Running benchmarks and related problems

General C8 call 05.05.2022

Nikos Karastathis

Scenarios under consideration:

1. A 10^16 eV proton shower 10^16, 30° zenith angle

2. A 10^15 eV proton shower, 45° zenith angle, with 200 radio antennas (under inspection)

3. A 10^15 eV proton shower, 85° zenith angle, with 200 radio antennas (failed)

Workstation specifications:

- CPU: AMD Ryzen Threadripper 2970WX 24-Core Processor
  Base clock speed: 3GHz
  Max clock speed: 4.2GHz
  L3 cache: 64 MB
- RAM memory: 64 GB

Run specifications:

- C7: FFLAGS=-O2 CFLAGS=-O2 ./coconut
  - SIBYL 2.3d, URQMD 1.3cr, (CoREAS)
- C8: vertical_EAS example
  - logging set to WARN
  - CUTS were set to appropriate values in order to avoid crushes in PROPOSAL

10^16 eV proton shower 10^16, 30° zenith angle

C7 runtime: 32m 32,830s

C8 runtime: 1h 2m 26.303s

These results are preliminary and will be rechecked!

Although, critical bugs were encountered...

LongitudinalWriter issue

The writer is hardcoded to work with the vertical shower case. If one changes the zenith angle the run will simply crash. The issue is traced here:

```
/**
 * Construct a new writer.
 */
LongitudinalWriter(ShowerAxis const& axis,
                   GrammageType dX = 10_g / square(1_cm), // profile binning
                   size_t const nBins = 200);             // number of bins
```

```
for (int bin = binStart; bin <= binEnd; ++bin)
```

for a quick fix we just skip the bins that extend to greater values than the ones hardcoded now:

```
if (bin >= profile_.size()) {
  break;
}
```

But this is clearly not good enough. The writer should be able to calculate on its own the number of bins it needs depending on the zenith angle of the shower.

ShowerAxis issue

By default the shower axis length has to be set to a value **1.5 larger** than its actual value in order to avoid corsika:errors like the following:

```
[corsika:error   (ShowerAxis.inl:78)] shower axis too short, cannot extrapolate (l / max_length_ = 2.043954963735 )
[corsika:error   (ShowerAxis.inl:68)] cannot extrapolate to points behind point of injection l=-88963.5228401087 m
```

When changing the zenith angle 1.5 times larger is not enough and we have to go even higher, depending on how big is the angle. These corsika:errors do not interrupt the simulation run, but probably the ShowerAxis class should figure out this information by itself.

Lessons learned

- I encourage everyone to please run some real world scenarios!
- The lab tests we run do not reveal the true problems of the C8 core code and its modules!
- There are critical bugs that should be addressed and fixed well before any actual release!
- Evolve these “examples” to validation tests.