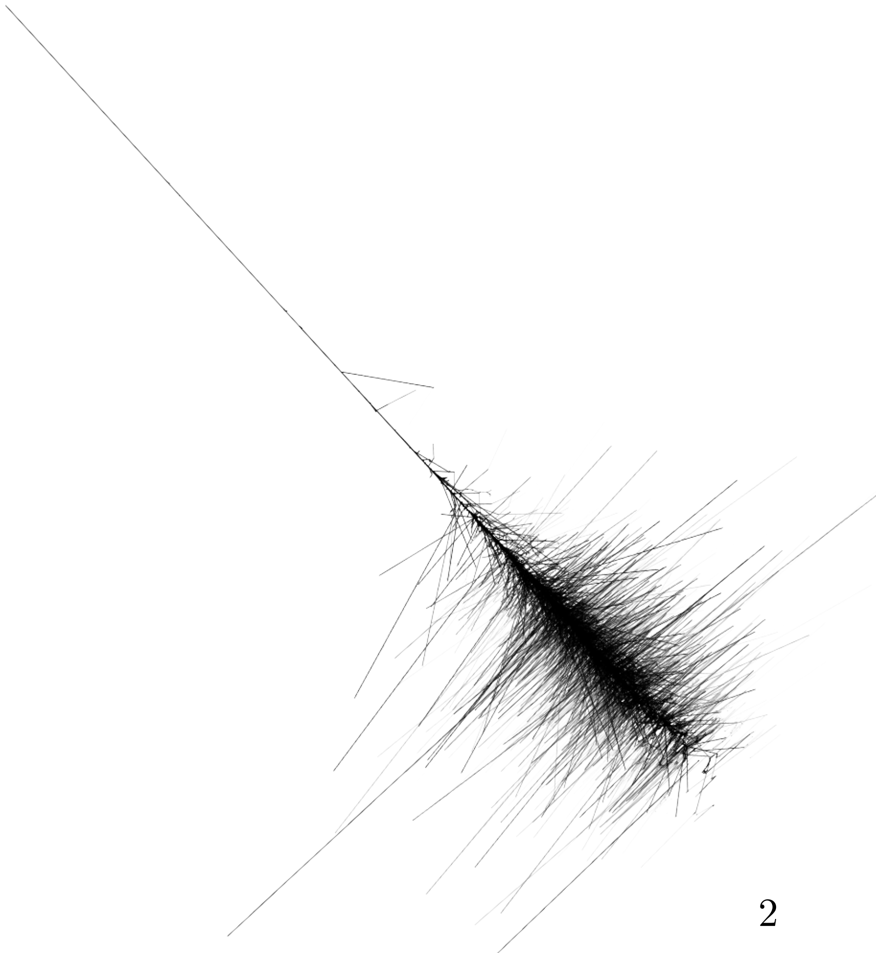
Transition radiation in cross-media showers

Juan Ammerman-Yebra, IGFAE, Universidade de Santiago de Compostela

CORSIKA 8 Air-Shower Simulation and Development Workshop,
Heidelberg 12 of July 2022

Outlook of the talk

- Transition radiation
- C8 implementation:
  1)Cross media shower
  2)Propagators implementation
- Results: very simple example to start testing the framework.



Note: all files mentioned in the presentation are available here.

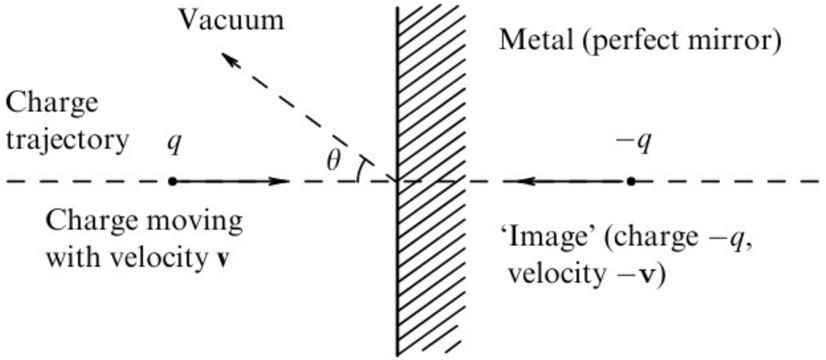
Transition radiation

- It is produced when a charge particle crosses the boundary between two dielectric media.

- Simplest example: Particle at constant speed from vacuum to metal (high conductivity).

  The field in the vacuum is then the sum of a charge q and an image charge -q.

  Thus, for this observer, two particles would annihilate at the boundary producing transition radiation.



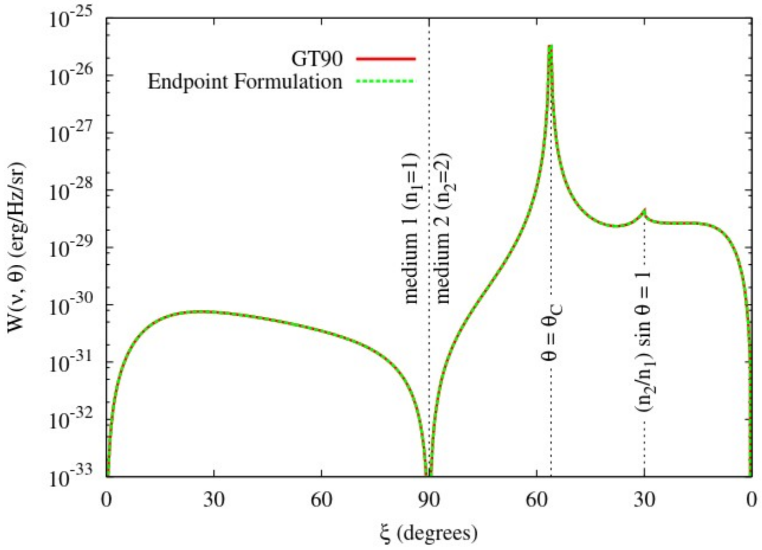
Ginzburg,Vitalii L, Physics-Uspekhi(1996),39(10).

Transition radiation

- At a microscopic level, transition radiation can be calculated as the sum of three contributions: direct, reflected and transmitted.

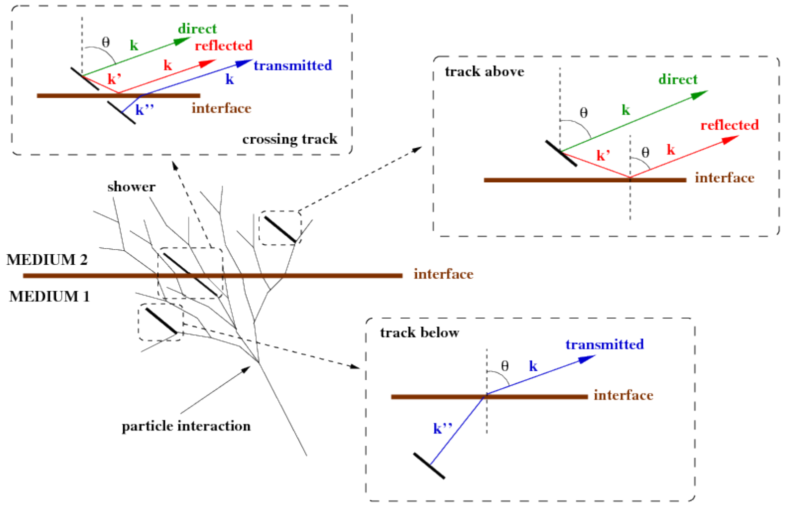
Comparison with theory:

Particle from vacuum to medium n=2



C. W. James, H. Falcke, T. Huege, and M. Ludwig, Phys.Rev. E84, 056602 (2011).

All possible cases in a shower



P. Motloch, J. Álvarez-Muñiz, P. Privitera, and E. Zas, Phys. Rev. D 93, 043010 (2016).

Implementation in Corsika 8

- As long as one knows the medium where the track is, we know which contributions to calculate.

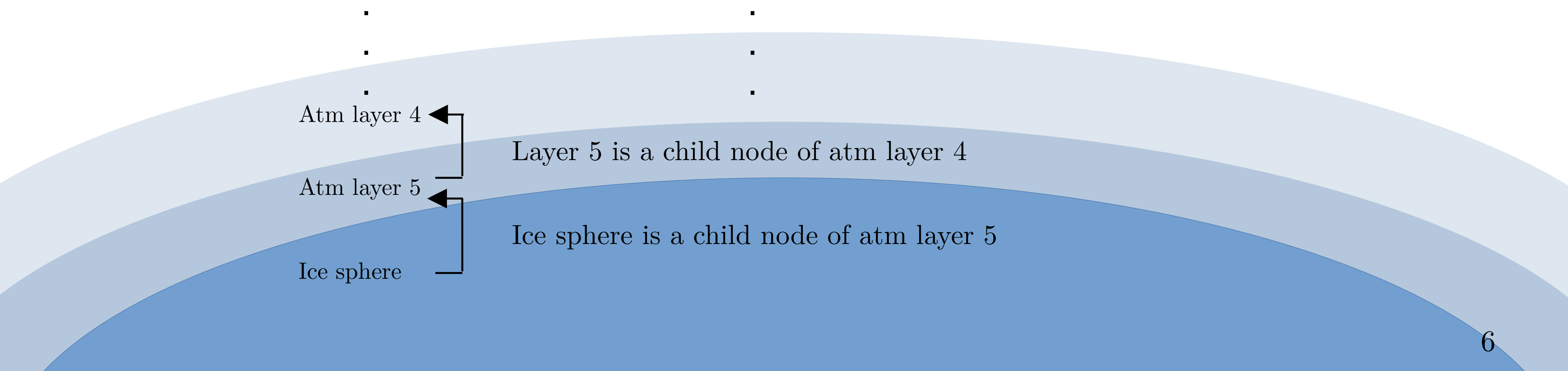
- The special case of a track crossing the boundary is automatically handled by the particle propagation algorithm in Corsika 8 (Cascade.inl line 166).

  - The distance to the next medium is calculated in each Monte Carlo step.
  - If this distance is the shorter than all interaction lengths, that is the one chosen to be advance.
  - This way we will always get tracks fully contained in one medium.

Shower from air to ice in C8

- For a shower that crosses several mediums we need to construct a layered environment in C8.

- This is done by adding one volume node to another. In this case if we want to have ice after the atmosphere, we add the ice sphere as a child node to the atmosphere sphere.



Shower from air to ice in C8

```cpp
MassDensityType const rho5 = grammage(1222.6562) / 994186.38_cm;
auto node5 = EnvType::createNode<Sphere>(center, EarthRadius+4_km);
auto const medium5 =
      std::make_shared<
      UniformRefractiveIndex<MediumPropertyModel<UniformMagneticField<SlidingPlanarExponential<EnvironmentInterface>>>>>(
      1.000327, Medium::AirDry1Atm, Vector(rootCS, 0_T, 0_T, 0_T), center, rho5, -994186.38_cm, nucl_comp_air, EarthRadius);
node5->setModelProperties(medium5);

auto ice_sphere = EnvType::createNode<Sphere>(center, EarthRadius);
NuclearComposition const nucl_comp_ice{{Code::Hydrogen, Code::Oxygen}, {0.11, 0.89}};
MassDensityType const density_ice = 0.918_g / (1_cm * 1_cm * 1_cm);
auto const medium_ice =
    std::make_shared<UniformRefractiveIndex<MediumPropertyModel<UniformMagneticField<HomogeneousMedium<EnvironmentInterface>>>>>(
      1.43, Medium::WaterIce, Vector(rootCS, 0_T, 0_T, 0_T), density_ice, nucl_comp_ice);
ice_sphere->setModelProperties(medium_ice);

VolumeTreeNode<EnvironmentInterface>* const universe = env.getUniverse().get();

node5->addChild(std::move(ice_sphere));
node4->addChild(std::move(node5));
node3->addChild(std::move(node4));
node2->addChild(std::move(node3));
node1->addChild(std::move(node2));
universe->addChild(std::move(node1));
```

→ Sphere for layer 5 of atm

→ Medium properties: refr. Index, type of medium, mag. Field, exponential density model

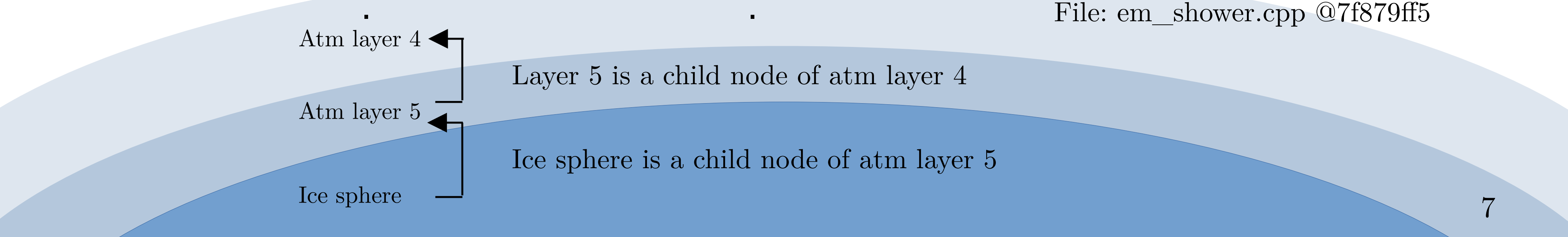
→ Add medium properties to layer 5 of the atm.

→ Medium properties: refr. Index, type of medium, mag. Field, uniform density model

Setting up the environment taking into account the hierarchy between volumes (from the outermost to the innermost).

→ Universe is the volume that englobes the “world” we create

File: em_shower.cpp @7f879ff5



Details of cross media showers

- If we use UrQMD for low energy hadronic interactions we only have tabulated cross-sections for nitrogen, oxygen and argon.

- This makes running showers in ice or water slow, could be added in the future since would only mean adding hydrogen to cover these two cases.

- Currently UrQMD needs to do many iterations to find the next interaction, failing to do so after reaching 50000 iterations, this makes us set very high hadronic thresholds.

- Electromagnetic interactions with PROPOSAL seems to be working without any issues from one medium to another.

Dificulties in implementing TR

- Transmitted ray paths are already not analytical in a planar interface between two mediums with constant refractive index (calculated through iterative process).

- Straight reflected paths cannot be analytically calculated if the boundary surface is spherical (can be calculated through an iterative process).

- Fresnel coefficients are usually not analytical, only in some approximations one can arrive at an analytical expression.

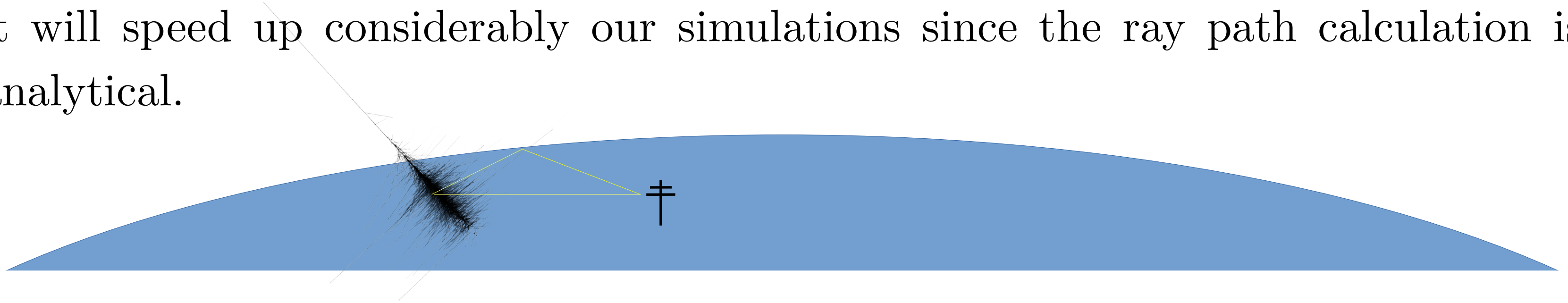
Specific media propagators in C8

- We will consider a few assumptions in order to get a simplified case:

  - Using a spherical ice surface, we can approximate the air-ice interface as flat. The shower dimensions are small compared to the Earth radius (not always true if very inclined).

  - We simplify our case to an homogenues ice sphere with constant refractive index.

  - For the test antenna we are looking at, the air contribution is negligible: neglecting it will speed up considerably our simulations since the ray path calculation is not analytical.



Specific media propagators in C8

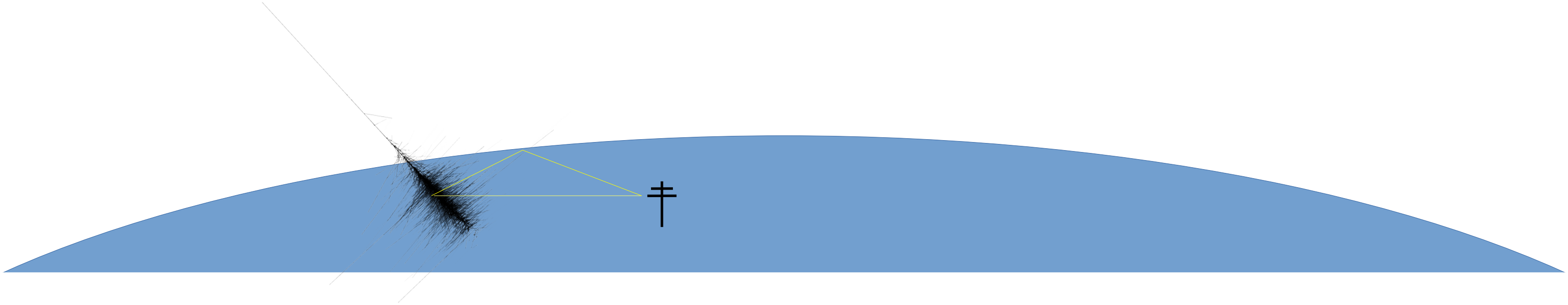
```
const auto& prop{node->getModelProperties()};
const auto& medium = mediumData(prop.getMedium());
if (medium.getName() == "water_ice"){
  CoordinateSystemPtr const& rootCS = Base::env_.getCoordinateSystem();
  auto const z_length = -destination.getZ(rootCS)+constants::EarthRadius::Mean;
  Point const destination_ref = {rootCS, destination.getX(rootCS), destination.getY(rootCS), destination.getZ(rootCS)+2*z_length};
  auto emit_ref_{(destination_ref - source).normalized()};
  Vector<dimensionless_d> k_ref{rootCS, {emit_ref_.getX(rootCS),
                                         emit_ref_.getY(rootCS),
                                         -emit_ref_.getZ(rootCS)}};
  auto receive_ref_{-k_ref};
  auto distance_ref_{(destination_ref - source).getNorm()};
```

$\longrightarrow$ Substracting the medium's properties

$\longrightarrow$ Checking if we are in the ice medium

Calculate path to reflected antenna as if the interface was a plane at fixed z

File: SimplePropagator.inl



Specific media propagators in C8

```
auto const beta_perp_perp_fresnel = beta_perp_perp*fresnel_perp;
auto const beta_perp_para_fresnel = beta_perp_para*fresnel_para;
auto const beta_perp_final = beta_perp_perp_fresnel +
                             beta_perp_para_fresnel;
```

We return two propagators, one for each ray path. Then the radio process will iterate over them.

```
return {SignalPath(time, averageRefractiveIndex_, ri_source, ri_destination, emit_,
                   receive_, beta_perp, distance_, points),
        SignalPath(time_ref, ri_destination, ri_source, ri_destination, emit_ref_,
                   receive_ref_, beta_perp_final, distance_ref_, points)};
}
```

File: SimplePropagator.inl

New argument that contains the direction of the field taking into account the Fresnel coef.

Fresnel coefficients in C8

- As the radio module is right now, they cannot be in principle implemented due to not having all the information in one place.

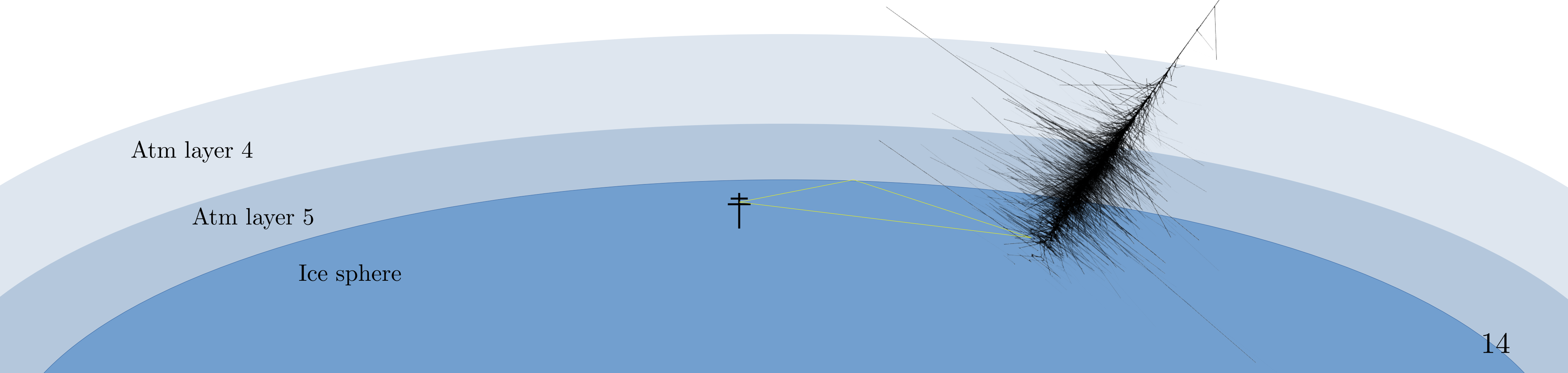
- In the radio process one is lacking the environment properties.

- In the section were the propagator is implemented you don't get the direction of the electric field.

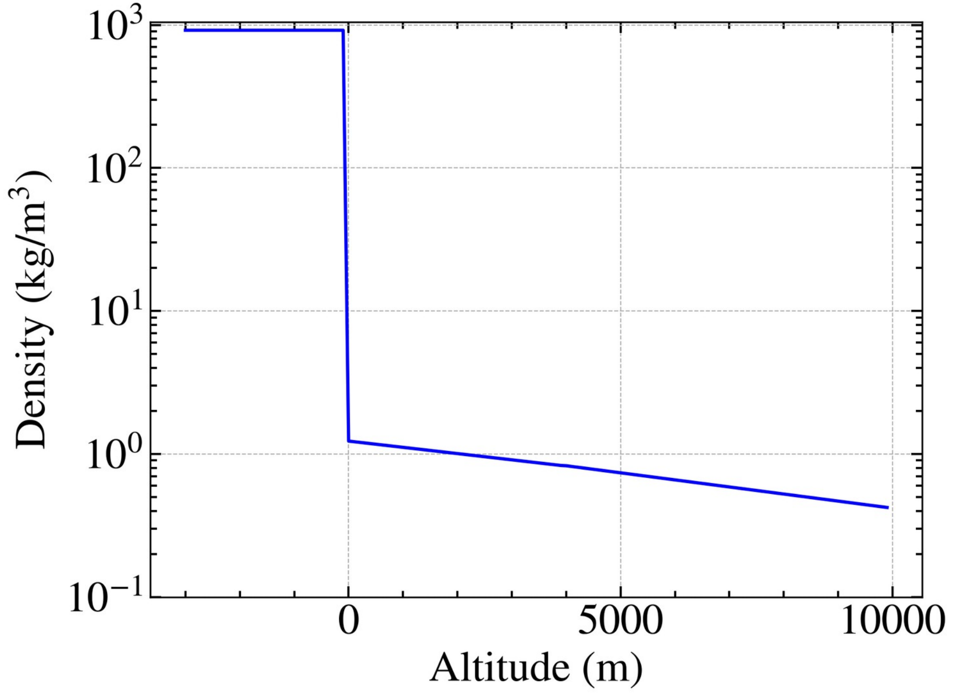
- Needs discussion to evaluate best solution. Currently I implemented a version that modifies the propagator to also calculate the electric field direction in it taking into account the Fresnel coefficients.

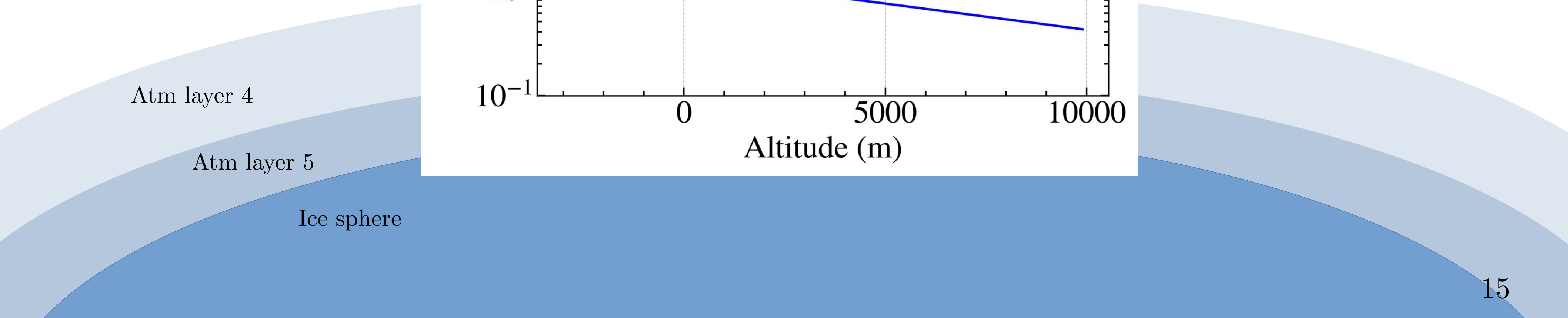
Results: Simulation setup

- e- 10 TeV shower with only electromagnetic interactions.
- 45º shower to have a more coherent emission in ice for both reflected and direct contributions.
- Ice antenna at (-4 km, 0 km, Earth Radius – 400 m), intersection of shower axis with the interface is at (0 km, 0 km, Earth Radius).
- Thresholds: 2 MeV for electron, positrons and photons.

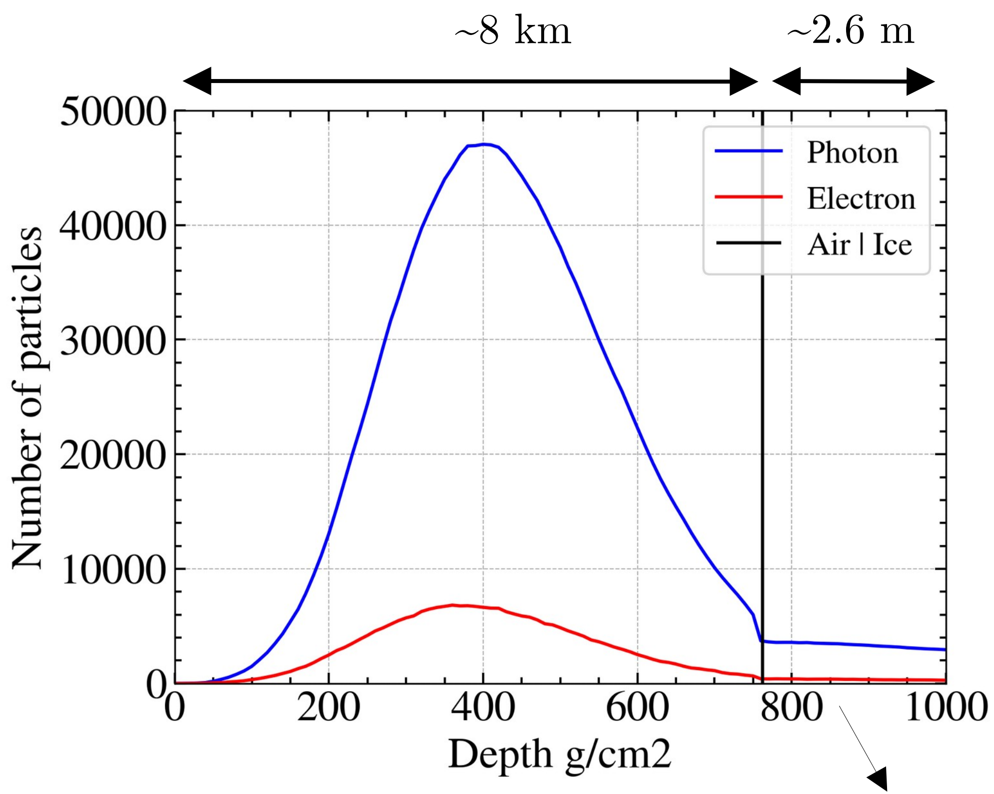


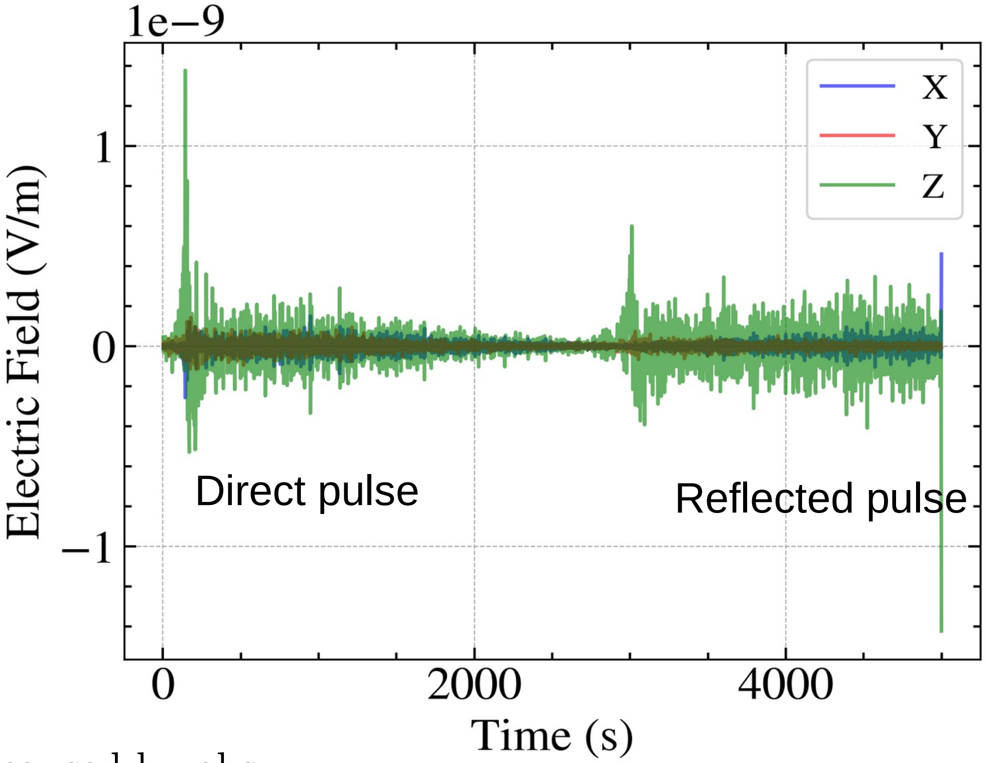
Results: Density profile





Results: 10 TeV electron EM shower at 45º





Not expected. Maybe caused by obs.
plane and shower axis in combination.

Summary

- Corsika 8 allows an easy, all in one, implementation of cross media showers.
- These showers need to be compared to other setups, like Corsika 7 to Geant4 for example.
- The radio interface is working as expected for multiple propagators and producing reasonable pulses.
- The radio interface needs some small changes to apply Fresnel coefficients.

Summary

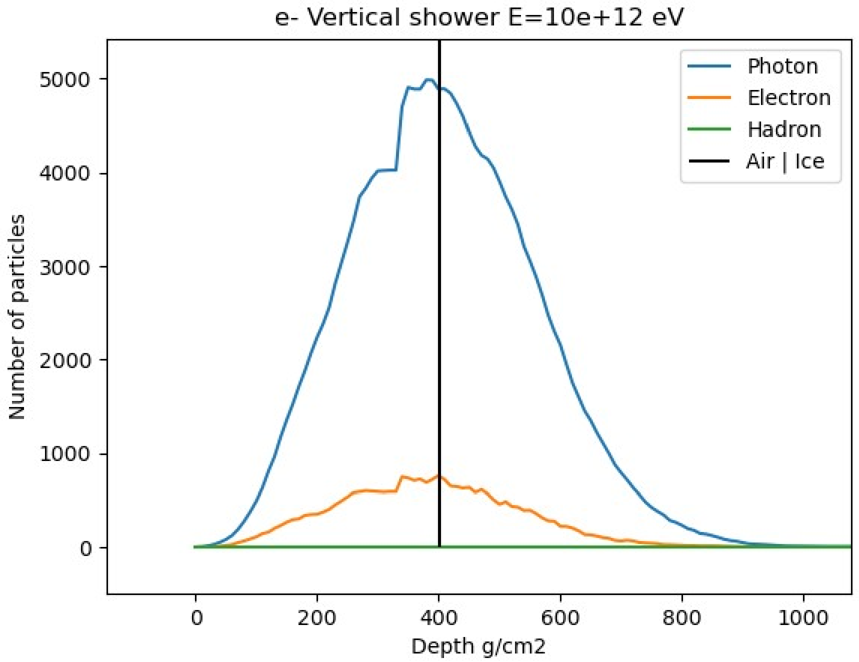
- Corsika 8 allows an easy, all in one, implementation of cross media showers.
- These showers need to be compared to other setups, like Corsika 7 to Geant4 for example.
- The radio interface is working as expected for multiple propagators and producing reasonable pulses.
- The radio interface needs some small changes to apply Fresnel coefficients.

Thank you for your attention !

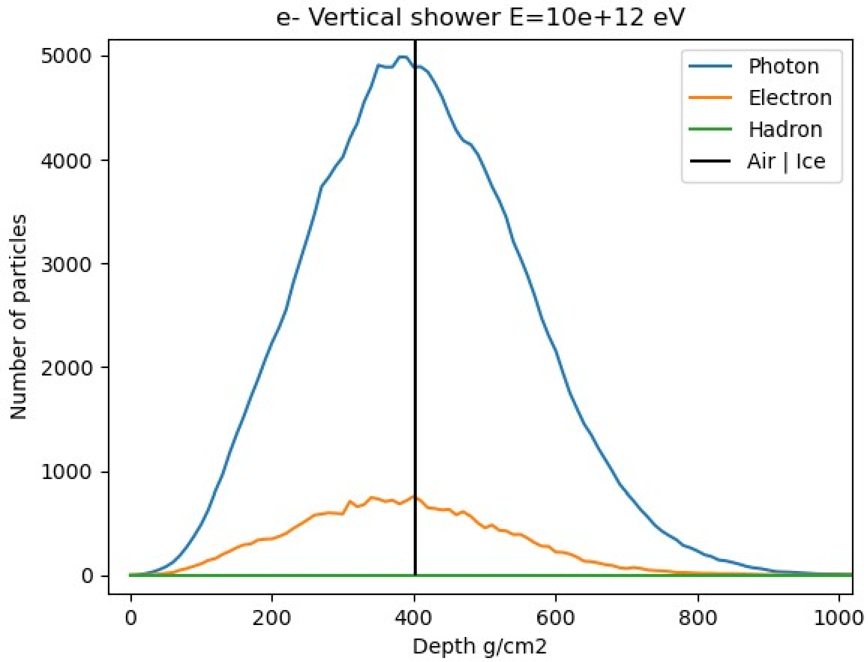
Backup slides

Vertical em shower

1000 steps in showerAxis



10000 steps in showerAxis



Simulation setup for EM shower