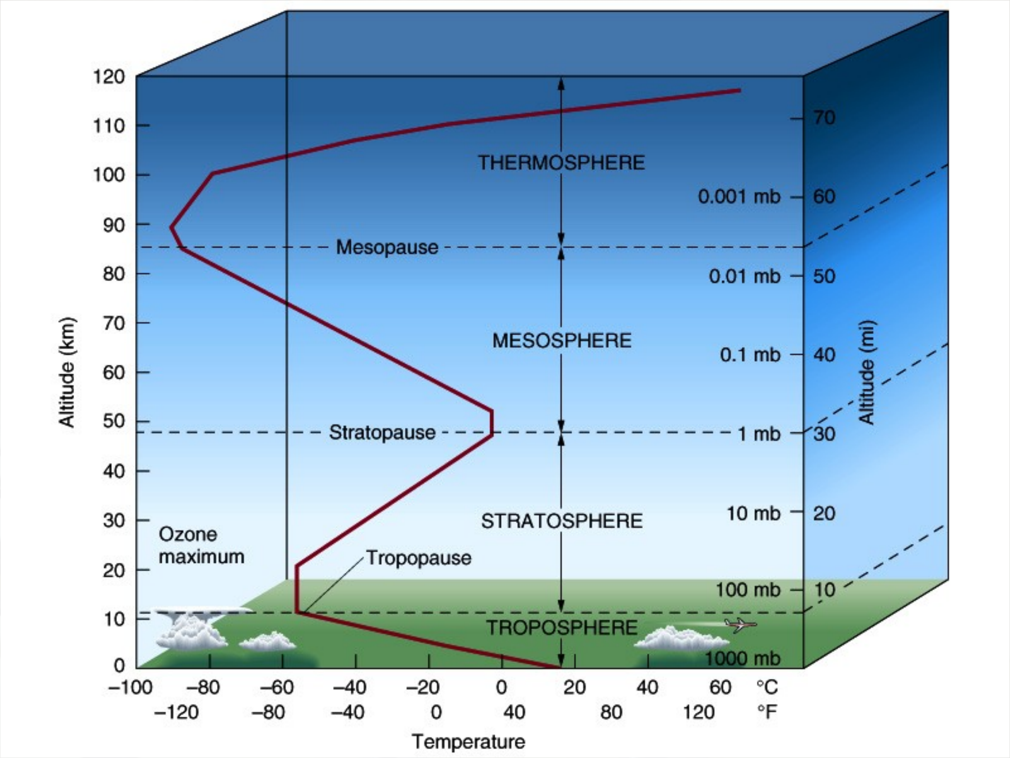
Atmosphere implementations and approximations

K. Bernlöhr
MPIK Heidelberg

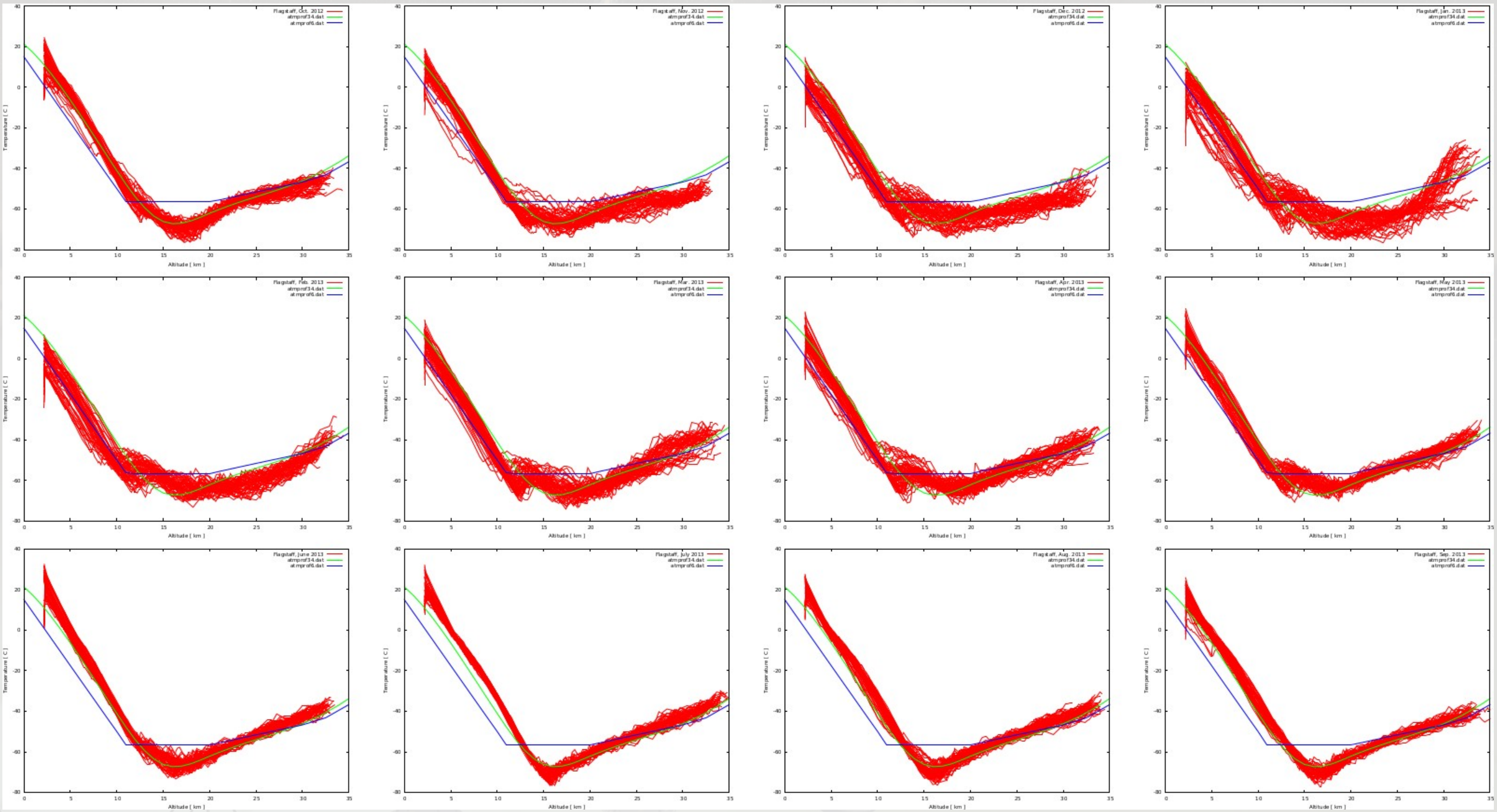


CORSIKA Cosmic Ray Simulation Workshop
Karlsruhe, 2019-06-19

Introduction

- Particle “detector” simulation codes like EGS4 and its modern derivatives or GEANT4 etc. want homogeneous “detector” components but not components with density and/or composition gradients.
  - Splitting up the atmosphere into thin layers is the usual solution but it comes at a big CPU impact.
  - Planar model preferred but spherical is manageable.
- The real atmosphere has a complex (and variable) vertical profile and if you want to cover large horizontal distances you may even face horizontal asymmetries and non-spherical shape.

Example temperature profiles



One year of radiosonde data (temperature versus altitude) for Flagstaff (USA).
Blue line: U.S. ‘standard’ atmospheric profile.

The traditional CORSIKA approach

- Realizing that splitting up the atmosphere into thin slices costs too much CPU time and using a single exponential profile is not a good approximation, CORSIKA came up with an intermediate solution:
  - 5 vertical zones (4 exponential, 1 linear gradient).
- Unfortunately, that is hard-coded and part of the data format – you cannot switch from 5 to 20.
- Transformations from atmospheric thickness (used for interactions, scattering) to geometric space (for decays, bending in B field, time delay ...) and back always need exp() and log() function calls.

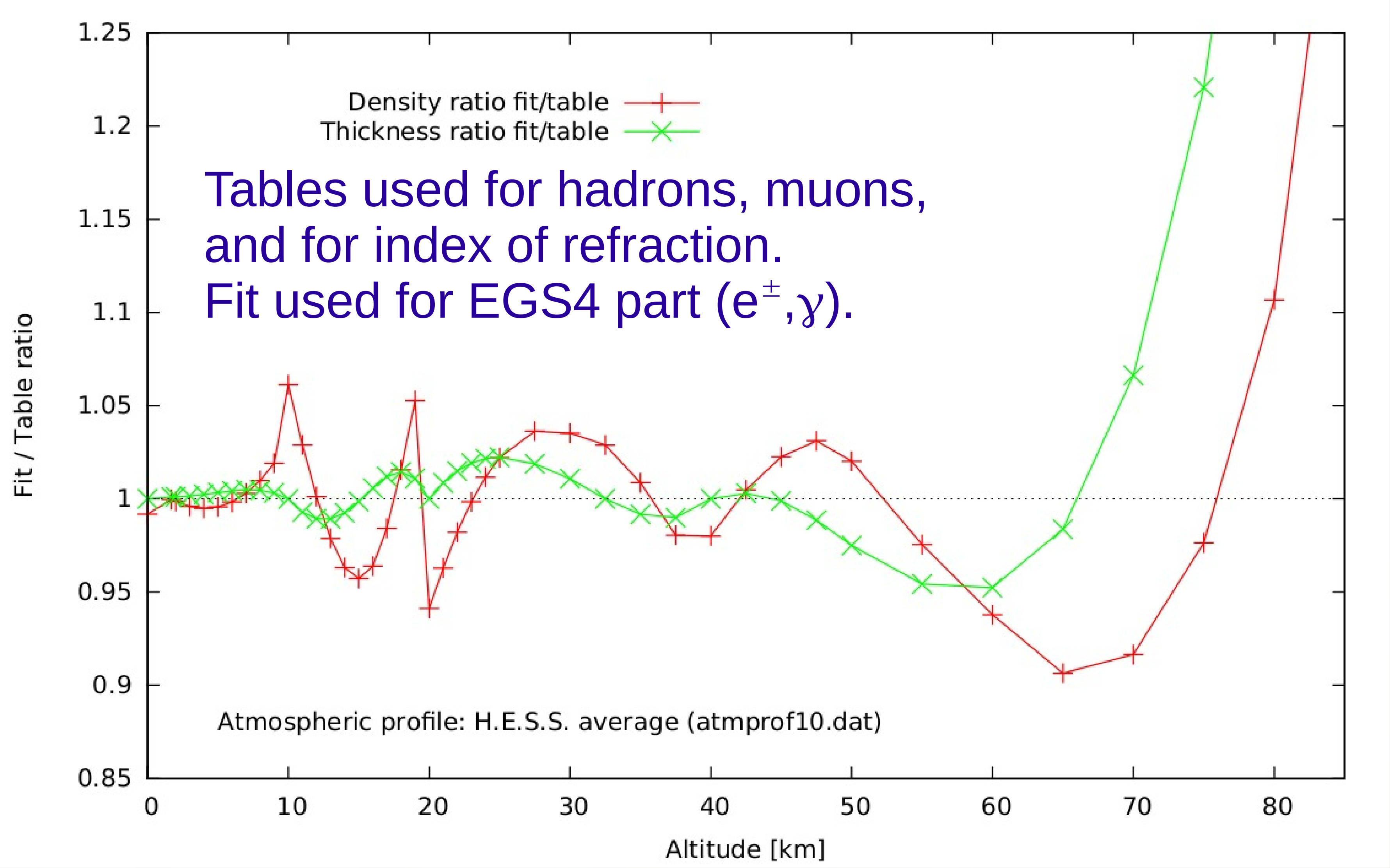
The ATMEXT approach

- With the ATMEXT option, the ‘ATMOSPHERE’ or ‘IACT ATMOFILE’ input card can be used to load a numeric table of the atmospheric profile.
  - ATMOSPHERE <n> Y
    loads file atmprof<n>.dat and enables refraction.
- For the CORSIKA-internal way, the numerical profile gets always fitted:
  - Fitting both density and atm.depth columns.
  - Optimizing boundary altitudes between layers.
- EGS part of CORSIKA can only use internal way.
- CURVED option also using internal only.

What is there to interpolate

- The essential information in the atmospheric profile tables includes:
  - height a.s.l. [km]
  - density [g/cm^3]
  - atmospheric depth (for vertical) [g/cm^2]
  - index of refraction minus 1 (n-1)
- The other columns (temperature, pressure, water vapor partial pressure) are not used.

Fit to tabulated atmospheric profiles



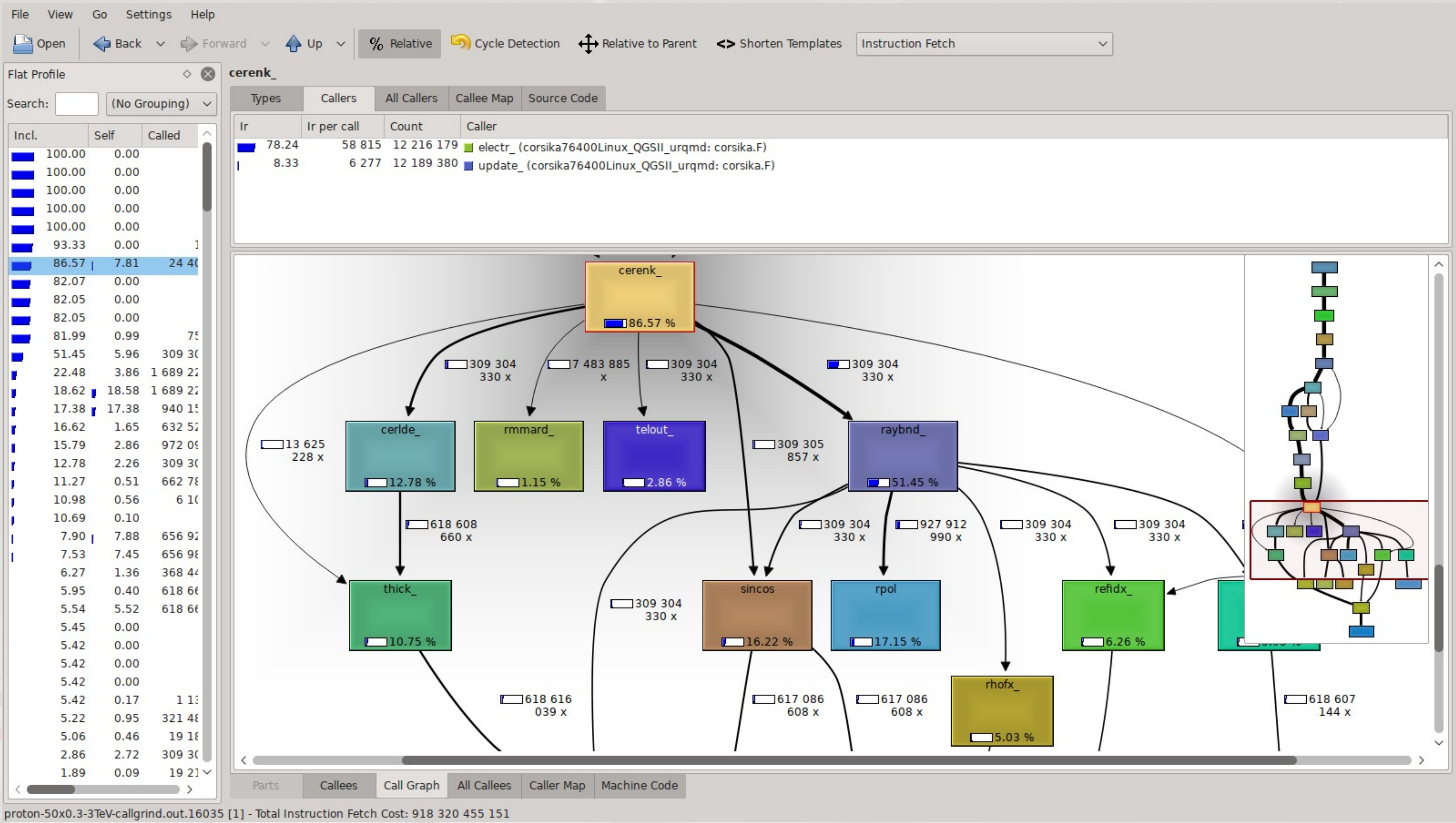
Energy scale and other problems ?

- Height corresponding to atmospheric depth of, say, 300 g/cm (typ. shower max.) can differ by up to ~100 meters between table and fit.
  - Example: atmprof36.dat: 70 m.
  - For comparison: linear and cspline differ by 2 m.
- This is no longer negligible in the energy scale systematics budget for CTA.
- Mixed approach (fit for EGS part, table interpolation elsewhere) has its own problems.
  - A hadron traversing 300 g/cm^2 is at a different altitude than an electron traversing 300 g/cm^2.

Areas where we wanted to improve with the latest IACT/ATMO release

- Atmospheric refraction correction taking too much (~50%) CPU time.
  - Initially added to check on the impact of refraction.
  - Correction based on known zenith-angle dependence and one-time ray-tracing (z=0° & 45°).
  - Involved interpolation with binary search of interval.
- FAST_INTERPOLATION not applied everywhere
- Lots of exp() calls, even for built-in profiles.
  - density / thickness / n-1 nearly prop. exp(-h/s), thus interpolation in log(ρ) versus h etc.
- CERLDE (long. dist.) still taking 14% CPU time.
  - Used to be ~90% until CORSIKA 5.x.
  - Partial solution with CORSIKA patch (factor 2 red.).

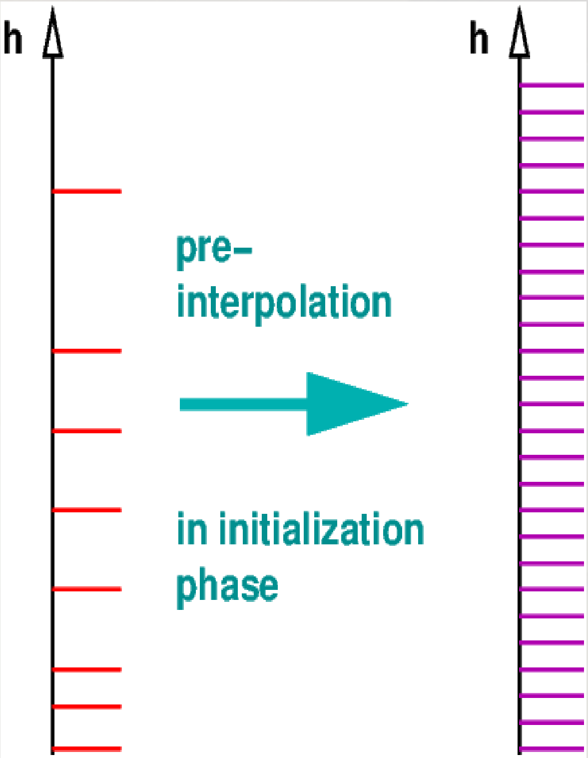
Callgrind + kcachegrind



CERENK subroutine and below used to take ~87% of the CPU time.

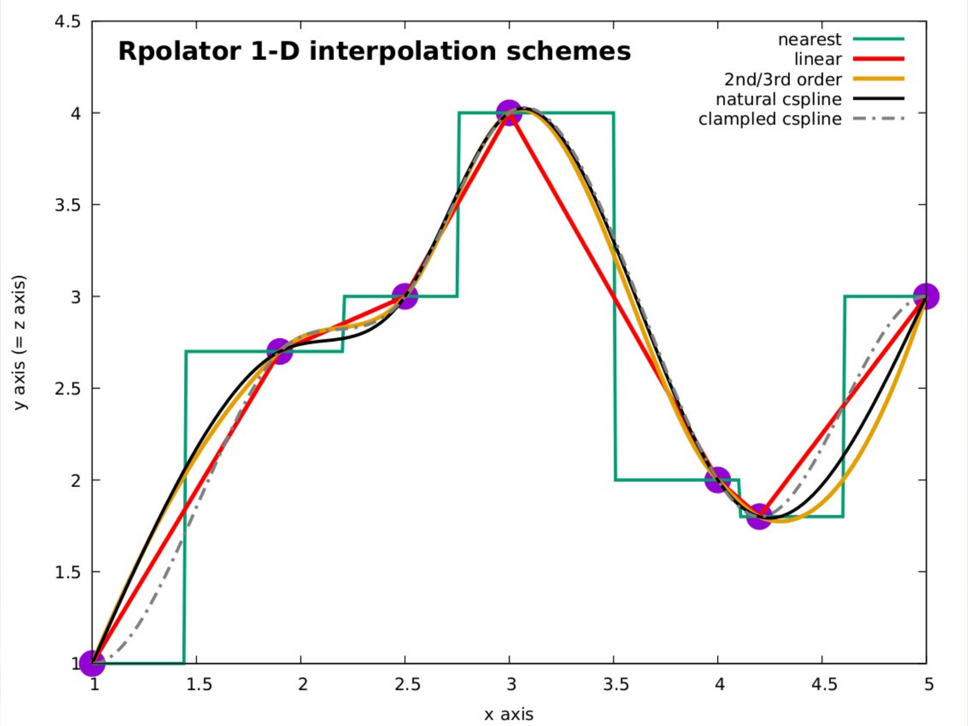
FAST_INTERPOLATION

- Compile-time option in atmo.c (used by default) to pre-interpolate profile parameters from non-uniform support altitudes to fine uniform stepping (used to be 10 000 steps).
- Non-uniform needs a (binary) search to find the interval in which to interpolate.
- Uniform can calculate the interval number directly.
- Steps needed is question of accuracy, memory, cache efficiency.



Different interpolation methods

- Interpolation methods at hand (“rpolator”):

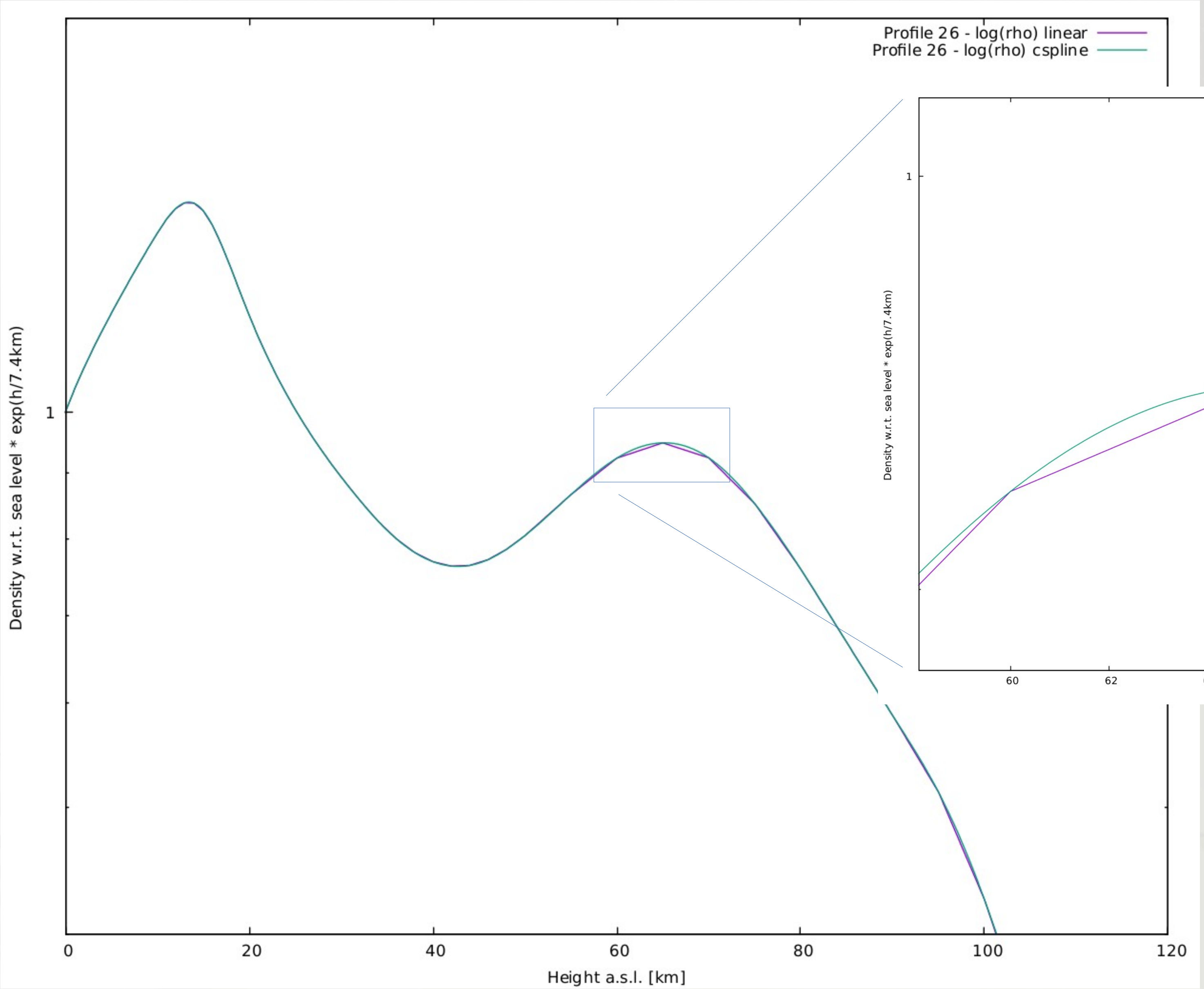


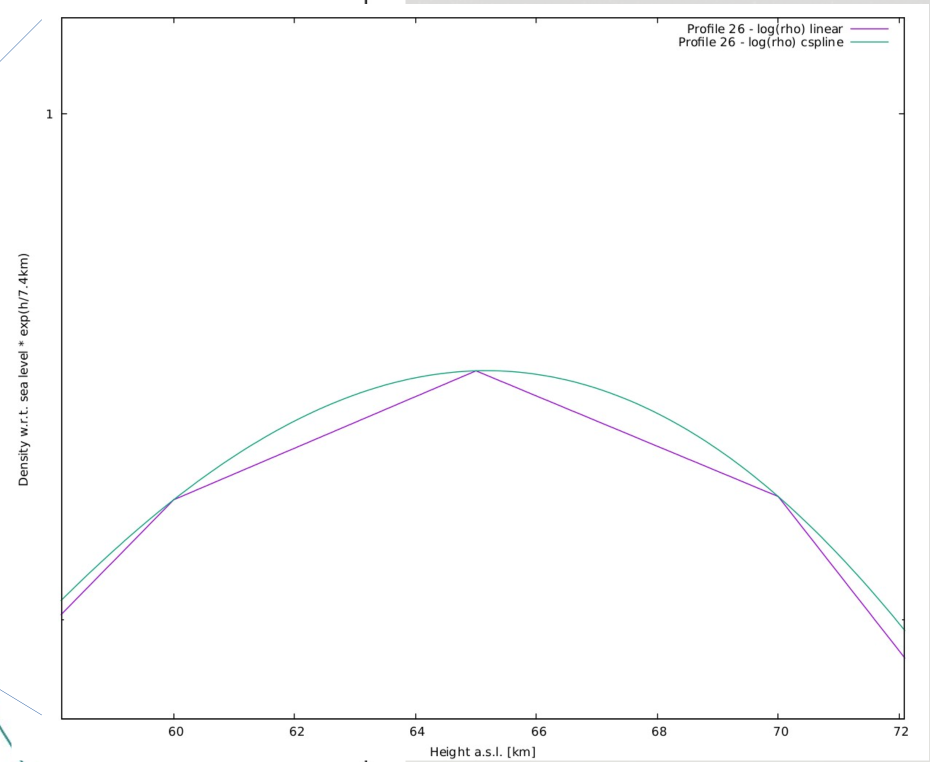
- 0th to 3rd order available, linear being the default.
- Equidistant supporting points are faster, non-equidistant need binary search for interval.

Linear versus cubic splines

- Cubic splines can produce smoother profiles than linear interpolation in log(ρ) versus h etc.
- Might be more accurate but no guarantees.
- Csplines should work with fewer support points (e.g. if cache efficiency is a problem).
- Cspline is slower:
  - Cspline needs to get 4 parameters and do 3 multiplications and 3 additions while
  - linear needs 2 parameters and do 1 multipl. and 1 addition.
- No perfect inversion with csplines.
- Some oddities of csplines at top of atmosphere.

Linear versus cubic splines





Can we avoid most exp() calls ?

- YES – but:
  - We need more support points in pre-interpolation and/or cubic splines in final interpolation.
  - Need to check for drop of cache efficiency with more points, e.g. 40 000 instead of 10 000.
    - Apparently not a problem. At least not when running a single process on my notebook or the MPIK cluster.
- Significant speedup of RHOF(), THICK(), REFIDX() and HEIGHT() functions.
- Also works for raybnd() function interpolations.
- As a result, CORSIKA now faster with tabulated atmospheric profiles than with built-in profiles.

Try options yourself ?

- Use version 1.58 from
  https://www.mpi-hd.mpg.de/hfm/~bernlohr/iact-atmo/
- Compile with your choice of compiler defines
  - -DNO_RPOLATOR / -DWITH_RPOLATOR
  - -DNO_FAST_INTERPOLATION(|2|3)
  - -DNO_RPOLATOR_CSPLINE
  - -DNO_THICKX_DIRECT
  - -DWITH_THICKX_DIRECT_CSPLINE
  - -DOLD_RAYBND
  - ...
- Version 1.59 stripped away most test options.
  - Less complexity. Easier to maintain.

Accuracy ?

- Note that even tiny differences in the interpolation of atmospheric profiles will change the shower evolution completely after some decision (e.g. decay instead of interaction).
- Instead of comparing average of many showers,
  - `gcc -DCHECK_REFRACT atmo.c fileopen.c straux.c -lm -o atmo_cf`
  - looking at evaluated profiles at some random and some critical points, also HEIGHT(THICK(h)) <-> h,
  - compared refraction correction versus ray-tracing ...
- HEIGHT(THICK(h)) generally better than 0.2 mm except near the second last support (at typ. 115 km) where it can be a few cm.
- Refraction correction better than 2 mm ($2 \cdot 10^{-4}$ of displacement) and 10 picoseconds for $z < 60°$.

Combined speed-up

- With IACT/ATMO package 1.59 – before vectorizing (see Luisa’s talk) – the typical CPU time with IACT + ATMEXT options, LONGI enabled is reduced by 35% (speed-up by a factor of ~1.54).
- On top of that another factor ~1.5 can be achieved by AVX* vectorization (not in distributed package yet).

Climate change

- Climate change is on-going!
  - Keep your CO_2 footprint small, if you can, for the sake of your children.
- The impact on shower simulations will be much less dramatic (order ~m in Hmax) but still ...
  - Use the proper composition for deriving atmospheric profiles and interaction cross sections.
  - Simplest approximation: reduce O_2 as CO_2 increases, although there are additional source and sink terms.

Lessons for CORSIKA 8 ?

- Don’t waste time with exponential layers.
- Direct interpolation can be faster, after pre-interpolation during start-up.
  - No exp() / log() calls; no binary search.
- HEIGHT(THICK(h)) round-trip error (0.2 mm) is negligible w.r.t. fit residuals for old-style layers (can be 100 meters).
- For spherical atmosphere need 2-D interpolation.
  - What should be the second coordinate? Sec(z)?

Conclusions

- Atmospheric profile is too complex for a good representation in the classical 5-layer scheme.
- Numerical table representation (e.g. based on weather forecast or radiosonde data) preferred.
- Differences between numerical input and best 5-layer fit may have non-negligible energy scale consequences.
- Numerical interpolation can even be faster than the 5-layer scheme and should be the only way in the CORSIKA 8 future.