

# Methods

## Data Analysis

Transect Data Converting to SWE??

## 1 Data Processing

### 1.1 Linear and Curvilinear Transects

Snow depth measurements along the linear and curvilinear transects were taken at locations a certain distance from marked waypoints. Since only the coordinates of the waypoints (WP) were recorded, the measurement coordinates needed to be estimated. The measurement locations were assumed to be 10, 20, and 30 m behind the marked WP, in a straight line between the marked WP and the previous WP. In cases with only two observers, locations were assumed to be 10 and 20 m behind the marked WP. For the first marked WP of a pattern, it was assumed that the locations were along the same line as that between the first and second WPs. The following methodology was used to determine measurement locations (each step corresponds with a section of the Matlab code ‘MeasurementLocations.m’):

1. Waypoint (WP) locations were exported from the GPS units using the BaseCamp program. They were then imported to QGIS and exported with UTM coordinates. This file was then used in the Matlab script and is entitled ‘GlacierWP\_UTM.xlsx’.
2. In order to obtain the measurement locations for the first WPs of each pattern, an “imaginary” WP was created that was along the line between the first and second WPs, but located ahead of the first WP. These waypoints were then inserted into the original data.
3. A set of 1000 equally spaced points was created along a straight line between each set of subsequent WPs (including the “imaginary” WPs from the previous step) using the function `linspaceNDim.m` created by Steeve Ambroise and downloaded from the MathWorks File Exchange. The Euclidean distance between these interpolated points and the marked WP was then calculated and the points with distances closest to the assumed separation between observers were retained. The final matrix has the easting and northing of each measurement location and is labelled with the marked WP and a decimal that corresponds to the relative observer (e.g. label 45.2 means that the location was determined from the marked WP #45 and is 20 m behind this WP because it is the second observer).

The data recorded by each observer in the field books was transcribed to a spreadsheet format and then imported and processed in Matlab according to the following steps (each step corresponds with a section of the Matlab code ‘Import\_Transect.m’):

1. A spreadsheet was created with a sheet for data from each field book (SD#1, SD#2, SD#3, and SWEDepth). For each reference WP there were values for all snow depth measurements and their quality (1 for good, 0 for bad), comments written, field book name, glacier name, observer, pattern, and date collected.
2. The quality, comments, book name, glacier name, observer, pattern, and date entries were categorized. This allows for efficient grouping and data searching in future analysis.
3. The depth data was then assigned the corresponding measurement location UTM from the 'MeasurementLocations.m' script. This was done by matching the WP number from the field books and that of the marked WPs and then assigning the coordinates from the WP ending with .1 to depths recorded in book SD#1, and likewise for the remaining books. The matrices for each set of observations were then made to be the same dimensions by inserting empty cells for WPs where no data was recorded in that set of observations.
4. The data was then arranged in a structure variable (called SD) with rows corresponding to each book (e.g. row 1 is data from book SD#1) and columns corresponding to the various types of data (e.g. depth values or glacier category). For example, the matrix with the glacier category for each value recorded in the book SD#1 can be accessed with 'SD(1).glacier'.

Subsets of the transect data can be pulled using the function 'pulldata.m'. The function is called with `pulldata(data, book, glacier, person, pattern, quality, format)`. Here, `data` is the full SD structure, `book`, `glacier`, `person`, and `pattern` are all strings that refer to desired categories, `quality` differentiates between good (1), bad (0), or 'all' data, and `format` specifies the formatting of the full depth matrix as being either a column vector ('skinny') or a matrix with depth values for one WP in a single row.

## 1.2 Zigzag

Data from zigzag measurements, which includes the measured snow depth and the distance of that measurement from the previous measurement or vertex, was transcribed to a spreadsheet. The data were then processed using the following procedure (each step corresponds with a section of the Matlab code 'Import.Zigzag.m'):

1. Data were imported into Matlab.
2. Descriptive data, including glacier name, zigzag zone label, reference vertex, data quality, observer name, date collected, and book name were categorized.
3. A structure was created with the depth and categorical data.
4. The distance of each measurement point from its reference vertex was then calculated. These locations were assumed to be a cumulative sum of distances in a straight line between two subsequent vertices. Two options exist for the location of the reference vertex:

- (a) Option 1 calculates the distance of each point from the UTM coordinates of the reference vertex.
  - (b) Option 2 calculates the distance of each point from the end of the previous line of measurements. Since the total distance measured with the avalanche probe did not always equal the distance between vertices (likely due to error in GPS units), this option takes the reference for each line to be from the end of the previous line. The coordinates of the vertices were used for the start of each ‘Z’ shape (ZZ01 and ZZ05).
5. The final processing step involves removing bad quality data, obtaining the index for the start of each zigzag (needed for future analysis), as well as converting snow depth to snow water equivalent (SWE) based on the density calculated from the average SWE values measured with the Federal Sampler in each zigzag (if Option 2 is selected for that section).

### 1.3 Density

The density data from snowpit and Federal Sampler measurements were compiled into a spreadsheet and the density from each measurement calculated and summarized in a spreadsheet.

Integrated snow density was calculated from snowpit measurements. This was done by multiplying the measured density from each wedge sample by the depth of the sample and summing these values. A density of  $917 \text{ kg m}^{-3}$  was applied to ice layers and a density of  $600 \text{ kg m}^{-3}$  was applied to layers that were described as ‘hard’ and were too difficult to sample. When varying the ice density between 700 and  $917 \text{ kg m}^{-3}$ , the ice thickness by  $\pm 1 \text{ cm}$ , and the ‘hard’ layer density between 500 and  $600 \text{ kg m}^{-3}$ , a maximum error of 10% was found. This error was applied to all values of integrated snow density.

Density values determined from Federal Sampler measurements were filtered based on quality and then averaged for each measurement location. Measurements that were deemed to be unrepresentative of the local snow pack, which included measurements where the inner core length was less than 70% of the snow depth or where density values were exceptionally high (e.g.  $490 \text{ kg m}^{-3}$ ), were marked as bad quality data. The data was processed in Matlab as follows (each step corresponds with a section of the Matlab code ‘Import\_Density.m’):

1. Data were imported into Matlab and only good quality data were kept. Indices for data subsets (e.g. only density values from snowpits) were identified manually.
2. The mean density, standard deviation, and number of good measurements was calculated for the zigzag locations (Federal Sampler).
3. The mean density, standard deviation, and number of good measurements was calculated for the snowpit locations (Federal Sampler) and combined in a matrix with the corresponding snowpit derived density values.
4. A structure with the processed data was created.