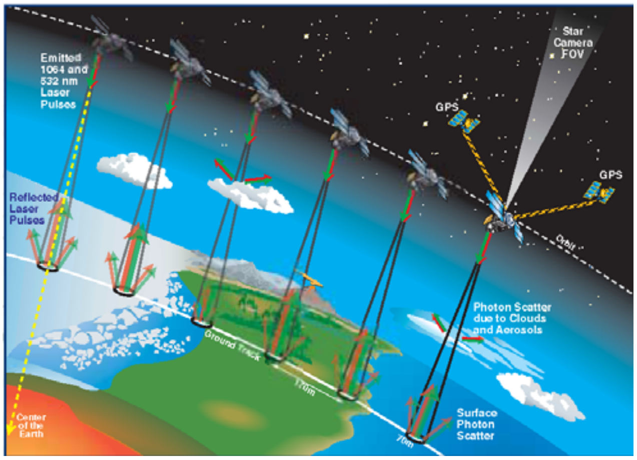
# AE1106: Programming I

## Graded Assignment #1

December 1, 2010

**Please read these instructions first:**

As an individual assignment, the code and solutions you submit should represent your own work. Discussion with others is not allowed, and neither is the use of online resources other than the standard Matlab help pages (i.e., no Wikipedia, Google searches, chat windows, etc.). During this assignment, the student assistants and the lecturer will only answer questions related to what the question is asking you to do, and will not assist you with how to actually complete the assignment. When finished, you will submit your Matlab source code file(s) to BlackBoard for grading (see below for submission instructions). **The deadline to complete the assignment is 12:00 for morning sessions, and 17:00 for afternoon sessions.** These deadlines are managed automatically by BlackBoard, so it is important that you submit your documents before the deadline in order to receive credit for the assignment (i.e., manage your time so that you start the submission process at least 15 minutes before the deadline). Those students who have been approved by the student counselors to receive extra time on their exams will receive an extra 30 minutes to complete the assignment.



This individual assignment is designed to test your ability to conduct a series of basic tasks that are common when working with experimental data sets. In short, you will perform some simple statistical analysis on a provided data set, after which you will create a plot summarizing the computed values, and then you will write your new results to a text file. The example we will use involves data collected from the Ice, Cloud and land Elevation Satellite (ICESat), which was the first satellite laser altimetry mission every launched. The primary instrument onboard ICESat was a laser that was used to make extremely precise measurements of Earth's surface (i.e., topography). Each transmitted laser pulse is reflected off the surface and is received again by the satellite, with the shape of the returned energy dependent on the height profile within the laser footprint. The data set you will work with is a full waveform profile from one of these returned pulses, and the values have been extracted from the actual data products that can be downloaded from the ICESat mission data archives. By the end of the exercise, you will write two m-files (one script and one function) which will generate one figure and a text output file. Please follow the instructions below carefully in order to avoid deductions on the assignment.

1. (2 pts) On BlackBoard is a text file called “Waveform.txt” that contains the digitized values of the returned energy from a single ICESat laser shot. The data consists of two columns, with the first being the relative time in nano-seconds (1 ns = 1e-9 s), and the second column being the “gate counts.” In the Matlab editor, create a new script file called “main.m” and use this to import the contents of this data file. Before we can analyze the waveform properly, the “gate counts” column needs to be converted to voltages using a lookup table that is located in the file “cnt2volts.txt,” which is also available from BlackBoard. This file contains 255 different rows, with the row number corresponding to a gate count. For example, a gate count of 1 would correspond to a voltage reading of -0.1953, and gate count of 5 would correspond to a voltage of -0.1686.

2. (3 pts) Now that the data has been loaded and converted, you will create a function that will compute some basic statistics of the waveform. Create a function with the following format:

   ```
   function [maxv, maxt, meanv, skew] = waveprop(t,v)
   ```

   This function will take the time and voltage vectors of your waveform data as the input variables and return as output variables the items explained below. Compute all of the items using only basic programming techniques, such as for-loops and if-statements. **This means you are NOT allowed to use built-in Matlab functions such as mean, sum, std, find, etc., in your final submitted work, although you may use them to verify your code.**

   a. Search the input data to find the value (in V) and location (in time) of the maximum point of the waveform. The function output variable **`maxv`** should contain the value of the maximum point, while the function output variable **`maxt`** should contain the time when this maximum occurs.

   b. After conversion, the minimum value of the waveform is assumed to be zero volts (0 V). The times that this minimum value is reached on either side of the maximum value will define the starting and ending points of the waveform. As such, the points outside of the waveform (i.e., the “flat” regions) should NOT be used to compute the statistical values described next (such as the mean and skewness). For the data set provided, the start and end times were found to be at **231** and **470** ns respectively. Using only the data between these starting and ending times, compute the mean (i.e., average) value of the waveform, and return this value in the function output variable **`meanv`**.

   c. Compute the skewness of the waveform (**`skew`**), which gives us information about the degree of symmetry in the signal. The skewness is defined as:

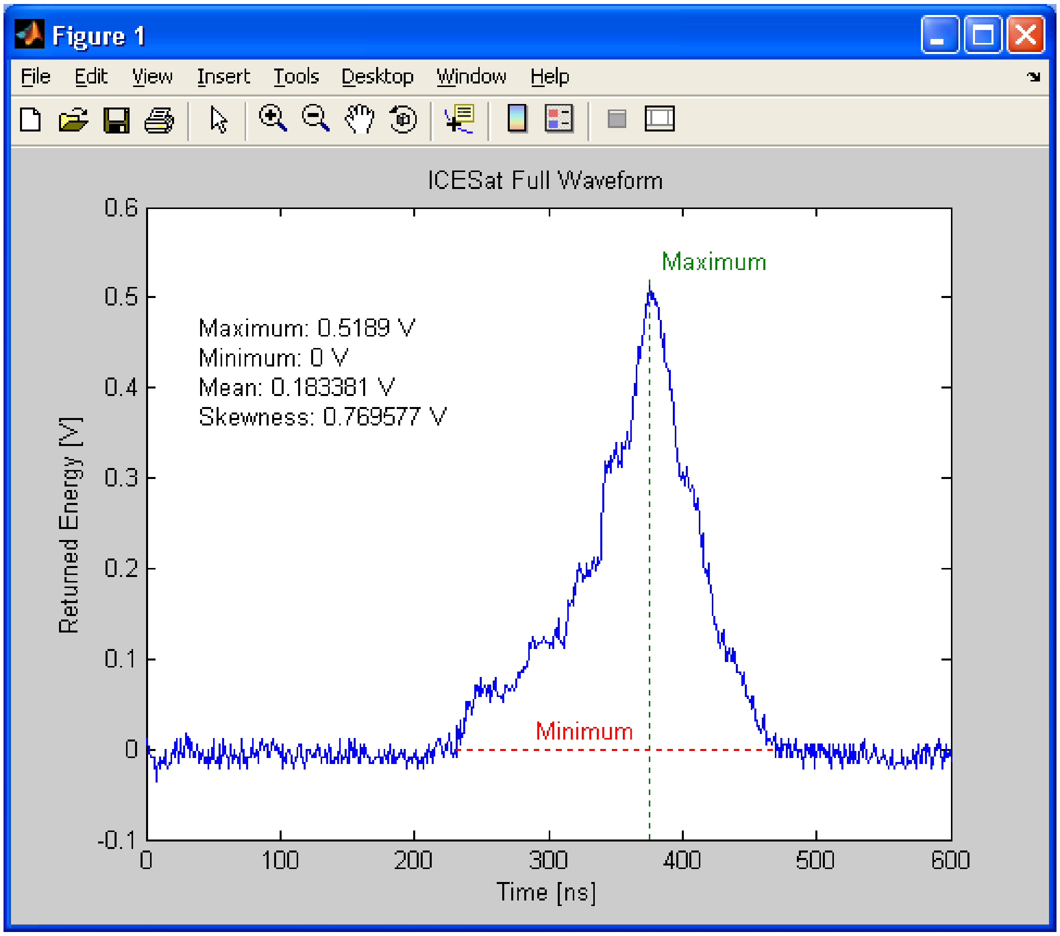
   $$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\bar{x}}{\sigma_x}\right)^3$$

   where $\bar{x}$ is the mean value of the waveform (as computed in part b above), N is the number of data points, and $\sigma_x$ is the standard deviation of the waveform values, defined as

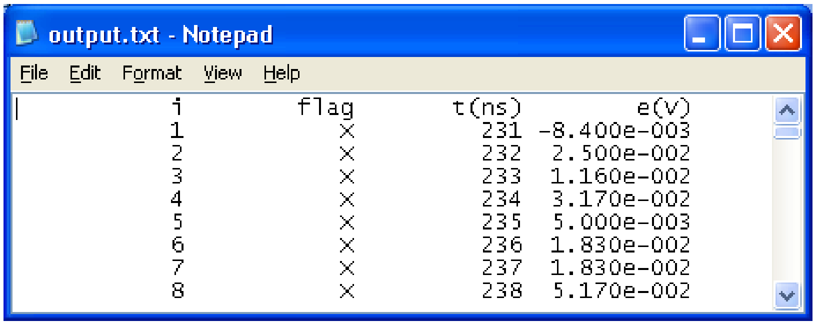
   $$\sigma_x=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i-\bar{x}\right)^2}$$

3. (3 pts) Your main script should now have loaded the data and called the function waveprop to perform some basic analysis on the waveform data. Now you will plot the results. Again, from your main script, attempt to duplicate every detail shown in the plot below, including line styles, colors, text, etc. (Hint: for the text, do not manually type the numerical values…use **`sprintf`** and the output from waveprop). Be sure to consult the course materials or Matlab help files for the detailed

plotting controls. For the green color shown for the maximum text and line, use the RGB combination [0 0.5 0].



4. (2 pts) Using the data from part 2, have your script write out the time and (converted) energy values of only the waveform (i.e., not the “flat” part of the data) to a text file called “output.txt”. The text file should contain a one line header and have two additional columns with a row counter (called “i”) and a string value (called “flag”). The flag column should contain an “X” for all values. All columns should have a width of 12. Because each row contains both text and numerical values, you cannot use **`dlmwrite`**; you must use **`fopen`** and **`fprintf`** instead. When opened in a text editor, your file should look like the image below. As with the plot from part 3, try to duplicate all aspects of the output file, including column alignments, header labels, and the display of numerical values.



```
           i        flag       t(ns)        e(V)
           1           X         231 -8.400e-003
           2           X         232  2.500e-002
           3           X         233  1.160e-002
           4           X         234  3.170e-002
           5           X         235  5.000e-003
           6           X         236  1.830e-002
           7           X         237  1.830e-002
           8           X         238  5.170e-002
```

## Grading

1) The value for each question is provided above, totaling to 10. Each question will be mostly evaluated on how closely the plots and output match the requested format; however, deductions will be made for code that does not run properly or produces unclear results.
2) Remember to only submit two m-files for this assignment, “main.m” and “waveprop.m”. Running the main script should generate all of the plots and output for all of the questions.

## Submission Instructions

1) When you are ready to submit your files to BlackBoard, go to the Assignments and then click on the “Graded Assignment 1” item. This will open a page where you can upload your assignment files.
2) By clicking on the “Browse My Computer” button next to the “Attach File” label, you can add one or more files to your assignment submission. You can save the current state of your submission at any time using the “Save as Draft” but your work will not be officially recorded in the system until you click the “Submit” button. After you have submitted the assignment, BlackBoard should then show you a summary page, which lets you know the submission went successfully. **Please be patient after submitting your files, as sometimes it takes a few seconds before BlackBoard registers everything properly**. We recommend that after submission, you wait 15-20 seconds, then go back to the course homepage and proceed again through to the “Graded Assignments 1” item. After doing this, you should see all of your submitted files in the summary page if everything went smoothly.
3) Remember that the submission folder automatically disappears at the deadline, so please make sure you start the submission process well before this time.
4) If you made an error and you wish to change a file after your have hit the submit button, you can go back to the “Graded Assignment 1” link and click on the “Start New Submission” link to repeat the entire process (i.e., do not just submit the updated file, but completely resubmit all files). Note that if you make more than one submission, we will only grade the last attempt.
5) As a reminder, please do NOT include special characters in the names of your files, such as “%”, brackets “[ ]”, underscores “_”, etc., as BlackBoard may have problems with these. Also, do NOT upload a “.mat” or “.asv” file…only upload your “.m” files, or other documents (e.g., MSWord file) if requested.