Notes from the CORSIKA Workshop

June 25./26. 2018 at KIT
Indico: https://indico.scc.kit.edu/event/426
Participants: Alex Olivas, Annarita Margiotta, Blahoslav Pastircak, Darko Veberic, David Sierra-Porta, Dieter Heck, Dominik Baack, Doris Wochele, Francesca Bisconti, Gernot Maier, Ghazala LAKEL, Hans Dembinski, Isabel Astrid Goos, Jakob van Santen, Jiri Blazek, Johan Bregeon, Johannes Hulsman, Johannes Knapp, Luigi Antonio Fusco, Luisa Arrabito, Malek Mastafa, Markus Roth, Martin Alfonso Subieta Vasquez, Maximilian Reininghaus, Michael Schmelling, Michael Unger, Miroslav Kulifaj, Muhammad Ajaz, Nepomuk Otte, Pantea Davoudifar, Paolo Desiati, Pierpaolo Savina, Piotr Mijakowski, Ralf Ulrich, Ralph Engel, Shahid Ullah, Tanguy Pierog, Tim Huege, Tobias Winchen, Yanis Fontenla Barba

Thank you very much to all participant for the large amount of information, and the lively discussions. Here is a summary of the presentations, and in of topics that were raised in the related discussions. Please write me if there are further relevant points to be added.

General input from the community and needs of experiments

In general there are three major tasks for CORSIKA in the next many years:

1. Shower production for gamma-ray and neutrino experiments down to very very low primary energies. The number of events will be very large, the size per event very small.
2. Shower production at ultra-high energies for ground and space based experiments. Single showers can require enormous computing resources, and huge disk space.
3. Development, physics interpretation, event reconstruction algorithms, etc.: flexibility, modularity, fast development cycles.

It is a challenge to suit these different requirements, and will require a very flexible software package. Since shower production on major computing facilities is also a big cost factor for future collaborations and experiments, efficient code is as important as accuracy and flexibility. From the very beginning, computing efficiency will be checked and enters into the programming, but not on the cost of the usefulness of the code. This requires continuing discussions, and regular re-thinking of basic concepts.

CTA presented that already in 2018 (January to June) they produced about 120M (HS06) CPU hours of simulation on the grid, of which 70% is CORSIKA. Even small improvements in efficiency on the order of 10% will make a big impact. Dedicated groups are looking in detail for such improvements.

In the gamma ray community, accuracy of electromagnetic showers and Cherenkov photon simulations are key parameters. The goals of CTA are to reduce modelling uncertainty from now ~5% to 1..2%. CTA will be an open observatory with individual observation requests on the order of ~hours. CORSIKA will be used to perform run-based shower production for the signal region on a massive scale.

For all gamma ray experiments (CTA, Hess, Magic, FACT) there is a need to incorporate a better and more fine grained Cherenkov photon emission model, and in particular more powerful atmospheric models. Photon transport should include absorption and scattering, next to the refractive effects. To increase efficiency, also a basic detector implementation is needed to reduce the number of photons to actually write to disk. Such simulations will benefit tremendously from GPU acceleration.

The same Cherenkov physics will also benefit the neutrino telescopes (IceCube, Antares, KM3net, Baikal), where Cherenkov emission from single particles and small showers in deep water or ice, needs to be tracked through the medium to the detector. There could be a general large benefit, if the Cherenkov tracking code could be made available in the best possible implementation as part of CORSIKA for the whole community.

At least for the gamma-ray experiments, the Cherenkov photon tracking through the atmosphere to the detectors are crucial parts of the shower simulation and cannot be moved to the “detector simulation” without losing precision on a critical level.

The atmosphere cannot be handled in exponential layers as done so far. Effects of this are already now limiting the simulation precision on the level of ~5%. We need a better approach, that suits all communities. Also radio simulations are much affected by this. Options are simple tabulated atmospheres, or potentially spline-interpolated version of this.

More performance

It was demonstrated that in the current CORSIKA, vectorization is partly used or can be further added with little effort. Speedup by compiler optimization: in >3000 tested combination of compiler option combinations did never exceed ~1.06.
Optimizing the atmosphere by removing calls of the “exp” function has the potential to speedup by a factor of 1.15. Splines can be even better, however, there was a question if this is always true wrt. to tables, or just in some cases.
Speedup by explicitly vectorization of “exp” in the Cherenkov part of CORSIKA leads to a speedup of 1.16.
May consider precision reduction, where precision is not critical, i.e. double → float. But this is very dangerous to do by hand. Things can change with time, and it is very easy to make things worse. Maybe automatic analysis code for this purpose? But it is also not clear what is the actual performance benefit; besides the fact that it would be faster in specific cases.

More flexibility

Different shower and transport media must be made available. To add water, ice, rock and other materials will enormously increase the physics reach of CORSIKA. For the in-ice and in-water, or upward-looking experiments these are critical developments. But beyond this, much more will be possible with such a code, e.g. shower on the moon surface, etc. One aspect of more materials will be to be able to define volumes in the environment, containing specific models of media inside. This will also allow 3D atmospheres etc., which can be relevant for very inclined showers developing over >100km extend in the atmosphere.

It is wished that there is a feedback possible from each physics process into the shower process. For example: kill particles, adjust thinning levels, change tracking step length, etc.

Such general feedback will also allow to change the environment itself. E.g. by the buildup of positive ions and subsequent modification of electric fields.

Technically it was asked to provide “particle guns” that can inject arbitrary particles at any place and geometry in CORSIKA. To have a super-simple event generator for tests (Null-model, full elastic model, Heitler model, etc.) for tests and development. To save the full state of the simulation at any time, for later restart with changed parameters.

Technical comments

- On GPU: deterministic execution very difficult. Define requirements. Cannot be too strict.
- Automated testing is super important.
- Be careful about CMake versions and potential conflicts.
- Input, output modular, and offer TCP support.
- Use exception, evtl. with memory dump. Think about collecting crashes centrally. Discuss with community.
- Python bindings with pybind11. Can be included in project.
- Use python as wrapper for configuration and command line parsing → removed dependencies on other C++ libraries. Python is always there….
- Template error messages. May use static assert. Look into this. Error messages are important.
- Either adopt units consistently everywhere, or nowhere. Consistent code is important.
- Cleanup printout. Too much ASCII output.
- Remove old and “harmful” options, like NKG.

Remove limitations, add functionality

- Consider the current myon problem as one of the main physics cases for CORSIKA. Provide extra diagnostics! Help to understand and solve the issue!
- CORSIKA needs to fully support upward going Cherenkov photons.
- Nuclear masses are restricted right now, to Z<60 and A<99
- Tremendously extend the physics validation. In particular “interaction test”, to be done in same environment and with fully consistent interface, with standardized set of plots and important accelerator data. Also air shower data must be used here.
- Preshower in combined field of earth and sun.
- Deep shower analysis, sub-sampling. Rejecting uninteresting showers.
- For example simulate first hadron core to find interesting showers, then do e.m. part.
- Fix the hard cut between low- and high-energy models, which leads to known artefacts.
- Interaction model registry to clearly link particles to specific models, for example electron-photons vs. hadrons, low- vs. high-energies, muons, or other phase-space regions.
- Study effects of random number generator on shower physics.
- Add more standardized visualization tools. Visualization is important.
- Investigate: general quality steering with single parameter?
- Investigate: Interactions of heavy quark particles at ultra-high energies?
- Extra diagnostics on energy conservation, nuclear breakup, buildup of ions, etc.
- Maybe use particle IDs as simple version of history

Additional comments

- HESS collaboration has already a package to simulate e.m. cascades for a detector simulation. Check, if can be used and adapted for CORSIKA.
- Output format:
  - Provide legacy output format for some while.
  - Note: for example GEANT4 has no output format. It is just a framework, you do your own output.
  - CrPropa: no ROOT output, this is provided, but not supported!
  - HDF5 interface in pythia and c++ is not optimal...!
- Provide option to steer from legacy steering cards.
- Also move classic CORSIKA to gitlab, and add testing etc. Help for this was offered.
- boost may cause problems. Difficulty with many version and changing interfaces. Be careful, minimize use, consider copying needed code.
- Be very careful about: versioning, data preservation, reproduction, archiving.

Final remarks

The current CORSIKA will be fully maintained until the new version actually reaches a better physics performance in real applications. Physics performance is the only parameter that counts. The manpower and effort needed to reach this point was discussed, and is not insignificant. Different ways to collaborate on the project have been mentioned and need to be discussed and clarified in the next time; help from the community has been offered in principle, and also in concrete cases. Many details need to be discussed; it could be excellent if we find a way where the CORSIKA upgrade project is suited as baseline for specific funding requests in different groups and institutions.