Status update of IRT 2.1 code adaptation to ePIC software stack

Alexander Kiselev

ePIC Collaboration Meeting @ BNL, 01/23/2026

Progress since the JLab meeting

- IRT 2.0 -> … -> 2.1c; quite some effort over August-October 2025 time frame
  - Codes “re-based” against *EDM4eic*, *epic* and *EICrecon* repositories by hand (twice)
  - Shown to work with recent Docker images

- Generalization in all places - *no pfRICH/dRICH/{QBF}RICH specifics*
  - ACTS tracking interface has been re-worked, in particular

- Functionality without a standalone .C script at the end of the chain demonstrated
  - IRT reconstruction engine called within a generic EICrecon *IrtInterface* plugin, …
  - … with a fully functional detector-specific JSON config parser

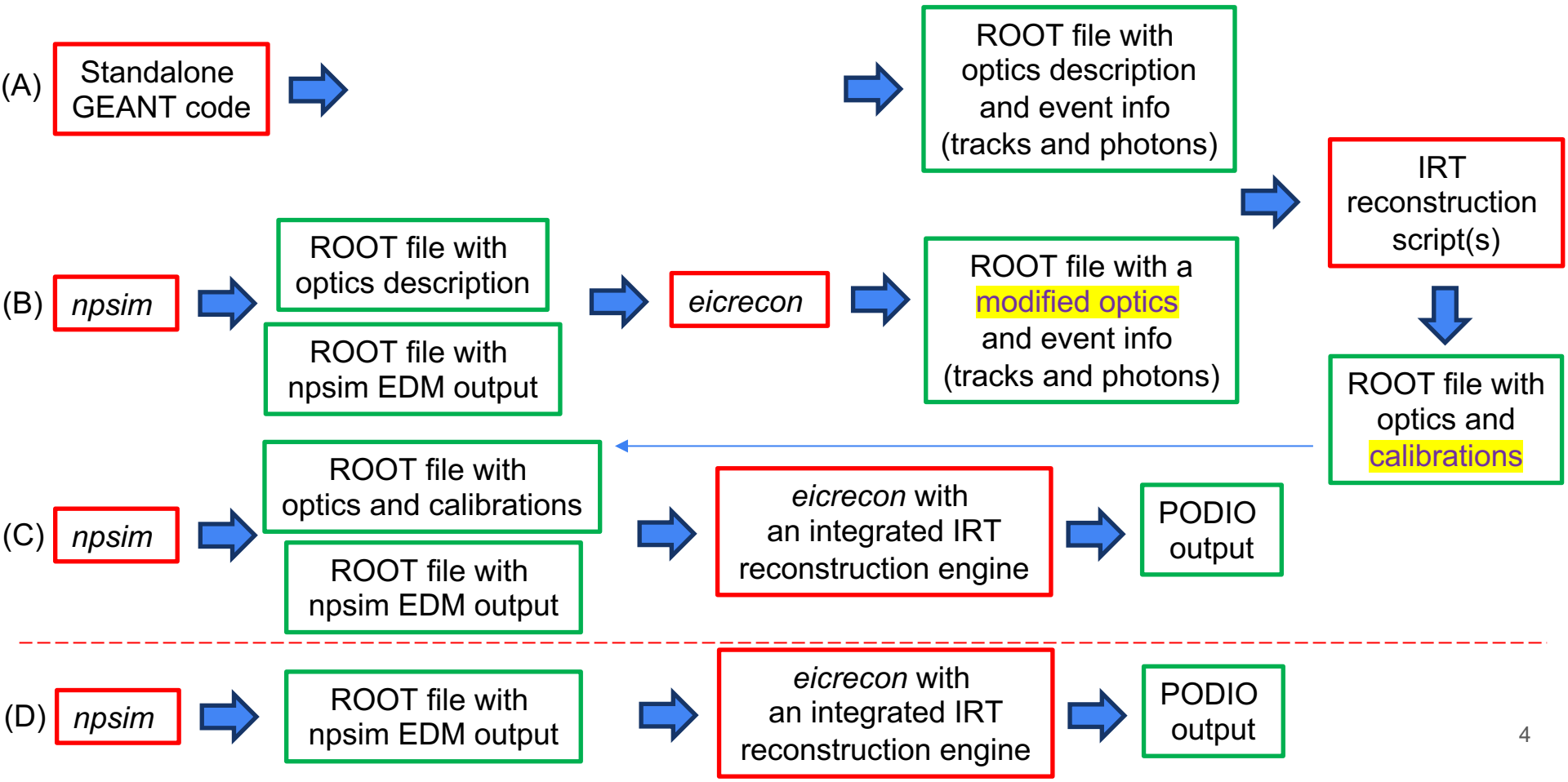
- IRT engine output is propagated to the respective PODIO tables

Progress since the JLab meeting

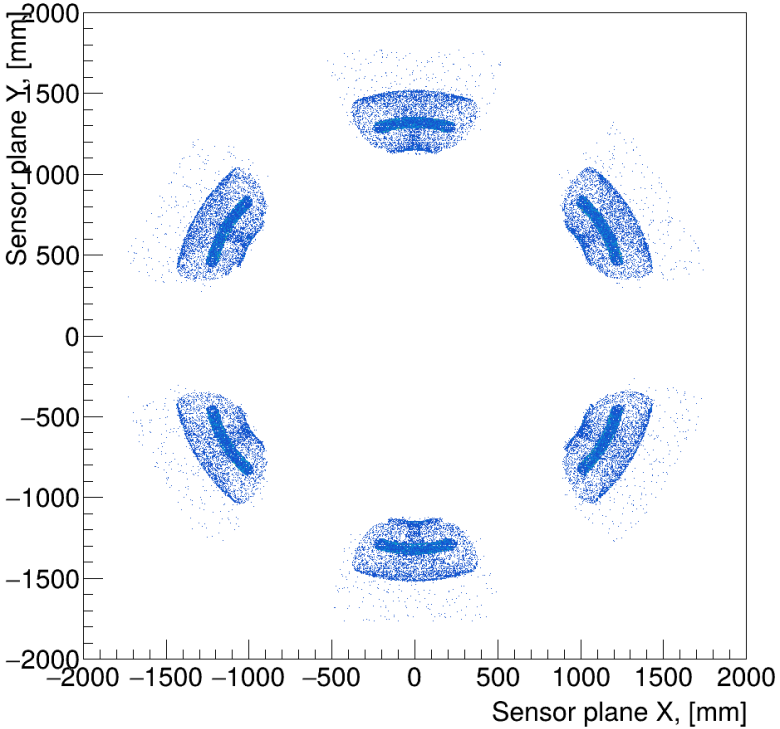
- Optics ROOT file exchange between *npsim* & *eicrecon* “eliminated”
  - All what is needed is created on-the-fly upon *eicrecon* startup
- Calibration data made importable as a 2D histogram in a ROOT file
  - Caveat: its creation requires running a ROOT .C script on a non-PODIO *eicrecon* output

- dRICH geometry adapted to IRT2 optics description
  - Was a very easy exercise based on Chris’s work (thanks!)
  - A unified gas+aerogel case confirmed to work (including separated evaluation output)
- pfRICH geometry followed recently
  - Based on Bill’s work (thanks!), with some complications
  - Pyramid mirrors and aerogel delimiters not yet backported
  - A cleanup and a thorough material cross-check pending

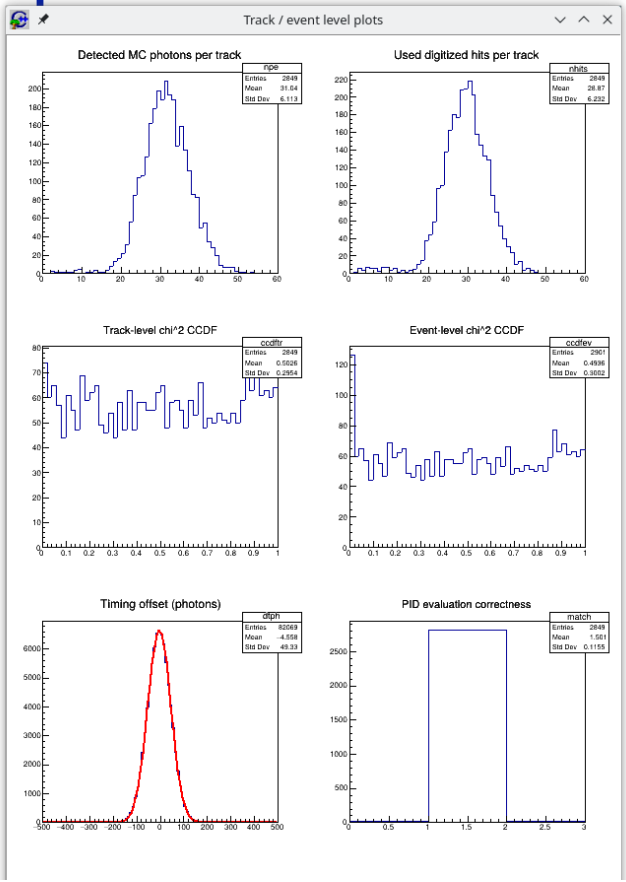
IRT-2.0(1) sandbox setup evolution



dRICH IRT 2.1 reconstruction example

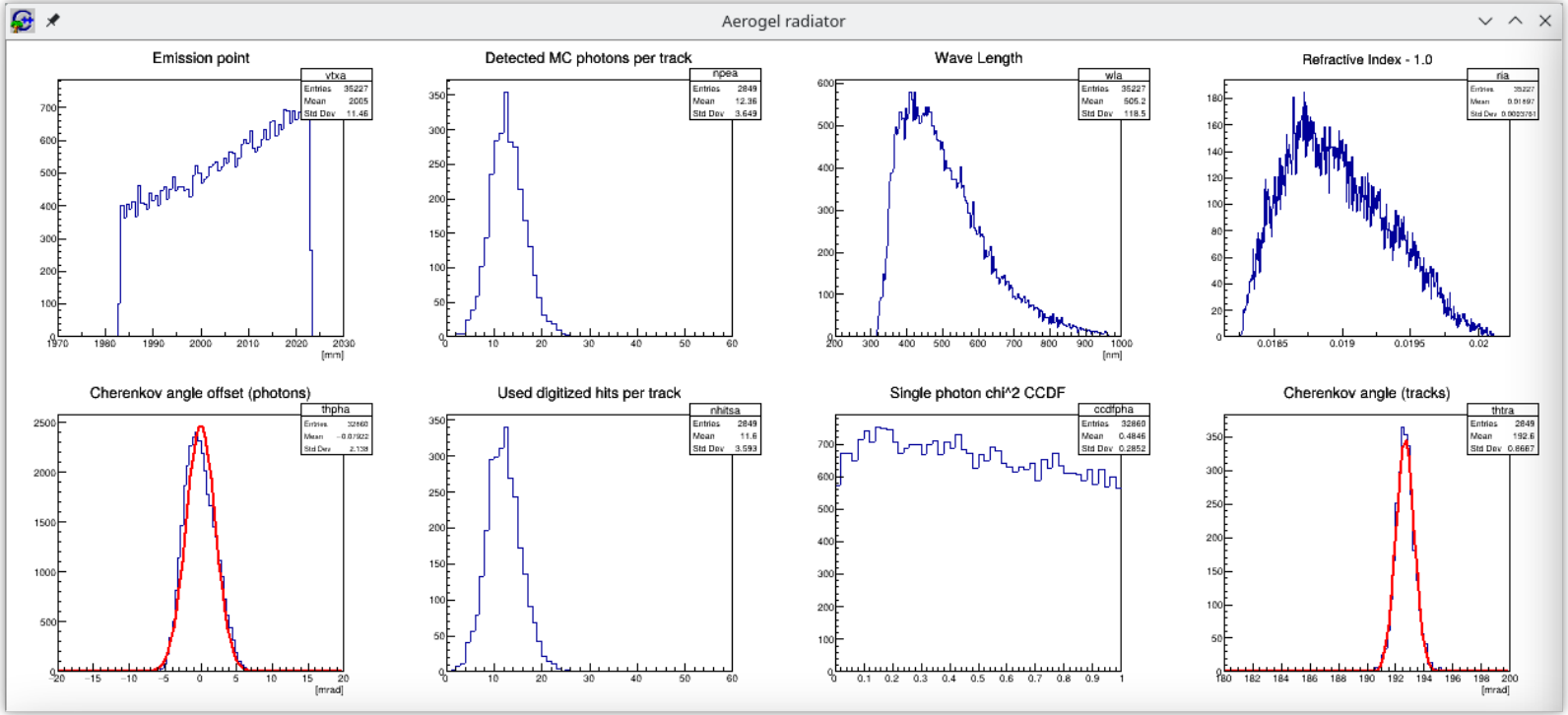


Digitized hit map (10 GeV/c pions, η=2.0)



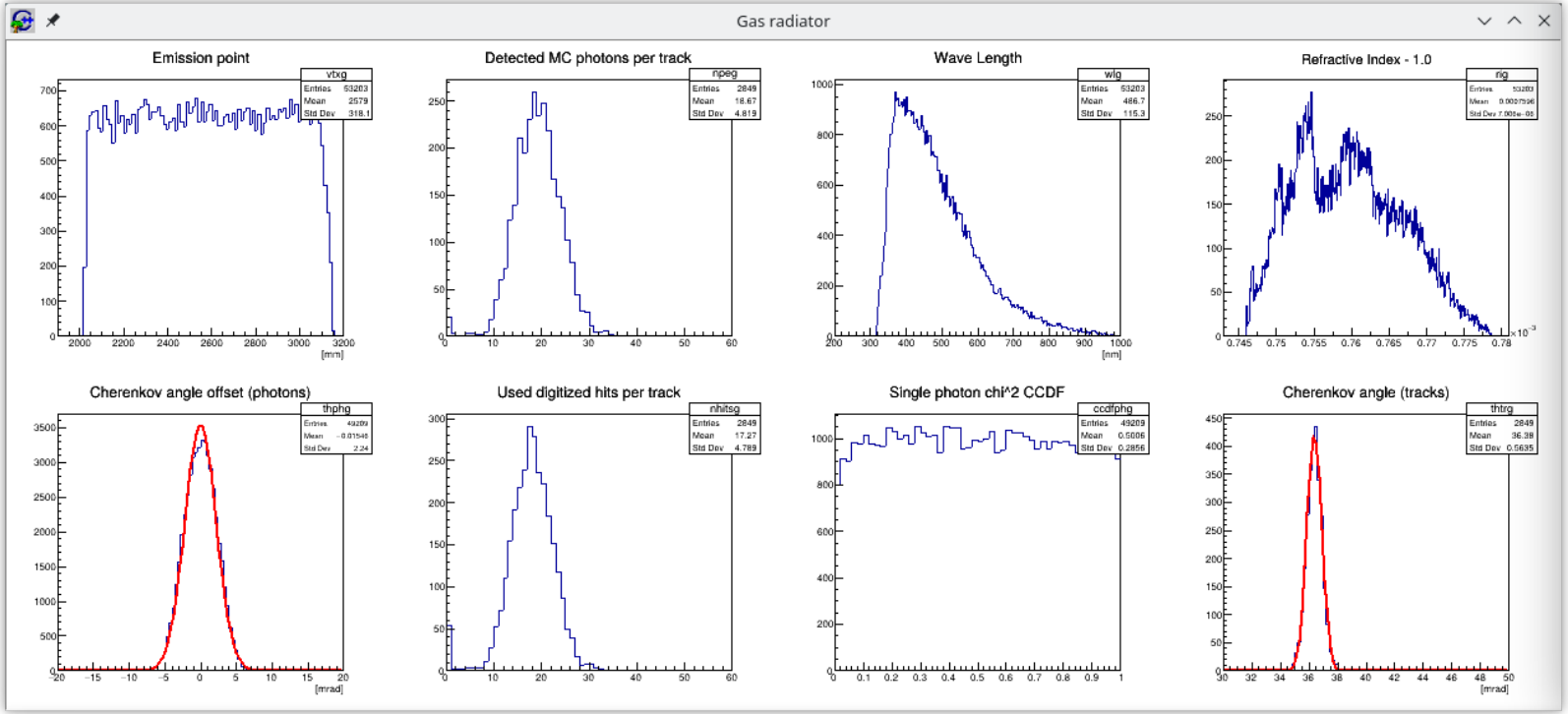
Combined IRT pass plots

Aerogel radiator evaluation plots



Like in a standalone IRT use case,
these plots can now be seen at the end of *eicrecon* pass

Gas radiator evaluation plots



A unified aerogel/gas/tof PID engine, yet with a per-radiator stat info available

Summary & Outlook

- Porting & coding exercise per se is converging to a “final product” state
  - With intrinsic limitations caused a combinatorial nature of the IRT engine

- Remaining implementation topics (as seen by AK)
  - Digitization needs to be extracted from the IRT engine itself into a separate step
  - Some adjustments in CMakeLists.txt files are still needed
  - Fine details in pfRICH & dRICH material & optics description missing
  - Calibrations & configuration may be made more sophisticated

- Debugging, case studies, limitations
- PODIO output handling & benchmarks
- Ongoing integration into the main branches