(1) File naming scheme

Currently, file names have some base name and an ending of the form "_xxx" specifying the projection angle for this data. With the projection angle now specified in the header of each file, I would prefer if the "xxx" was simply a sequential number starting from 0 (or 1) and increasing up to the number of files - 1 (or number of files if starting from 1). To read the data right now, I have to specify the data format and the angular interval between successive projection angles and then use this to determine what files to look for. However, now that the header has all of the information I need to determine both the data format and projection angle, it would be much easier to just always iterate through files from 1 until there are no more files to read. This would eliminate the need to know the interval between projection angles, since this is only used to determine which files to read (though it could easily be calculated if desired), and would not need to remain constant throughout the scan. This would also make it possible to use non integral rotation intervals (e.g. 2.5 degree rotations) without complicating the file naming (e.g. can't use "." in middle of a file name) or the file reading procedure. Although not a strict requirement at the moment (and perhaps ever), it is probably easiest and best to keep the files in order of increasing angle, but it is technically possible to process them in any order right now. I can't see how it would be easier to use any other ordering and a deterministic ordering would be useful with ordering histories for BIP reconstruction algorithms (where orderings such that $< a_i, a_j >= 0 \ \forall \ i, j \in \text{block}_k$ improves convergence considerably), so I think an inreasing numbering for an increasing angle also makes the most sense.

It would also be convenient if either (1) a consistent base name was used for data sets (at least among those generated by a particular person) or (2) the base name of the data files are the same as the name of the data set as a whole (i.e. the archive, folder, zip, ... name containing the files). There are parameters in the header file for a data set's directory, folder name, and file basename and some of the data sets I have used have had long names with no correspondence to the file names, so changing a data set means having to change the parameters accordingly. To make this faster, I have added the folder/file names for the data sets I have use and comment/uncomment appropriately to work with the desired data, but the header file is now filling up with the various values so it would be nice to reduce/eliminate this if possible. This is just a matter of convenience for me, it doesn't take a huge amount of time to type it in every time and I could change the names of the folder/files to something more convenient, so if there is no practical alternative that is fine. I only bring it up because with each file having a header, perhaps it is possible and not terribly inconvenient to standardize the naming to some degree.

- Header:
  - * Format version identifier (integer, 1)
  - * Projection angle (float — degrees)
  - * Number of events in file N (integer)
  - * U coordinates of tracker planes (4 * float — mm)
  - Magic number identifier: "PCTD" (4-byte string)

  - Run number (integer)
  - Beam energy (float — MeV)
  - Acquisition/generation date (integer — Unix time)
  - Pre-process date (integer — Unix time)
  - Phantom name or description (variable length string)
  - Data source (variable length string)
  - Prepared by (variable length string)

- Data: (For each history)
  - * t_1
  - * t_2
  - * t_3
  - * t_4
  - * v_1
  - * v_2
  - * v_3
  - * v_4

Initially I thought that having a header in every file instead of a single file was redundant, and much of this information is unnecessary for reconstruction, but I am really starting to like the idea. I see several ways these headers can be useful for me, such as simplifying the file naming scheme as mentioned above, and if the order/structure of the header can be modified in a couple ways, I can avoid most of the inefficiences/difficulties associated with the redundancy and unnecessary information. Above is the ordering I would prefer (where "*" denotes the data I need) but the ordering of the t/v values among themselves doesn't matter to me, this is just the order I came up with first; the only ordering that affects me is the header. So, I have a few requests/suggestions:

(1) Could we place the format version identifier, projection angle, number of events, and U-coordinates first in the header?

- The entire input data set typically cannot be processed on the GPU at once, so the data has to be broken up into pieces and read/processed iteratively. The first thing my program does is read the files to determine how many histories there are in each file/total so it can determine how many files can be read simultaneously (without exceeding the GPU memory) and how to break up the data. For this step, I only need the first three values, so it would be great if these were first so I can easily read these and ignore the remaining information without skipping around. I like the format identifier first, but the closer to the top the better. Wherever it goes, it should stay in the same location in future formats so the format does not need to be known up front but can instead be determined automatically. Ideally, it would be with the other three sequentially in the order shown so I don't have to skip around between reading each of them. The only 1 of the 4 I might prefer elsewhere is the U-coordinates. If they remain constant throughout the scan, then I prefer them with the other 3 (as shown above) so I can read them once and skip them after that. If these coordinates can vary during a scan, then I would prefer them to be at the end of the header and directly above the other tracker data.

(2) Could the variable length strings be changed to fixed length strings?

- When I get to reading the data, I would like to be able to skip past all the information I either already have read or do not need. To seek forward in the file to the data I need, I have to know exactly how many bytes there are between the beginning of the file and the first piece of data I need. The data preceding these variable length strings are always the same size so I do not need to read the data, I can simply seek forward by the associated number of bytes. On the other hand, once I get to the variable length strings, I must first read their sizes before I can skip past them and these sizes must also be included in the header as well. I realize that the name of a phantom or a person will vary and is not known ahead of time, but could these be made large enough to fit any desired name? The name could be initialized to all space characters and then overwritten from the beginning so in the end it has the name followed by a bunch of blank spaces. Another option would be to write in the name and explicitly fill the remainder with spaces. Even if these were made 100 (or even 1000!) characters long, this still wouldn't be a large waste of space, the header would no longer need to include their sizes, and it would be much easier and efficient for me to read the data.