Status of Merlin++ and 5.02 release

Sam Tygier*, Rob Appleby*
Scott Rowan+, Roger Barlow+

*Manchester Cockcroft Accelerator Group, UK
*Cockcroft Institute, Daresbury, UK
+University of Huddersfield, UK

29 April 2019

sam.tygier@manchester.ac.uk
Merlin++

- Accelerator simulation library
- Modular and extensible C++
  - Class hierarchy:
    - Beam line elements: Drift, Dipole, Quad ... 
    - Bunches: Electron, Proton ... 
    - Physics processes: Trackers, Collimation, Synchrotron radiation ...
  - easy to add new physics, tracking, element types
- 39,000 Lines of code (+5500 of examples)
- Git revision control (github.com/MERLIN-Collaboration)
- Test suite, nightly tests of range of systems
- New release 5.02 in March
Features

- Standard range accelerator components
- Multiple tracking integrators
- Optics measurements
- Collimation
  - Online aperture checking
  - Advanced proton scattering physics
- Synchrotron radiation
- Hollow electron lens
- MADX lattice import
LHC Squeeze simulations

- Validation for TCT losses during squeeze
  - Run 1 and 2
  - Range of steps in optics
  - Comparison with BLM data
- Predictions for HL-LHC
- Publication:
  - Performance of the Large Hadron Collider cleaning system during the squeeze: simulations and measurements - 10.1103/PhysRevAccelBeams.22.023001
What new in 5.02

● Main focuses:
  – Sustainability
  – Refactoring
  – Cleanup
  – Merging
  – Documentation

● Changes to:
  – Apertures, MADInterface, Bunches, RNG, DataTable, LatticeFunctions, Tests, Tutorials

● Plus:
  – Bug fixes, memory leaks, code formatting, warnings ...

● New website

● DOI: 10.5281/zenodo.2598429
Software sustainability

- We often focus on using a piece of software right now
- Sustainability is about how software will be used in the future

https://software.ac.uk/

- **Community**: There is a community infrastructure with a common investment (required for sustainability)
- **Open**: Software has permissive license (required for modification)
- **Defined**: Accurate metadata defines the software and its functionality, dependencies and constraints (required for preservation)
- **Extensible**: The software is usable, modifiable for different data, pipelines, purposes (required for reproducibility)
- **Runnable**: The software is available and provides the information to operate it (required for publication)

- Sustainability of the Merlin++ particle tracking code – CHEP18
Documentation and tutorials

- New website
  - www.accelerators.manchester.ac.uk/merlin/

- Quickstart guide
  - Installation
  - IDE setup
  - Tutorials

- Doxygen
  - www.accelerators.manchester.ac.uk/merlin/doxygen
  - Full library API

- Examples – distributed with source code
Refactoring

- Reducing complexity
  - Deeply nested branching
  - Long methods
- Reducing propagation cost
  - Additional changes required by amendments
- ‘Code smells’
  - Duplicated code, large switch statements …
- Find worst hotspots to target first
  - Metriculator, ArchDia DV8, Valgrind
- Performance
  - Often cleaner design helps
  - Sometimes need to compromise
Bunch construction

• Previously
  – one large method in ParticleBunchConstructor() - over 200 lines
  – Any new bunch type need to be added to this core code

• Now
  – ParticleBunchConstructor class removed, implementation in ParticleBunch
  – Class for each distribution generator
  – New distributions can be in user or core code
  – Just needs to define method to produce a single particle GenerateFromDistribution()
DataTable

- Class to hold table of data with named columns of mixed types
  - Designed with TFS files in mind
  - Input and output to TFS (other formats can easily be added)
- Replaces TFS parsers in several places in Merlin++
  - Reduce duplication
  - Separate parsing and usage
  - More robust error handling
Random Numbers

• Used:
  - Bunch generation, scattering, element misalignment, ...

• Merlin used to contain a random number implementation

• C++11 standard library now include RNGs
  - Strong Mersenne Twister algorithm
  - Easy to define new distributions

• Switching Merlin++ to stdlib
  - Removed many lines of code
  - Allow multiple random streams
  - Smarter seeding
  - Easier to switch generator in future
Future plans

- Continue making improvements
  - Documentation, code quality, performance
- C++14 and C++17
  - Allows further clean-ups, parallel STL algorithms
- Performance
  - Currently similar to sixtrack
  - Improvements to vectorisation, intra-node concurrency, multi-threading
  - removing some known inefficiencies
- Ion tracking
  - WIP
- Publish Merlin++ paper
  - In progress – Computer Physics Communications
References

- Recent publications

- [www.accelerators.manchester.ac.uk/merlin/](http://www.accelerators.manchester.ac.uk/merlin/)
- [github.com/Merlin-Collaboration/Merlin](http://github.com/Merlin-Collaboration/Merlin)