

CORSIKA8 development - Status report

Source code re-factory

A. Augusto Alves Jr

Presented at CORSIKA development meeting - KIT, Karlsruhe

May 5, 2020



- Impressions from the first source code assessment
- Re-factory
- Status and file tree structure
- Comments

Source code assessment

- Source tree hierarchy not very clear
- Many different types of files mixed in sub-directories: FORTRAN, C, C++, Python.
- Some files containing two classes:
`COASTStack.hpp`, `CombinedStack.hpp`, `StackIteratorInterface.hpp`
- Not clear separation between interface and implementation details.
- Files including other files inside `namespace` declarations: `ParticleProperties.hpp`
- Some methods and classes are implemented in `.cc` files, but the absolute majority is implemented “in-place”.
- Test-cases not in dedicated directories
- Many other minor inconsistencies

High level structure assessment

- Many classes are not “first class”. Many missing copy/move constructors and assignment operators.
- Is RAII enforced?
- Design patterns seems mixed or partially implemented. For example, many classes have a non-default constructors and a `Init(...)` method.
- Not clear who manages the resource pointed by `CoordinateSystem const* cs` in `BaseVector.hpp`
- Iterator idiom not fully implemented in `Stack.hpp`
- Many other minor inconsistencies

- Project sitting between a framework and library.
- C++ part is effectively header-only.
- FORTRAN part will probably require some building step.
- The source tree structure and overall code stack difficult to maintain at long term due absence of idioms, patterns so on.

Re-factory: first round

Focused on the source tree structure and aiming to enforce CORSIKA as a framework. Status: mostly done.

- All `C++` headers have extension `.hpp` now. **Rationale:** to make clear it is `C++` stuff.
- Headers containing only declarations now. DOXYGEN documentation should be stored in the headers. **Rationale:** expose the public interface together with its documentation.
- Implementations moved to `.inl` files, which are post included. No DOXYGEN documentation there, only normal comments. **Rationale:** hide implementation details.
- Eliminate include guards and use `\#pragma once`. **Rationale:** faster compilation for header-only, self-referencing projects. Easier to maintain at long-term.
- All test cases moved into a dedicated `tests` folder. Tests should be built on demand.
- Eigen and some other dependencies eliminated. **Rationale:** These projects are consolidated and are present in most of the distributions.

Re-factory: first round

Pending or incomplete tasks:

- Define `namespace xyz::detail` and move some code into it.
- Get ride of some namespaces and folders and mirror the directories hierarchy at the namespace hierarchy.
- Split and rename some files to mirror keep representing one `class` per file.
- Make all classes “first class”.
- Analyze all classes for resource management.

A glimpse into the current structure

```
Arquivo  Editar  Exibir  Pesquisar  Terminal  Ajuda
[augalves@LabHome corsika] $ tree --dirsfirst -A
.
├── cmake
│   ├── CodeCoverage
│   ├── CodeCoverage.cmake
│   ├── CorsikaUtilities.cmake
│   └── FindPythia8.cmake
├── corsika
│   ├── coast
│   │   ├── COASTProcess.hpp
│   │   ├── COASTStack.hpp
│   │   └── ParticleConversion.hpp
│   ├── framework
│   │   ├── core
│   │   │   ├── detail
│   │   │   │   └── Cascade.inl
│   │   │   ├── Cascade.hpp
│   │   │   ├── ParticleProperties.hpp
│   │   │   ├── PhysicalConstants.hpp
│   │   │   └── PhysicalUnits.hpp
│   │   ├── geometry
│   │   │   ├── detail
│   │   │   │   ├── BaseVector.inl
│   │   │   │   ├── CoordinateSystem.inl
│   │   │   │   ├── FourVector.inl
│   │   │   │   ├── Helix.inl
│   │   │   │   ├── Line.inl
│   │   │   │   ├── Plane.inl
│   │   │   │   ├── Point.inl
│   │   │   │   ├── QuantityVector.inl
│   │   │   │   ├── Sphere.inl
│   │   │   │   ├── Trajectory.inl
│   │   │   │   └── Vector.inl
│   │   │   ├── BaseVector.hpp
│   │   │   ├── CoordinateSystem.hpp
│   │   │   ├── FourVector.hpp
│   │   │   ├── Helix.hpp
│   │   │   ├── Line.hpp
│   │   │   └── Plane.hpp
```


- You can watch my work at: <https://gitlab.ikp.kit.edu/AAAlvesJr/corsika>
- Pending:
 - Prune unnecessary files.
 - Reflect about the fate of the FORTRAN files and related building process.
 - Update `CMAKE` scripts.
- CORSIKA8 should be distributed as a header-only framework.
- Tests and examples: built on demand.