LHCb Physics Event Model Comments on the MathCore Vector and the Root Linear Algebra Classes

Introduction

The review of the physics event model is the right opportunity to also revise the usage of CLHEP in the Physics Event Model (and also an incent to totally abandon CLHEP in LHCb software). The merger of SEAL and ROOT is supposed to provide a new library of vectors and linear algebra (matrices) that overcome the known CLHEP problems (see below). There is a test version of the vector classes available, while the liear algebra matrices are part of the ROOT TObject classes for quite some time.

We have tried to use these classes in the new Physics Event Model but have found too many problems.

The contact person far all these issues is Juan Palacios.

MathCore Vector Classes

- MathCore doxygen

Although they have a little complicated structure (many templates and typedefs) which makes it sometimes difficult to know which class one is actually using, the MathCore Vector Classes are quite well designed. Also the classes have sometimes silly names (XYZTVector for instance), which we suggest to hide behind an additional layer of typedefs.

There are a few small problems that can be easily fixed:

- We found an inconsistency in the behaviour of the typedefs for Vector3Ds and LorentzVectors. If Ione includes MathCore/Vector3D.h in the code, this includes MathCore/Vector3Dfwd.h, so one gets the typedefs XYZVector, XYZVectorF, etc. However, if one includes MathCore/LorentzVector.h, one does not get access to the typedefs as this file does not include MathCore/LorentzVectorfwd.h.
- A XYZTPoint is missing. That would be nice to have for MC truth.
- There is an Eta() method in the LorentzVector, but no Gamma() and Beta().

Root Linear Algebra Classes

- ROOT 4.04 doxygen

- All classes are TObjects, which pulls the whole of ROOT into our event model. Is that necessary?
- Internally there are exported typedefs like Assert which overwrites the Assert method in GaudiTool.
- We would like to be able to get a sub-matrix by reference. This is presently implemented, but the interface of the method is quite strange.
- The linear algebra classes seem not to be compatible with the mathcore classes. There is no way to multiply a matrix and a vector. (?)

Use case for linear algebra package presented to SEAL/ROOT group

The minimal algebra package is to be optimised for small (typically < 10) size matrices and vectors. In what follows, is is assumed that the software must be fast and efficient. This use-case has been compiled from communications between Patrick K., Vanya, Matt N., Juan and Lorenzo Moneta.

Main requirements

- Square matrices up to at least 7x7 dimension
- Row and column vectors up to 7x1
- Matrix inversion with defined failure indication
- Calculation of matrix determinant
- Basic matrix operations: *, +, +=, -, transpose
- Specialisations for the case of symmetric matrices and their operations
    - Important: symmetric matrix similarity transformation specialisation: M*S*M^-1 is faster for symmetric matrices S and otrhogonal matrix M than for standard ones. This is used extensively in the track fit.
- Easy construction of identity and zero matrices
- Assignment without temporaries (generaliseable to = operator?)
- Access to individual elements
- Access to size (rows, columns)
- Access by reference to rows and columns
- Access by reference to matrix sub-blocks (with clear syntax!)
    - Diagonal sub-blocks of symmetric matrices returned as symmetric matrices
    - Off-diagonal sub-blocks returned as standard matrices
- Easy coupling to MathCore 3D and 4D vectors and points (with compile time dimentionality check?)
    - Multiplication operations
    - Vectors to matrix columns or rows
    - Matrix columns or rows to vectors

Other requirements

These are requirements where speed might not be of the escence, since they relate to operations not needed in typical reconstruction algorithms

- Evaluation of eigenvalues and eigenvectors
- Decompositions
- Conversion to GSL matrices? This would allow to use the GSL eigenvalues and decompositions for free.

Open questions

Which of the operations above should be non-mutating? Should inverse and transpose return a new matrix, or perform the operation on itself?

-- JuanPalacios - 04 Oct 2005

What do we expect from a Linear Algebra package?

First brain dump by PK at the LHCb Software week (slides) summarized by Juan:

- Package must deal with matrixes of dimension up to 7x7
    - Vanya: I would reformulate such as "package must deal efficiently with matrices of dimension upto 7x7". Also one need to indicate what operations we need. In addiiton to "obvious" and "natural" operations we definitely needs :
        - ◊ efficient matrix inversion (with the proper and convinient way to indicate the inversion errors)
        - ◊ efficient evaluation of eigenvalues and eigenvectors. I do not know if we needs other operations, like numerouse decomposition (e.g. LU,...etc)

- Package must allow for efficient retrieval of arbitrary submatrices with clear, easy to use semantics.
  - Vanya's comments:
    - For symmetrix matrices one needs to get access BY REFERENCE to all diagonal sub-blocks as 'diagonal' submatrices and for off-diagonal blocks BY REFERENCE as 'general' matrices. In general we do not require access to ARBITRARY subblock.
    - For 'general matrix one also does not need access to ARBITRARY sub-blok. Only the diagonal and off-diagonal block are really needed for majority of applications.
    - The non-diagonal blocks, which containes at least 2 diagonal elements usually are out of interest for all cases.
- Must have easy coupling to MathCore vector and point classes (obviously).

"Obvious requirements" by Matt for tracking:

- Matrices and Symmetric matrices
- Basic matrix algebra functionality: *, + , += ,-, transpose
- Easy construction of identity and zero matrices
- CLHEP had an assign function, so you could do

```
tC.assign((HepDiagMatrix(tC.num_row(), 1) - (m_K * m_H.T()))*tC);
```

- Matrix inversion (maybe copy the dsinv from cernlib as done by root and CLHEP ?). Failure of the inversion should be clearly indicated.
- Access to individual elements
- CLHEP provides a similarity function m1*s*m1.T() for symmetric matrices. We use this in the fit alot. I was once told that the reason that this is there is because this function is optimized for speed
- Access the size (number of rows and columns)

-- PatrickKoppenburg - 30 Sep 2005

Links

- LHCbPhysicsEventModelTaskForce
- MathCore doxygen
- ROOT dogygen
- CLHEP dogygen
- Physics Event Model Proposal with new classes

Appendix

CLHEP problems

- CLHEP dogygen

A brief reminder of the problems with CLHEP:

- No constructor from Point to Vector. A point minus a point is a point (!). The following code calculating an impact parameter does not compile anymore:

```
HepPoint3D P(0,0,0), B(1,2,3);
Hep3Vector V(0.8,0.6,0.0);
Hep3Vector D = B - A;                                          // does not compile with
double IP = (D.cross(V.unit()).mag();
```

One needs to:

```
HepPoint3D TMP = B - A ;                                          // point - point is point (!?)
Hep3Vector D(TMP.x(),TMP.y(),TMP.z()) ;  // object-oriented programming ?
```

CRJ : Note, this does work if you use HepVector3D instead of Hep3Vector

```
HepPoint3D P(0,0,0), B(1,2,3);
HepVector3D V(0.8,0.6,0.0);
HepVector3D D = B - A;                                                  // does work
double IP = (D.cross(V.unit()).mag();
```

I think the confusion is "why are there 2 versions of 3-vectors in CLHEP ?" -- ChrisRJones - 20 Sep 2005

- Because of a bug of the new CLHEP 1.9, whenever one includes `HepSymMatrix.h` *after* `*Point*.h`, one does not get the method HepSymMatrix sub(int min_row, int max_row); which results in the run-trime error

```
relocation error: .../slc3_ia32_gcc323/lib*.so: undefined symbol:
```

Conclusion: Include `SymMatrix` at the very beginning of the file.

- The internal representation of a HepLorentzVector is (momentum, energy) which causes precision problems for photons for instance.

-- PatrickKoppenburg - 09 Sep 2005

This topic: LHCb > MathCore
Topic revision: r7 - 2005-10-18 - PatrickSKoppenburg