

## LEFA Worksheet 9/15/2010

This plus the LMA worksheet is Assignment 4 -- due 9/30

- 1) Open terminal in your home directory
- 2) `cp /usr/local/nfs/lefa/code/* ~/`
- 3) Pull in the data you need for the first flash you want to analyze. It is in `/usr/local/nfs/lefa/data`.
- 4) The data naming convention is as follows:

### **puma7\_091020-194000.bin**

**puma** [name of computer being used] **7** the number of the station, date: (20)**091020** - time **19:40:00** (UT). **bin** (binary). All files are one minute long..

- 5) Binary data files do not explicitly have time set up. It is embedded in the file as GPS information, but not yet put in a form you can use. So before using matlab, on each file, do:

**./process\_bin\_file.py -vb puma8\_091020-223300.bin**

- 6) Now the data file is in a form you can use. You should have a **puma8\_091020-223300.bdat** file in your home directory.

- 7) The following Matlab scripts will crunch the data. Feel free to extend these for further processing. In fact, I recommend you do extend them, but first you need to look at the data.

- 8) All of the .m files have descriptions which explain in detail how they work. Just edit them or type `>>help mfilename`. Below I summarize them. You can get your first figure on the screen using only `plot_fieldchange_20090915.m`

### PLOT\_FIELDCHANGE\_20090915.m

`%usage: plot_fieldchange_20090915('load')` Loads and converts to E-field

`% plot_fieldchange_20090915('plot')` Replots already loaded data

`% plot_fieldchange_20090915('save')` Saves data for future work

`%`

`%% USES:`

`% BDAT_READ, FIT_60HZ, REMOVE_60HZ, SEGMENTED_MEDIAN2, DEDROOP, STAMPIT, SET_FONTS`

**BDATREAD.M** Reads a .bdat file. You need to specify the number of columns.

`%usage: data=bdat_read('puma7_091020-194000.bdat',5)`

**FIT\_60HZ** fits the amplitude and phase of any 60 Hz contamination in the data

**REMOVE\_60HZ** Subtracts out the fitted noise.

**SEGMENTED\_MEDIAN2.** Divides the data into 10 segments. Selects the "quietest" segment and assumes there is no lightning there. Uses that as a reference to remove DC offsets from the data.

**DEDROOP.** Removes the effects of the high-pass filter in the instrument. Tries to restore the electric field to actual electric field.

**SET\_FONTS.** Changes the default Matlab fonts to more readable fonts for publications quality figures.

**STAMPIT.** Time stamps your figure. Good scientific practice.

**PRINTIT.M** Prints the figure to an EPS or JPEG file for later browsing or printing.

- 9) Inside `plot_fieldchange` is a section you have to edit by hand. It starts at line 28. Here it is.

```

%=====THINGS YOU NEED TO CHANGE=====
instrument='puma2'; %Change this for different instruments
hr=18; %Set to UTC hour of data file
minute=27; %Set to UTC minute of data file
start_sec=1; % Leave alone
stop_sec=59; % Leave alone
cname='S'; % Usually use the 'S' sensitive channel, but if flash saturates, switch to 'M' or 'I'
% If the voltage plot (Figure 1) exceeds 10 V (or -10 V) then you are
% saturating that channel and need to move to a less sensitive one.
%Change start_plot and stop_plot to highlight the flash of interest after
%you figure out where it is.
start_plot=35;
stop_plot=40;
%=====END OF THINGS YOU NEED TO CHANGE=====

```

Hopefully it is self explanatory.

### Things to do:

For each of the six LMA plots you selected last time, plot the field change data. Once you see the flash, you can set start\_plot and stop\_plot to be 1 second before and 1 second after the flash. You may want to continue to make these limits tighter to see different parts of the flash. For each of the flashes try to determine the following.

If it is CG, you should see a return stroke ... or more than one. Find them, count them. Positive or negative CG? You should also see the stepped and dart leaders. You probably have to zoom the data to find them.

You may find my software frustrating ... you didn't write it. I just gave you the bare bones to get the data in reasonable shape for further analysis. That is why I encourage you to use the .save function. You can then load the .mat file ... already processed, and plot it any way you like with your own code.

### Final result:

Look through all six of your flashes on at least one station (stations 2, 5, and 7 data are provided). Pick two with the best quality or most understandable data. If it's a CG flash, see if you can find multi-return strokes ... or a positive CG. For the two flashes you select, plot data for all three stations.

You should end up with one IC and one CG for which you have confirmed with E-field and LMA that they are an IC and a CG. Make several plots at different time resolutions to show off important features (leaders, return strokes, preliminary breakdown ... other features you think are interesting) Using the printit function, you should have .JPG or EPS of this data and should be able to e-mail it or bring it to class and show what you have learned.

You will present your data for discussion on 9/30.