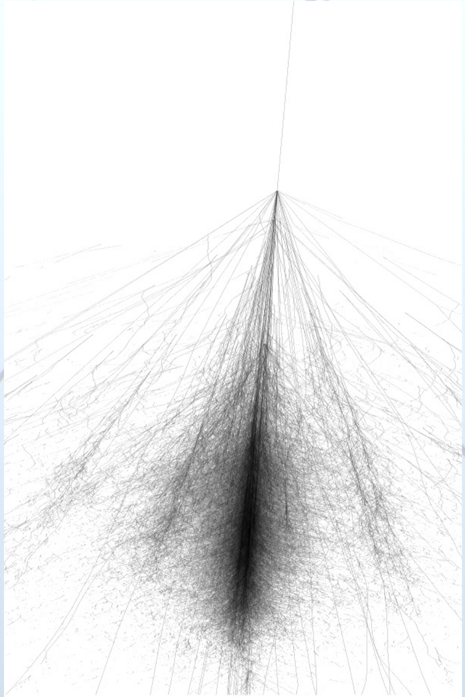
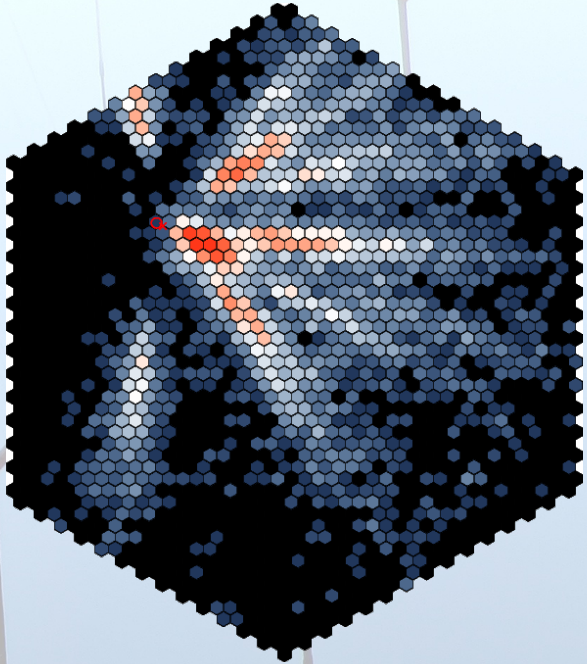
A rather personal view on

Needs for Cherenkov Light in CORSIKA 8





K. Bernlöhr

Max-Planck-Institut für Kernphysik

CORSIKA 8 Air-Shower Simulation and Development Workshop, 2022-07-13

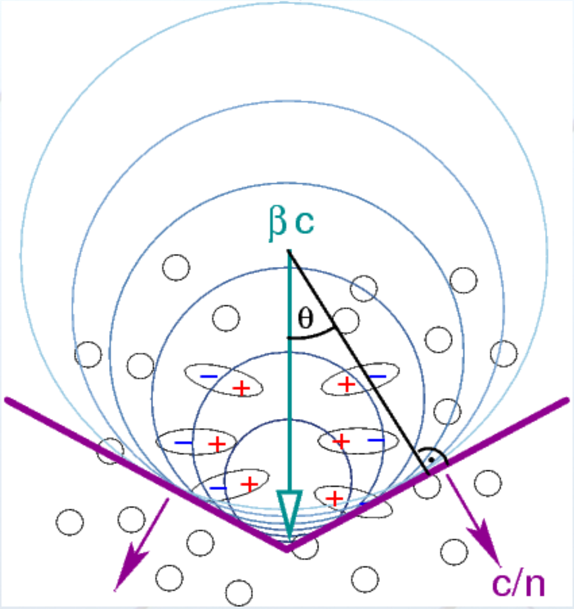
Why personal?

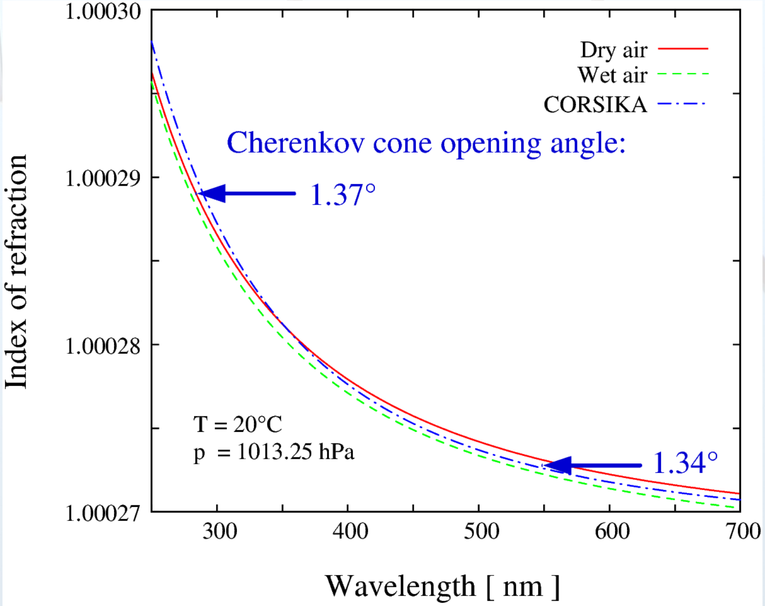
- Speaking as the (perhaps) main author of the current C7 Cherenkov code, in particular in view of IACTs.
- No time to adjust to CTAO/CTAC/HESS/... presentation/publication policies since agreeing on giving this talk.
  - Thus staying out of experiment-/observatory-specific details.

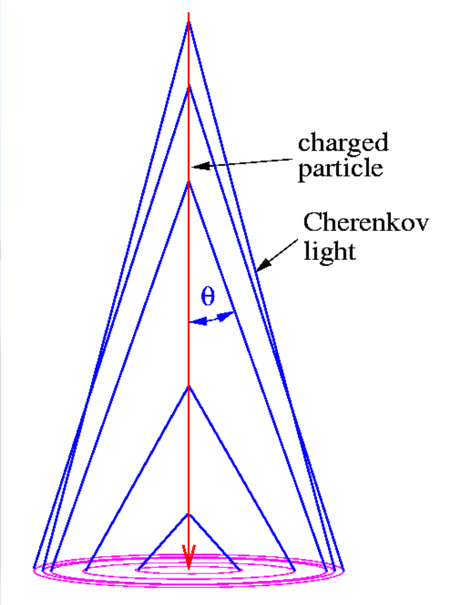
What is special about Cherenkov?

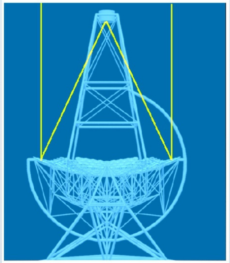
- Emission focused on a cone.
- Emission angle depends on particle velocity and index of refraction (density, water vapor content, ...).
- IACTs with high angular resolution (typ. pixel 0.06° to 0.2°, sources to be located within arc seconds).
- Therefore, small changes in particle direction and in emission angle must be followed (smaller steps).
- Detectors can trigger on a few photons, perhaps by a single particle. Need to keep artificial fluctuations low.

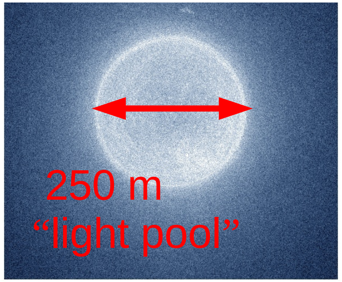
Cherenkov light



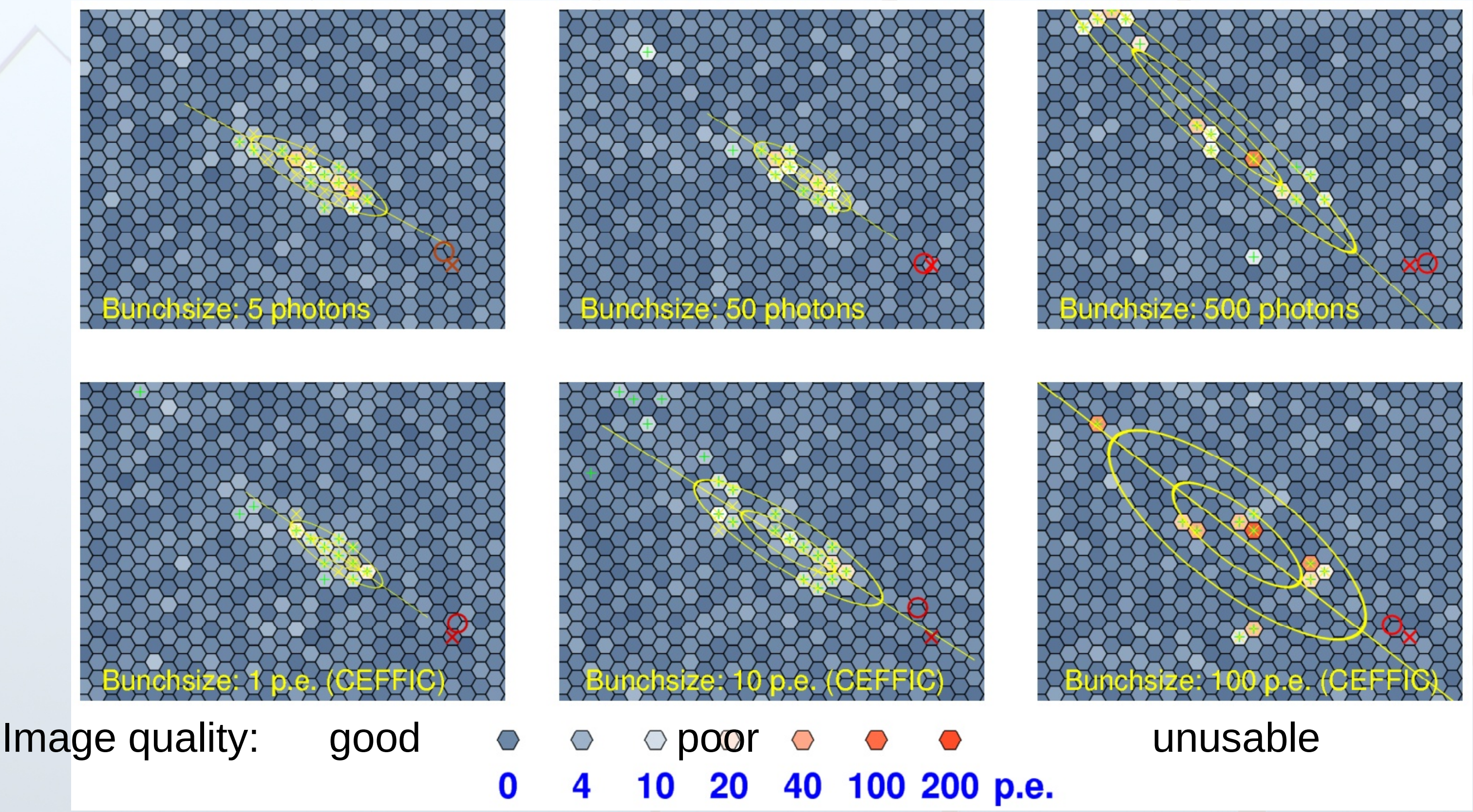








Artificial fluctuations from bunch size



Shown:
Part of the focal plane of an IACT for the same event.

Note that thinning in CORSIKA also increases the effective bunch size. Reducing the step size for 'thinned' particles would be counter-productive.

A brief outline of an implementation

Input: particle info, state at start and end positions

- particle info includes charge numb., mass/ID, ...,
- states include position and velocity/4-mom. vectors, and time.

- Check for Cherenkov emission at start, middle, end of the provided track segment. $n\beta > 1$?
- Decide on wavelength strategy:
  - W.l. dependent more complex, needs smaller bunch size (more CPU time),
  - Cherenkov emission may be possible at one wavelength but not at another.
- Decide on sub-step strategy:
  - Little change in $n\beta$: Simple loop
  - Minor change: interpolation?
  - Too much changes: sub-step by sub-step eval.
    Imagine some worst case to be covered, e.g.
    - Atmosphere too thin at start
    - Density increasing, emission initially only in UV
    - Particle slows down, eventually no emission.

For each sub-step along the track segment:

- Random photon “bunch” emitted on cone around the current velocity vector. Transform to obs. coord. syst.
- Straight-line propagation down to observation level:
  - No extinction (handled later in detector sim.)
  - No scattering (little relev. for most IACT obs.)
  - Refraction is handled by a set of correction terms, not by ray-tracing. There is only one initial ray-tracing to set up the correction terms.
    Accuracy: millimeters, picoseconds, milli-arc sec.

- Only photon “bunches” found to hit a detector fiducial sphere get recorded:
  - Detector search speed-up (by grid etc.) if there are many detectors.
  - Additional info about emitting particle may be recorded along with the photon “bunch”.
  - Fluorescence photons can be recorded the same way.
  - Scattered light probably needs extra path info.

Current status in CORSIKA 7

- CERENK subroutine and helpers pretty much stable since the days of CORSIKA 6.9.
  - Supports a range of compile-time and run-time options.
  - Supports several output options/formats.
  - Vectorized implementation for high performance (not supporting all of the extra options but IACT/ATMO).
  - Works well together with an experimental fluorescence implementation by Madrid group. No scattering though.
  - Includes atmospheric refraction correction.
  - Atmospheric extinction left for detector program.

A few notable options in C7

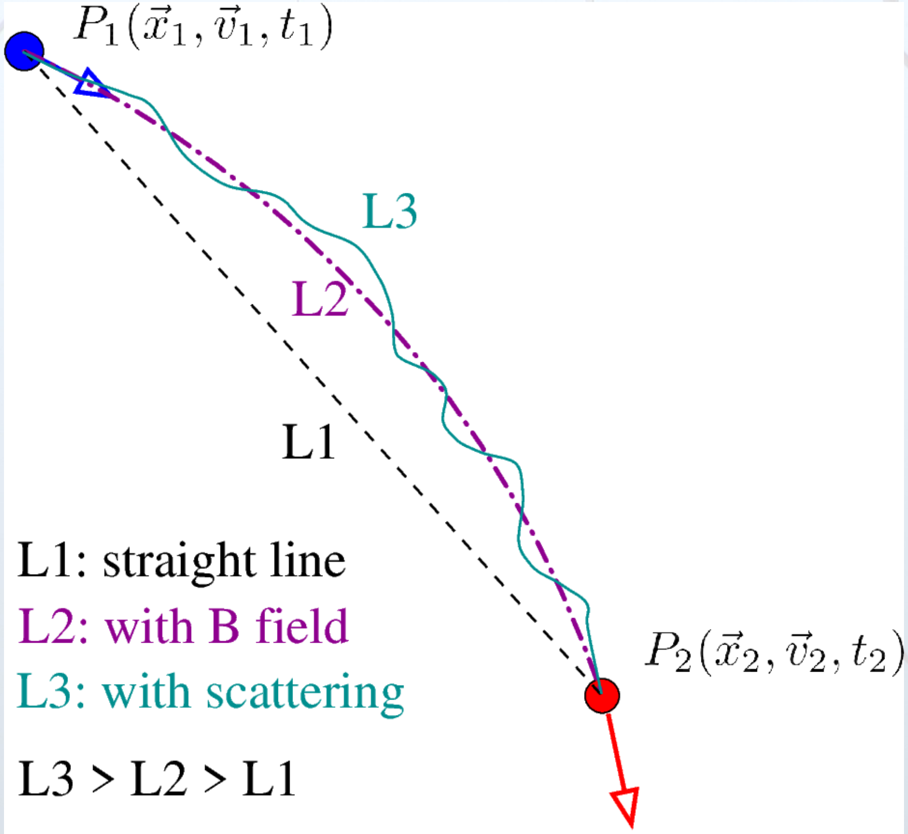
- Compile-time:
  - CERENKOV, **IACT**, **ATMEXT**, CURVED (za>65°), SLANT, THIN.
  - IACTEXT for particle output and extra info on light emitting particles.
  - CERWLEN for wave-length dependent index of refraction.
- Run-time (with IACT/ATMO package, at least):
  - multiple use of showers at random offsets, optionally with energy-dependent importance sampling,
  - tabulated atmospheric density/index of refraction profiles,
  - atmospheric refraction correction to photon arrival dir./pos./time,
  - direct pipe of data stream into (multiple) detector simulation(s), without writing CORSIKA data to disk first.

What we absolutely need

- Step lengths adapted to the angular resolution, ... and obeyed by continuous energy loss, multiple scattering, bending in geomagnetic field.
- The initial and final state of a charged particle in each track segment, including
  - position and time (with look-up of index of refraction as f(h)),
  - velocity vector (or energy-momentum 4-vector),
  - particle charge number,

  and the complete path length in the track segment.

Length of a track segment ?



Straight line between point 1 and point 2 will always underestimate the path length traveled (and thus Cherenkov light emitted). Helix-style path should be available (since this is the path segment calculated) while the path length with scattering can only be known on a statistical basis (see PDG).

If the lengths differ so much that we care, should be use a the more accurate calculation or reduce the step length?

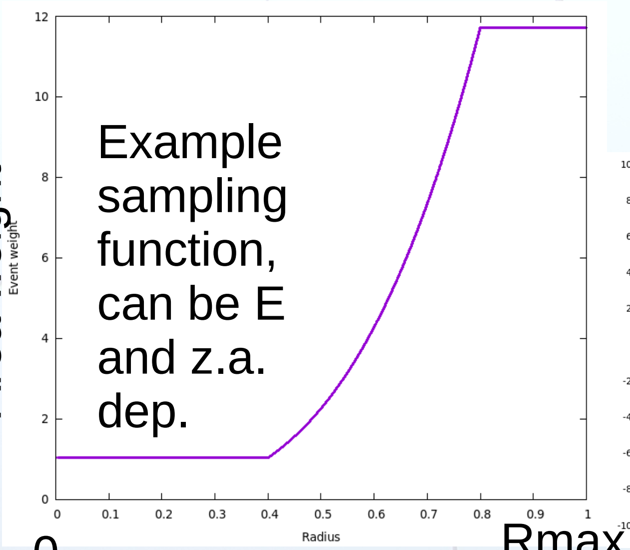
Which one is the Cherenkov emission cone to follow?

What we really want to keep fr. C7

- 3-D detector array as fiducial sphere.
- Multiple shower re-use with random array offsets (optionally importance sampling).
- Complete record of what was simulated – current simulation chain keeps the full set of CORSIKA input cards even beyond the telescope simulation, with record of command lines and all parameters involved.
- Choice(s) between speed-optimized (but good enough) and more sophisticated with additional information.

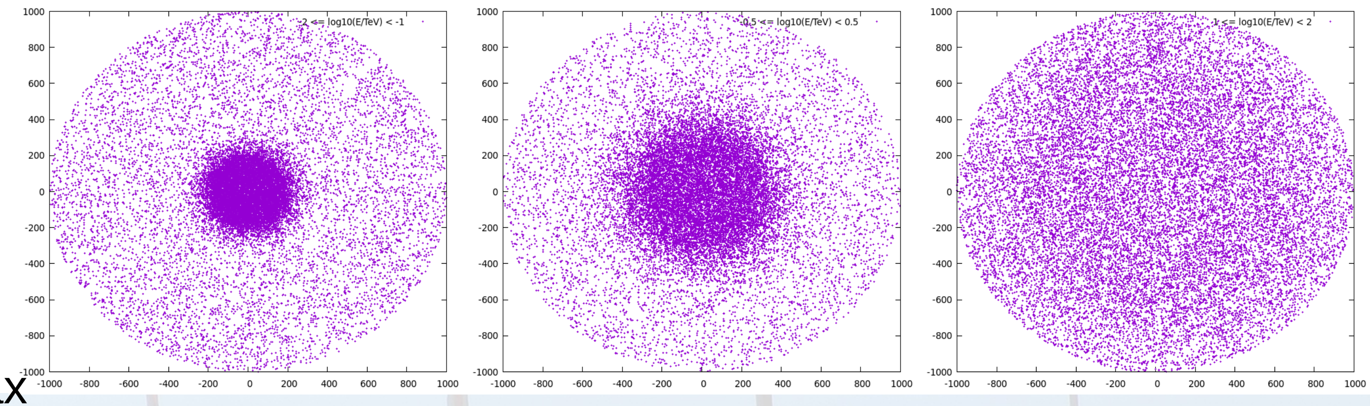
Example energy dependent importance sampling:

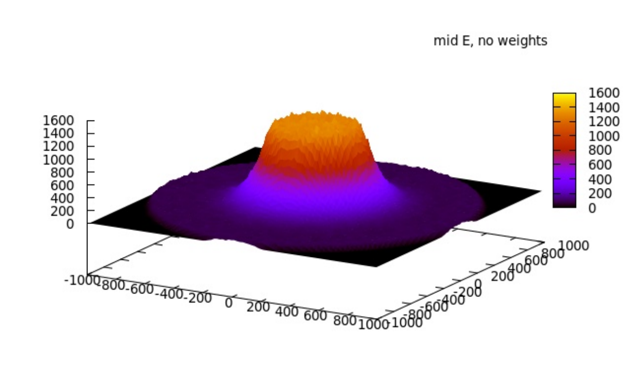
Area weight

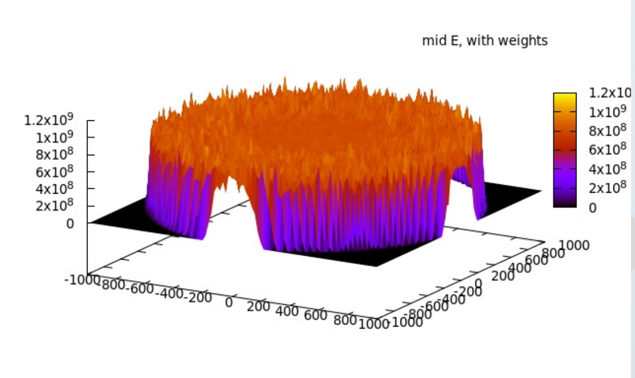


0 Rmax

Low E Medium E High E







Raw core position distribution (medium E)

Distribution after applying “area weight” factor

What we never cared about

- Parallel computing
  - Poor man’s parallelization with one job per core is extremely efficient and the longest shower takes perhaps an hour. For jobs running days to complete, the occasional shower taking up to an hour is no problem – just leave a few hours margin for batch system limits.
  - Thus never made any serious effort in that direction.

What we do not want/need to keep

- Parameterized atmospheric profiles.
  - The current 5-layer limitation is a real source of systematic problems and table interpolation is faster anyway.
- The (equivalent of) ATMEXT and IACTEXT compile-time options. Better always on. With information passed by reference (C8), the run-time penalty of not using the extra features should be negligible (C7 puts data on stack).
- The distinction between IACT mode and non-IACT mode seems obsolete if step lengths get calculated as functions of the required angular resolution.

What would be a real obstacle in fast C8 adoption

- Enforcing an incompatible output format without an easy and efficient way for conversion into existing formats.
- In other words, any experiment/observatory with significant investment in their detector simulation code would prefer to have a C8 mode working as a drop-in replacement for C7 – and verifying that first. Then start adapting for new features in C8.

What we may wish for

- Atmospheric and geomagnetic properties entirely table-based, no further parameterizations.
  - Initially 1-D (height) but eventually 3-D in-homogeneous configurations might be asked for.
  - Geomagnetic field directly derived from IGRF (geomag), time-dependent (3D?).
- In addition to extinction losses (now part of the detector simulation), an efficient implementation of scattered Cherenkov light, complementing fluorescence emission, would be welcome, although not much of an issue for most IACT use cases.

What we can only hope for

- That CORSIKA 8 will become the robust and reliable work horse for air-shower simulations in the coming decades, like the FORTRAN implementation has been for more than 30 years.
- Given that I was taking care of (most of) the Cherenkov part in C4/5/6/7 over the last 25 years, I hope we can find the developers who – after the initial C8 Cherenkov implementation – will take care of the maintenance and fine-tuning for the next 25 years.