Data Types

Details of each datatype used by the Dosimetry App can be found at the Dosimetry Manifest Guide.

Some more complex data types have detailed descriptions and example usages shown below:

Dij Matrix

A Dij matrix is a container used to store the dose from each beamlet (bixel) to each calculation point. It's stored sparsely for efficiency purposes as each beamlet will only provide dose to a small subset of the field points. Since the matrix is stored sparsely, it's important to understand the proper means for extracting the data it contains. The matrix data is stored as a list of *dij_entry* values. These values are properly indexed to beamlets and points by using the *dij_row* values also held as a list within the *dij_matrix* container. There is one *dij_row* value for each dose point and each value contains the offset position for accessing the list of entries as well as the number of entries for this point. This is best explained by reviewing the code example provided below that shows how to loop over the complete *dij_matrix*, summing the dose to compute the total dose to each calculation point.

```cpp
std::vector<double> pdose_total(dij.n_points, 0.0);
for (uint32 i = 0; i != dij.n_points; ++i)
{
    dij_row const& row = dij.rows[i];
    dij_entry const* entry = dij.entries.elements + row.offset;
    dij_entry const* end = entry + row.n_entries;
    // Loop over the beamlets (entries) for point i
    for (; entry != end; ++entry)
    {
        // dose to this point from entry->beamlet_index is entry->dose, so add
it's value to the point dose
        dose_total[i] += entry->dose;
    }
}
```