please enter your AIPS user ID number: 456

```
>>>import AIPS
```

```
>>>mandl = AIPSTask('mandl')
```

```
>>>mandl.go()
```

```
>>>tv = AIPSTV()
```

```
>>>tv.start()
```

```
>>>kntr = AIPSTask('kntr')
```

```
>>>kntr.inname = 'MANDLELBROT'
```

```
>>>kntr.inclass = 'MANDL'
```

```
>>>kntr.inseq = 1
```

```
>>>kntr.indisk = 1
```

```
>>>kntr.dovec = -1
```

```
>>>kntr.docont = -1
```

```
>>>kntr.dotv = 1
```

```
>>>kntr.go()
```

```
>>>tv.kill()
```

ParselTongue - get started interactive

AIPS input

```
task 'mandl'  
>inp
```

```
AIPS 1: MANDL      Create a sub-section of the MANDLEBROT Set  
AIPS 1: Adverbs   Values                               Comments  
AIPS 1: -----  
AIPS 1: OBJECT    ' '                               Source name  
AIPS 1: IMSIZE    0           0           Output image size (cells)  
AIPS 1: CELLSIZE  0           0           Cellsize (0=Fill X-Y Range)  
AIPS 1: OUTNAME   ' '                               Output image name (name)  
AIPS 1: OUTCLASS  ' '                               Output image name (class)  
AIPS 1: OUTSEQ    0                               Output image name (seq. #)  
AIPS 1: OUTDISK   1                               Output image disk unit #.  
AIPS 1: CPARM     *all 0           Set Region selection,  
AIPS 1:                               1:XMin;2:YMin;3:XMax;4:YMax  
AIPS 1:                               5:Max Value for Set (0=> 256)
```

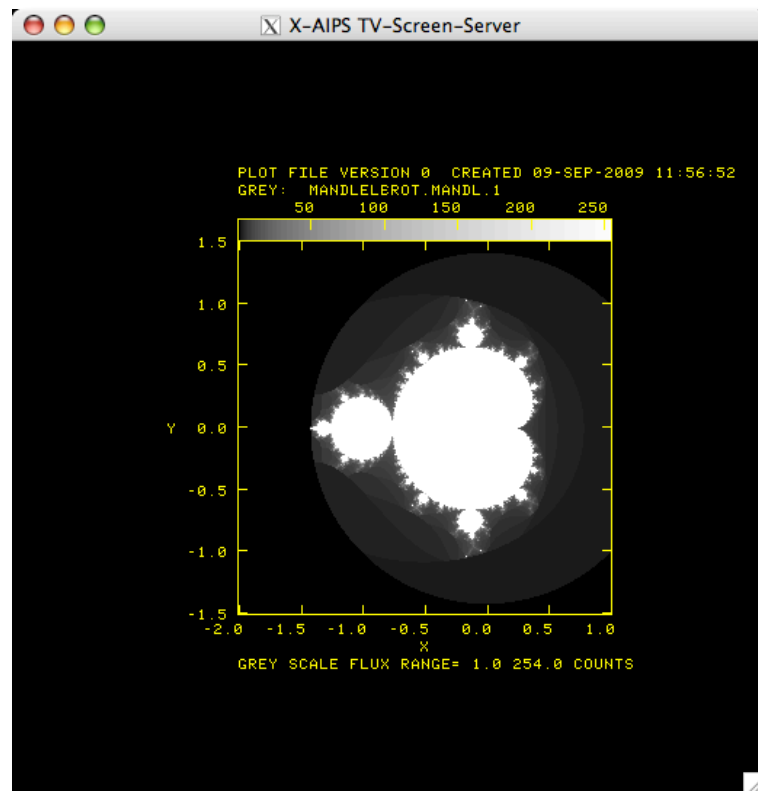
ParselTongue - get started interactive

MANDL.HLP

```
; MANDL
;-----
;! creates an image of a subset of the Mandelbrot Set
;# Task Imaging
;-----
;; Copyright (C) 1995
;; Associated Universities, Inc. Washington DC, USA.
;;                               Charlottesville, VA 22903-2475 USA
;-----
;-----
MANDL      LLLLLLLLLLLLLLUUUUUUUUUUUU CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
MANDL      Create a sub-section of the MANDLEBROT Set
OBJECT          Source name
IMSIZE          Output image size (cells)
CELLSIZE        Cellsize (0=Fill X-Y Range)
OUTNAME         Output image name (name)
OUTCLASS        Output image name (class)
OUTSEQ          -1.0      9999.0 Output image name (seq. #)
OUTDISK         0.0      9.0  Output image disk unit #.
CPARM          Set Region selection,
                1:XMin;2:YMin;3:XMax;4:YMax
                5:Max Value for Set (0=> 256)
-----
MANDL
Task: Mandelbrot set capability for AIPS. Can make an image as ...
```


ParseITongue - get started scripting

ParseITongue PT_TUT1.py



```
PT_TUT1.py
import AIPS
from AIPSTask import AIPSTask
from AIPSTV import AIPSTV

AIPS.userno = 456 # set the aips user number

AIPS.log = open('./TUT1.log','w') # define the log file

mandl = AIPSTask('mandl')
mandl.go()

tv = AIPSTV()
tv.start()

kntr = AIPSTask('kntr')
kntr.inname = 'MANDELBROT'
kntr.inclass = 'MANDL'
kntr.inseq = 1
kntr.indisk = 1
kntr.dovec = -1
kntr.docont = -1
kntr.dotv = 1
kntr.go()

print 'AIPS TV is goin down in 10 seconds'
sleep(10)

tv.kill()
```

ParselTongue - get information

```
>pca
AIPS 1: Catalog on disk 1
AIPS 1: Cat Usid Mapname      Class  Seq Pt      Last access      Stat
AIPS 1:      1  456 MANDLELBROT .MANDL .      1 MA 09-SEP-2009 12:02:39

>imhe
AIPS 1: Image=                (MA)          Filename=MANDLELBROT .MANDL .      1
AIPS 1: Telescope=           Receiver=
AIPS 1: Observer=           User #= 456
AIPS 1: Observ. date=09-SEP-2009  Map date=09-SEP-2009
AIPS 1: Minimum= 1.00000000E+00  Maximum= 2.54000000E+02 COUNTS
AIPS 1: -----
AIPS 1: Type      Pixels  Coord value      at Pixel      Coord incr  Rotat
AIPS 1: X          256  -2.0000000E+00      1.00  1.1718750E-02  0.00
AIPS 1: Y          256  -1.5000000E+00      1.00  1.1718750E-02  0.00
AIPS 1: -----
AIPS 1: Maximum version number of extension files of type HI is 1
```

AIPS information

ParselTongue information

```
>>>image = AIPSImage('MANDLELBROT', 'MANDL', 1, 1)
```

```
>>>image.header
```

```
{'ndim': 2, 'observer': '', 'object': '', 'ctype': ['X', 'Y', '', '', '', '', ''], 'cdelt': [0.01171875, 0.01171875, 1.0, 1.0, 1.0, 1.0, 1.0], 'crval': [-2.0, -1.5, 0.0, 0.0, 0.0, 0.0, 0.0], 'date_obs': '2009-09-09', 'crota': [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0], 'bpa': 0.0, 'niter': 0, 'crpix': [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0], 'telescop': '', 'epoch': 0.0, 'bmaj': 0.0, 'naxis': [256, 256, 1, 1, 1, 1, 1], 'instrume': '', 'bmin': 0.0, 'bunit': 'COUNTS', 'date_map': '2009-09-09', 'datamin': 1.0, 'datamax': 254.0}
```

ParselTongue - get information

it's a python dictionary

```
>>> image.header.keys()
['ndim', 'observer', 'object', 'ctype', 'cdelt', 'crval', 'date_obs', 'crota', 'bpa', 'niter', 'crpix', 'telescop', 'epoch', 'bmaj',
'naxis', 'instrume', 'bmin', 'bunit', 'date_map', 'datamin', 'datamax']
>>> image.header['datamax']
254.0
>>> image.header.has_key('object')
True
```

let's start with the next script (PT_TUT2.py)

-- Header information of simulated UV dataset --

uvdata.header gives:

```
{'ndim': 5, 'observer': '', 'object': '', 'ptype': ['UU-L', 'VV-L', 'WW-L', 'TIME1', 'BASELINE', '', '', '', '', '', '', '', ''], 'crota': [0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0], 'cdelt': [1.0, -1.0, 100000.0, 0.0, 0.0, 0.0, 0.0], 'date_obs': '2000-01-01', 'ctype': ['COMPLEX',
'STOKES', 'FREQ', 'RA', 'DEC', '', ''], 'ncorr': 1, 'nrparm': 5, 'crpix': [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0], 'telescop': '', 'sortord':
'TB', 'naxis': [3, 1, 1, 1, 1, 0, 0], 'instrume': '', 'epoch': 2000.0, 'bunit': 'JY', 'date_map': '2009-09-09', 'crval': [1.0, -1.0,
1667000055.3131104, 0.0, 60.0, 0.0, 0.0]}
```

uvdata.header.keys() provide the keyword in the header:

```
['ndim', 'observer', 'object', 'ptype', 'crota', 'cdelt', 'date_obs', 'ctype', 'ncorr', 'nrparm', 'crpix', 'telescop', 'sortord', 'naxis',
'instrume', 'epoch', 'bunit', 'date_map', 'crval']
```

uvdata.header.has_keys() logical feedback is keyword exists:

```
False
```

uvdata.header['ctype'] gives:

```
['COMPLEX', 'STOKES', 'FREQ', 'RA', 'DEC', '', '']
```

ParseITongue - delete table and file

PT_TUT2.py

The following task are used:

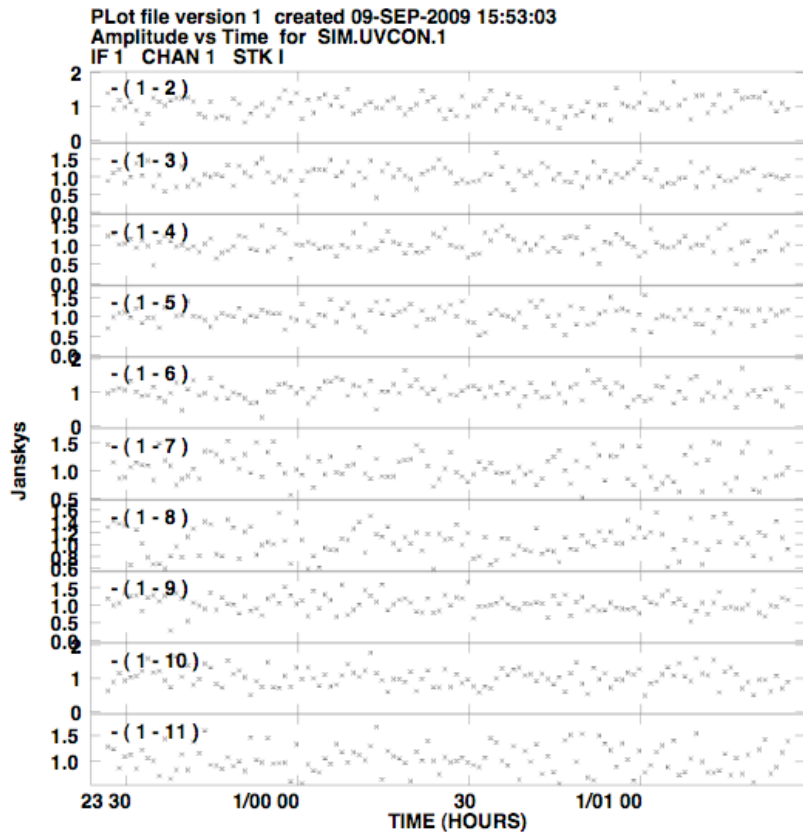
```
uvcon      = AIPSTask('uvcon')
vplot      = AIPSTask('vplot')
lwpla      = AIPSTask('lwpla')
uvmod      = AIPSTask('uvmod')
```

- outputs the UVDATA header information
- produces 2 PS files
- deletes an attached PL table and UVDATA set

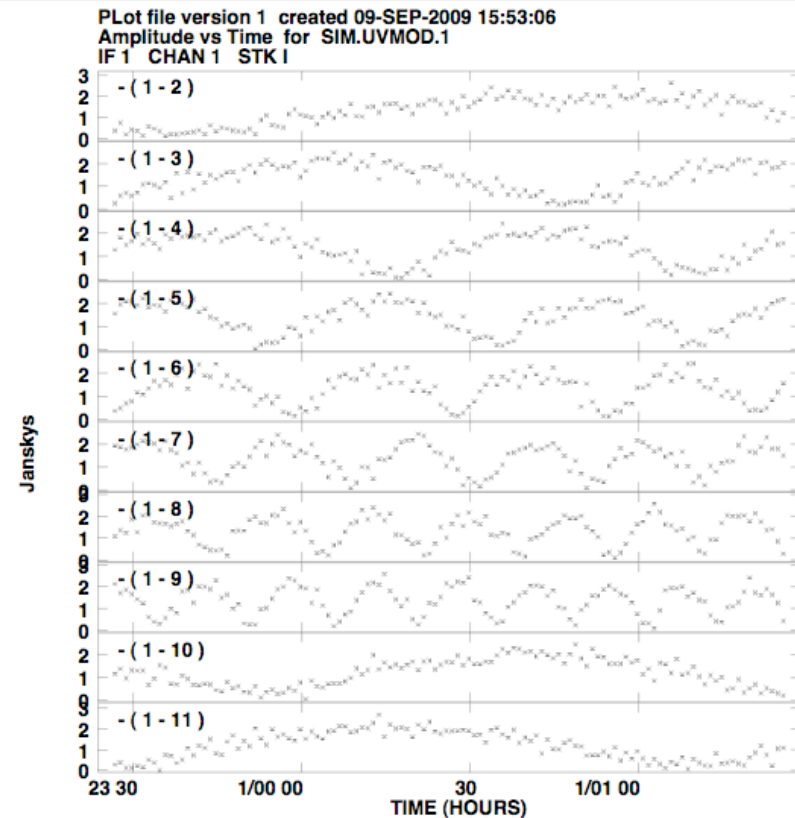
```
uvdata = AIPSUVDData('SIM','UVCON', 1, 1)
uvdata.zap_table('PL', -1)
uvdata.zap()
```

ParselTongue - delete table and file

Output PS files of PT_TUT2.py



PT_TUT2_1_1JY.PS



PT_TUT2_2_1JY.PS

ParselTongue - tables & visibilities

content of a table

PT_TUT3.py

```
antab = uvdata2.table('AN',1)

x,y,z = [],[],[]
for row in antab:
    x.append(row.stabxyz[0])
    y.append(row.stabxyz[1])
    z.append(row.stabxyz[2])
```

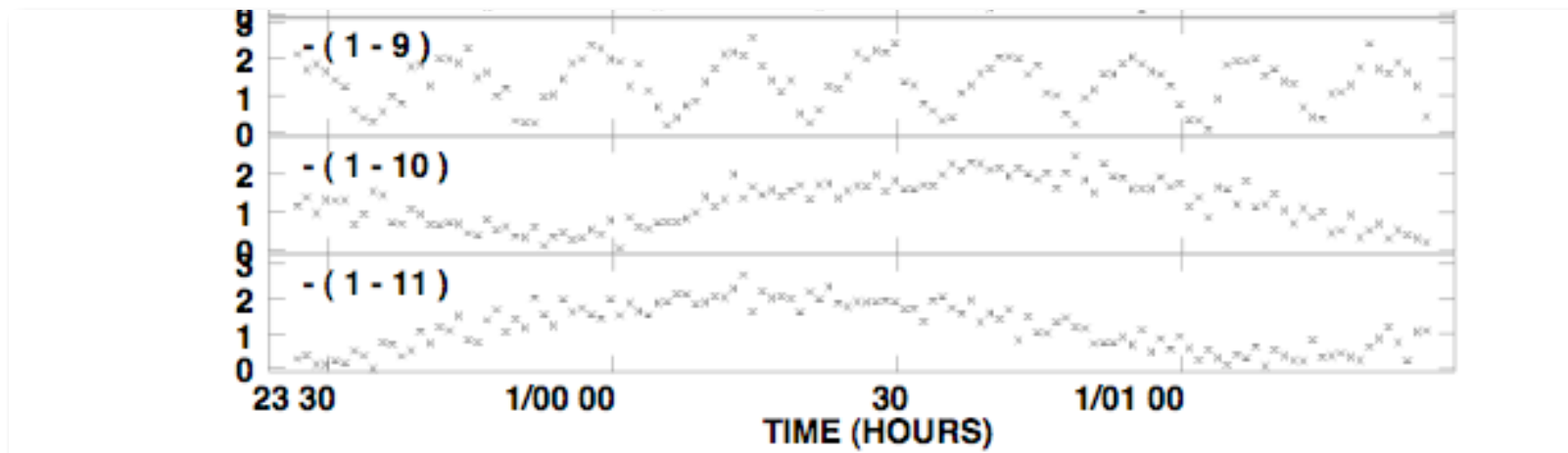
get the visibilities

```
from Wizardry.AIPSDData import AIPSUVDData as AIPSDDataVIS

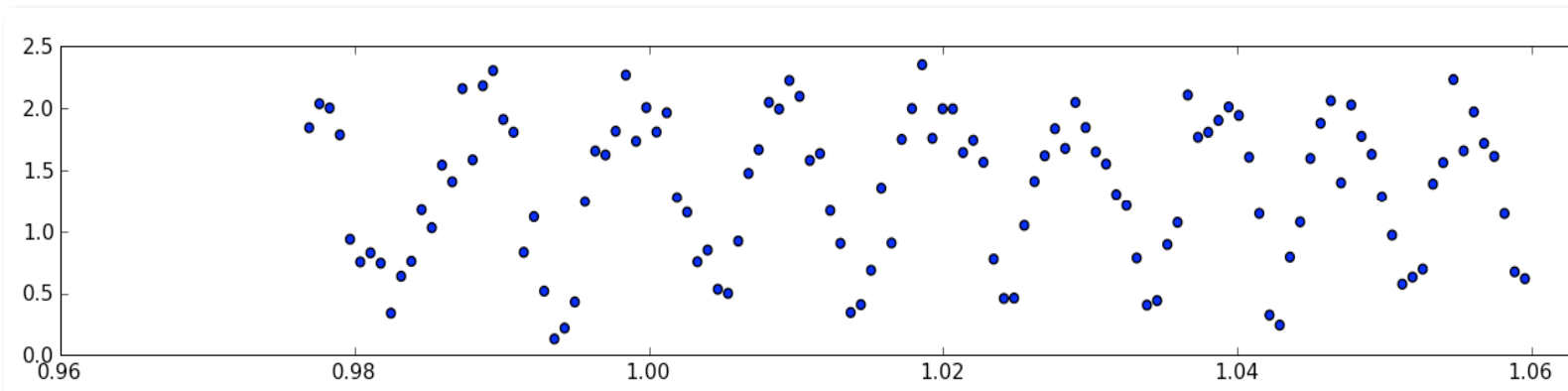
visdata = AIPSDDataVIS(data[0],data[1],data[2],data[3])

for visibility in visdata:
    u    = visibility.uvw[0]
    v    = visibility.uvw[1]
    w    = visibility.uvw[2]
    real = visibility.visibility[IF,chan,st,0]
    imag = visibility.visibility[IF,chan,st,1]
```

ParselTongue - tables & visibilities



Baseline 1-9



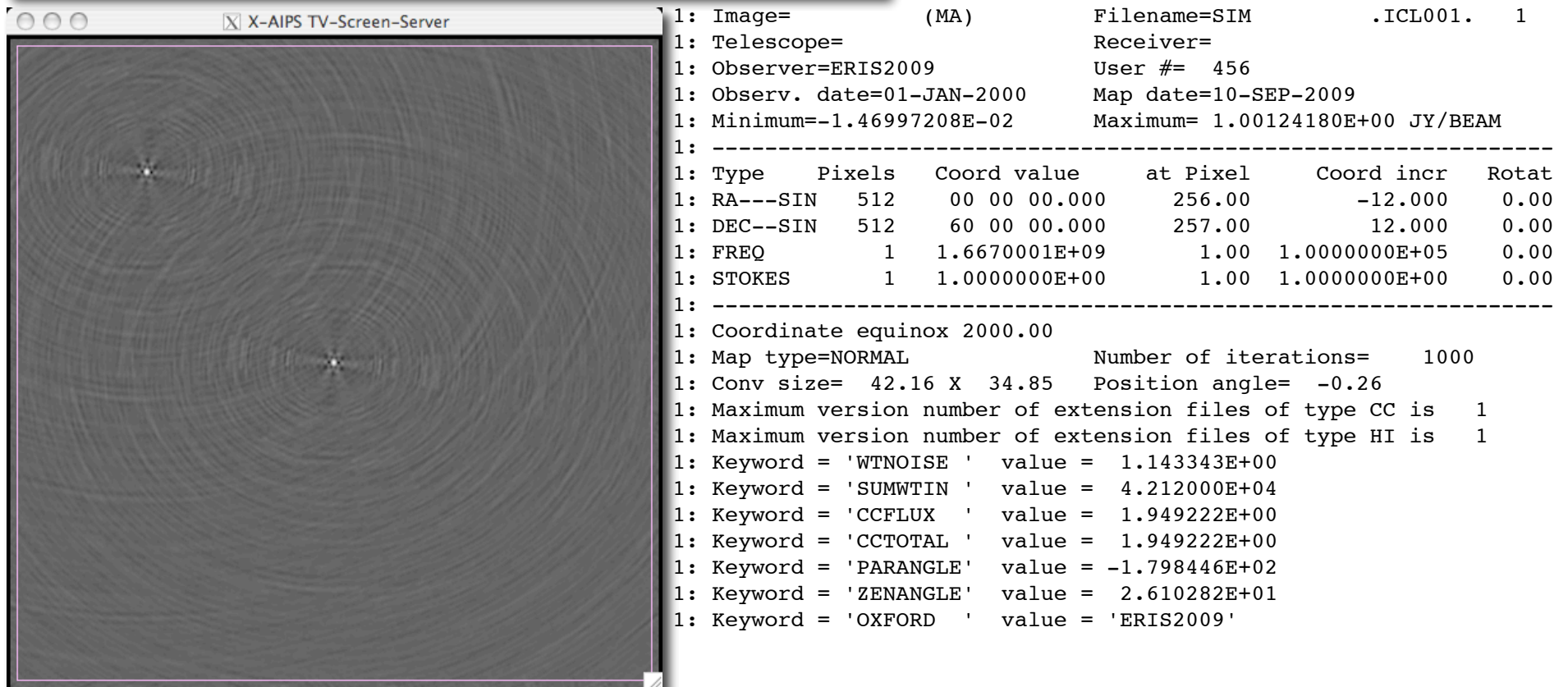
ParselTongue - edit header & tables

```
from Wizardry.AIPSDData import AIPSIImage as WizAIPSIImage
edimage2 = WizAIPSIImage('SIM','ICL001', 1, 1)
```

```
edimage2.header['observer'] = 'ERIS2009'
edimage2.header.update()
```

```
edimage2.keywords['OXFORD'] = 'ERIS2009'
edimage2.keywords.update()
```

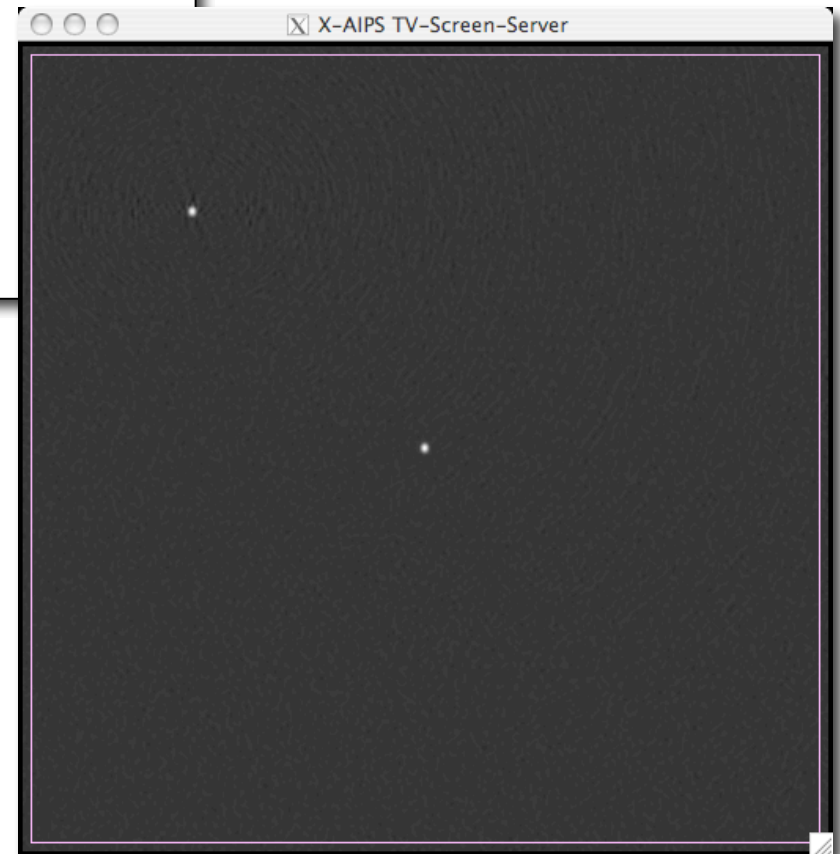
PT_TUT4.py



ParselTongue - edit header & tables

```
from Wizardry.AIPSDData import AIPSImage as WizAIPSImage
edimage2 = WizAIPSImage('SIM','ICL001', 1, 1)
```

```
edmftab2 = edimage2.table('MF',2)
for row in edmftab2:
    row.i_flux = -999
    row.update()
    row['err_qflx'] = -999
    row.update()
```



i_flux, err_flux, peak_int

```
[0.9993823766708374, 0.0027634163852781057, 1.0016094446182251,
1.0089050531387329, 0.0028133804444223642, 0.98438650369644165]
```

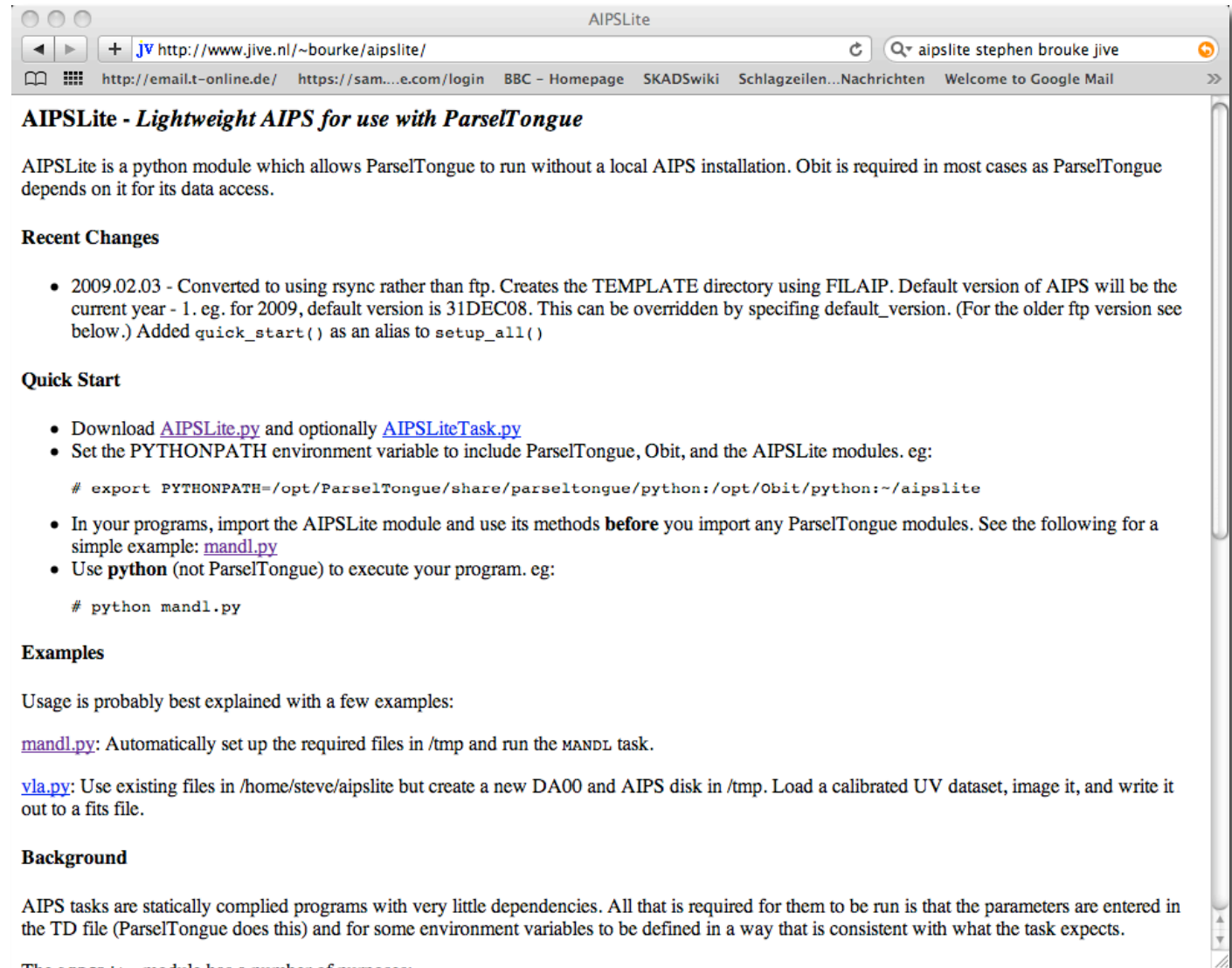
ParselTongue - modify visibilities

PT_TUT5.py

```
visdata = AIPSDataVIS(data[0],data[1],data[2],data[3])
for visibility in visdata:
    if visibility.baseline[0] == basel[0] and visibility.baseline[1] == basel[1]:
        #
        # Here the weights of a visibility set has been changed to 0
        # in AIPS that indicate that the data has been flagged
        #
        weight.append(visibility.visibility[IF,chan,st,2])
        visibility.visibility[IF,chan,st,2] = 0
        visibility.update()
```

ParselTongue - AIPSLite

Use python instead of ParselTongue to run scripts !



AIPSLite - Lightweight AIPS for use with ParselTongue

AIPSLite is a python module which allows ParselTongue to run without a local AIPS installation. Obit is required in most cases as ParselTongue depends on it for its data access.

Recent Changes

- 2009.02.03 - Converted to using rsync rather than ftp. Creates the TEMPLATE directory using FILAIP. Default version of AIPS will be the current year - 1. eg. for 2009, default version is 31DEC08. This can be overridden by specifying default_version. (For the older ftp version see below.) Added `quick_start()` as an alias to `setup_all()`

Quick Start

- Download [AIPSLite.py](#) and optionally [AIPSLiteTask.py](#)
- Set the PYTHONPATH environment variable to include ParselTongue, Obit, and the AIPSLite modules. eg:

```
# export PYTHONPATH=/opt/ParselTongue/share/parseltongue/python:/opt/Obit/python:~/aipslite
```
- In your programs, import the AIPSLite module and use its methods **before** you import any ParselTongue modules. See the following for a simple example: [mandl.py](#)
- Use **python** (not ParselTongue) to execute your program. eg:

```
# python mandl.py
```

Examples

Usage is probably best explained with a few examples:

[mandl.py](#): Automatically set up the required files in /tmp and run the MANDL task.

[vla.py](#): Use existing files in /home/steve/aipslite but create a new DA00 and AIPS disk in /tmp. Load a calibrated UV dataset, image it, and write it out to a fits file.

Background

AIPS tasks are statically compiled programs with very little dependencies. All that is required for them to be run is that the parameters are entered in the TD file (ParseITongue does this) and for some environment variables to be defined in a way that is consistent with what the task expects.

PT - additional information

<http://wiki.astrogrid.org/pub/Astrogrid/RadioAgenda/pt-demo.html>

<http://www.jive.nl/parseltongue/codex.html#using-scripting>

parseltongue-1.1.2/NEWS file in the PT - distribution

ParselTongue - part II

http://www-astro.physics.ox.ac.uk/~hrk/AIPS_TUTORIAL/HRK_AIPS_1.html

Download the data file DAY1.CH0.UV.FITS

AIPS task need:

`fitld, indxr, listr, uvplt, snplt, uvflg, vplot,`
`quack, setjy, calib, getjy, clcal, split, splat,`
`imagr, fittp`

