SpectraReading_Day1_Part1

March 21, 2016

- First day: Dealing with spectra
  - Reading/writing ascii files; handling fits files
  - Dealing with arrays
  - Interpolation
  - Fitting models to data
  - Plotting
- Second day: Dealing with tables and imaging
  - Displaying images
  - Cross-matching tables
  - Querying archives
  - Astrometry and WCS

** Reading an ascii file **

```
In [8]: %matplotlib inline
```

```
In [9]: # https://github.com/astropy/specutils/raw/master/specutils/io/tests/files/multispec_equispec.1
```

```
In [10]: %%bash
         curl -O https://raw.githubusercontent.com/astropy/specutils/master/specutils/io/tests/files/mul
```

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 27461  100 27461    0     0   146k      0 --:--:-- --:--:-- --:--:--  146k
```

```
In [11]: with open('multispec_equispec.11.dat','r') as fh: # fh is short for file handle
             # file consists of two columns, e.g.:
             # 14740.266391838  0.8220932
             # 14743.8622868028  -1.856567

             # declare empty list
             all_lines = []
             for line in fh:
                 # each line will split on a whitespace
                 all_lines.append(line.split())
```

It's more convenient to work with arrays. Also, we want the values to be floats intead of strings as they are now:

```
In [12]: all_lines[:2]
```

```
Out[12]: [['14740.266391838', '0.8220932'], ['14743.8622868028', '-1.856567']]
```

```
In [13]: float_lines = [list(map(float,x)) for x in all_lines]
```

```
In [14]: print("first two float lines: ",float_lines[:2])
         print("length(float_lines): ",len(float_lines))
```

```
first two float lines:  [[14740.266391838, 0.8220932], [14743.8622868028, -1.856567]]
length(float_lines):  1024
```

What if we want the array to have dimensions [2,1024] instead of [1024,2]?

```
In [15]: float_lines_inverted = list(zip(*float_lines))
```

```
In [16]: float_lines[0][:10]
```

```
Out[16]: [14740.266391838, 0.8220932]
```

Numpy arrays are much more convenient to work with and are generally faster. As long as you have a list of numbers (not a list of strings), they are easy to use:

```
In [17]: import numpy as np
         float_lines_array = np.array(float_lines)
```

```
In [18]: float_lines_array.shape
```

```
Out[18]: (1024, 2)
```

For example, transposing an array is much easier with numpy:

```
In [19]: float_lines_array.T.shape
```

```
Out[19]: (2, 1024)
```

```
In [20]: xaxis, yaxis = float_lines_array.T
```

With nested lists, you need to index each layer separately, whereas with numpy arrays you can index them together:

```
In [21]: float_lines_array[:5,1]
```

```
Out[21]: array([ 0.8220932, -1.856567 , -2.0807   , -2.75078  , -1.882897 ])
```

```
In [22]: float_lines[:5]
```

```
Out[22]: [[14740.266391838, 0.8220932],
          [14743.8622868028, -1.856567],
          [14747.4581817676, -2.0807],
          [14751.0540767325, -2.75078],
          [14754.6499716973, -1.882897]]
```

```
In [23]: # difficult to access the second column:
         list(zip(*float_lines[:5]))[1]
```

```
Out[23]: (0.8220932, -1.856567, -2.0807, -2.75078, -1.882897)
```

Arrays can be manipulated like any other number, and arithmetic operations will be applied to each element:

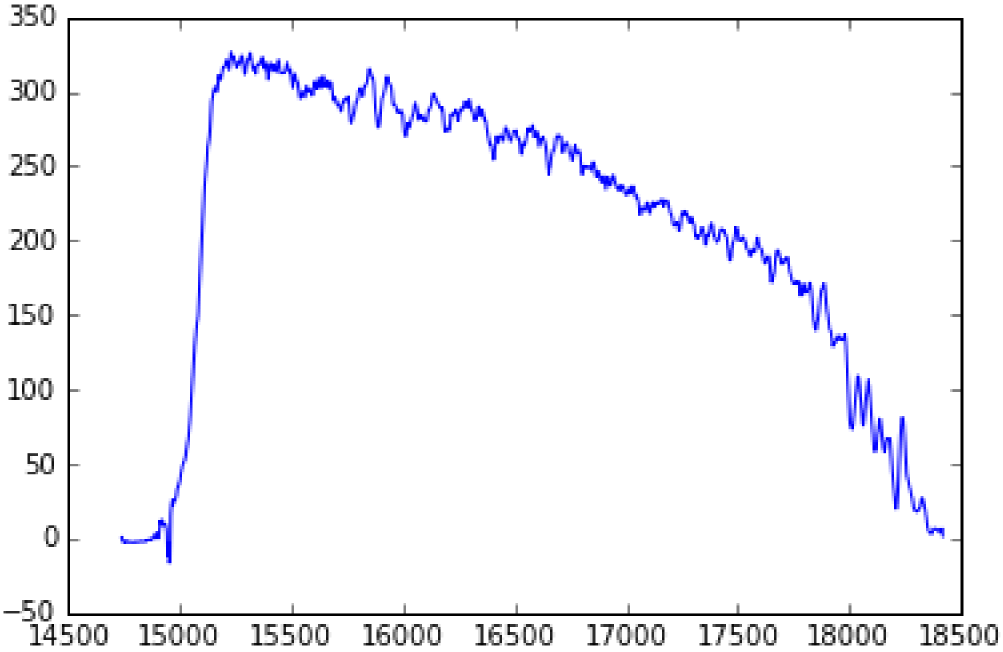
```
In [24]: 5 * float_lines_array[:5,1]
```

```
Out[24]: array([  4.110466,  -9.282835, -10.4035  , -13.7539  ,  -9.414485])
```

1 Plotting

```
In [25]: import pylab as pl
```

```
In [26]: pl.plot(xaxis, yaxis)
         pl.savefig("my_first_spectrum_plot.pdf")
```



2 Tools for reading ASCII files

```
In [27]: import numpy as np
```

```
In [28]: arr = np.loadtxt('multispec_equispec.11.dat')
         arr
```

```
Out[28]: array([[  1.47402664e+04,   8.22093200e-01],
                [  1.47438623e+04,  -1.85656700e+00],
                [  1.47474582e+04,  -2.08070000e+00],
                ...,
                [  1.84116752e+04,   5.27366100e+00],
                [  1.84152710e+04,   6.57225800e+00],
                [  1.84188669e+04,   1.60453100e+00]])
```

```
In [29]: arr = np.genfromtxt('multispec_equispec.11.dat')
         arr
```

```
Out[29]: array([[  1.47402664e+04,   8.22093200e-01],
                [  1.47438623e+04,  -1.85656700e+00],
                [  1.47474582e+04,  -2.08070000e+00],
                ...,
                [  1.84116752e+04,   5.27366100e+00],
                [  1.84152710e+04,   6.57225800e+00],
                [  1.84188669e+04,   1.60453100e+00]])
```

```
In [30]: arr = np.genfromtxt('multispec_equispec.11.dat', delimiter=" ", comments="#",
                             skip_header=0, skip_footer=0)
         arr
```

```
Out[30]: array([[  1.47402664e+04,              nan,   8.22093200e-01],
                [  1.47438623e+04,              nan,  -1.85656700e+00],
                [  1.47474582e+04,              nan,  -2.08070000e+00],
                ...,
                [  1.84116752e+04,              nan,   5.27366100e+00],
                [  1.84152710e+04,              nan,   6.57225800e+00],
                [  1.84188669e+04,              nan,   1.60453100e+00]])
```

```
In [31]: from astropy.table import Table
         from astropy.io import ascii
```

```
In [32]: tbl = Table.read('multispec_equispec.11.dat', format='ascii.no_header', delimiter=' ')
         tbl
```

```
Out[32]: <Table length=1024>
              col1        col2
            float64     float64
         ------------- ---------
         14740.2663918 0.8220932
         14743.8622868 -1.856567
         14747.4581818   -2.0807
         14751.0540767  -2.75078
         14754.6499717 -1.882897
         14758.2458667 -1.653645
         14761.8417616 -2.496639
         14765.4376566 -2.216392
         14769.0335516 -1.711144
         14772.6294465 -2.086175
                   ...       ...
         18386.5038862  6.753047
         18390.0997811  6.417622
         18393.6956761  6.072701
         18397.2915711  5.728085
          18400.887466  4.878081
          18404.483361  3.940828
         18408.0792559  4.006176
         18411.6751509  5.273661
         18415.2710459  6.572258
         18418.8669408  1.604531
```

```
In [33]: import pandas as pd
         ptbl = pd.read_csv('multispec_equispec.11.dat', delim_whitespace=True, header=None)
         ptbl
```

```
Out[33]:                  0         1
         0     14740.266392  0.822093
         1     14743.862287 -1.856567
         2     14747.458182 -2.080700
```

```
3     14751.054077  -2.750780
4     14754.649972  -1.882897
5     14758.245867  -1.653645
6     14761.841762  -2.496639
7     14765.437657  -2.216392
8     14769.033552  -1.711144
9     14772.629447  -2.086175
10    14776.225341  -2.388523
11    14779.821236  -2.401196
12    14783.417131  -2.646510
13    14787.013026  -2.633347
14    14790.608921  -2.327991
15    14794.204816  -2.146435
16    14797.800711  -2.082651
17    14801.396606  -2.673334
18    14804.992501  -2.370189
19    14808.588396  -2.146415
20    14812.184291  -2.337629
21    14815.780186  -2.144238
22    14819.376081  -1.720990
23    14822.971976  -1.555961
24    14826.567871  -2.047852
25    14830.163766  -2.028217
26    14833.759661  -1.917684
27    14837.355556  -2.099508
28    14840.951451  -1.696484
29    14844.547346  -1.599167
...            ...        ...
994   18314.585987  19.464080
995   18318.181882  21.615080
996   18321.777777  25.198370
997   18325.373672  27.101250
998   18328.969567  26.054030
999   18332.565462  24.030930
1000  18336.161357  22.149230
1001  18339.757252  19.462560
1002  18343.353147  15.519730
1003  18346.949042  11.143000
1004  18350.544937   7.884392
1005  18354.140831   5.898630
1006  18357.736726   4.498694
1007  18361.332621   3.895802
1008  18364.928516   3.249683
1009  18368.524411   3.085544
1010  18372.120306   4.522633
1011  18375.716201   5.928256
1012  18379.312096   6.679966
1013  18382.907991   6.624547
1014  18386.503886   6.753047
1015  18390.099781   6.417622
1016  18393.695676   6.072701
1017  18397.291571   5.728085
1018  18400.887466   4.878081
1019  18404.483361   3.940828
```

```
1020  18408.079256   4.006176
1021  18411.675151   5.273661
1022  18415.271046   6.572258
1023  18418.866941   1.604531

[1024 rows x 2 columns]
```

2.1 Speed Comparison

```
In [34]: %timeit pd.read_csv('multispec_equispec.11.dat', delim_whitespace=True, header=None)
```

```
1000 loops, best of 3: 1.06 ms per loop
```

```
In [35]: %timeit Table.read('multispec_equispec.11.dat', format='ascii.no_header', delimiter=' ')
```

```
1000 loops, best of 3: 1.38 ms per loop
```

2.2 Exercises

1. Read this text file:

   https://raw.githubusercontent.com/astropy/specutils/master/specutils/io/tests/files/AAO_11.txt which contains both a header and two data columns. Plot it, and save as a `.png` and as a `.pdf`

2. Write your own function for file reading. Based on the original example, write a function that reads a 2-column (or n-column) space-separated text file into a numpy array. Compare its execution time to that of `pandas.read_csv` and `astropy.io.table.Table.read`.

   URL for notebook from this session: `goo.gl/EIbNDg`

```
In [29]: %%bash
         curl -O https://raw.githubusercontent.com/astropy/specutils/master/specutils/io/tests/files/AA
```

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 87908  100 87908    0     0  99536      0 --:--:-- --:--:-- --:--:-- 99556
```

```
In [45]: !head -n 180 AAO_11.txt
```

```
BITPIX  =                    8  /  8-bit ASCII characters
NAXIS   =                    1  /  Number of Image Dimensions
NAXIS1  =                 2746  /  Length of axis
ORIGIN  = 'NOAO-IRAF: WTEXTIMAGE'  /
IRAF-MAX=                   0.  /  Max image pixel (out of date)
IRAF-MIN=                   0.  /  Min image pixel (out of date)
IRAF-B/P=                   32  /  Image bits per pixel
IRAFTYPE= 'REAL FLOATING     '  /  Image datatype
OBJECT  = 'TW HYA            '  /
FILENAME= 'AAO_11.0011.FITS  '  /  IRAF filename
FORMAT  = '5G14.7            '  /  Text line format
EXTEND  =                    F / File may contain extensions
ORIGIN  = 'NOAO-IRAF FITS Image Kernel July 2003' / FITS file originator
DATE    = '2012-11-26T14:48:43' / Date FITS file was generated
IRAF-TLM= '2012-11-26T14:48:43' / Time of last modification
OBJECT  = 'TW Hya  '           / Name of the object observed
COMMENT   FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT   and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
```

```
DCT_DATE= 'Oct 28 2009'        / DCT release date
DCT_VER = 'r3_110  '           / DCT version number
DETECXE =                 2048 / Last column of detector
DETECXS =                    1 / First column of detector
DETECYE =                 4096 / Last row of detector
DETECYS =                    1 / First row of detector
FIRMVSYS= 'System: AAO2 CCD Controller System Controller V1.23 151104' / System
FIRMVSEQ= 'Sequencer: AAO2 CCD Controller Sequencer V1.3 161203' / Sequencer fir
DETECTOR= 'EEV2    '           / Detector name
XPIXSIZE=                 13.5 / X Pixel size in microns
YPIXSIZE=                 13.5 / Y Pixel size in microns
CONFIGID=                    7 / Controller configuration Id
DETECTID=                    4 / Controller detector Id
ABCKPLID=                    3 / Analog backplane Id
VIDPBID =                    2 / Video personality board Id
CLKPBID =                   21 / Clock personality board Id
BRDID_1 =                65537 / Controller board #1 Id
BRDSN_1 =                   66 / Controller board #1 serial #
BRDID_2 =               327681 / Controller board #2 Id
BRDSN_2 =                   70 / Controller board #2 serial #
BRDID_3 =               131074 / Controller board #3 Id
BRDSN_3 =                   82 / Controller board #3 serial #
BRDID_4 =               196610 / Controller board #4 Id
BRDSN_4 =                   92 / Controller board #4 serial #
BRDID_5 =               262146 / Controller board #5 Id
BRDSN_5 =                   75 / Controller board #5 serial #
BRDID_6 =               458754 / Controller board #6 Id
BRDSN_6 =                  105 / Controller board #6 serial #
BRDID_7 =               393217 / Controller board #7 Id
BRDSN_7 =                   58 / Controller board #7 serial #
METHOD  = 'Normal ccd control method' / Observing method
SPEED   = 'fast    '           / Readout speed
READAMP = '        '           / Readout amplifier
EXPOSED =                 600. / Exposure time (seconds)
ELAPSED =                600.3 / Elapsed time (seconds)
TOTALEXP=                 600. / Total exposure (seconds)
RO_GAIN =                  1.0 / Readout amplifier (inverse) gain (e-/ADU)
RO_NOISE=                 5.35 / Readout noise (electrons)
TELESCOP= 'Anglo-Australian Telescope' / Telescope Name
ALT_OBS =                 1164 / Altitude of observatory in metres
LAT_OBS =            -31.27704 / Observatory latitude in degrees
LONG_OBS=             149.0661 / Observatory longitude in degrees
RCT_VER = 'r3_62L  '           / Run Control Task version number
RCT_DATE= '27-Oct-2009'        / Run Control Task version date
RUNCMD  = 'RUN     '           / Run command
RADECSYS= 'FK5     '           / FK5 reference system
EQUINOX =                2000. / J2000 equinox
INSTRUME= 'UCLES   '           / Instrument in use
GRATID  = '79      '           / The grating ID, Either 31 or 79
LAMBDAC =              5497.31 / Central Wavelength (angstroms)
GORDER  =                   41 / Spectrum Order. Range 20 to 189
BR_STATE= 'INACTIVE'           / Tracking state of the beam de-rotator
TEL_PA  =                  -1. / position angle. -1 not used
RUN     =                   52 / Run number
```

```
OBSNUM  =                   52 / Observation number
GRPNUM  =                   52 / Group Number
GRPMEM  =                    1 / Group member
GRPMAX  =                    0 / Group maximum
OBSTYPE = ’OBJECT  ’           / Observation type
UTDATE  = ’2011:04:18’         / UT date
EPOCH   =      2011.2933533356 / Current Epoch, Years A.D.
UTSTART = ’09:32:07’           / UT start
UTEND   = ’09:42:08’           / UT end
STSTART = ’09:13:01’           / ST start
STEND   = ’09:23:03’           / ST end
UTMJD   =     55669.3973058276 / Modified Julian Date (UTC)
TOPEND  = ’F/36    ’           / Telescope top-end
AXIS    = ’REF     ’           / Current optical axis
AXIS_X  =                   0. / Optical axis x (mm)
AXIS_Y  =                   0. / Optical axis y (mm)
TRACKING= ’TRACKING’           / Telescope is tracking.
MEANRA  =     165.465172255035 / 11 01 51.64
MEANDEC =    -34.7031303545904 / -34 42 11.3
HASTART =    -27.3532949569471 / HA at start of run
ZDSTART =     23.1258066125982 / ZD at start of run
APPRA   =     2.89039527913243 / Current apparent place position right ascension
APPDEC  =   -0.606841574053954 / Current apparent place position declination
WINDOW  = ’eev_planet_bx2.txt’ / Observing window (file name)
CPIXEL_S=                  27. / CPIXEL_SIZE
CSLIT_PR=                 8.63 / CSLIT_PROJ
DPIXEL_S=                 13.5 / DPIXEL_SIZE
DSLIT_PR=                13.25 / DSLIT_PROJ
ECHGAMMO=                   0. / ECH_GAMMA_OFF
ECHTHETO=                   0. / ECH_THETA_OFF
FM_FACTO=                 1.37 / FM_FACTOR
PLATE_SC=                0.712 / PLATE_SCALE
SLITANGO=                   0. / SLIT_ANGLE_OFF
CAMSHUT = ’OPEN    ’           / Camera shutter state
COLLIMAT= ’WIDE    ’           / Collimator selected
ECHELLE = ’31      ’           / Echelle selected
ECHTHETA= ’0.199087’           / Echelle theta in degrees
HARTMANN= ’OUT     ’           / Hartmann state
HARTPOS = ’UP      ’           / Hartmann position
LFILT1  = ’1 (CLEAR)’          / Lamp filter 1 selected
LFILT2  = ’1 (CLEAR)’          / Lamp filter 2 selected
PRISMPOS= ’50.712251’          / Prism position in mm
BEAMDROT= ’OUT     ’           / Beam rotator state
ROTANGLE=           -77.419355 / Rotator angle in degrees
SLITANGL= ’-6.862440’          / Slit angle in degrees
SFILT2  = ’1 (CLEAR)’          / Slit filter 2 selected
TVFIL1  = ’1 (CLEAR)’          / TV filter 1 selected
TVFIL2  = ’        ’           / TV filter 2 selected
UCAMSHUT= ’SHUT    ’           / UHRF Camera shutter state
UHRFCAMR= ’        ’           / UHRF Camera Resolution selected
UHRFECH = ’OUT     ’           / UHRF Echelle selected
UFM     = ’6E5     ’           / UHRF Focal Reducer selected
UHRFHLOW= ’OPEN    ’           / UHRF Hartmann lower position
UHRFHUP = ’SHUT    ’           / UHRF Hartmann upper position
```

```
UHRFXD  = ’MR      ’           / UHRF X Disperser selected
SLITMODE= ’SLIT    ’           / Slit selected
SLITSHUT= ’SHUT    ’           / Slit shutter state
ECHGAMMA= ’-1.174436’          / Echelle gamma in degrees
SLITLONG= ’3.494471’           / Slit length in mm
SOURCE  = ’NONE    ’           / Which lamp is switched on
TVMIRROR= ’Slit Viewing’       / TV mirror position
SFILT1  = ’1 (CLEAR)’          / Slit filter 1 selected
SLITWIDE=             0.974965 / Slit width in mm
FOCALMOD= ’CLEAR   ’           / Focal modifier selected
COLFOCUS= ’-0.981031’          / Collimator focus position in mm
HAEND   =    -24.8446920519603 / HA at end of run
ZDEND   =     21.0661076095138 / ZD at end of run
WINDOXS1=                    1 / First column of window 1
WINDOXE1=                 2048 / Last column of window 1
WINDOYS1=                  676 / First row of window 1
WINDOYE1=                 3421 / Last row of window 1
FIELDXB1=                    2 / Columns/bin in x-binning field 1
FIELDXS1=                    1 / First column of x-binning field 1
FIELDXE1=                 2048 / Last column of x-binning field 1
WINDOXS2=                 2050 / First column of window 2
WINDOXE2=                 2069 / Last column of window 2
WINDOYS2=                  676 / First row of window 2
WINDOYE2=                 3421 / Last row of window 2
FIELDXB2=                    2 / Columns/bin in x-binning field 2
FIELDXS2=                 2050 / First column of x-binning field 2
FIELDXE2=                 2069 / Last column of x-binning field 2
XEFFSIZE=                  27. / Effective X pixel size in microns
YEFFSIZE=                 13.5 / Effective Y pixel size in microns
FILEORIG= ’/data/aatobs/OptDet_data/110418/ccd_2/18apr20052.fits’ / The filename
APSCATTE= ’Scattered light subtracted’
WCSDIM  =                    2
CTYPE1  = ’MULTISPE’
CTYPE2  = ’MULTISPE’
CDELT1  =                   1.
CDELT2  =                   1.
CD1_1   =                   1.
CD2_2   =                   1.
LTM1_1  =                   1.
LTM2_2  =                   1.
WAT0_001= ’system=multispec’
WAT1_001= ’wtype=multispec label=Wavelength units=angstroms’
WAT2_001= ’wtype=multispec spec1 = "11 78 2 7338.5078087124 -0.040949778386667 ’
WAT2_002= ’2746 0. 185.66 197.66 1. 0. 1 5 1. 2746. 7282.91234708959 -56.216484’
WAT2_003= ’3947003 -0.608130816699156 0.0129135590007818 2.16037765569379E-5"’
DCLOG1  = ’REFSPEC1 = a002t’
BANDID1 = ’spectrum - background none, weights none, clean no’
WAXMAP01= ’1 0 0 0 ’
END
```

```
7338.50780871237  703.5236
7338.46870555523  721.9175
```

How do we exclude the first N lines?

```
In [44]: with open('AAO_11.txt','r') as fh:
             for ii,line in enumerate(fh):
                 if 'END' in line:
                     last_header_line_number = ii

         aao_data = np.genfromtxt('AAO_11.txt', skip_header=last_header_line_number+1)
         print(last_header_line_number)
```

```
174
```

```
In [48]: float('nan') # works
         float('1.234') # works
         float('blah') # fails
```

```
        ---------------------------------------------------------------------------

        ValueError                                Traceback (most recent call last)

        <ipython-input-48-d8f49fd65559> in <module>()
          1 float('nan') # works
          2 float('1.234') # works
    ----> 3 float('blah') # fails


        ValueError: could not convert string to float: 'blah'
```

```
In [46]: with open('AAO_11.txt','r') as fh:
             for ii,line in enumerate(fh):
                 try:
                     float(line.split()[0])
                     break
                 except (ValueError, IndexError) as ex:
                     print(ex)
                     continue
         last_header_line_number = ii
         print(last_header_line_number)
```

```
could not convert string to float: 'BITPIX'
could not convert string to float: 'NAXIS'
could not convert string to float: 'NAXIS1'
could not convert string to float: 'ORIGIN'
could not convert string to float: 'IRAF-MAX='
could not convert string to float: 'IRAF-MIN='
could not convert string to float: 'IRAF-B/P='
could not convert string to float: 'IRAFTYPE='
could not convert string to float: 'OBJECT'
could not convert string to float: 'FILENAME='
could not convert string to float: 'FORMAT'
could not convert string to float: 'EXTEND'
could not convert string to float: 'ORIGIN'
could not convert string to float: 'DATE'
could not convert string to float: ’IRAF-TLM=’
could not convert string to float: ’OBJECT’
could not convert string to float: ’COMMENT’
could not convert string to float: ’COMMENT’
could not convert string to float: ’DCT_DATE=’
could not convert string to float: ’DCT_VER’
could not convert string to float: ’DETECXE’
could not convert string to float: ’DETECXS’
could not convert string to float: ’DETECYE’
could not convert string to float: ’DETECYS’
could not convert string to float: ’FIRMVSYS=’
could not convert string to float: ’FIRMVSEQ=’
could not convert string to float: ’DETECTOR=’
could not convert string to float: ’XPIXSIZE=’
could not convert string to float: ’YPIXSIZE=’
could not convert string to float: ’CONFIGID=’
could not convert string to float: ’DETECTID=’
could not convert string to float: ’ABCKPLID=’
could not convert string to float: ’VIDPBID’
could not convert string to float: ’CLKPBID’
could not convert string to float: ’BRDID_1’
could not convert string to float: ’BRDSN_1’
could not convert string to float: ’BRDID_2’
could not convert string to float: ’BRDSN_2’
could not convert string to float: ’BRDID_3’
could not convert string to float: ’BRDSN_3’
could not convert string to float: ’BRDID_4’
could not convert string to float: ’BRDSN_4’
could not convert string to float: ’BRDID_5’
could not convert string to float: ’BRDSN_5’
could not convert string to float: ’BRDID_6’
could not convert string to float: ’BRDSN_6’
could not convert string to float: ’BRDID_7’
could not convert string to float: ’BRDSN_7’
could not convert string to float: ’METHOD’
could not convert string to float: ’SPEED’
could not convert string to float: ’READAMP’
could not convert string to float: ’EXPOSED’
could not convert string to float: ’ELAPSED’
could not convert string to float: ’TOTALEXP=’
could not convert string to float: ’RO_GAIN’
could not convert string to float: ’RO_NOISE=’
could not convert string to float: ’TELESCOP=’
could not convert string to float: ’ALT_OBS’
could not convert string to float: ’LAT_OBS’
could not convert string to float: ’LONG_OBS=’
could not convert string to float: ’RCT_VER’
could not convert string to float: ’RCT_DATE=’
could not convert string to float: ’RUNCMD’
could not convert string to float: ’RADECSYS=’
could not convert string to float: ’EQUINOX’
could not convert string to float: ’INSTRUME=’
could not convert string to float: ’GRATID’
could not convert string to float: ’LAMBDAC’
could not convert string to float: ’GORDER’
could not convert string to float: ’BR_STATE=’
could not convert string to float: ’TEL_PA’
could not convert string to float: ’RUN’
could not convert string to float: ’OBSNUM’
could not convert string to float: ’GRPNUM’
could not convert string to float: ’GRPMEM’
could not convert string to float: ’GRPMAX’
could not convert string to float: ’OBSTYPE’
could not convert string to float: ’UTDATE’
could not convert string to float: ’EPOCH’
could not convert string to float: ’UTSTART’
could not convert string to float: ’UTEND’
could not convert string to float: ’STSTART’
could not convert string to float: ’STEND’
could not convert string to float: ’UTMJD’
could not convert string to float: ’TOPEND’
could not convert string to float: ’AXIS’
could not convert string to float: ’AXIS_X’
could not convert string to float: ’AXIS_Y’
could not convert string to float: ’TRACKING=’
could not convert string to float: ’MEANRA’
could not convert string to float: ’MEANDEC’
could not convert string to float: ’HASTART’
could not convert string to float: ’ZDSTART’
could not convert string to float: ’APPRA’
could not convert string to float: ’APPDEC’
could not convert string to float: ’WINDOW’
could not convert string to float: ’CPIXEL_S=’
could not convert string to float: ’CSLIT_PR=’
could not convert string to float: ’DPIXEL_S=’
could not convert string to float: ’DSLIT_PR=’
could not convert string to float: ’ECHGAMMO=’
could not convert string to float: ’ECHTHETO=’
could not convert string to float: ’FM_FACTO=’
could not convert string to float: ’PLATE_SC=’
could not convert string to float: ’SLITANGO=’
could not convert string to float: ’CAMSHUT’
could not convert string to float: ’COLLIMAT=’
could not convert string to float: ’ECHELLE’
could not convert string to float: ’ECHTHETA=’
could not convert string to float: ’HARTMANN=’
could not convert string to float: ’HARTPOS’
could not convert string to float: ’LFILT1’
could not convert string to float: ’LFILT2’
could not convert string to float: ’PRISMPOS=’
could not convert string to float: ’BEAMDROT=’
could not convert string to float: ’ROTANGLE=’
could not convert string to float: ’SLITANGL=’
could not convert string to float: ’SFILT2’
could not convert string to float: ’TVFIL1’
could not convert string to float: ’TVFIL2’
could not convert string to float: ’UCAMSHUT=’
could not convert string to float: ’UHRFCAMR=’
could not convert string to float: ’UHRFECH’
could not convert string to float: ’UFM’
could not convert string to float: ’UHRFHLOW=’
could not convert string to float: ’UHRFHUP’
could not convert string to float: ’UHRFXD’
could not convert string to float: ’SLITMODE=’
could not convert string to float: ’SLITSHUT=’
could not convert string to float: ’ECHGAMMA=’
could not convert string to float: ’SLITLONG=’
could not convert string to float: ’SOURCE’
could not convert string to float: ’TVMIRROR=’
could not convert string to float: ’SFILT1’
could not convert string to float: ’SLITWIDE=’
could not convert string to float: ’FOCALMOD=’
could not convert string to float: ’COLFOCUS=’
could not convert string to float: ’HAEND’
could not convert string to float: ’ZDEND’
could not convert string to float: ’WINDOXS1=’
could not convert string to float: ’WINDOXE1=’
could not convert string to float: ’WINDOYS1=’
could not convert string to float: ’WINDOYE1=’
could not convert string to float: ’FIELDXB1=’
could not convert string to float: ’FIELDXS1=’
could not convert string to float: ’FIELDXE1=’
could not convert string to float: ’WINDOXS2=’
could not convert string to float: ’WINDOXE2=’
could not convert string to float: ’WINDOYS2=’
could not convert string to float: ’WINDOYE2=’
could not convert string to float: ’FIELDXB2=’
could not convert string to float: ’FIELDXS2=’
could not convert string to float: ’FIELDXE2=’
could not convert string to float: ’XEFFSIZE=’
could not convert string to float: ’YEFFSIZE=’
could not convert string to float: ’FILEORIG=’
could not convert string to float: ’APSCATTE=’
could not convert string to float: ’WCSDIM’
could not convert string to float: ’CTYPE1’
could not convert string to float: ’CTYPE2’
could not convert string to float: ’CDELT1’
could not convert string to float: ’CDELT2’
could not convert string to float: ’CD1_1’
could not convert string to float: ’CD2_2’
could not convert string to float: ’LTM1_1’
could not convert string to float: ’LTM2_2’
could not convert string to float: ’WAT0_001=’
could not convert string to float: ’WAT1_001=’
could not convert string to float: ’WAT2_001=’
could not convert string to float: ’WAT2_002=’
could not convert string to float: ’WAT2_003=’
could not convert string to float: ’DCLOG1’
could not convert string to float: ’BANDID1’
could not convert string to float: ’WAXMAP01=’
could not convert string to float: ’END’
list index out of range
list index out of range
list index out of range
178
```

```
In [37]: ll = [’a’,’b’,’c’]
         for ii,x in enumerate(ll):
             print(ii, x)
```

```
0 a
1 b
2 c
```

```
In [35]: aao_data
```

```
Out[35]: array([[ 7338.50780871,   703.5236    ],
                [ 7338.46870556,   721.9175    ],
                [ 7338.42960094,   757.0006    ],
                ...,
                [ 7226.18595796,   914.5856    ],
                [ 7226.14331306,  1079.059     ],
                [ 7226.10066704,   928.7758    ]])
```

```
In [53]: # inf vs nan
         np.inf, -np.inf, np.nan
         (np.isfinite([np.inf, -np.inf, np.nan, 0]),
          np.isnan([np.inf, -np.inf, np.nan, 0]),
          np.nan == np.nan)
```

```
Out[53]: (array([False, False, False,  True], dtype=bool),
          array([False, False,  True, False], dtype=bool),
          False)
```

```
In [55]: None, bool(None), None == np.nan
```

```
Out[55]: (None, False, False)
```

```
In [59]: def f():
             pass
         f() is None # preferred
         f() == None # can result in problems in some situations
```

```
Out[59]: True
```

3 Reading FITS files

```
In [40]: %%bash
         curl -O https://raw.githubusercontent.com/astropy/specutils/master/specutils/io/tests/files/gb
```

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 40320  100 40320    0     0  51021      0 --:--:-- --:--:-- --:--:-- 50973
```

FITs files can be read. . . .

```
In [41]: from astropy.io import fits
         from astropy.wcs import WCS
```

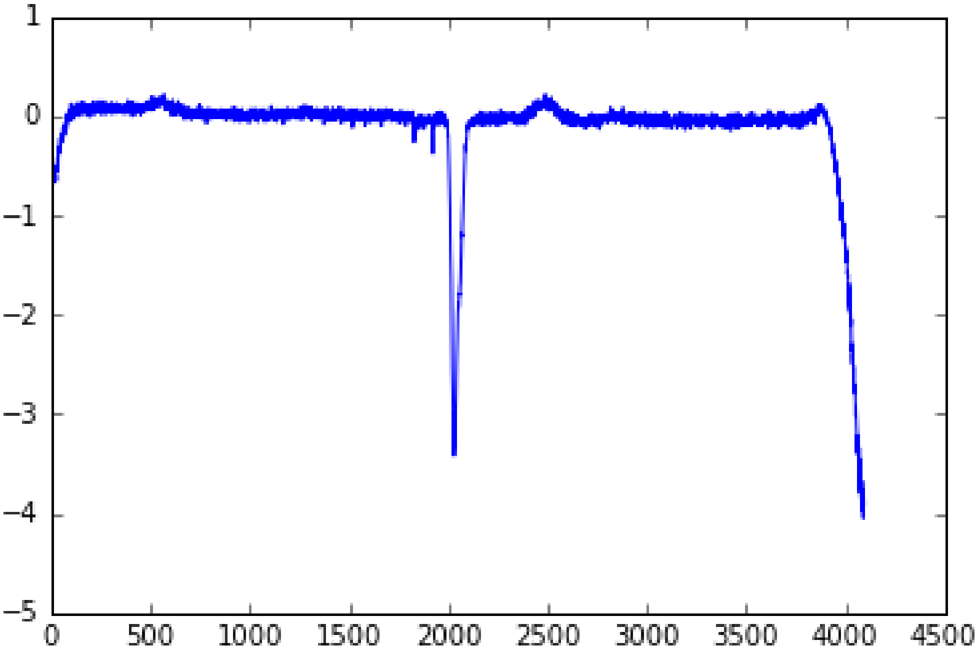
```
In [42]: fh = fits.open('gbt_1d.fits')
         fh

Out[42]: [<astropy.io.fits.hdu.image.PrimaryHDU at 0x11263edd8>]

In [43]: data_hdu = fh[0]

In [44]: data_hdu.header

WARNING: VerifyWarning: Verification reported errors: [astropy.io.fits.verify]
WARNING:astropy:VerifyWarning: Verification reported errors:
WARNING: VerifyWarning: Card 'RESTFRQ' is not FITS standard (invalid value string: '1.4488479e+10').  Fi
WARNING:astropy:VerifyWarning: Card 'RESTFRQ' is not FITS standard (invalid value string: '1.4488479e+1
WARNING: VerifyWarning: Note: PyFITS uses zero-based indexing.
 [astropy.io.fits.verify]
WARNING:astropy:VerifyWarning: Note: PyFITS uses zero-based indexing.

WARNING: VerifyWarning: Card 'CRVAL1F' is not FITS standard (invalid value string: '1.4488303e+10').  Fi
WARNING:astropy:VerifyWarning: Card 'CRVAL1F' is not FITS standard (invalid value string: '1.4488303e+1
WARNING: VerifyWarning: Card 'RESTFRQF' is not FITS standard (invalid value string: '1.4488479e+10').  F
WARNING:astropy:VerifyWarning: Card 'RESTFRQF' is not FITS standard (invalid value string: '1.4488479e+
WARNING: VerifyWarning: Card 'RESTFRQV' is not FITS standard (invalid value string: '1.4488479e+10').  F
WARNING:astropy:VerifyWarning: Card 'RESTFRQV' is not FITS standard (invalid value string: '1.4488479e+
WARNING: VerifyWarning: Card 'RESTFRQT' is not FITS standard (invalid value string: '1.4488479e+10').  F
WARNING:astropy:VerifyWarning: Card 'RESTFRQT' is not FITS standard (invalid value string: '1.4488479e+
WARNING: VerifyWarning: Card 'ZSOURCE' is not FITS standard (invalid value string: '2.6151426e-05').  Fi
WARNING:astropy:VerifyWarning: Card 'ZSOURCE' is not FITS standard (invalid value string: '2.6151426e-0

Out[44]: SIMPLE  =                    T / Written by IDL:  Fri Aug 19 18:36:44 2011
         BITPIX  = -64
         NAXIS   =                    1 / number of array dimensions
         NAXIS1  =                 4096
         CDELT1  =   -0.25258831
         CRPIX1  =   2049.0000
         CRVAL1  =   7.5845751
         CTYPE1  =  'VRAD'
         CUNIT1  =  'km/s    '
         SPECSYS =  'LSRK'
         RESTFRQ =        1.4488479E+10
         VELOSYS =   -3940.7291
         CDELT1F =   12207.031
         CRPIX1F =   2049.0000
         CRVAL1F =        1.4488303E+10
         CTYPE1F =  'FREQ'
         CUNIT1F =  'Hz'
         SPECSYSF=  'LSRK'
         RESTFRQF=        1.4488479E+10
         VELOSYSF=   -3940.7291
         CDELT1V =   -0.25258524
         CRPIX1V =   2063.4260
         CRVAL1V =   3.9407291
         CTYPE1V =  'VRAD'
         CUNIT1V =  'km/s'
         SPECSYSV=  'LSRK'
         RESTFRQV=        1.4488479E+10
         VELOSYSV=   -3940.7291
         CDELT1T =   -0.25258524
         CRPIX1T =   2063.4260
         CRVAL1T =   0
         CTYPE1T =  'VRAD'
         CUNIT1T =  'km/s'
         SPECSYST=  'TOPO'
         RESTFRQT=        1.4488479E+10
         VELOSYST=   -3940.7291
         VDEF    =  'RADI-LSR'
         SRCVEL  =   7.8400000
         ZSOURCE =        2.6151426E-05
         BUNIT   =  'K'
         OBJECT  =  'G43.17+0.01'
         TELESCOP=  'GBT'
         TSYS    =   28.0243
         TSYSFD0 =   27.6077
         TSYSFD1 =   28.4602
         TATM    =   275 / assumed
         TREC    =   20.1375 / assumed
         ELEV    =   58.294203
         AIRMASS =   1.1754225
         TAU     =   0.0236965
         TAUCOR  =   1.0282450
         LINE    =   'h2co'
         FREQ    =   14.488303
         TARGLON =   287.55837
         TARGLAT =   9.1044256
         MJD-AVG =   55425.080
         CONTINUU=   27.6056
         CONTERR =   0.0598740
         CONTFD0 =   26.151077
         CONTFD1 =   29.155884
         SMTHOFF =   16
         ETAMB   =  0.88607179
```

```
In [45]: data_hdu.data.shape
```

```
Out[45]: (4096,)
```

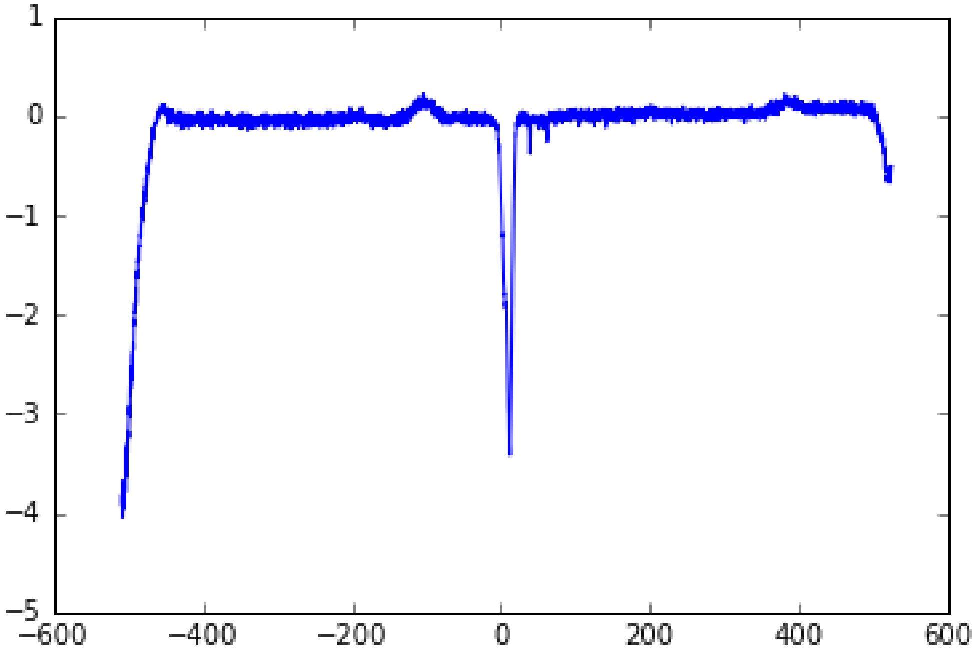
```
In [46]: pl.plot(data_hdu.data)
```

```
Out[46]: [<matplotlib.lines.Line2D at 0x10bc24358>]
```



Generate the x axis:

```
In [47]: hdr = data_hdu.header

In [48]: xarr = (np.arange(hdr['NAXIS1']) - hdr['CRPIX1'] + 1) * hdr['CDELT1'] + hdr['CRVAL1']

In [49]: pl.plot(xarr, data_hdu.data)

Out[49]: [<matplotlib.lines.Line2D at 0x110ac6630>]
```



```
In [50]: from astropy import units as u
         xarr_u = xarr*u.Unit(hdr['CUNIT1'])
```

```
In [51]: xarr_u
```

```
Out[51]:
```

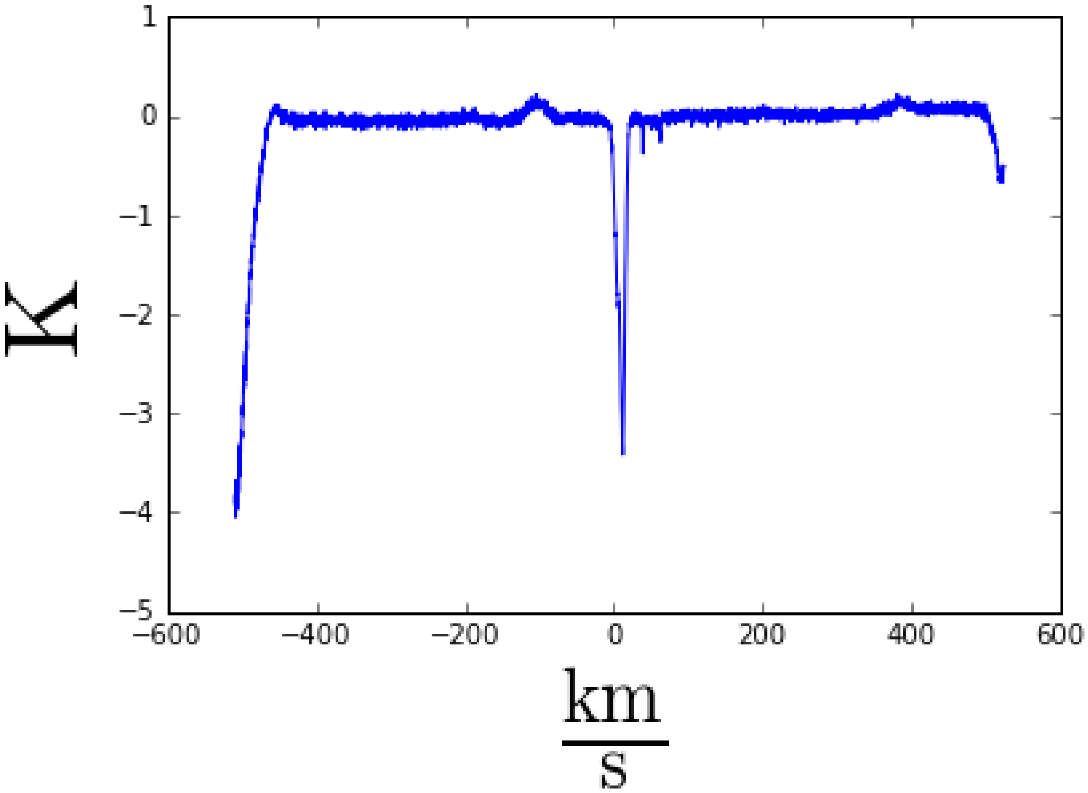
$$[524.88543, \; 524.63285, \; 524.38026, \; \ldots, -508.95852, \; -509.21111, \; -509.4637] \; \mathrm{\frac{km}{s}}$$

```
In [52]: xarr_u.unit.to_string(format='latex')
```

```
Out[52]: '$\\mathrm{\\frac{km}{s}}$'
```

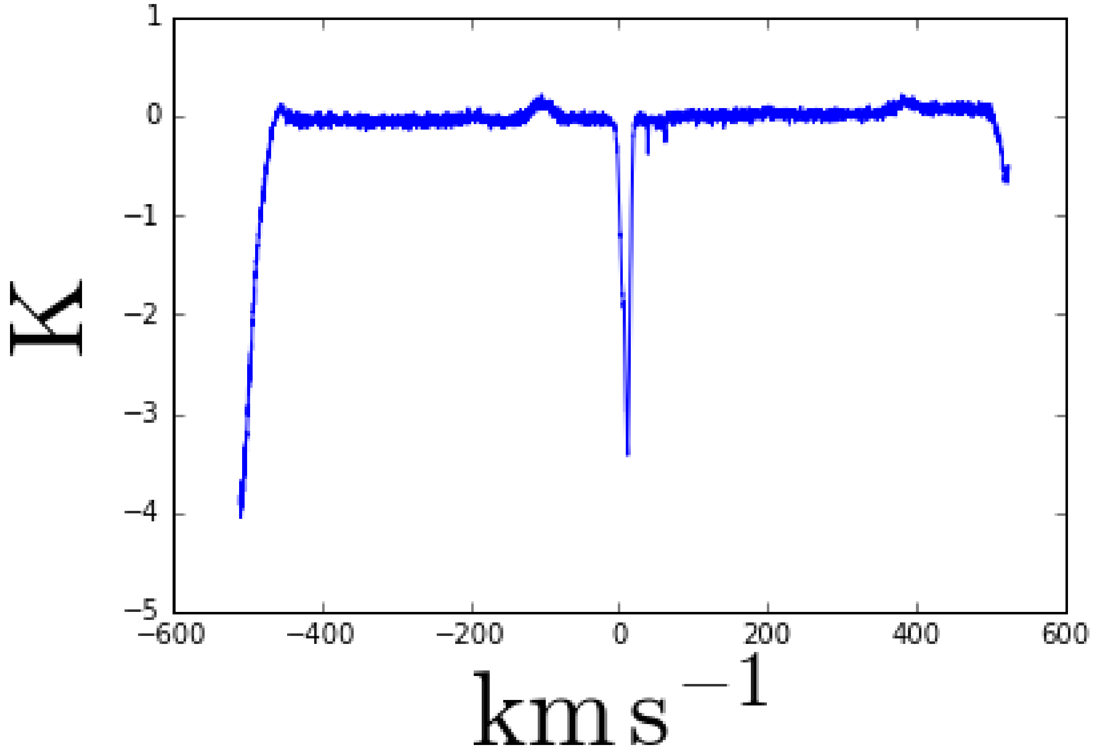
```
In [53]: pl.plot(xarr_u, data_hdu.data)
         pl.xlabel(xarr_u.unit.to_string(format='latex'), fontsize=40)
         pl.ylabel(u.Unit(hdr['BUNIT']).to_string(format='latex'), fontsize=40)
```

```
Out[53]: <matplotlib.text.Text at 0x1107c7470>
```



```
In [54]: pl.plot(xarr_u, data_hdu.data)
         pl.xlabel(xarr_u.unit.to_string(format='latex_inline'), fontsize=40)
         pl.ylabel(u.Unit(hdr['BUNIT']).to_string(format='latex'), fontsize=40)
```

```
Out[54]: <matplotlib.text.Text at 0x11401bd30>
```



```
In [55]: mywcs = WCS(hdr)
         mywcs

Out[55]: WCS Keywords

         Number of WCS axes: 1
         CTYPE : 'VRAD'
         CRVAL : 7584.5751
         CRPIX : 2049.0
         PC1_1  : 1.0
         CDELT : -252.58830999999998
         NAXIS    : 4096 0
```

```
In [56]: xarr_again = mywcs.wcs_pix2world(np.arange(hdr['NAXIS1']), 0) * u.Unit(mywcs.wcs.cunit[0])
```

```
In [57]: xarr_u

Out[57]:
```

$$[524.88543,~524.63285,~524.38026,~\ldots,-508.95852,~-509.21111,~-509.4637]\;\mathrm{\frac{km}{s}}$$

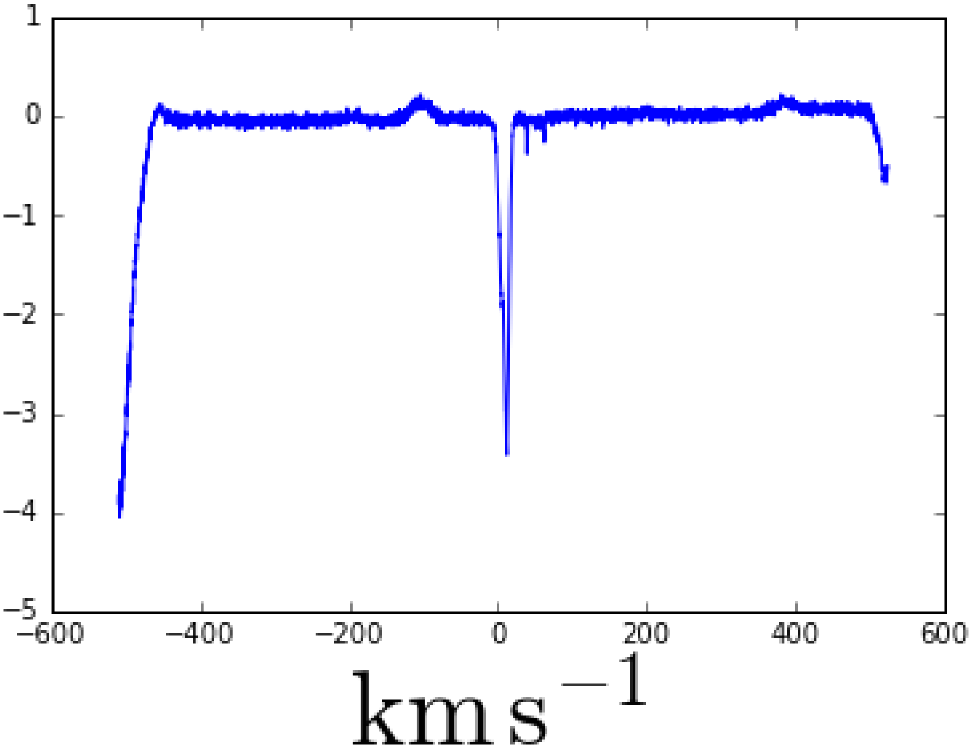
```
In [58]: xarr_again

Out[58]:
```

$$[[524885.43,~524632.85,~524380.26,~\ldots,-508958.52,~-509211.11,~-509463.7]]\;\mathrm{\frac{m}{s}}$$

```
In [59]: np.isclose(xarr_again, xarr_u, atol=0)

Out[59]: array([[ True,  True,  True, ...,  True,  True,  True]], dtype=bool)

In [60]: from specutils.io import fits

WARNING: AstropyDeprecationWarning: astropy.utils.compat.odict.OrderedDict is now deprecated - import O
WARNING:astropy:AstropyDeprecationWarning: astropy.utils.compat.odict.OrderedDict is now deprecated - i

In [61]: spec = fits.read_fits_spectrum1d('gbt_1d.fits')

In [62]: import specutils
         import numpy
         import astropy
         specutils.__version__, astropy.__version__, numpy.__version__

Out[62]: ('0.2.dev596', '1.2.dev14793', '1.10.4')

In [63]: spec.velocity

Out[63]:
```

$$[524.88543,\ 524.63285,\ 524.38026,\ \ldots, -508.95852,\ -509.21111,\ -509.4637]\ \mathrm{\frac{km}{s}}$$

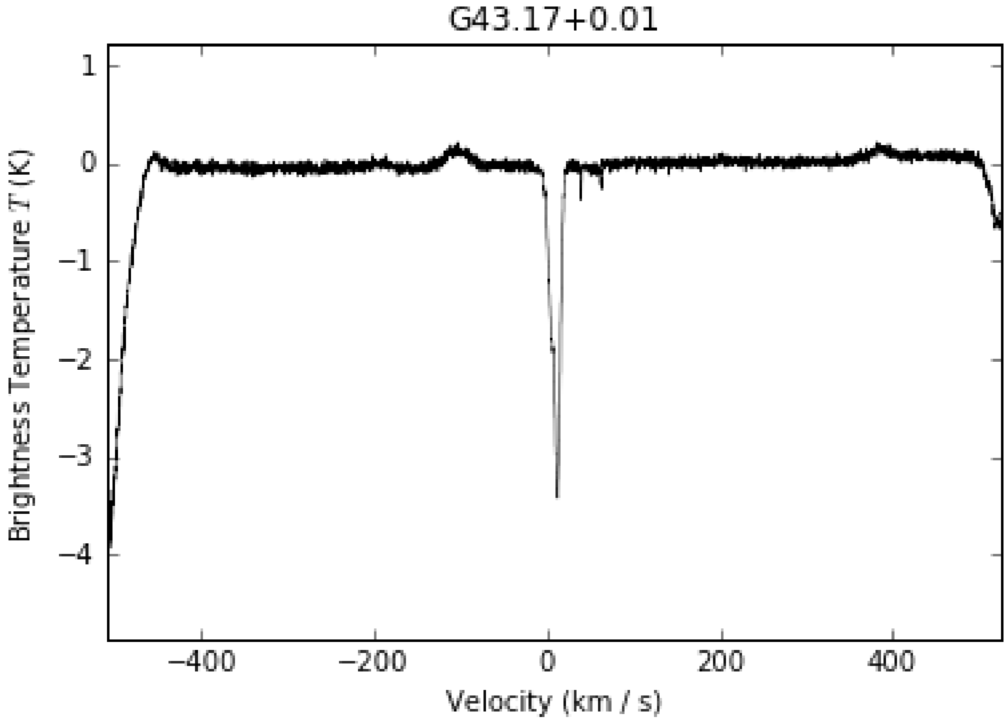
```
In [64]: pl.plot(spec.velocity, spec.flux)
         pl.xlabel(spec.velocity.unit.to_string(format='latex_inline'), fontsize=40)
         pl.ylabel(spec.flux.unit.to_string(format='latex'), fontsize=40)

Out[64]: <matplotlib.text.Text at 0x11462ada0>
```



```
In [65]: # you’ll need to ‘‘pip install pyspeckit‘‘ to get access to this
         import pyspeckit
```

```
In [66]: sp = pyspeckit.Spectrum('gbt_1d.fits')
```

```
In [67]: sp.plotter()
```



4 Interpolation

Functions for interpolation:

- `np.interp1d`
- `scipy.interpolate`
- `pyspeckit.interpolation`

5 Convolution

- `np.convolve`
- `scipy.ndimage.convolve`
- `astropy.convolution`
- `pyspeckit.smooth`

6 Exercises

1. Load the `gbt_1d.fits` spectrum and plot it
2. Interpolate the spectrum onto a new finer grid from -50 to 50 km/s with 1000 channels
3. Smooth the spectrum by 8 km/s, then interpolate it onto a coarser grid from -400 to 400 km/s with 200 channels

```
In [1]: import specutils
        from specutils.io import fits
```

```
In [60]: spec = fits.read_fits_spectrum1d('gbt_1d.fits')
```

```
In [61]: spec.flux
```

```
Out[61]:
```

$$[-0.50829212, \; -0.49870891, \; -0.52269076, \; \ldots, -3.8145078, \; -3.8554833, \; -3.9074813]$$

```
In [62]: spec.velocity
```

```
Out[62]:
```

$$[524.88543, \; 524.63285, \; 524.38026, \; \ldots, -508.95852, \; -509.21111, \; -509.4637] \; \mathrm{\frac{km}{s}}$$

```
In [6]: from astropy import units as u
```

```
In [63]: new_xarr = np.linspace(-50, 50, 1000)*u.km/u.s
```

```
In [66]: new_xarr
```

```
Out[66]:
```

$$[-50, \; -49.8999, \; -49.7998, \; \ldots, 49.7998, \; 49.8999, \; 50] \; \mathrm{\frac{km}{s}}$$

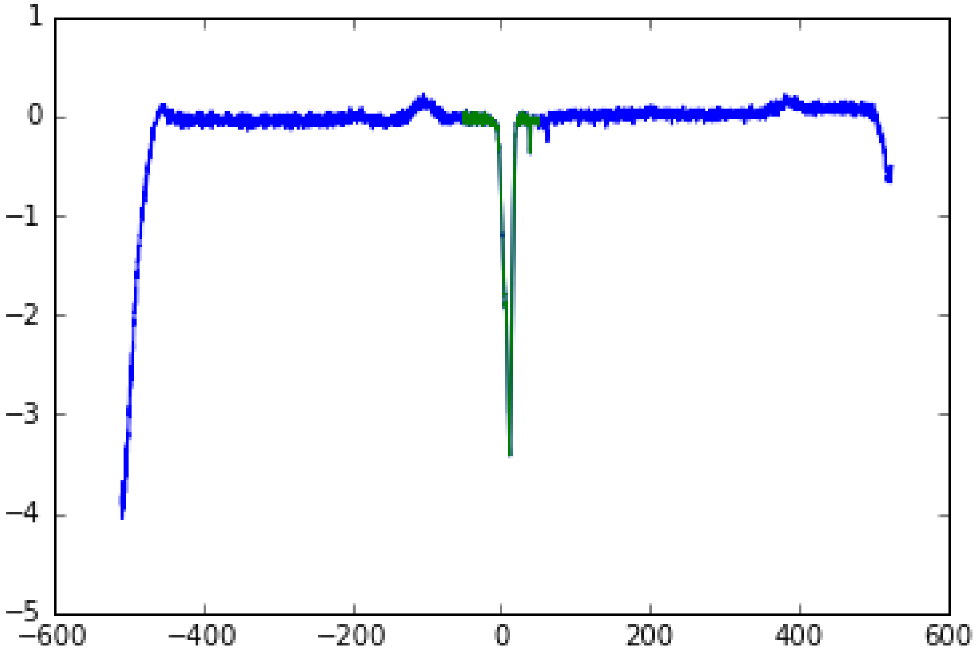
```
In [68]: np.interp?
```

```
In [23]: interpolated_data = np.interp(new_xarr[::-1], spec.velocity[::-1], spec.flux[::-1])
```

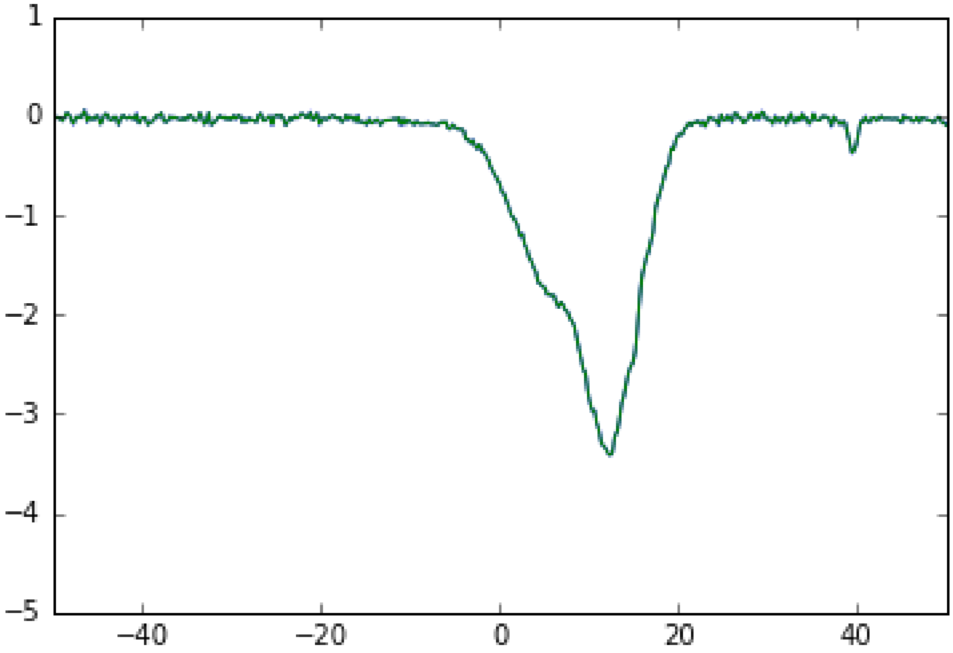
```
In [24]: %matplotlib inline
         import pylab as pl
```

```
In [27]: pl.plot(spec.velocity, spec.flux)
         pl.plot(new_xarr[::-1], interpolated_data)
```

```
Out[27]: [<matplotlib.lines.Line2D at 0x10e7fa710>]
```



```
In [28]: pl.plot(spec.velocity, spec.flux)
         pl.plot(new_xarr[::-1], interpolated_data)
         pl.xlim(-50, 50)

Out[28]: (-50, 50)
```



```
In [ ]:
```