

Data Analysis: IDL meets Python



Dominic Zarro (ADNET Systems, Inc.)

GSFC SunPy Users Group Meeting
April 2017

Outline

- Setting up Python-IDL Bridge Environment
 - Windows
 - Linux
- Calling IDL from Python:
 - Plotting IDL array
 - Copying variables from IDL to Python
 - Copying variables from Python to IDL
- Real World Examples:
 - Read FITS file
 - Calibrate SDO/AIA image
 - Plot GOES light curves
 - Search Virtual Solar Observatory (VSO)

Setting up the Bridge Environment

- Minimum requirements: Python 2.7 and IDL 8.5
- Python 3.4 supported in IDL 8.6
- Anaconda distribution preferred
- Useful links:
 - https://github.com/dzarro/python-idl/blob/master/IDL_meets_Python.pdf
(this presentation)
 - <https://www.harrisgeospatial.com/docs/pythontoidl.html>
 - <http://blog.rtwilson.com/ten-little-idl-programs-in-python/>
 - <http://mathesaurus.sourceforge.net/idl-numpy.html>
 - <http://sunpy.org>

Windows

Set environment variables via Control Panel or use [pyidl.bat](#)

IDL_DIR =>

C:\Program Files\Exelis\IDL85

SSW => C:\ssw

SSW_INSTR => aia hmi secchi

PYTHONPATH => where Python modules live

%IDL_DIR%\bin\bin.x86_64;%IDL_DIR%\lib\bridges;%SSW%\gen\python
\bridge

Linux

```
# Set environment variables via .cshrc or .login
```

```
setenv IDL_DIR /usr/local/exelis/idl85
```

```
setenv SSW /solarsoft
```

```
setenv SSW_INSTR "aia hmi secchi"
```

```
setenv PYTHONPATH
```

```
    $IDL_DIR/bin/bin.linux.x86_64:$IDL_DIR/lib/bridges:$SSW/gen  
    /python/bridge
```

```
setenv LD_LIBRARY_PATH
```

```
    /usr/local/lib/python2.7:$IDL_DIR/bin/bin.linux.x86_64
```

Calling IDL from Python

```
# load Python-IDL bridge module

>>> import bridge

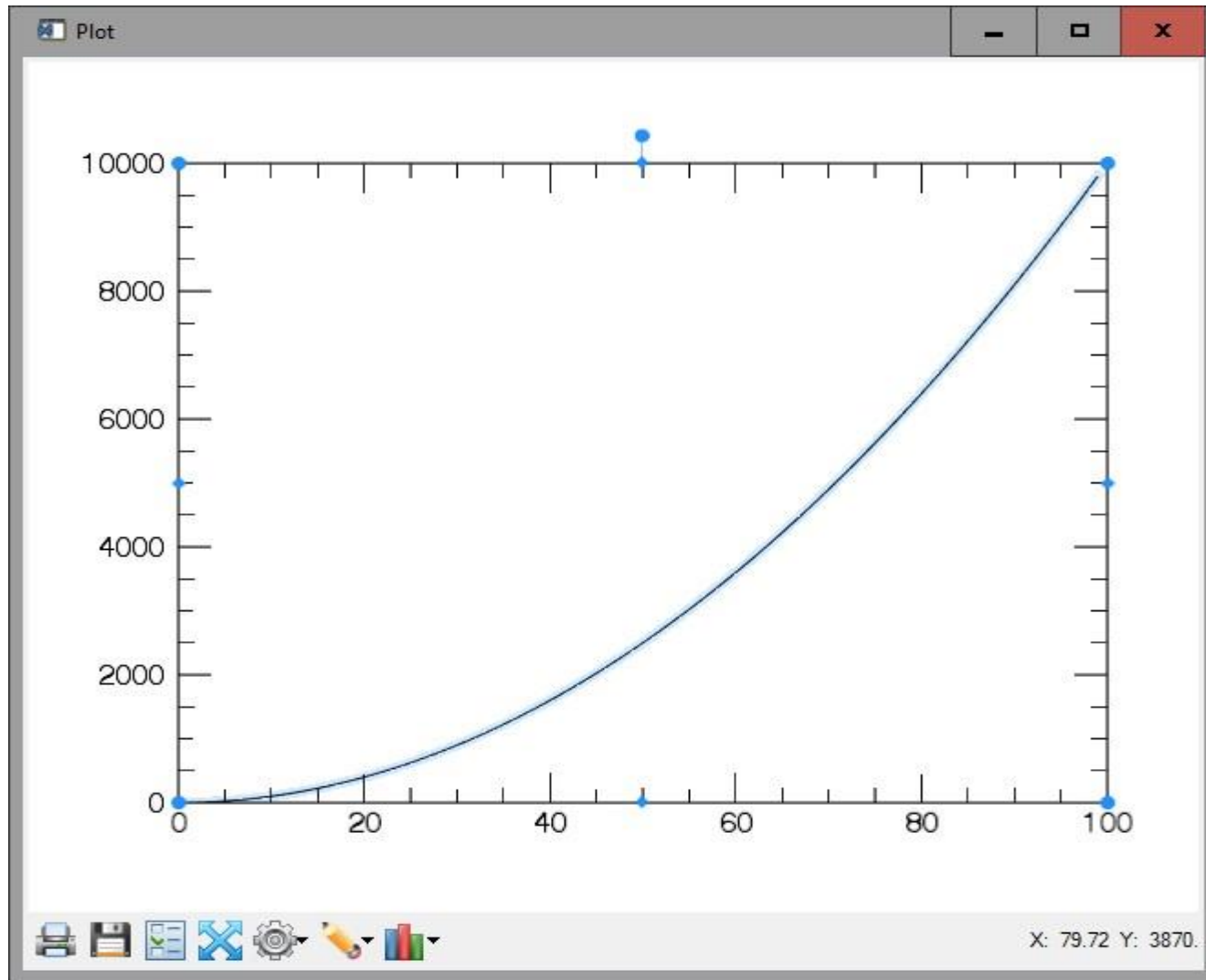
# execute IDL startup within Python

>>> IDL=bridge.startup()

# use IDL object run method to create and plot array in IDL

>>> IDL.run("x = FINDGEN(100)")
>>> IDL.run("y = x^2")
>>> IDL.run("z=plot(x,y)")
```

Result in IDL graphics



Copying variables from IDL to Python

```
# copy IDL variables as properties of IDL object

>>> x=IDL.x
>>> y=IDL.y

# IDL arrays become NumPy arrays

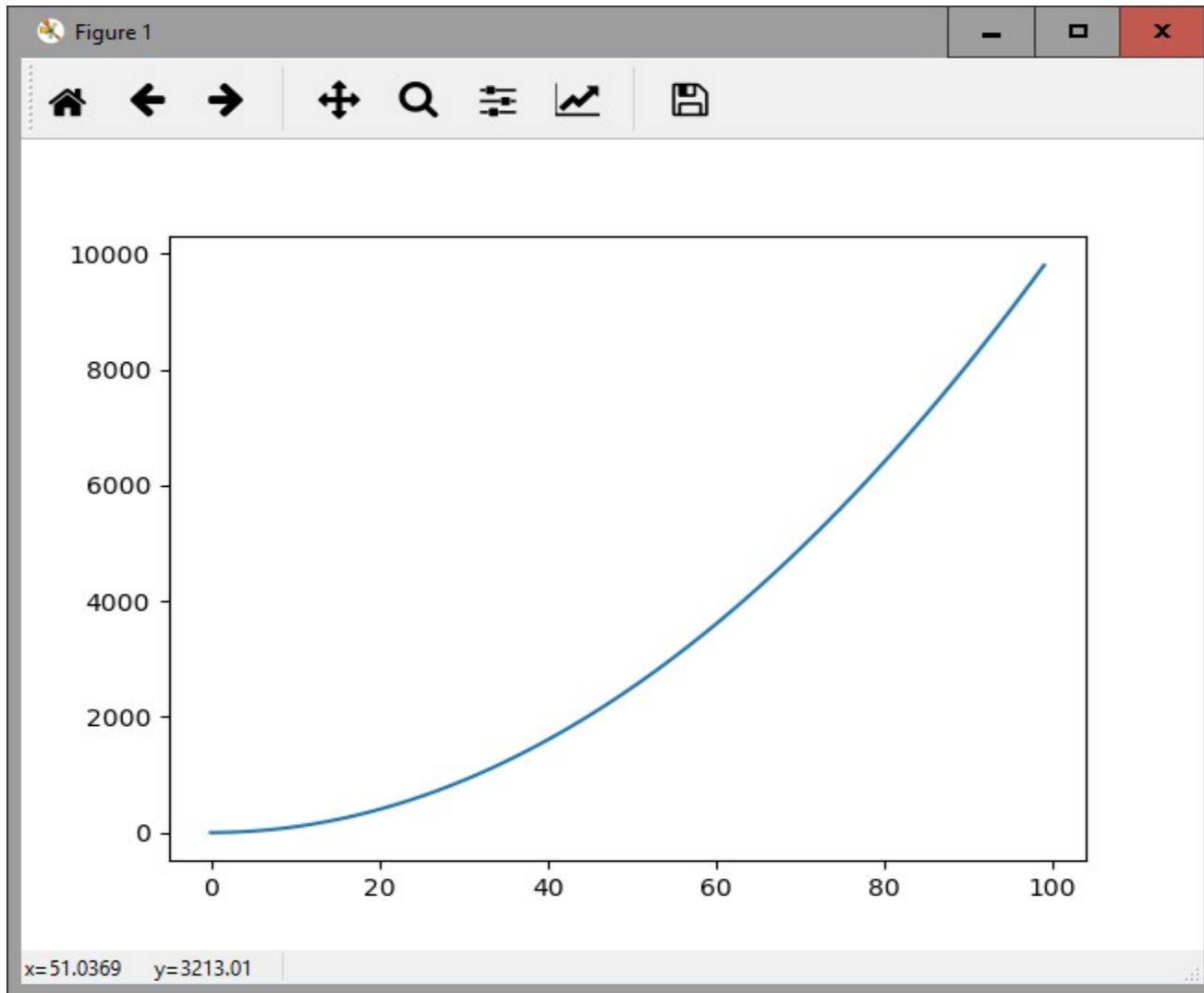
>>> x[0:5]
array([ 0.,  1.,  2.,  3.,  4.], dtype=float32)

>>> y[0:5]
array([ 0.,  1.,  4.,  9., 16.], dtype=float32)

# plot NumPy arrays using Matplot Pyplot

>>> import matplotlib.pyplot as plt
>>> plt.plot(x,y)
>>> plt.show()
```


Result in Python graphics



Copying variables from Python to IDL

```
# create NumPy arrays

>>> x=np.arange(1000)
>>> y=np.sin(2*np.pi*x/1000)

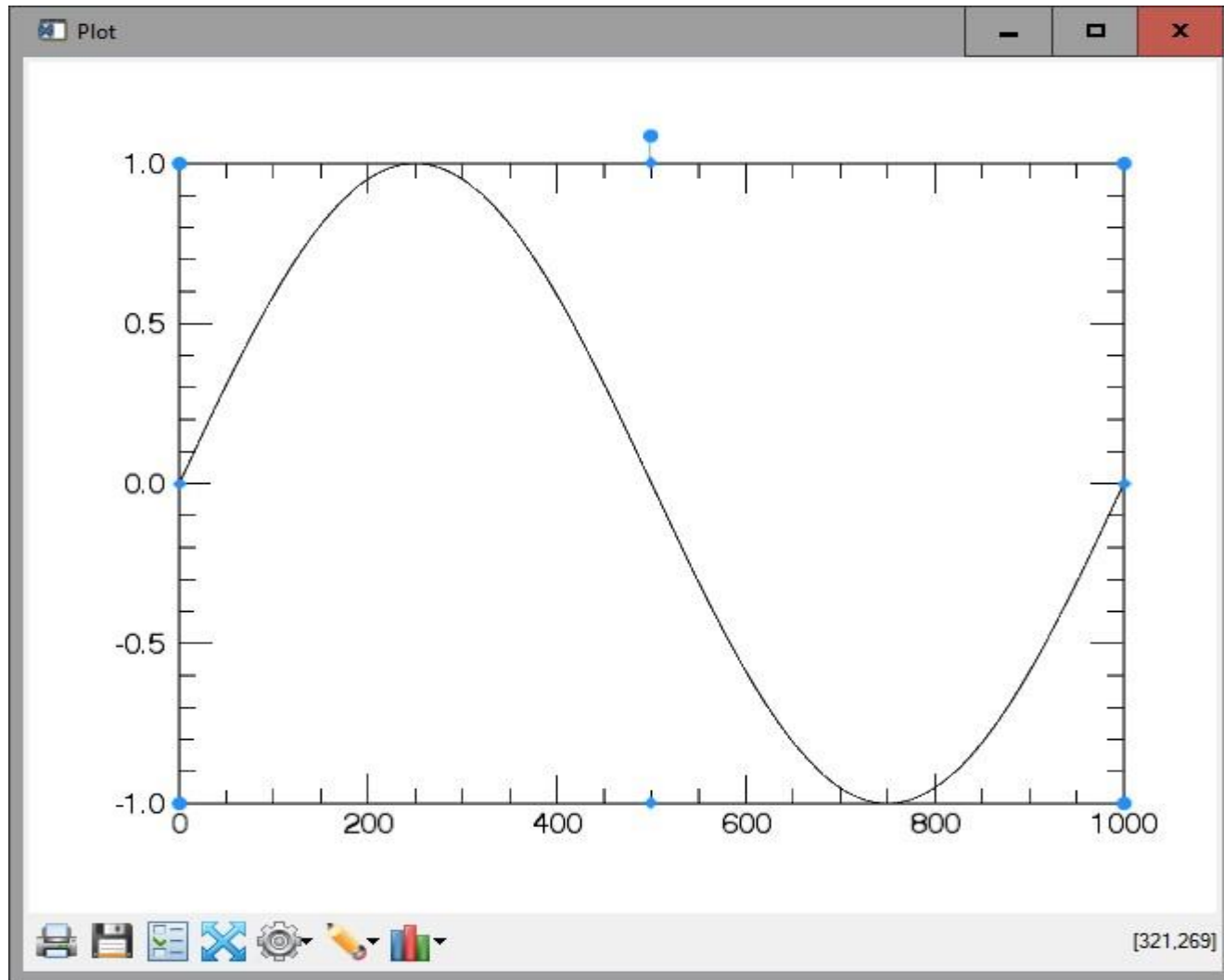
# copy arrays as properties to IDL object

>>> IDL.x=x
>>> IDL.y=y

# call IDL plot command as a method

>>> IDL.plot(x,y)
```

Result in IDL graphics



Read FITS file

```
# call IDL FITS reader from Python

>>> IDL.file="bbso\_halph\_f1\_20150312\_185727.fts"
>>> IDL.run("data=readfits(file,header)")

# copy FITS header Python

>>> header=IDL.header
>>> type(header)
<type 'list'>

>>> header[0:4]
['SIMPLE =                               T / Written by IDL:  Thu Mar
 12 12:38:26 2015                        ', 'BITPIX =                               16 /
 ', 'NAXIS =                               2 /
 ', 'NAXIS1 =                              2048 /
 ']
```

Display FITS image

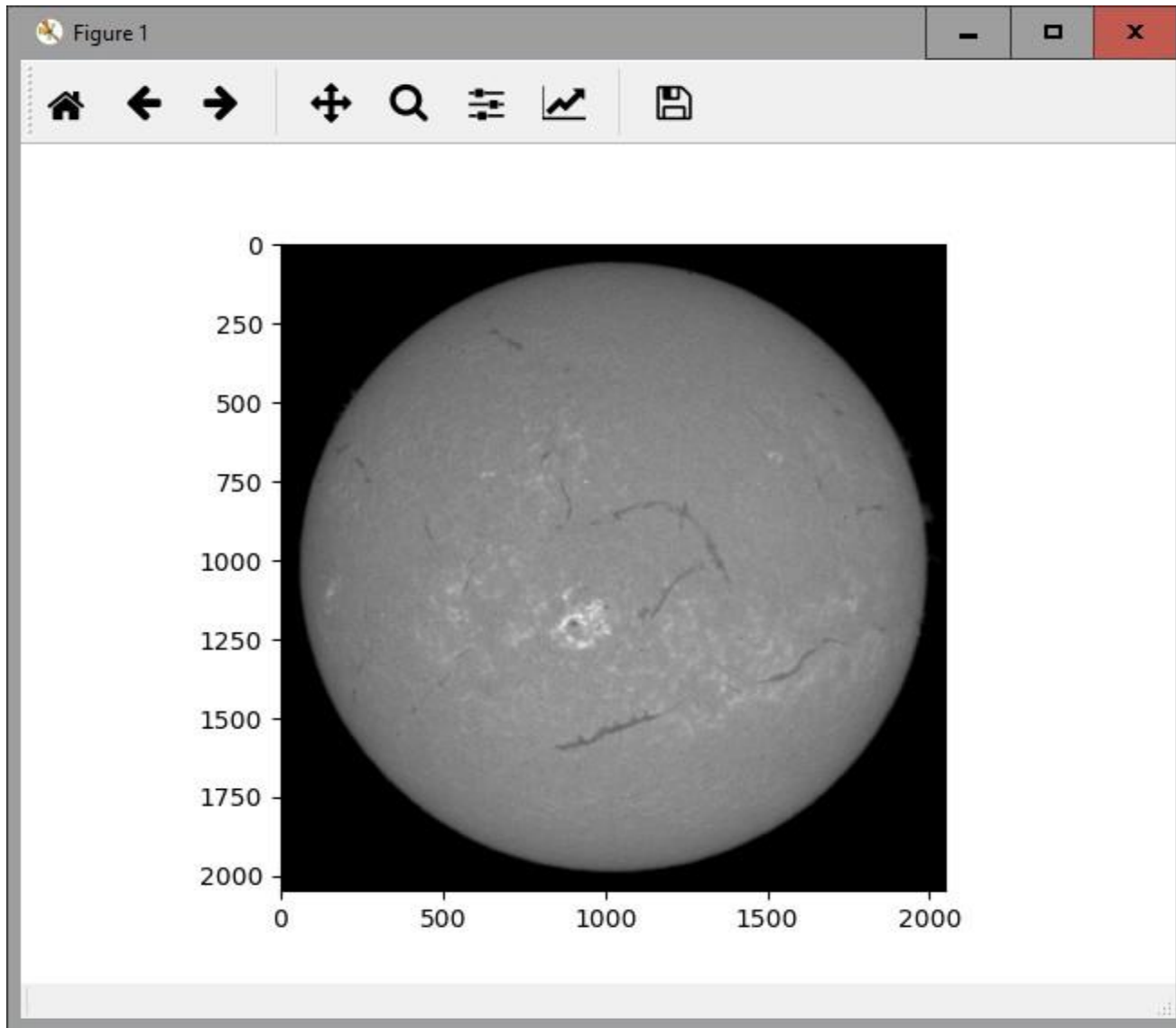
```
# copy FITS data to Python

>>> data=IDL.data
>>> type(data)
<type 'numpy.ndarray'>

# flip image North-South

>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> plt.imshow(np.flipud(data), cmap='grey')
>>> plt.show()
```

Result in Python graphics



Read and prep SDO/AIA image

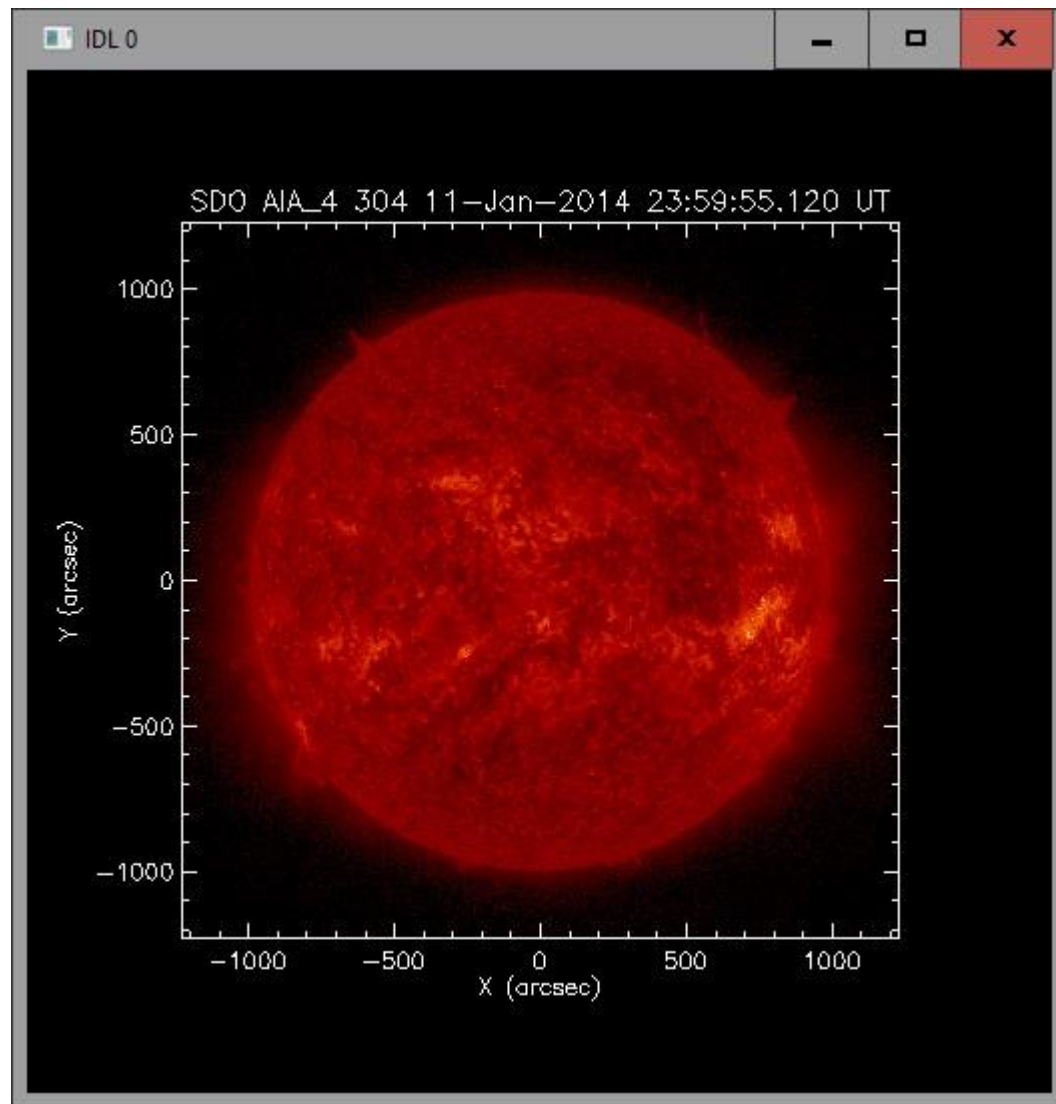
```
# Insert IDL commands into a batch file (e.g. script.pro)

file='aia.lev1.304A 2014-01-11T23 59 55.12Z.image lev1.fits'
read_sdo,file,index,data
aia_prep,index,data,pindex,pdata
index2map,pindex,pdata,amap
aia_lct,wave=304,/load
plot_map,amap,/log

# Execute batch file from Python

>>> IDL.run("@script")
```

Result in IDL graphics



Extract SDO/AIA image map

```
# copy IDL map structure to Python dictionary

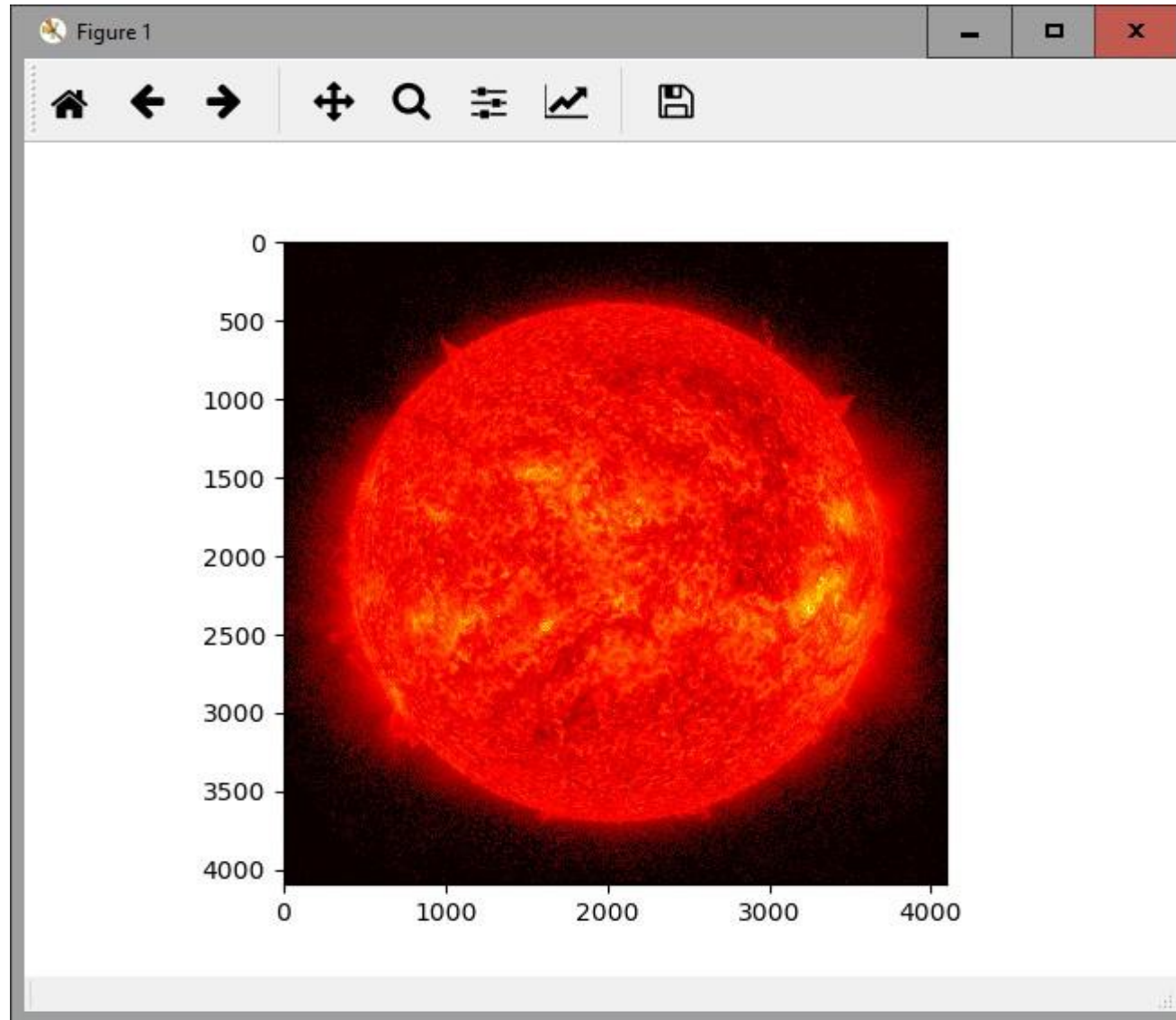
>>> map=IDL.amap
>>> type(map)
<type 'dict'>

>>> map.keys()
['XUNITS', 'RSUN', 'L0', 'ROLL_ANGLE', 'XC', 'YC', 'DY',
 'ROLL_CENTER', 'YUNITS', 'SOHO', 'DX', 'TIME', 'B0', 'DUR',
 'DATA', 'ID']

# plot on log scale (flip and remove zeroes)

>>> data=np.flipud(map['DATA'])
>>> (min, max) = (data[data > 0].min(), data.max())
>>> ldata = np.log10(data.clip(min,max))
>>> plt.imshow(ldata,cmap='hot')
>>> plt.show()
```

Result in Python graphics



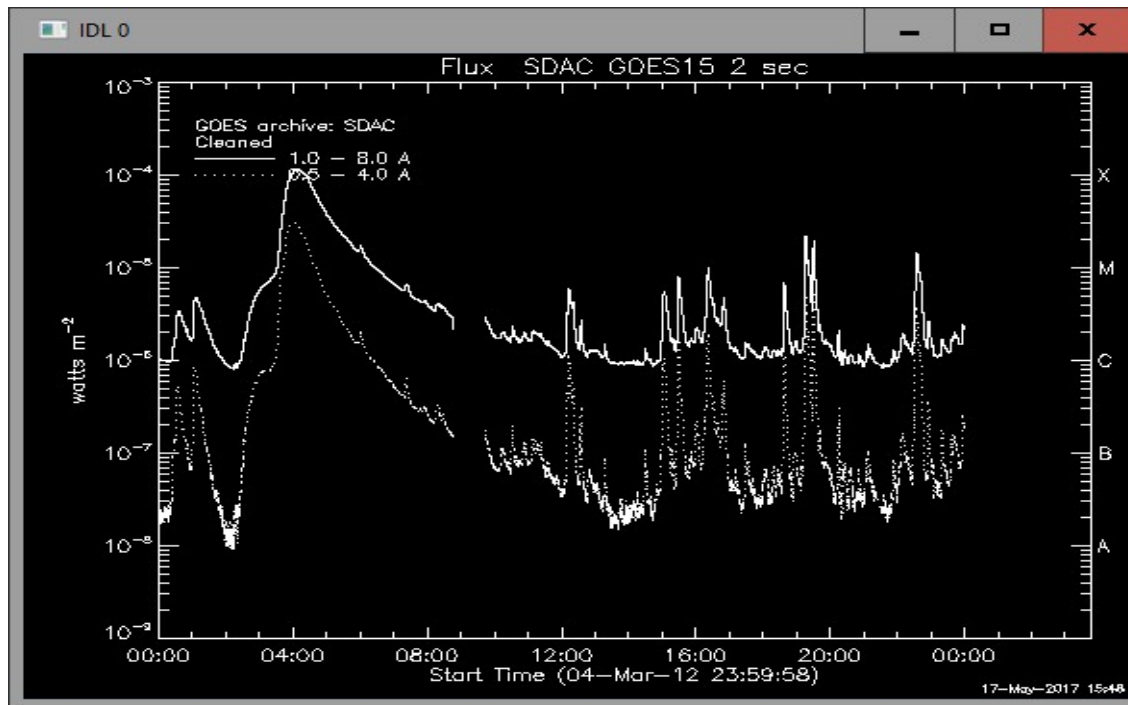
Read and plot GOES lightcurves

```
# create IDL GOES object
```

```
>>> IDL.run("goes=ogoes()")
```

```
>>> IDL.run("goes->read, '5-mar-12', '6-mar-12'")
```

```
>>> IDL.run("goes->plot")
```



Copy GOES lightcurve data into Python

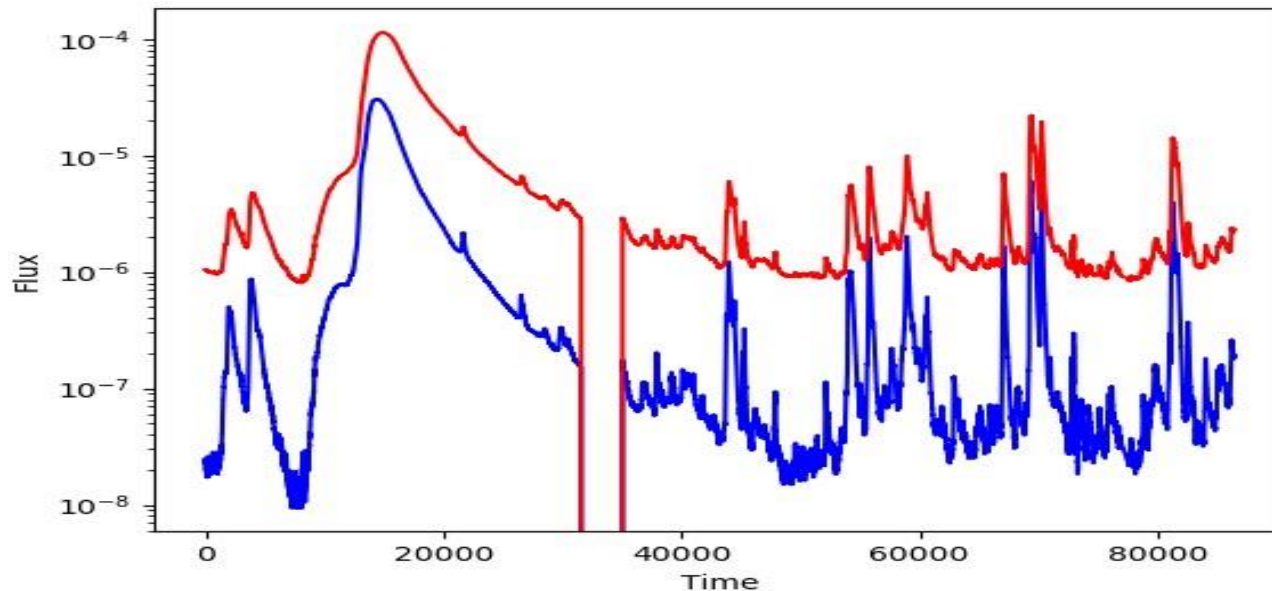
```
# copy IDL data array into NumPy array

>>> flux=IDL.goes.data
>>> flux
array([[ 1.05180004e-06,   1.05699996e-06,   1.05439995e-06, ...,
         2.32139996e-06,   2.32139996e-06,   2.32139996e-06],
       [ 2.38909994e-08,   2.27769998e-08,   2.50060008e-08, ...,
         1.89950001e-07,   1.88830001e-07,   1.88830001e-07]],
      dtype=float32)

>>> time=IDL.goes.time
>>> time
array([ 0.00000000e+00,   2.05000007e+00,   4.09600008e+00, ...,
        8.63958000e+04,   8.63978500e+04,   8.63998960e+04])
```

Plot GOES lightcurves in Python

```
>>> import matplotlib.pyplot as plt
>>> plt.plot(time,flux[0,:],color='r')
>>> plt.plot(time,flux[1,:],color='b')
>>> plt.yscale('log')
>>> plt.xlabel('Time')
>>> plt.ylabel('Flux')
>>> plt.show()
```



Search VSO from Python

```
# copy search command for STEREO/EUVI from Python to IDL

>>> cmd="files=vso_search('1-apr-17','2-apr-17',inst='euvi',/url)"
>>> IDL.cmd=cmd
>>> IDL.run(cmd)

# copy IDL results back to Python as List

>>> files=IDL.files

# each list element is a Dictionary

>>> file[0].keys()
['INFO', 'SOURCE', 'URL', 'PROVIDER', 'ECLIPSE', 'PERCENTD', 'WAVE',
 'DARK', 'INSTRUMENT', 'GETINFO', 'EXTENT', 'TIME', 'PHYSOBS',
 'DETECTOR', 'SIZE', 'EXPTIME', 'FILEID']

>>> file[0]['URL']
'https://stereo-ssc.nascom.nasa.gov/data/ins\_data/secchi/L0/a/img/euvi/20170401//20170401\_000530\_n4euA.fts'
```



IDL

Works great!