Vectors and matrices in LArSoft

Planning for a recommendation

Gianluca Petrillo

Fermi National Accelerator Laboratory

LArSoft architecture meeting, May 18th , 2016

Introduction to the problem

- LArSoft contains code representing vectors by many data structures: `float*`, `double*`, `std::vector<double>`, `TVector3`, `std::array<double, N>`, `CLHEP::Hep3Vector`, ...
- there have been major instances where the wrong choice caused serious performance issues
- this proliferation yields
  - growth of the interface when trying to support many of them, or
  - data conversion when moving from one context to another

Goal of the task

Define a recommendation the developers can refer to when choosing:

- which data structures to write on disk
- which libraries to use for linear algebra and geometry calculations

Areas of application

Goals of this meeting

1. agree on a process, also considering a context wider than LArSoft
2. identify which areas to cover with the recommendation
3. identify and prioritise requirements for the candidate libraries
4. collect past experiences

Different recommendations might be needed for different areas.
Areas I can think of:

geometry calculations 2D and 3D vectors, their transformations

physics calculations 4-vectors in Minkowski space

linear algebra calculations *N*-vectors, matrices, tensors, solutions to linear equations, principal component analysis...

Requirements: “static”

Example of requirements (not necessarily in any order):

1. ☐ license compatible with LArSoft
2. ☐ multi-platform
3. ☐ serialisable by ROOT I/O (with custom streamers)

Example of desirable features:

1. ☐ actively maintained
2. ☐ no memory overhead
3. ☐ fully featured (one-fits-all)
4. ☐ implemented in or aware of C++
5. ☐ header-only (or not)
6. ☐ support for sparse data

Requirements: “dynamic”

Example of “dynamic” requirements (not necessarily in any order):

1. ☐ usability
2. ☐ resources (memory, CPU)

Some names: brainstorming

Name of relevant libraries in no particular order:

- C++ Standard Template Library
- ROOT (global namespace objects)
- ROOT::Math (geometry and linear algebra)
- CLHEP
- Eigen
- Armadillo
- Elemental
- BLAS-related
  - boost::uBLAS
  - ...
- GNU Scientific Library
- FLENS
- Math Template Library (MTL4)
- PETSc
- Generic Graphic Toolkit