dz currently set to 64 instead of 50 to get power of 2.

Subroutines again - which lines need to be tweaked every now and then? Which ones will stay the same? Which ones are the same at a certain level, depending on which specific set of data is being plotted? For example, plotted both power and flux as functions of time. Only the y data (and y titles) changed for this entire code. Everything else (setting size and position of panels, x-axis data/title, colors, etc.) was exactly the same. Pull out graphics lines that need to be tweaked, vs. those that never change. Whatever makes it easier to re-make plots, whether saving to file with a bigger font or different resolution, or changing the time covered by my lightcurve, or adding a dotted line showing the background, etc. Should I subtract the background?? Go back farther in time? Shouldn't have to worry that I'll accidentally change data values when doing graphics. Science part needs to go in a completely different routine! If I need to makes changes, there shouldn't be any variables buried somewhere where it's easy to miss them. Better yet, only change variable ONCE (like when feeding into a subroutine). Having lots of little subroutines seems like the way to go, so need to figure out PATHs in IDL. That way I can make subdirectories for each part of project. Separate for plotting important things and making plots pretty. This might be easier to deal with; adjust position, tick length, all that crap in a different routine. Apply customizations as properties, after graphics are created. *Develop* code at Main Level, then put in subroutine. Also would be helpful to split so that some things can be tweaked without having to re-create the entire graphic. Need to work out methods, how to select different graphics. Need to restore variables and whatever calculations are necessary to get data in proper form for desired graphics. Combine all of these in structures (maybe use a subroutine just for this purpose) and pull whatever is needed, assuming this doesn't use too much memory. Easy way to add vertical lines, change axis label, etc. Needs to be set up to save to file in case I'm making quick changes for figures that need to go in a paper or poster right away. All pieces that stand alone should be in a subroutine, especially bits that define variables that are only needed for that one little thing. Goal is to *separate* code so each bit is easier to change, or at least be able to tell at a glance exactly what it's doing. Don't necessarily need to condense codes or make them as short as possible. Make sure the same block of code really can be used in multiple situations before trying to put it in a subroutine. Don't want to waste a bunch of time because I keep stumbling over undefined variables or arrays that don't line up.

Podcast of the day - "Intersection of disciplines"

Ideas for this thing:

- Free-writing, methods, and anything else that has a place in my research articles and/or dissertation can easily be copied into the file and combined with whatever else I have on the topic. But only copying, so still preserving my original mess.

Obviously, will need to use it for a while to see what really works and what doesn't, especially when it comes time to copy important bits to where they need to go. If I don't actually do this, there's no point.

`Resolve_routine` for functions inside procedures?

graphic2.pro /kw still works!

Shouldn't have to worry that I'll accidentally change data values when doing graphics. Science part needs to go in a completely different routine!

Calc. power map for every 5th timestep and showing all of them vs. Calc. every power map and showing every 5th one.

For next time:

Flux in second lightcurve should be over subset, not entire AR. Or just do the analysis for entire AR. Fix subroutines for position and window, so all images are the same size.

Plotting power vs. time, but with power covering the entire spread of time for which the fourier transform was calculated. Not feeling great about power as function of start time. Try WA? Maybe having *smoothed* data in lightcurves is how continuous frequencies are produced!

Flat corrections

Flow: integrated emission first (FT plots as function of time). Then spatial resolution to go with the temporal resolution.

Make $dz = 64$ (25.6 minutes) instead of 50 (20.0 minutes)? This would make it a power of 2 (to the 6th). How much faster is this, really? T = [170.7,192.0] instead of [171.4, 200.0].

Forgot to change z. Still indgen(100). So I'm doing every single step, but the calculations will stop too early. Should have done every 5th or so, up to z = $\sim$224.

Plot of power vs. time - one plot should probably be dotted as well as different color.

More motivation/previous research - Monsue "instability in chromosphere", cutoff frequency, significance of natural frequency. Why using AIA data? Background needs diagram or image. Lots of pre-flare stuff. Describe methodology/approach somewhere. Clearly state dz=50, dν. Correct *title*. Acronyms.

Simplify graphics routines. Figures out PATHs in IDL, so all the little ones can go elsewhere.

Read headers separately from rest of prep.pro

Make sure JD is reset every time I need to re-run interp on AIA 1700Å.

Separate for plotting important things and making plots pretty. This might be easier to deal with; adjust position, tick length, all that crap in a different routine. Apply customizations as properties, after graphics are created. *Develop* code at Main Level, then put in subroutine. Also would be helpful to split so that some things can be tweaked without having to re-create the entire graphic. Need to work out methods, how to select different graphics. Need to restore variables and whatever calculations are necessary to get data in proper form for desired graphics. Combine all of these in structuers and just pull whatever is needed assuming this doesn't use too much memory. Easy way to add vertical lines, change axis label, etc. Needs to be set up to save to file in case I'm making quick changes for figures that need to go in a paper or poster right away. All pieces that stand alone should be in a subroutine, especially bits that define variables that are only needed for that one little thing.

Answer JMac's email!

Flare times (based on GOES SXR, I think):
start 01:44 UT
peak 01:56 UT
end 02:06 UT Cycle 24
X2.2 (first X-class flare for cycle 24).
SOL2011-02-15T01:56 (based on peak time?)
NOAA AR 11158

Babble goes here.

- Run powermaps for entire active region.