

MAGNETIC RECONNECTION IN THE SUN'S CORONA

MSU Physics Department Solar Physics/Astronomy & Space Physics REU
Program

Jada Walters

School of Physics, Georgia Institute of Technology, Atlanta, GA, 30332, USA

August 9, 2018

Overview

This summer, I worked on an extension to the project described in the paper “Observations of Separator Reconnection to an Emerging Active Region” (Longcope et al. 2005) supervised by Marika McCarthy and Dr. Dana Longcope. The 2005 paper used TRACE data in the 171Å wavelength which did not have a full Sun field of view. Only one case of an active region emerging next to an existing active region was found in TRACE data due to the limitations of TRACE. With more recent SDO/AIA data, I used the 211Å wavelength to examine coronal loops connecting an existing active region to an emerging active region. During the ten week REU program, I was able to process data for 2 of 16 identified cases between 2011 and 2014 of an active region emerging alongside an existing active region. The two cases I focused on were Jan. 20th, 2011 and Feb. 17th, 2011. However, the codes I created can be used for the remaining cases once the data is downloaded. In this document, I will describe the programs I created, additional notes that may be useful for continuing this project, and examples of the plots I created using my programs.

Programs

All files/paths described in this document are found on the MSU Physics Department Helios server.

loop_pipeline_elaborate.pro

Location: /home/jwalters/loop_pipeline_elaborate.pro

Name: loop_pipeline

Purpose: Create and save plots of loop emergence, cumulative area, cumulative flux, and voltage (smooth and unsmoothed). Create and save histograms of loop diameter and duration.

Explanation: In order to use this program, you must have a file path to a save file containing stack and loops structures for the wavelength(s) you wish to plot. If you wish to plot multiple sets of data for one event, you must use the OVERPLOT keyword. The program will then ask you to type the wavelengths of the data so that legends can be prepared for each graph. You may iteratively change the bin size for the histograms until you have a histogram that you are satisfied with before it will be saved. Each plot will be saved to a directory that you specify. Note that this

program is designed to allow as many overplots of different wavelengths as are desired; however, the limited range of colors I have coded into the program means that it will throw an error if you attempt to plot more than 8 data sets together. In the unlikely event that you need to plot more than 8 data sets together, the code will need to be modified to include more colors for plotting. Other than that, this program is capable of plotting as many sets of data as are needed. For the loop flux plot, a 10 Gauss field is hard-coded into the program.

Calling Sequence: `loop_pipeline, /file_gui, /overplot, /means`

Input Parameters / Keywords: No input parameters, optional keywords `FILE_GUI`, `OVERPLOT`, and `MEANS`. If the `FILE_GUI` keyword is used, a window will pop up allowing you to navigate to your save file and select it rather than typing in the file path. If the `OVERPLOT` keyword is used, you may enter multiple file paths to different wavelengths of data for the same event and they will be plotted together in the loop emergence, cumulative area, flux, and voltage plots. If the `MEANS` keyword is used, the average value for voltage will be plotted as a straight line on the voltage graph.

Output: Displays and saves plots to a desired location

Example: In `SSWIDL`, the following lines will create and save plots of loop emergence, cumulative area, flux, and voltage as well as histograms of loop diameter and duration for 171Å and 211Å data for case 1 to your current working directory. Mean duration, diameter, and voltages for the two wavelengths will be printed in the terminal. See figures 3 - 9 for examples of the output produced by this code.

```
loop_pipeline, /overplot, /means
/disk/data/jwalters/case1/analysis_v4.sav
211
/disk/data/mimccart/jan2011_stack_loops.sav
171
n

y
y
y
y
```

stack_plot.pro

Location: `/home/jwalters/stack_plot.pro`

Name: `make_stackplot`

Purpose: Display a stackplot of an active region event with identified interconnecting coronal loops overlaid for two wavelengths.

Explanation: In order to use this program, you must have a file path to a hotter temperature wavelength (211Å in this project) save file with stack and loops structures retrieved from Marika's analysis pipeline, and a similar file path to a cooler temperature wavelength (171Å in this project). If you also want to use the `ARCSEC` keyword, you must have a file path to an index variable for each wavelength that includes the `cdelt1` index. `Cdelt1` allows you to convert pixels to arcseconds.

Calling Sequence: `make_stackplot, /arcsec`

Input Parameters / Keywords: No input parameters, optional keyword `ARCSEC` that allows the x-axis to be plotted in arcseconds rather than pixels starting from the initial arcsecond location. For clarification, that means that if the keyword isn't set and the initial slit started at -250

arcseconds in x and was 300 pixels across, the x-axis will start at -250 and go to 50.

Output: Displays a plot

Example: In SSWIDL, the following lines will display a stackplot with 211Å background, 211Å identified loops in red, and 171Å identified loops in blue for case 1. See figure 10 for the output of this code.

```
make_stackplot, /arcsec
/disk/data/jwalters/data211_2011_01_20/prep120211.sav
indol
/disk/data/jwalters/case1/analysis_v4.sav
/disk/data/krenzler/confirmation/preped_saves/20_2200_21_0150.sav
cutind
/disk/data/mimccart/jan2011_stack_loops.sav
```

create_quadrupoles.pro

Location: /home/jwalters/create_quadrupoles.pro

Name: create_quadrupoles

Purpose: Create a quadrupole model of a given active region event containing poles, connectivities, and time stamps.

Explanation: In order to use this program, you must have a file path to a save file containing a stack structure variable. This program uses the suite of MPOLE programs created by Dr. Longcope. For more information on the MPOLE programs, please visit <http://solar.physics.montana.edu/dana/mpole/>. The program will run through the time span specified by the stack structure and ask the user to select the desired region and trace around the emerging region for each time step. If you are on a Mac, in XQuartz X11 preferences you must have Click-Through Inactive Windows selected for these programs to work.

Calling Sequence: qp_cm_struct = create_quadrupoles(/cadence)

Input Parameters / Keywords: No input parameters, optional keyword CADENCE that allows the user to change the number of hours that pass before a magnetogram is pulled from the HMI archive. The default is 1 magnetogram every hour, but in certain cases it might be preferable to have 1 magnetogram every 5 hours to get a general trend as this code can be time-consuming depending on the duration of the event.

Output: A structure containing poles, connectivities, and time stamps.

Example: In SSWIDL, the following lines will create a structure called qp_cm_struct for case 1.

```
qp_cm_struct = create_quadrupoles()
/disk/data/jwalters/case1/analysis_v4.sav
```

quadrupole_flux_elaborate.pro

Location: /home/jwalters/quadrupole_flux_elaborate.pro

Name: quadrupole_flux

Purpose: Displays a plot of flux for a given active region event using loops identified from SDO data and flux modeled using a quadrupole model.

Explanation: In order to use this program, you must have a quadrupole structure from create_quadrupoles and file paths to stack/loop structures for the wavelengths you wish to plot. The program will plot the flux of P2 and N2 (the existing active region which stays relatively constant)

and the flux of P1 and N1 (the emerging active region which trends to greater magnitudes during the event) from `qp_cm_struct.poles`. The interconnecting flux from P1 to N2 is calculated from `qp_cm_struct.connectivities`. The interconnecting flux from SDO identified loops is found in `loops` structure variables.

Calling Sequence: `quadrupole_flux, qp_cm_struct, /broken_axis`

Input Parameters / Keywords: A quadrupole structure containing poles, connectivities, and time stamps is the only input parameter. There are many optional keywords that can change the type of plot that is created. The first of these is `BROKEN_AXIS`. This keyword creates a plot that extends from the minimum negative flux to the maximum negative flux from the quadrupole structure, breaks the axis, and then resumes from the minimum positive flux to the maximum positive flux from the quadrupole structure. The advantage of this plot is that it allows you to follow trends more clearly in the quadrupole flux as the default graph extends from the minimum negative flux to the maximum negative flux, leaving about 40 unused orders of magnitudes in the middle. I have also programmed the `BROKEN_AXIS` keyword to resize the positive and negative portions of the graph so they visually match the amount of data that each section spans. For example, in the broken axis plot for case 1 data, there is a greater range of data covered in the negative flux portion of the graph, so the negative portion of the graph is slightly taller than the positive portion of the graph. The second keyword is `ZOOM`. This keyword plots only the interconnecting flux, leaving out P2/N2 and P1/N1. This graph will cover a range of data from the 1% of the max value of interconnecting flux from the quadrupole model to 110% of the max value of the interconnection flux from the quadrupole model. The SDO identified loop flux is overplotted on this graph. The `ZOOM` keyword allows closer examination of the trends of interconnecting flux between the quadrupole model and observed data. The final keyword is the `TOTAL_FLUX` keyword, which must be used in conjunction with the `ZOOM` keyword. This keyword will do everything the `ZOOM` keyword does, with the addition of plotting a cumulative flux between the different wavelengths of data. Currently, this function is hard-coded to sum only two wavelengths of data (which was what I used for my project, to sum 211Å data and 171Å data). Adding the two fluxes is not as easy as it first seems because the two wavelengths of data had different cadences of data in the same time range. In my program, as long as the two pieces of data are within 35 seconds of each other, it will count them as the same time and add them together. Work still needs to be done on this keyword to make it compatible with more than two wavelengths.

Output: Displays a plot

Example: In `SSWIDL`, the following lines will display a flux plot for case 2 using the quadrupole model and stack/loop variables from Marika's analysis pipeline. Refer to figures 11 - 14 for examples of this code using the different keywords.

```
restore, 'qp_cm_case2_v1.sav'  
quadrupole_flux, qp_cm_struct[3:*]  
2  
/disk/data/jwalters/case2/case2_stack_loops.sav  
211  
/disk/data/jwalters/case2/case2_171_stack_loops.sav  
171
```

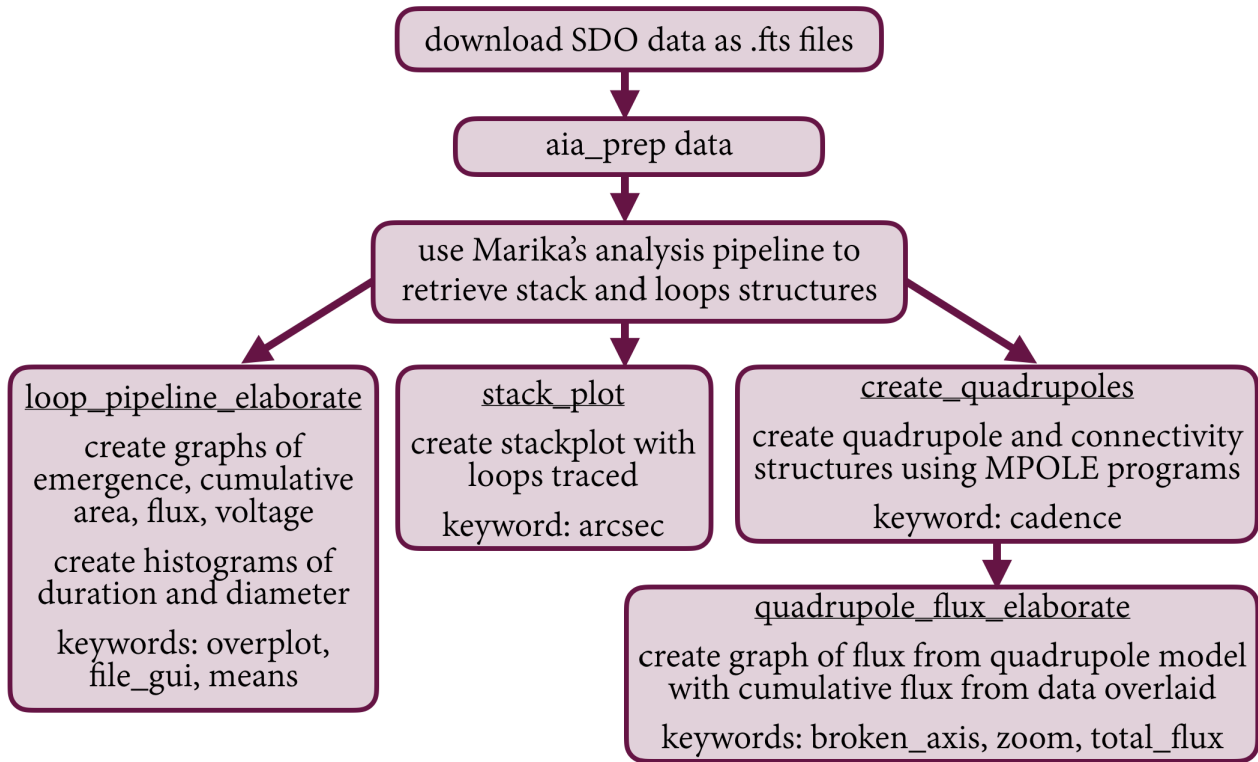


Figure 1: General outline of code and data flow for the project.

Other Notes

Case 1 Data

Case 1 .fts files for 211Å data can be found at `/disk/data/jwalters/data211_2011_01_20`

This data was used to create 3 aia_prepped files containing index/data variables found at the following locations:

`/disk/data/jwalters/data211_2011_01_20/prep120211.sav`

`/disk/data/jwalters/data211_2011_01_20/prep1202112c.sav`

`/disk/data/jwalters/data211_2011_01_20/prep1202113.sav`

This data was used to retrieve stack and loop structures which can be found at `/disk/data/jwalters/case1/analysis_v4.sav`

The stack and loops structures for 171Å data can be found at `/disk/data/mimccart/jan2011_stack_loops.sav`

The quadrupole model structure can be found at `/home/jwalters/qp_cm_v3.sav`

Case 2 Data

Case 2 .fts files for 211Å data can be found at `/disk/data/jwalters/case2_data`

This data was used to create 2 aia_prepped files containing index and data variables found at the following locations:

`/disk/data/jwalters/case2/case2_parta.sav`

`/disk/data/jwalters/case2/case2_part2_again.sav`

The stack structure and loops structure for 211Å data can be found at `/disk/data/jwalters/case2/case2_stack_loops.sav`

The stack structure and loops structure for 171Å data can be found at `/disk/data/jwalters/case2/case2_171_stack_loops.sav`

The quadrupole model structure can be found at `/home/jwalters/qp_cm_case2_v1.sav`. Note that for the first three indices of the data, I was unable to identify an emerging active region from the magnetogram. This is why in the example code for `quadrupole_flux_elaborate.pro`, the indexing for the structure starts at 3.

Kolmogorov-Smirnov Test

There was interest in conducting a Kolmogorov-Smirnov two sample test for diameters and durations between Case 1 and Case 2 to see whether it showed the cases were from different underlying populations. I ran this test using a function in Dr. Longcope's files called `kstwo`. The results are shown in the table below.

Comparing Case 1 and Case 2

	Kolmogorov-Smirnov Statistic	Significance Level (0-1)
Diameters (211 Å)	0.17168698	0.059535548
Diameters (171 Å)	0.13501932	0.093579732
Durations (211 Å)	0.075553656	0.88554215
Durations (171 Å)	0.21643838	0.00076538156

Comparing 171Å and 211Å Populations

	Kolmogorov-Smirnov Statistic	Significance Level (0-1)
Case 1 Diameters	0.25791529	0.0000098220708
Case 2 Diameters	0.21604633	0.010512816
Case 1 Durations	0.17566302	0.0068923305
Case 2 Durations	0.076178670	0.89997983

Figure 2: Results of the Kolmogorov-Smirnov two sample test between case 1 and case 2 as well as for 171 angstrom data compared to 211 angstrom data. Only case 1 diameters between 211 and 171 seem to definitively not be a part of the same population.

Movies

I used `.fts` files at `/disk/data/jwalters/data211_2011_02_17` in order to create a movie to show the case 2 event. All movies for case 1 and case 2 can be found at `/home/jwalters/movies`. If you are interested in creating movies of events, you can refer to `/home/jwalters/combined_movie.pro` as a template.

Obtaining Stack and Loops Structures

It is best to refer to Marika McCarthy if you have questions about retrieving the stack and loops variables which are necessary inputs for my programs described above. The programs I used this summer to retrieve stack and loops structures were found at `/disk/data/mimccart/pipeline`. There are four programs used in succession:

```

createdc.pro
createsp.pro
looptraced.pro
lof.pro

```

If you are interested in seeing how I used Marika's programs to create stack and loops structures for case 1 and case 2, refer to the following programs:

/home/jwalters/case1_v3.pro

/home/jwalters/case2_v2.pro

Website

My personal website for the REU program, which includes weekly reports of my progress, can be found at the following address: <http://solar.physics.montana.edu/REU/2018/jwalters/index.html>

Acknowledgements

This project was funded by the NSF REU grant to Montana State University. Special thanks to Dr. Qiu and Dr. Longcope for their work with the MSU Physics REU program as program coordinators. I would also like to acknowledge my supervisors for this project, Marika McCarthy and Dr. Dana Longcope.

References

Longcope, D. W., McKenzie, D. E., Cirtain, J., & Scott, J. (2005). Observations of Separator Reconnection to an Emerging Active Region. *The Astrophysical Journal*, 630, 596-614. doi: 10.1086/432039

Plots

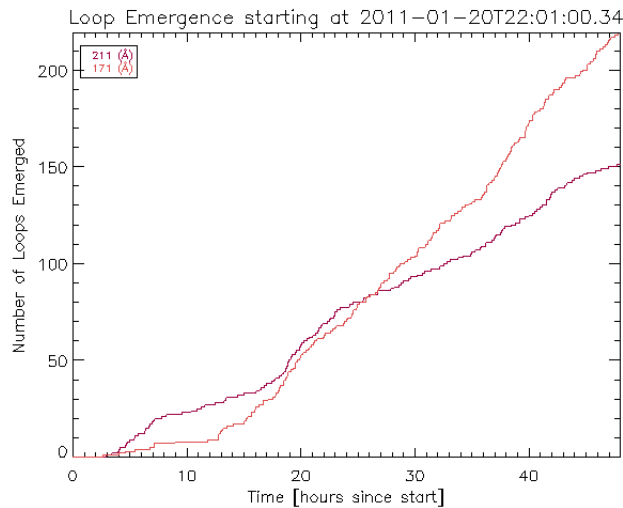


Figure 3: Plot of loop emergence for case 1 showing both 211Å data and 171Å data. This is one of the outputs of the example code for loop_pipeline_elaborate.pro.

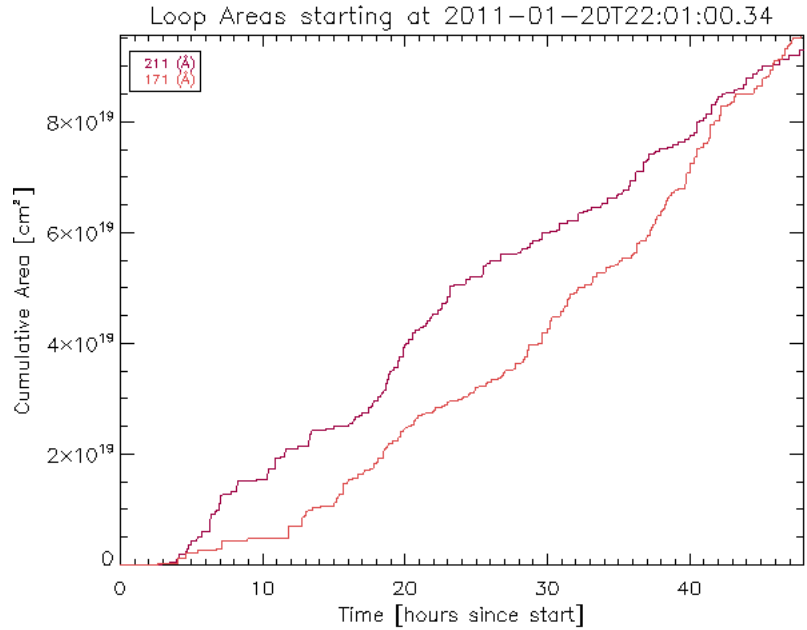


Figure 4: Plot of cumulative loop area for case 1 showing both 211Å data and 171Å data. This is one of the outputs of the example code for `loop_pipeline_elaborate.pro`.

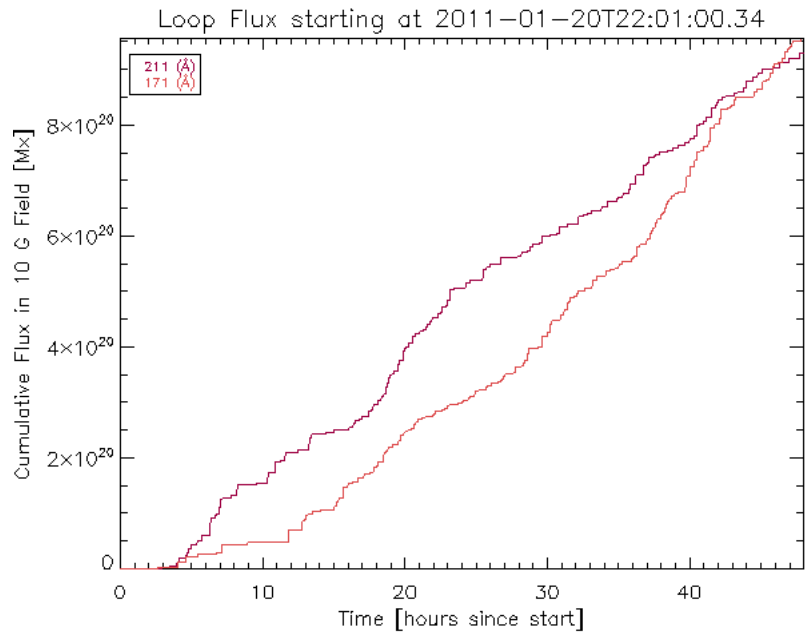


Figure 5: Plot of cumulative flux in a 10 Gauss field for case 1 showing both 211Å data and 171Å data. This is one of the outputs of the example code for `loop_pipeline_elaborate.pro`. It is interesting to note that in the both wavelengths, the final flux ends up being very close despite the difference in number of loops identified. This similar end flux can also be seen in case 2. It would be interesting to see if it occurs in any of the other 14 cases.

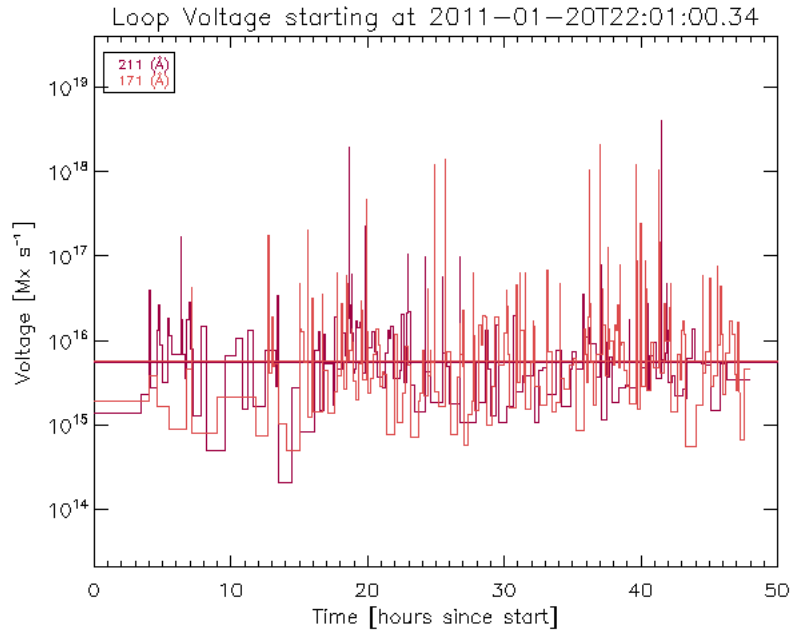


Figure 6: Plot of voltage for case 1 showing both 211Å data and 171Å data by taking the time derivative of the flux. This is one of the outputs of the example code for `loop_pipeline_elaborate.pro`. The mean values of unsmoothed voltage are printed in the terminal as the code is running.

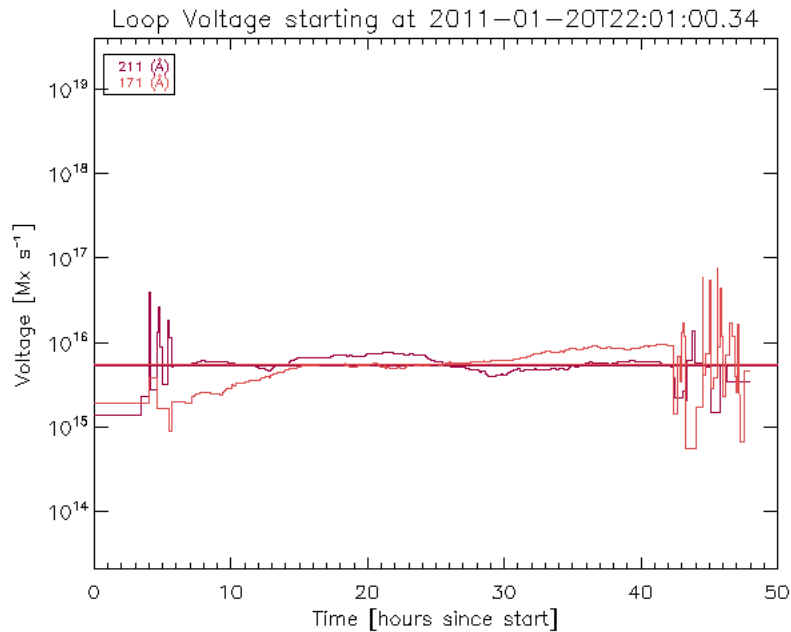


Figure 7: Plot of smoothed voltage for case 1 showing both 211Å data and 171Å data by taking the time derivative of the flux. This is one of the outputs of the example code for `loop_pipeline_elaborate.pro`. The mean values of smoothed voltage are printed in the terminal as the code is running. The smooth function is not good at smoothing the edges of the data, which is why there are discrepancies at the start and end.

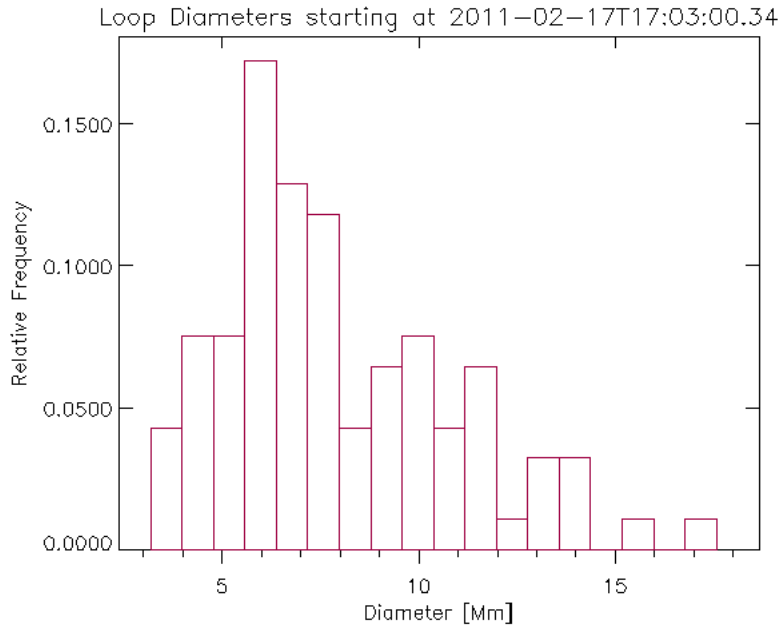


Figure 8: Histogram of 211Å data diameters for case 1. This is one of the outputs of the example code for `loop_pipeline_elaborate.pro`. The bin size was adjusted by the user as the code was run. For both case 1 and case 2, loop diameters were larger in 211Å.

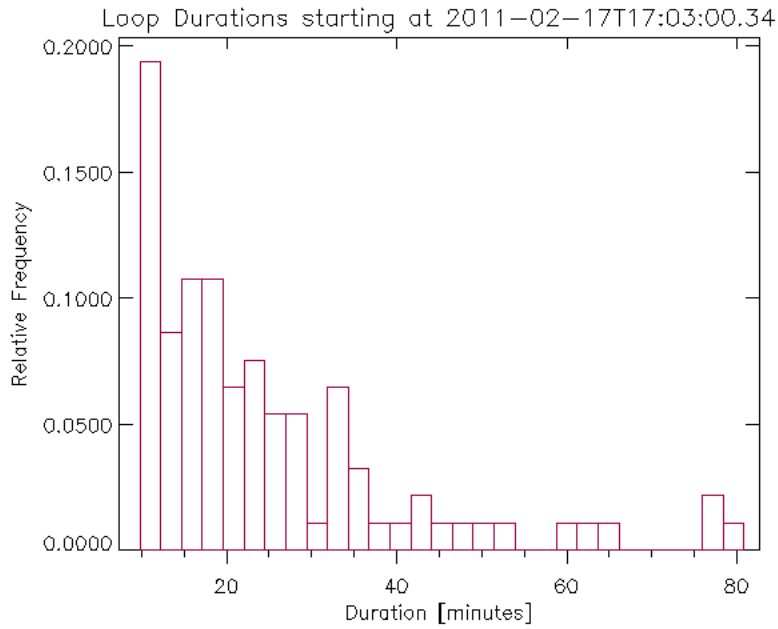


Figure 9: Histogram of 211Å data durations for case 1. This is one of the outputs of the example code for `loop_pipeline_elaborate.pro`. The bin size was adjusted by the user as the code was run. For both case 1 and case 2, average loop durations were nearly identical between 211Å and 171Å.

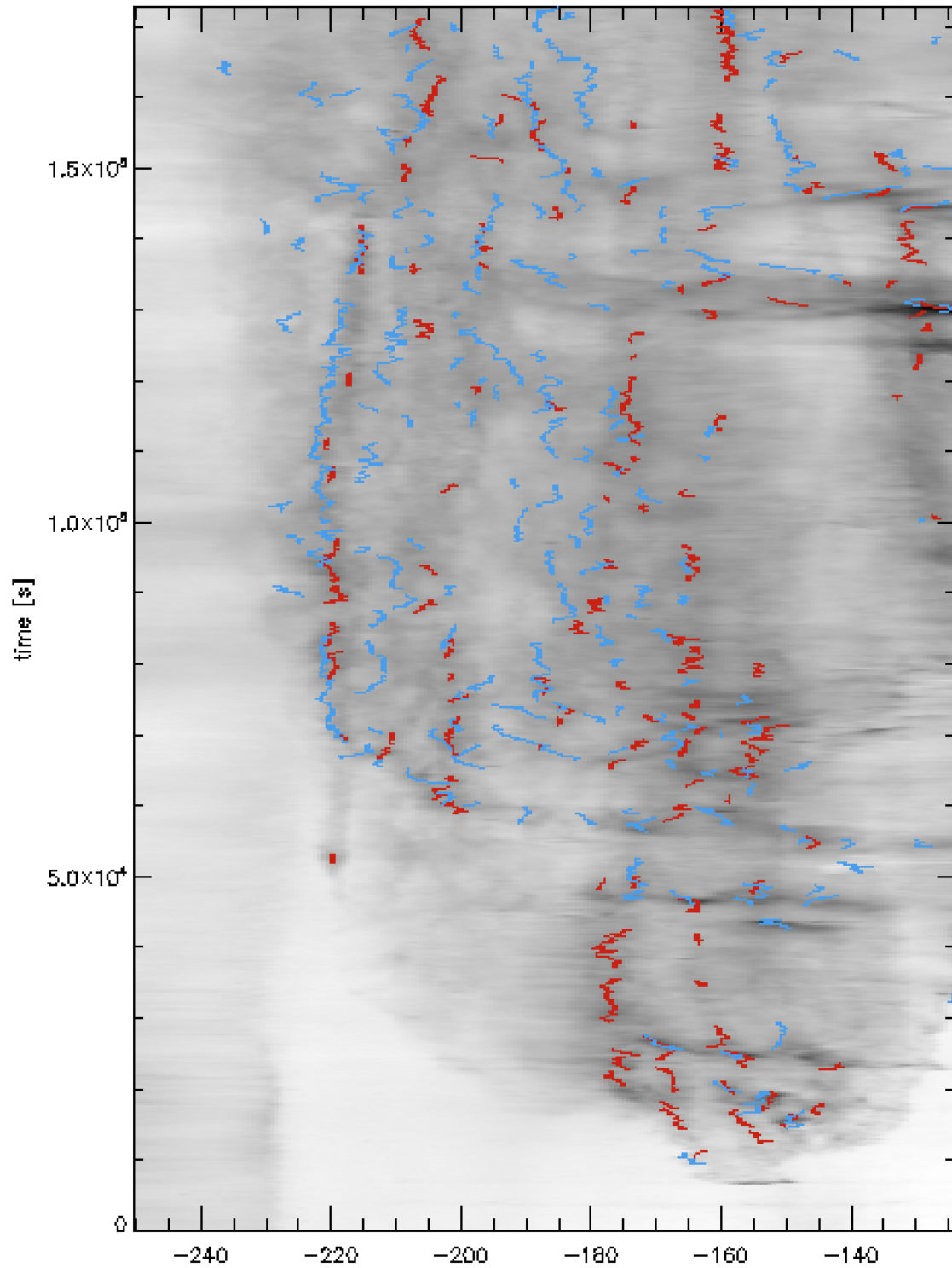


Figure 10: Output of the example code from `stack_plot.pro`. The `arcsec` keyword was used, so the x-axis is in arcseconds. This shows the stackplot for case 1 with 211\AA loops in red and 171\AA loops in blue. For future work on this project, a program could be created that tallies the number of loops between the two wavelengths that appear to be heating up or cooling down.

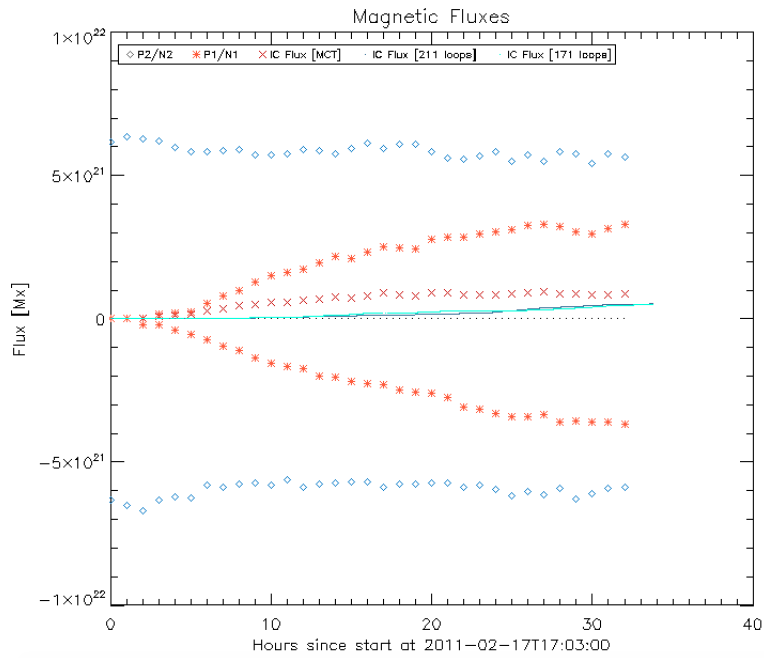


Figure 11: Output of the example code from `quadrupole_flux.pro`. This shows the flux from the quadrupole model and identified loops in 211Å data and 171Å data for case 2.

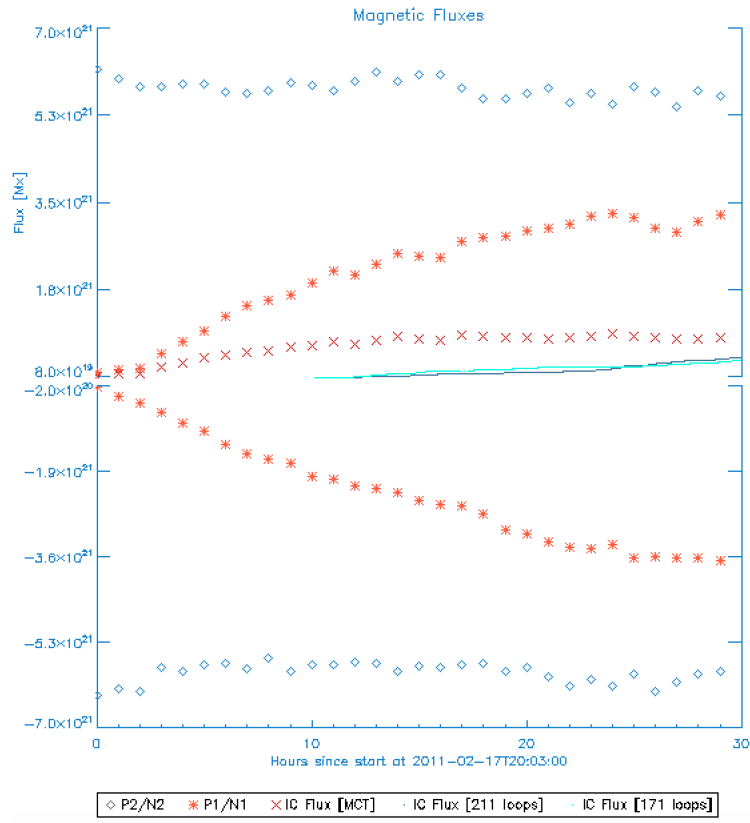


Figure 12: Output of the example code from `quadrupole_flux.pro` if the keyword `BROKEN_AXIS` was used.

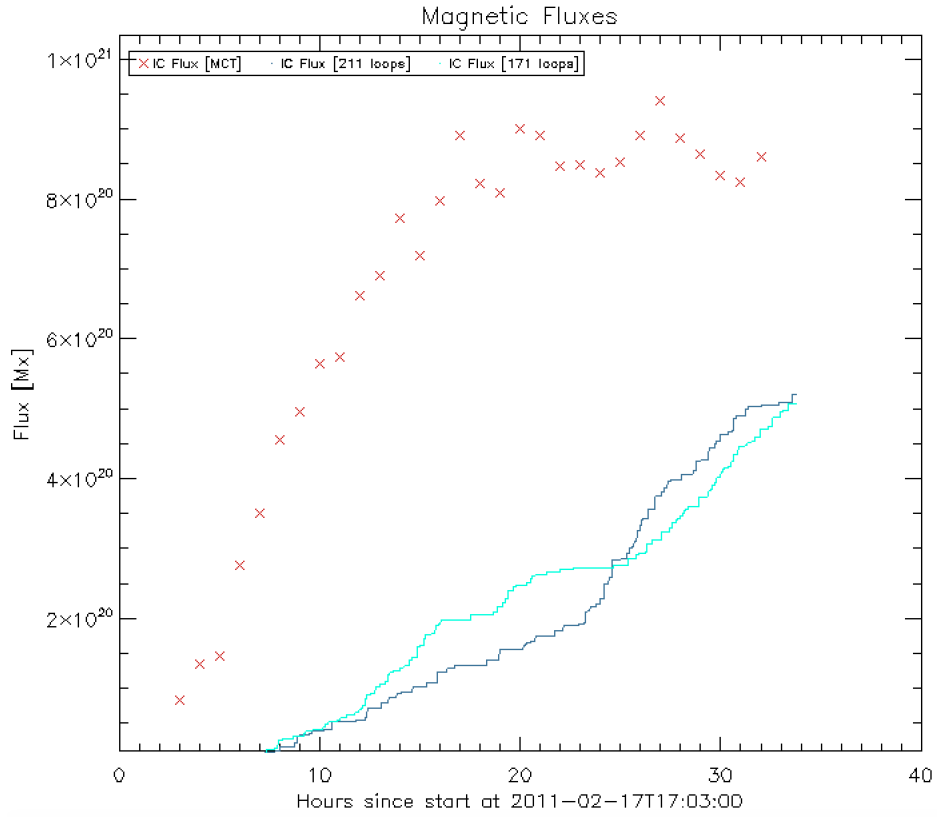


Figure 13: Output of the example code from `quadrupole_flux.pro` if the keyword `ZOOM` was used.

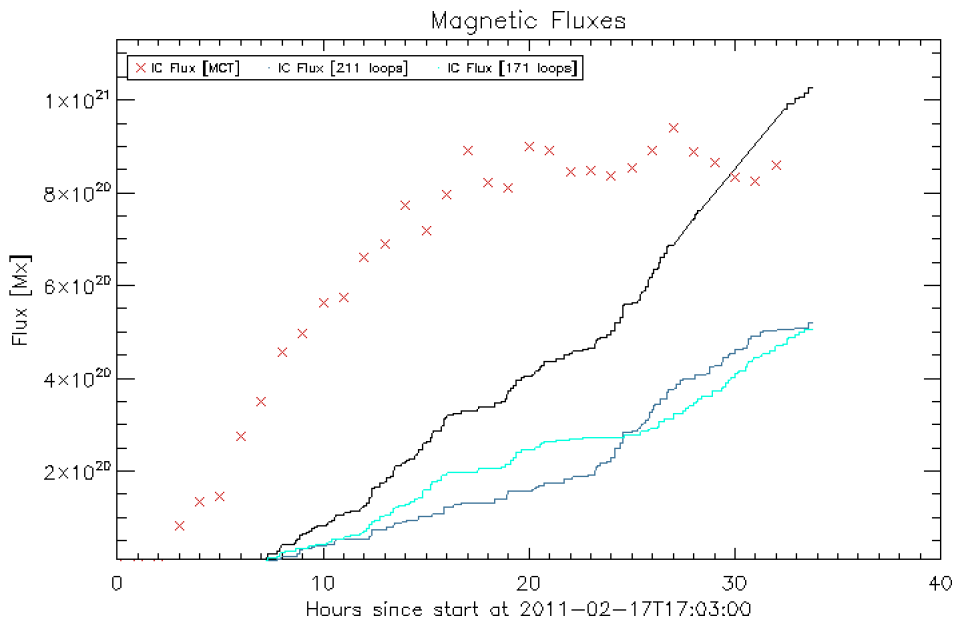


Figure 14: Output of the example code from `quadrupole_flux.pro` if the keywords `ZOOM` and `TOTAL_FLUX` were used.