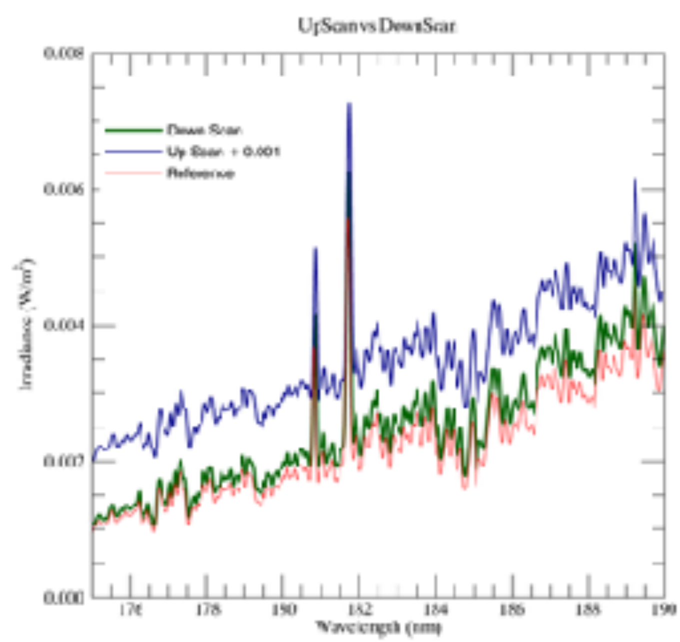
Please calculate the irradiance in watts/m^2 and compare the scan data. Provide plots of your results along with your code. Specifically, plot the region around the two emission lines at ~180nm. Also, calculate the ratio of each scan wrt the reference spectrum and plot the results.

All code is in IDL and is provided via this git repository: https://github.com/spenton/SORCE_penton.git . The plots presented here were created with "make_plots.pro", with the exception of the QuickScan analysis that appears later.

Read the raw telemetry: The routine get_telem.pro is an IDL routine to read the provided telemetry ascii file for each of the 5 orbits. It calls a base routine (get_instrumentTelemetry.pro) then uses the input start and end times to parse the full telemetry structure. An IDL structure with all telemetry items of interest is returned. As described in the instructions, the routine gp_to_wave.pro converts from grating position to wavelength.

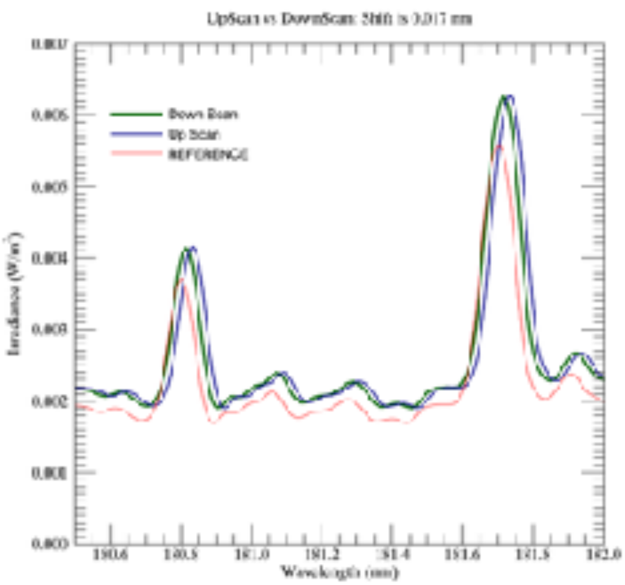
Determine Irradiance[1]:

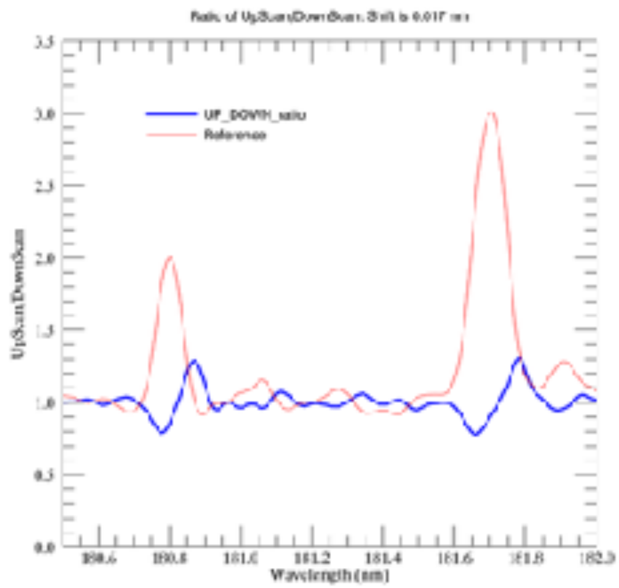
Watt/m^2 is obtained by using the energy (in J) per photon (get_energyperphoton.pro) and the number of photons per cm^2 per second per nm (get_ph_per_s_cm2_nm.pro), assuming a wavelength independent area of 0.01 cm^2. These routines determine the bin size (dispersion) from the telemetry to convert between irradiance and spectral irradiance (irradiance/nm). To the right is a comparison of the Up vs Down Scan Data (the Down Scan Data has been shifted up).

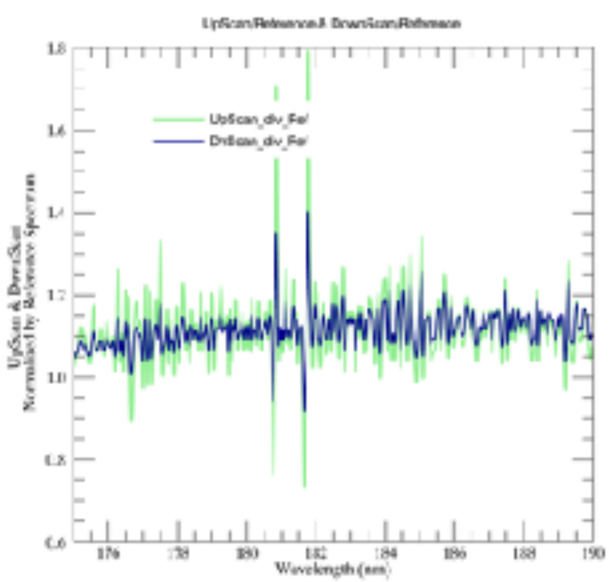


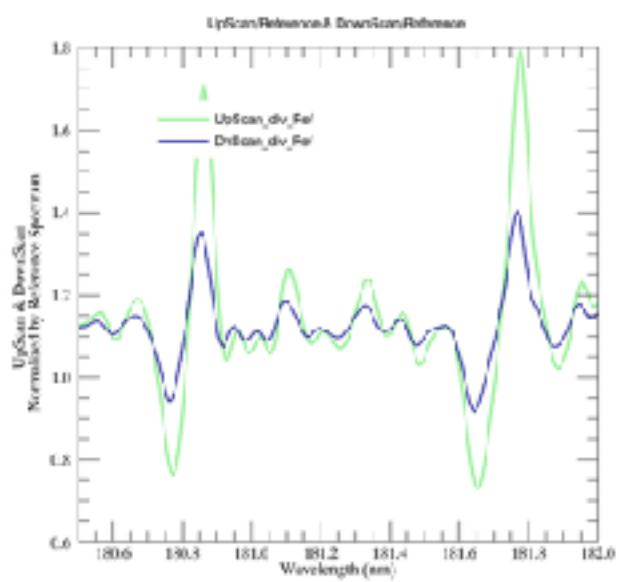
[1] The units of the reference spectrum (irradiance; watts/m^2) seem odd to be. I am accustomed to units of flux (ergs/s/cm^2/Å), so I was expecting spectral irradiance (watts/m^2/nm). I've matched my units to those of the reference spectrum.

Zooming in on the SiII lines (and losing the offset), a clear shift of about 0.17 nm is obvious. A ratio of the Up/Down Scan data clearly shows the offset. Both appear to be shifted wrt the reference spectrum, 0.014 nm (Up) and 0.031 nm (Down)









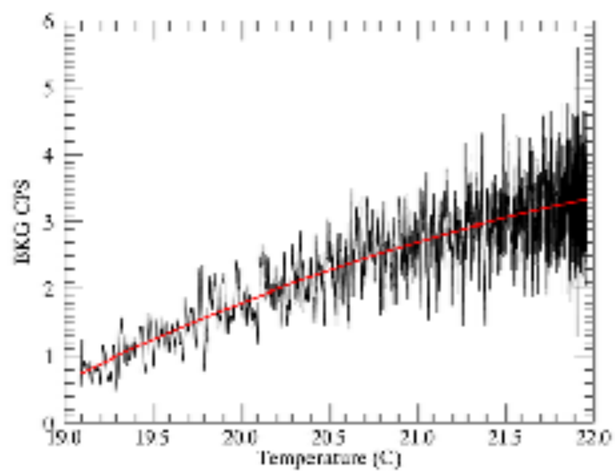
<u>Optional corrections :</u> It was unclear whether debugging this offset was part of the assignment, but I thought I would try a few basic calibrations anyway.

<u>Return temperatures as part of the telemetry.</u> The routines get_detectorTemp.pro and get_detTemp_bytime.pro interpolate the provided temperature data to include temperatures in the returned structure.

<u>Add doppler correction.</u> The routine DOPPCORR.pro provides a spacecraft doppler correction to the wavelengths. The routine get_distdopp_bytime.pro interpolates into the provided doppler data to return the appropriate doppler (and distance) factors. The doppler factor provided is assumed to be (1.0+v/c). The assumption is backed up by the scale of the corrections (~4 km/s) and the maximum expected for the orbit indicated (~8 km/s max velocity wrt the sun).

<u>Add distance correction.</u> The routine DISTCORR.pro applies a distance correction to the counts arrays. This routine also calls get_distdopp_bytime.pro. It assumes that the provided distance factor is (R/R0), and the correction is applied as 1./distancefactor^2.

<u>Add temperature-dependent background correction.</u> The routines darkcorr.pro and get_bkgcounts_byinttime.pro are called to interpolate the provided temperature data to and apply a temperature dependent background correction. Note that a simple parabola fit (using POLY_FIT) was constructed to model the background correction. This analysis can be found in eval_darkcorr.pro. The adopted solution was bkg_cps = -69.795 +6.119*temp -0.127*temp^2, as shown in the figure.



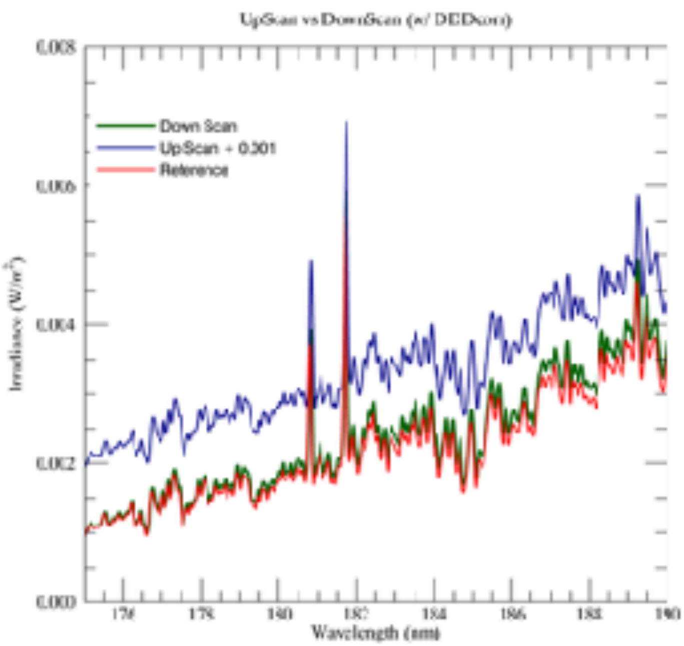
An example of the returned final structure of get_telem() is:

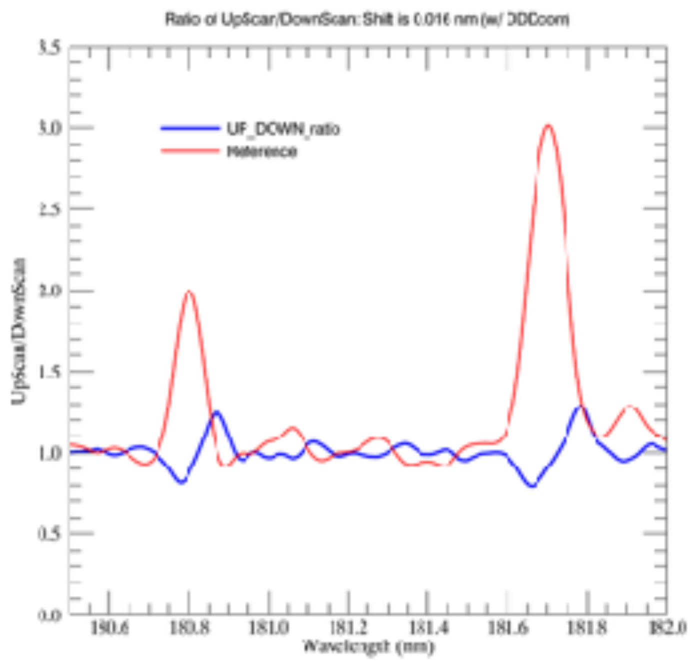
```
; EXAMPLE:
;            DownScan=[9.434134508500002E14, 9.434178736700002E14]
;            DS_telem=get_telem(DownScan,/add_darkcorr,/add_distcor,/add_doppcorr)
;            help,DS_telem,/str
;
;   START_TIME      FLOAT       9.43413e+14
;   END_TIME        FLOAT       9.43418e+14
;   TIMES_MS        DOUBLE    Array[2354]
;   TIMES_S         DOUBLE    Array[2354]
;   TIME_MS0        DOUBLE       9.4341346e+14
;   TIME_S0         DOUBLE       9.4341346e+11
;   TEMPS           DOUBLE    Array[2354]
;   GGRATPOS        FLOAT     Array[2354]
;   GWAVES          DOUBLE    Array[2354]
;   GCOUNTS         FLOAT     Array[2354]
;   IRR             DOUBLE    Array[2354]
;   WAVES           DOUBLE    Array[2354]
;   COUNTS          DOUBLE    Array[2354]
;   DELTA_WAVES     DOUBLE    Array[2354]
;   BINSIZE         DOUBLE        0.0063524922
;   DOPPCORR        INT                1
;   DWAVES          DOUBLE    Array[2354]
;   DW              DOUBLE    Array[2354]
;   DISTCORR        INT                1
;   DCOUNTS         DOUBLE    Array[2354]
;   DR              DOUBLE    Array[2354]
;   DARKCORR        INT                1
;   BKGCOUNTS       DOUBLE    Array[2354]
;   BKG             DOUBLE    Array[2354]
;   INTTIME_S       FLOAT     Array[2354]
;   EDGE_TRIM       INT                2
```

Repeating the above plots with DOPPCORR, DISTCORR, and DARKCORR enabled produces very similar results (although the offsets have all been reduced by 0.001 nm, and the flux calibration looks better).

<u>What are your thoughts?</u>

There is a 0.016nm zero point offset in the wavelength solution at the SiII lines (185nm). Since Doppler, distance, and dark corrections did not move the shift between the up and down scans, a simple zero point shift doesn’t solve the entire problem. My initial suspicion was that there is a thermal drift to the grating mechanism that has not been accounted for. The QuickScan data could be used to prove this (see eval_quickscan.pro). However, unless I missed something, even though the temperature of the QuickScan orbit was identical to the other orbits, I don't see any drift of the required magnitude of the wavelength solution wrt time (but I did see some residual doppler corrections that I must not be catching).





My next guess would be some hysteresis due to the change is scan direction, but I have not had a chance to examine this hypothesis.