Eisvogel and C8

Radio propagation through general media

Philipp Windischhofer, Christoph Welling, Cosmin Deaconu
University of Chicago

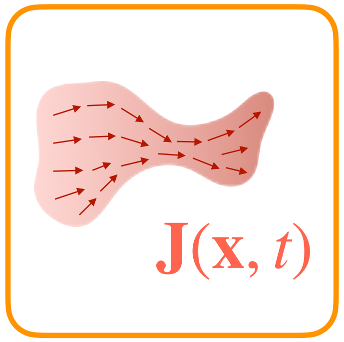
windischhofer@uchicago.edu

July 4, 2024

THE UNIVERSITY OF CHICAGO

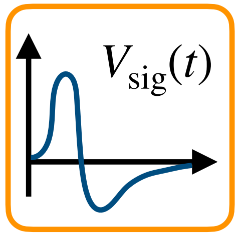
What is Eisvogel?

An electrodynamically-correct radio propagation code for general, linear media



Current density

Eisvogel



Antenna (voltage) signal

$$\nabla \times \mathbf{E} = -\partial_t \hat{\mu} \mathbf{H}$$

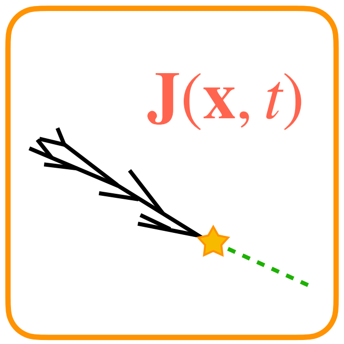
$$\nabla \times \mathbf{H} = \mathbf{J} + \hat{\sigma}\mathbf{E} + \partial_t \hat{\epsilon} \mathbf{E}$$

Maxwell's equations for linear medium

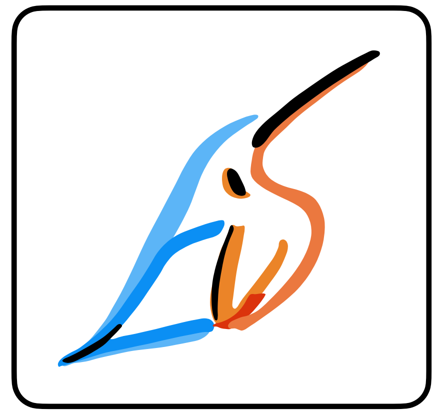
$\hat{\sigma}, \hat{\epsilon}, \hat{\mu}$

What is Eisvogel?

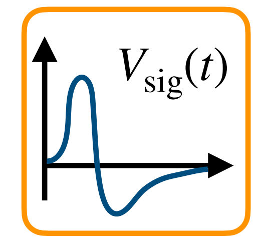
An electrodynamically-correct radio propagation code for general, linear media



Shower current density



Eisvogel



Antenna (voltage) signal

$$\nabla \times \mathbf{E} = -\partial_t \hat{\mu} \mathbf{H}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \hat{\sigma}\mathbf{E} + \partial_t \hat{\epsilon} \mathbf{E}$$

Maxwell's equations for linear medium

$\hat{\sigma}, \hat{\epsilon}, \hat{\mu}$

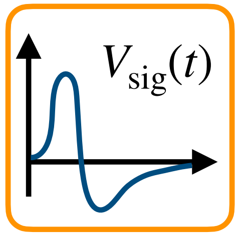
What is Eisvogel?

An electrodynamically-correct radio propagation code for general, linear media

corsika

Shower current density

Eisvogel



Antenna (voltage) signal

$$\nabla \times \mathbf{E} = -\,\partial_t \hat{\mu} \mathbf{H}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \hat{\sigma}\mathbf{E} + \partial_t \hat{\epsilon} \mathbf{E}$$

Maxwell's equations for linear medium

$\hat{\sigma}, \hat{\epsilon}, \hat{\mu}$

What is Eisvogel?

An electrodynamically-correct radio propagation code for general, linear media

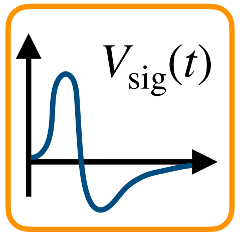
corsika

Shower current density

→

Eisvogel

→



Antenna (voltage) signal

C8 can simulate showers in very general media

↕

Eisvogel can simulate radio signals in very general media

(E.g. wave-optics effects for in-ice neutrino observatories)

What is Eisvogel?

An electrodynamically-correct radio propagation code for general, linear media

For today:

Eisvogel internals: *how does it work?*

Integration into C8: *demonstrator code*

Next steps: *missing features, dependency management, code structure…*

Eisvogel can simulate radio signals in very general media

(E.g. wave-optics effects for in-ice neutrino observatories)

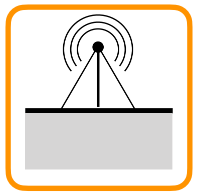
How does it work?

Isn't it hopeless to solve Maxwell's equations for large-scale geometries?

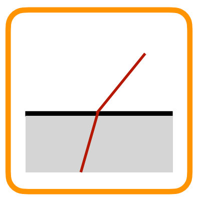
Yes, but …

km-scale simulation domains
cm-scale material features
cm-scale wavelengths

Antenna



Environment



$\mathbf{K}(\mathbf{x}, t)$

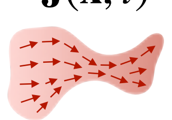
Green's function for Maxwell's equations
Encodes properties of environment and antenna

Performed once

Performed for every shower

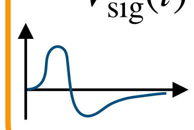
Shower current

$\mathbf{J}(\mathbf{x}, t)$



$$V_{\text{sig}}(t) = \int dt' \int d^3\mathbf{x}' \; \mathbf{K}(\mathbf{x}', t - t') \cdot \mathbf{J}(\mathbf{x}', t')$$

$V_{\text{sig}}(t)$



Antenna signal

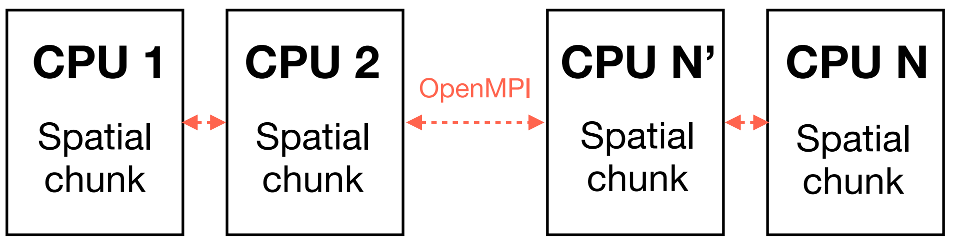
Convolution with shower is *fast*!

Calculation of complicated radiation propagation amortized into Green's function

How does it work in practice?

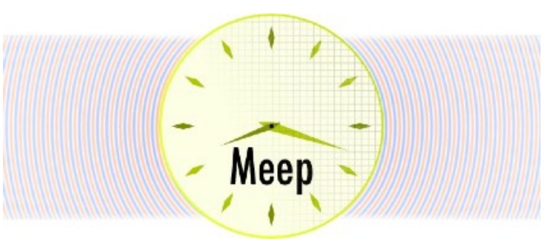
Green's function calculation …

Off-the-shelf numerical solvers for Maxwell's equations **applied to large-scale geometries**



Distributed-memory parallelism

Eisvogel interfaces with open-source solver *MEEP* [homepage] [code]

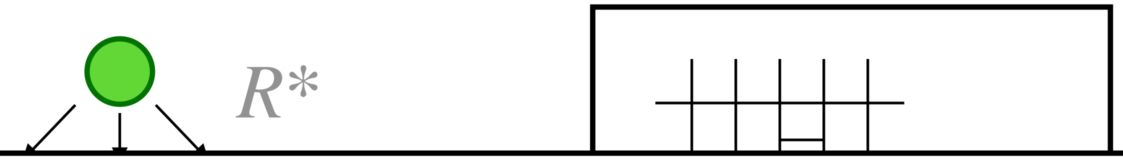


O(5 hours) on 128 cores, **O(100 GB)** disk space
(300m cylindrical geometry, 1.2cm resolution)

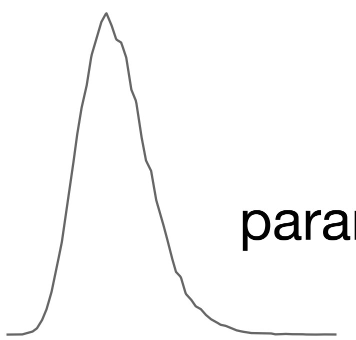
… and storage

Effectively a **custom file system for large sparse arrays**

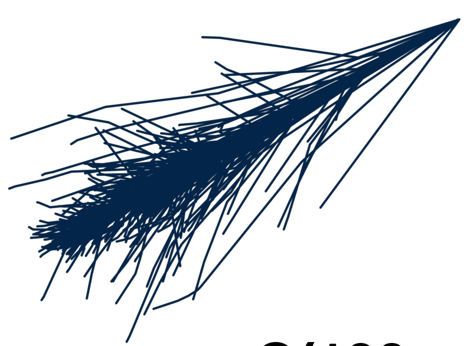
(Green's function is "lookup table", ≥ 10 TB when stored naively)



Shower signal calculation



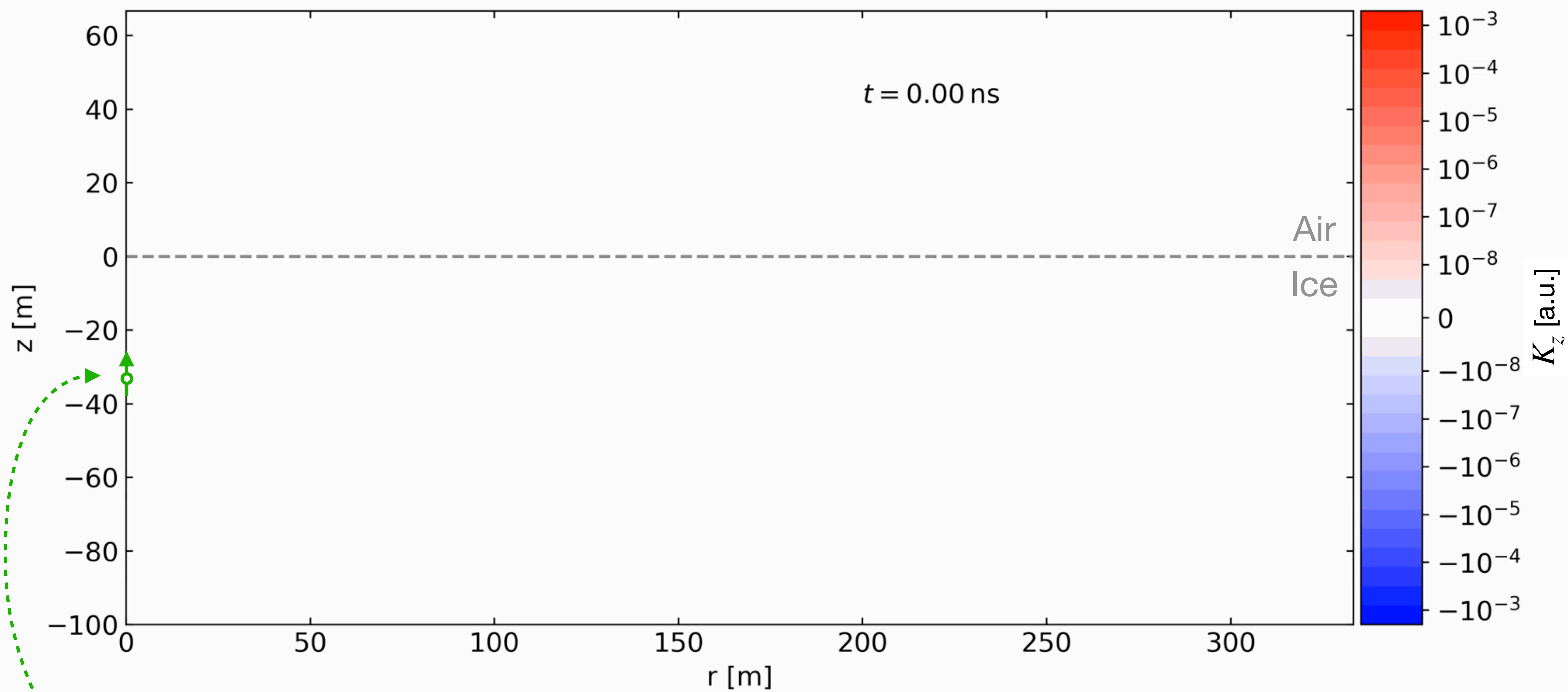
Starting from parametrized / binned **1-dim in-ice shower profile …** *(similar to NuRadioMC)*



… or microscopic showers *(with C8)*

O(100 ms) for **1-dim in-ice shower**
(Single-threaded, but vectorized)

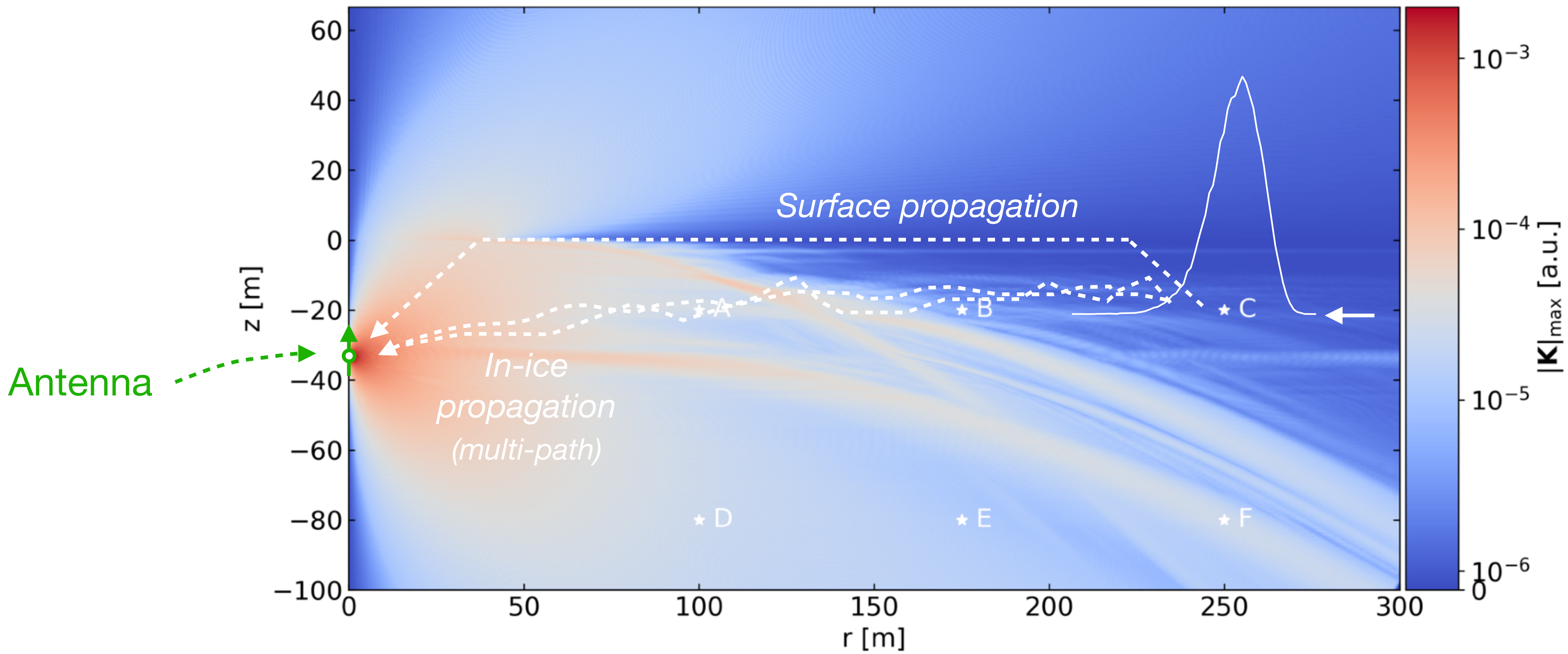
A Green's function for in-ice neutrino detectors

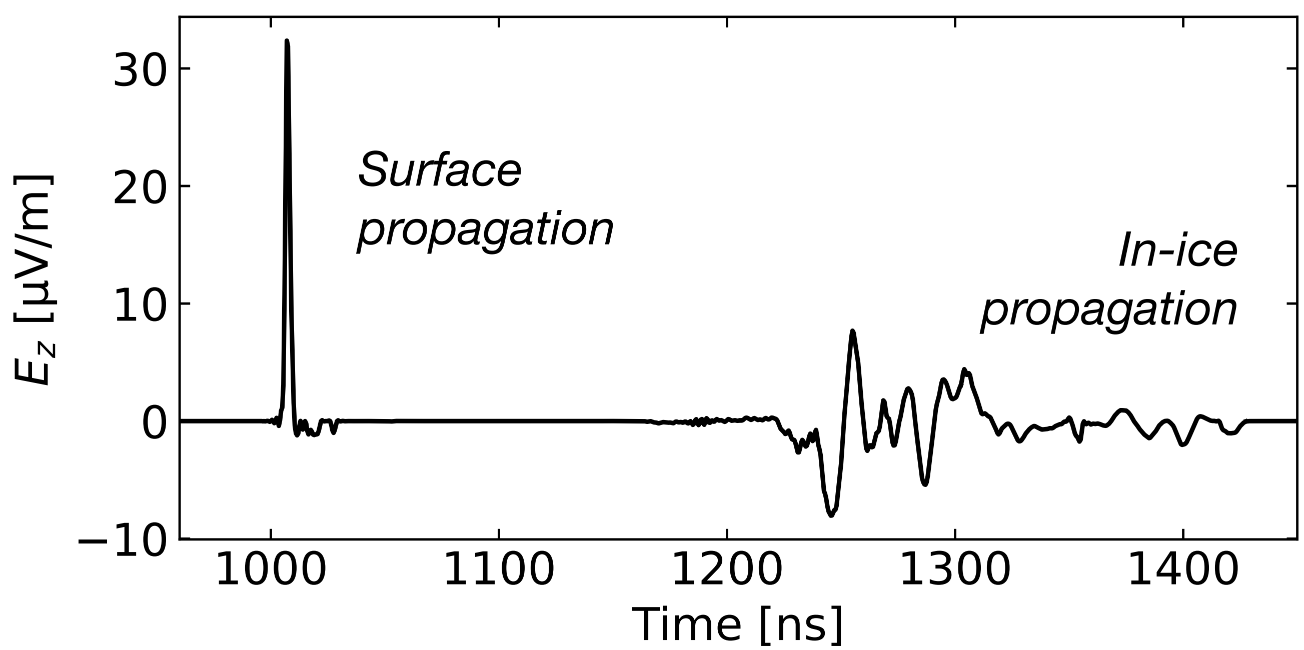


Vertically-polarized dipole antenna

Wavefront in the **topmost part of the ice** *(the "firn")* is **extremely complicated!**

Signals from neutrino-induced showers



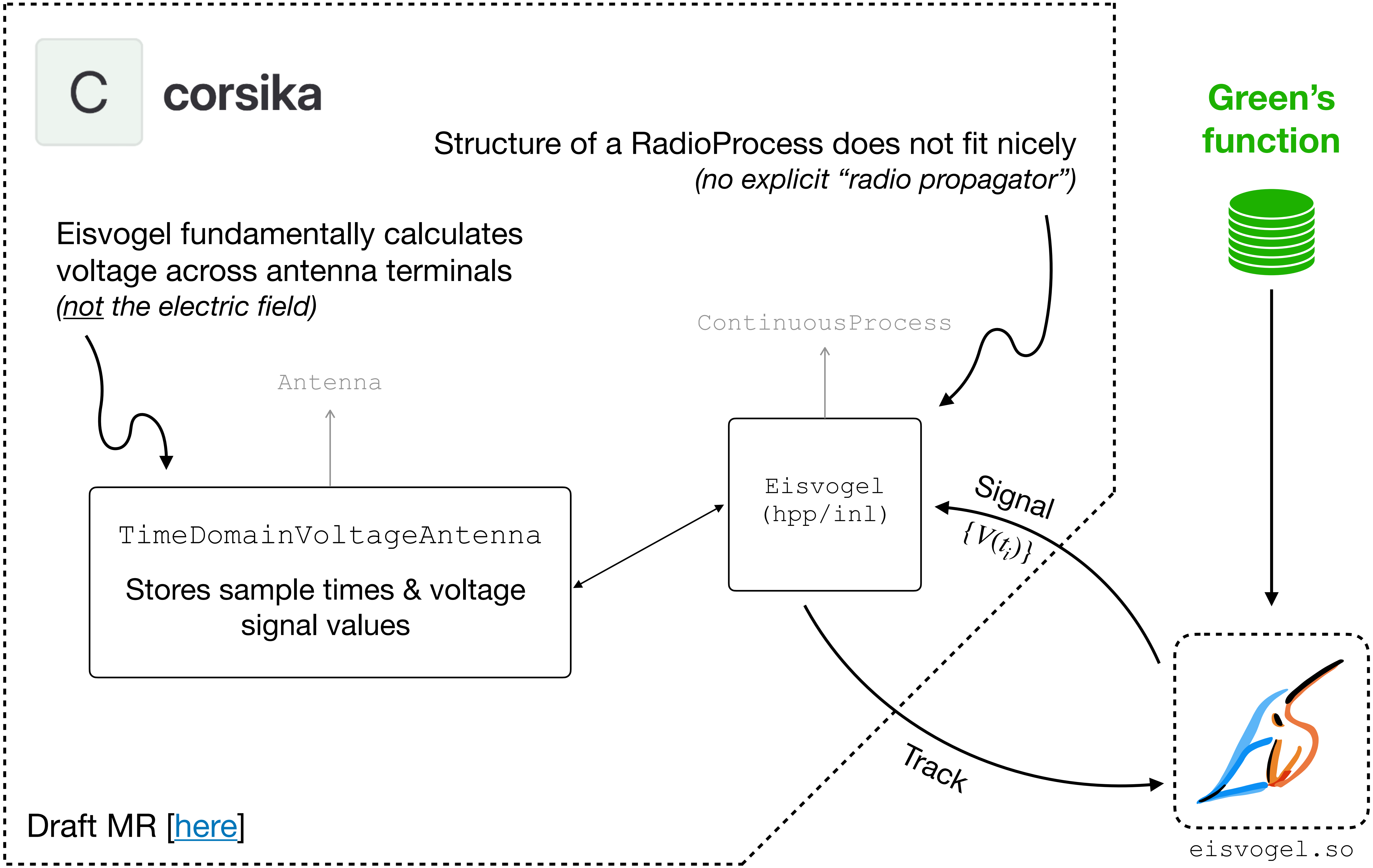


Expect **very complicated and dispersed** signals from **cosmic-ray air showers** impacting in **shadow zone**

10^{18} eV hadronic shower, 1-dim profile

A prototype C8 integration

Shower signal calculation



A prototype C8 integration

Shower signal calculation

corsika

Eisvogel fundamen
voltage across ant
(not the electric field

An

TimeDomainV

Stores sample
signa

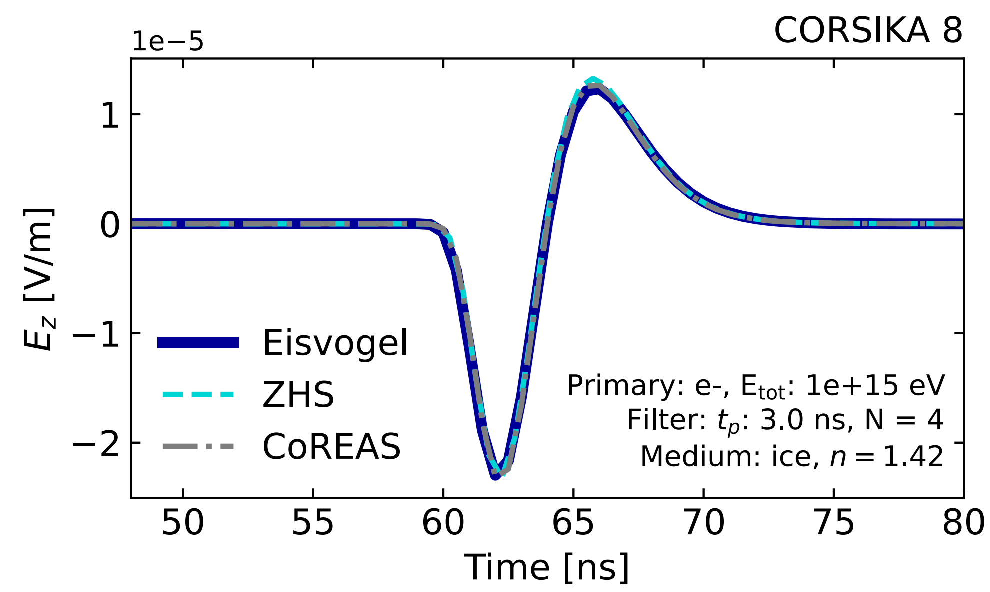
Draft MR [here]

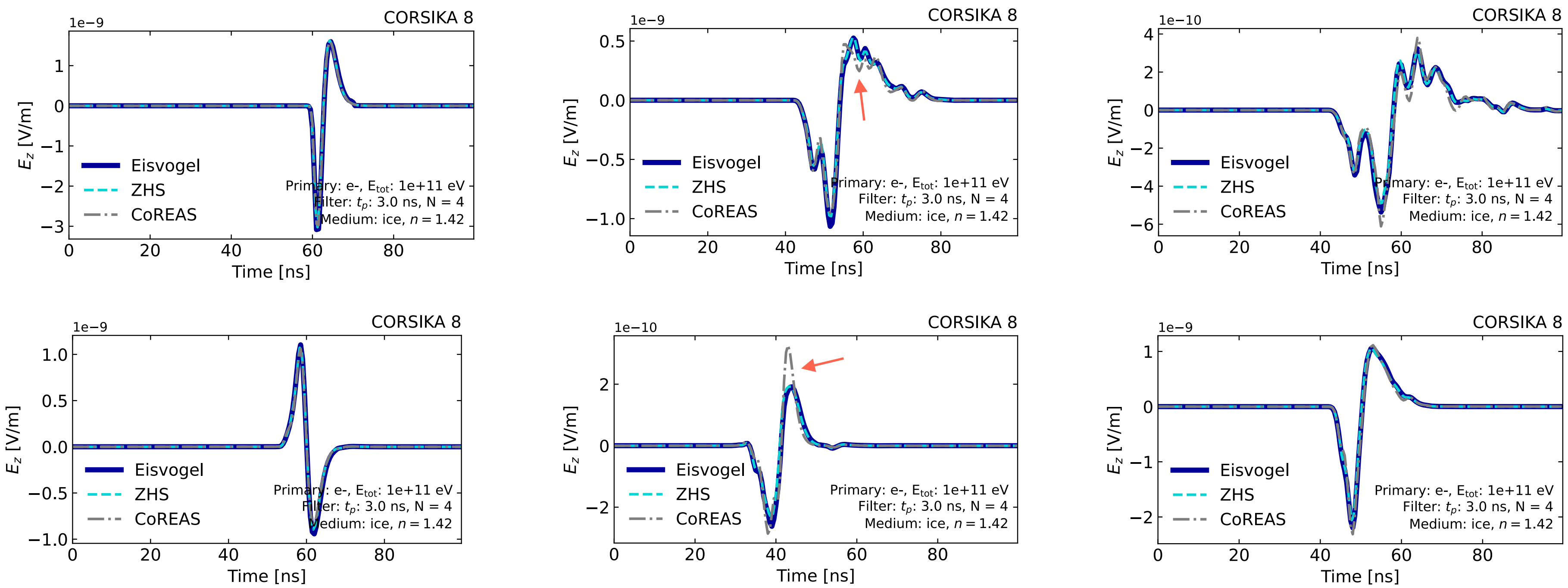
Quick test with showers in homogeneous ice

(Where `DummyTestPropagator` is accurate)

Near-perfect agreement with ZHS

Good agreement with CoREAS (`approxThreshold = 0.4`)





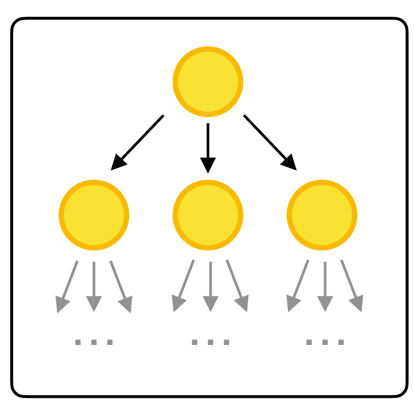
Coming next: environment dump?

Green's function calculation: Need to ensure consistency between geometry for shower simulation and Green's function!

C corsika

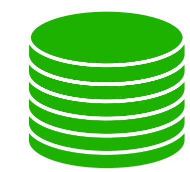
Ideal solution: have C8 environment serialize itself to a file
(symbolically if possible, rasterize if needed)

`Environment`



`environment.yaml` (?)

Green's function



`build_greens_function`

(Standalone executable running on cluster)

Git issue [here]

Questions from me / next steps

Any thoughts on the general structure of the Eisvogel integration?

Inheriting from `ContinuousProcess` instead of `RadioProcess`;
new `TimeDomainVoltageAntenna`

Important outstanding feature: dump of environment to parseable file

(Found some discussion in a git issue; is anybody already working on it? If not, I'm happy to help.)

Build practicalities

(Eisvogel uses C++20, C8 uses C++17 → build Eisvogel into `.so`, then link C8 against it)

Eventually, need to discuss "how" to include Eisvogel code into C8:

→ Build `.so` from source in separate build step (Conan?)

→ Alan: Include Eisvogel repository as submodule in C8?

https://github.com/eisvogel-project/Eisvogel

Summary

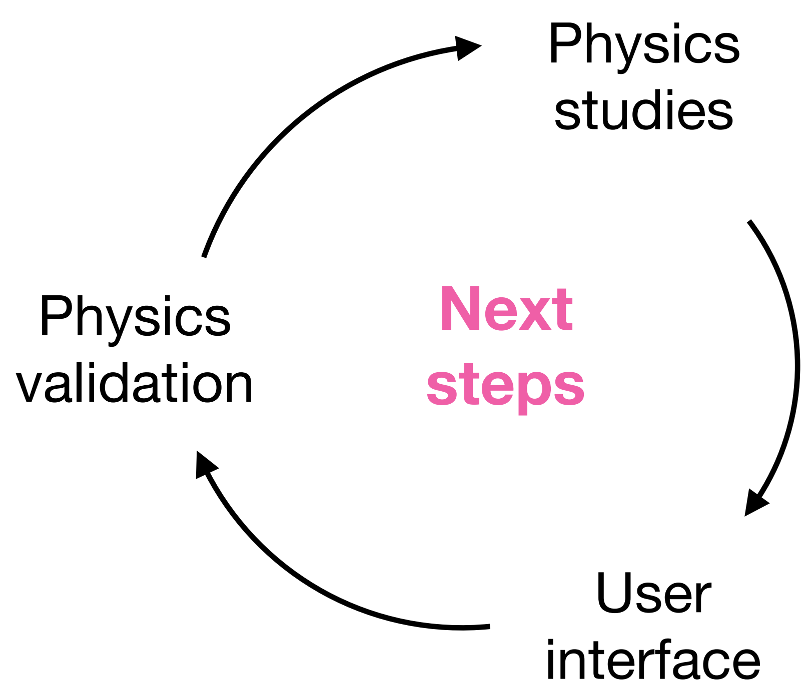
We can now calculate electrodynamically-correct radio signals in general media

Can test scenarios where ray-tracing does not apply

→ *Should make this as easily-accessible as possible!*

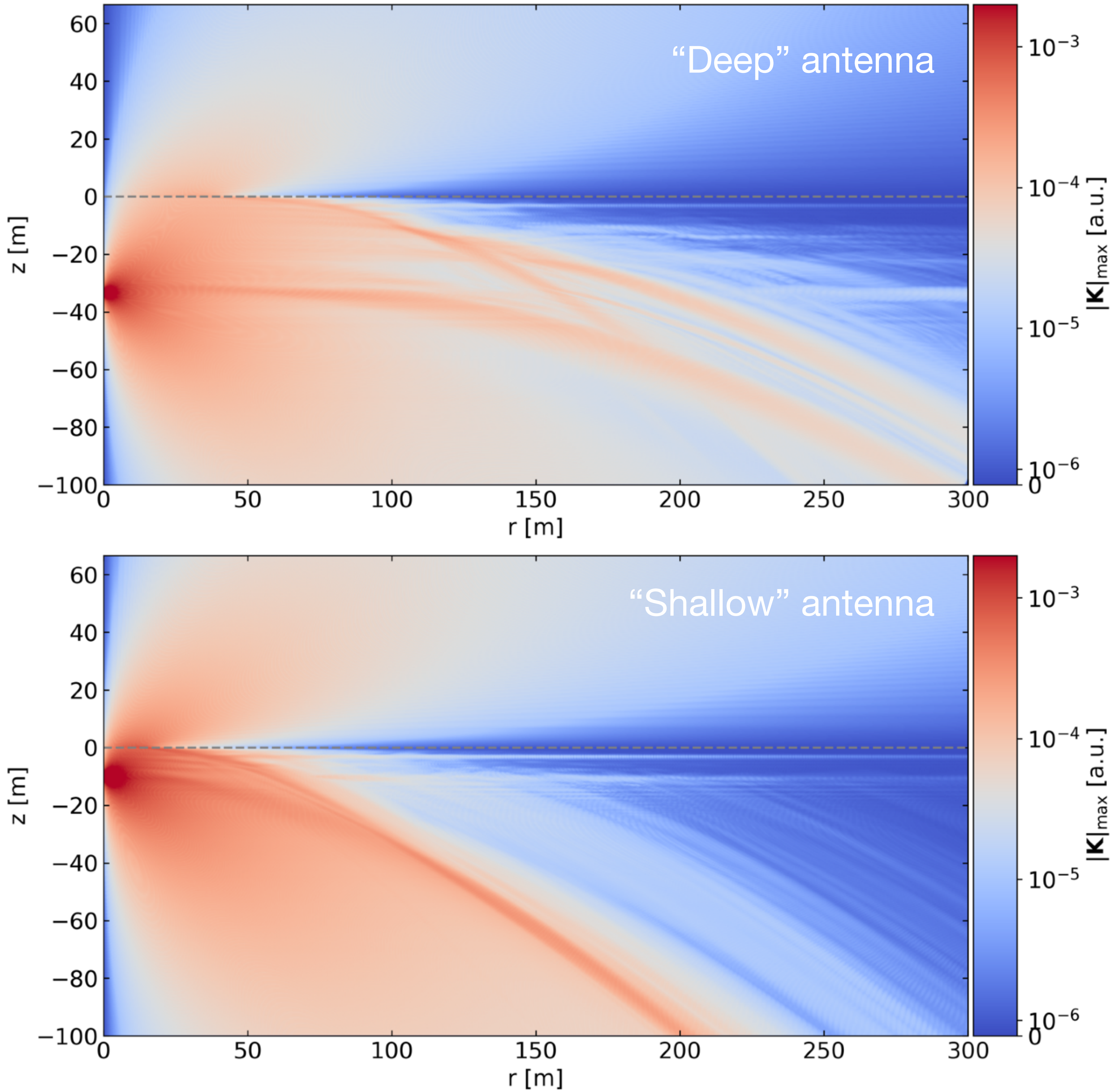
Have working prototype integration of Eisvogel in C8

Works if you know what you are doing … but no guardrails yet!

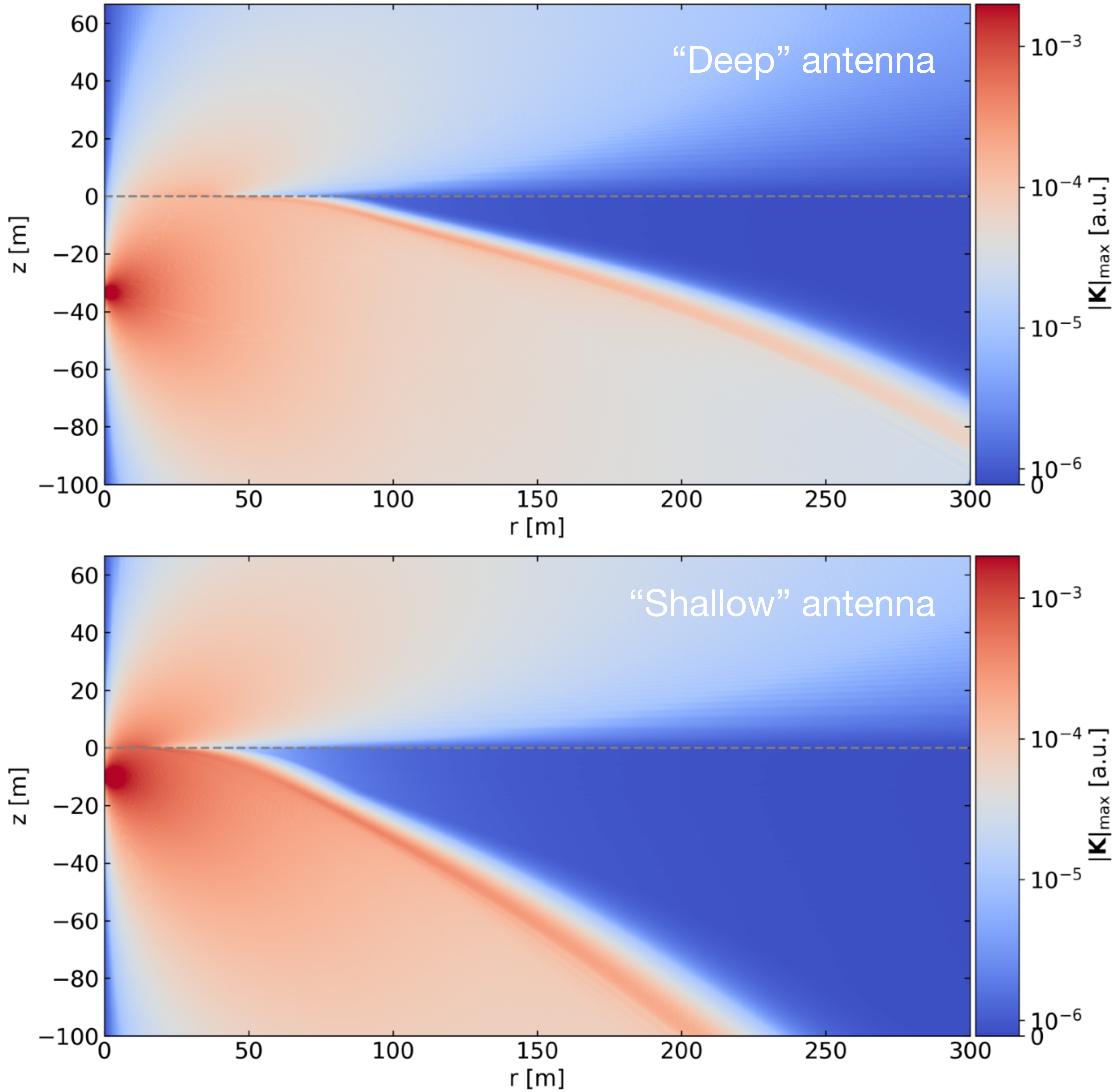


Backup

“Deep” vs “shallow” antenna placement



“Deep” vs “shallow” antenna placement



Dependencies

```
ldd ./bin/c8_air_shower

linux-vdso.so.1 (0x00007ffd99579000)
librt.so.1 => /lib64/librt.so.1 (0x00007f4cb644d000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f4cb622d000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f4cb6029000)
libCONEXsibyll.so => /home/windischhofer/C8/corsika-build/lib/libCONEXsibyll.so (0x00007f4ca1729000)
libgfortran.so.5 => /project/software/gcc-13.2.0-el8-x86_64/lib64/libgfortran.so.5 (0x00007f4ca1253000)
libfluka.so => /home/windischhofer/C8/corsika-build/lib/libfluka.so (0x00007f4c97c89000)
libquadmath.so.0 => /project/software/gcc-13.2.0-el8-x86_64/lib64/libquadmath.so.0 (0x00007f4c97a43000)
libstdc++.so.6 => /project/software/gcc-13.2.0-el8-x86_64/lib64/libstdc++.so.6 (0x00007f4c975e1000)
libm.so.6 => /lib64/libm.so.6 (0x00007f4c9725f000)
libgcc_s.so.1 => /project/software/gcc-13.2.0-el8-x86_64/lib64/libgcc_s.so.1 (0x00007f4c9703b000)
libc.so.6 => /lib64/libc.so.6 (0x00007f4c96c76000)
/lib64/ld-linux-x86-64.so.2 (0x00007f4cb6655000)
libmvec.so.1 => /lib64/libmvec.so.1 (0x00007f4c96a4b000)
```

```
ldd libeisvogel-corsika.so

linux-vdso.so.1 (0x00007ffd1c434000)
libuuid.so.1 => /lib64/libuuid.so.1 (0x00007f97e53e4000)
libtbb.so.2 => /lib64/libtbb.so.2 (0x00007f97e51a6000)
libstdc++.so.6 => /lib64/libstdc++.so.6 (0x00007f97e4e11000)
libm.so.6 => /lib64/libm.so.6 (0x00007f97e4a8f000)
libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f97e4877000)
libc.so.6 => /lib64/libc.so.6 (0x00007f97e44b2000)
/lib64/ld-linux-x86-64.so.2 (0x00007f97e5815000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f97e42ae000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f97e408e000)
librt.so.1 => /lib64/librt.so.1 (0x00007f97e3e86000)
```

Only used to generate filenames (libuuid)
→ if problematic, can be substituted

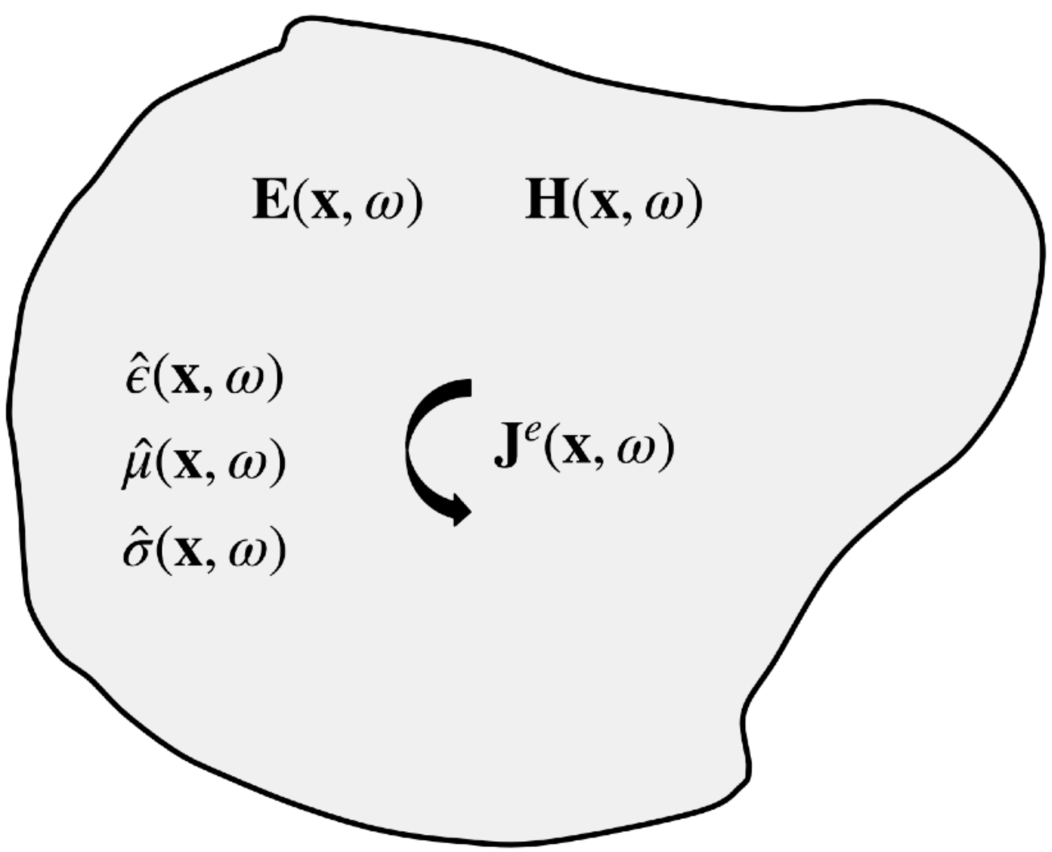
Picked up through liberal usage of `std::execution::unseq` but not actually called into (libtbb)
→ can probably be avoided entirely

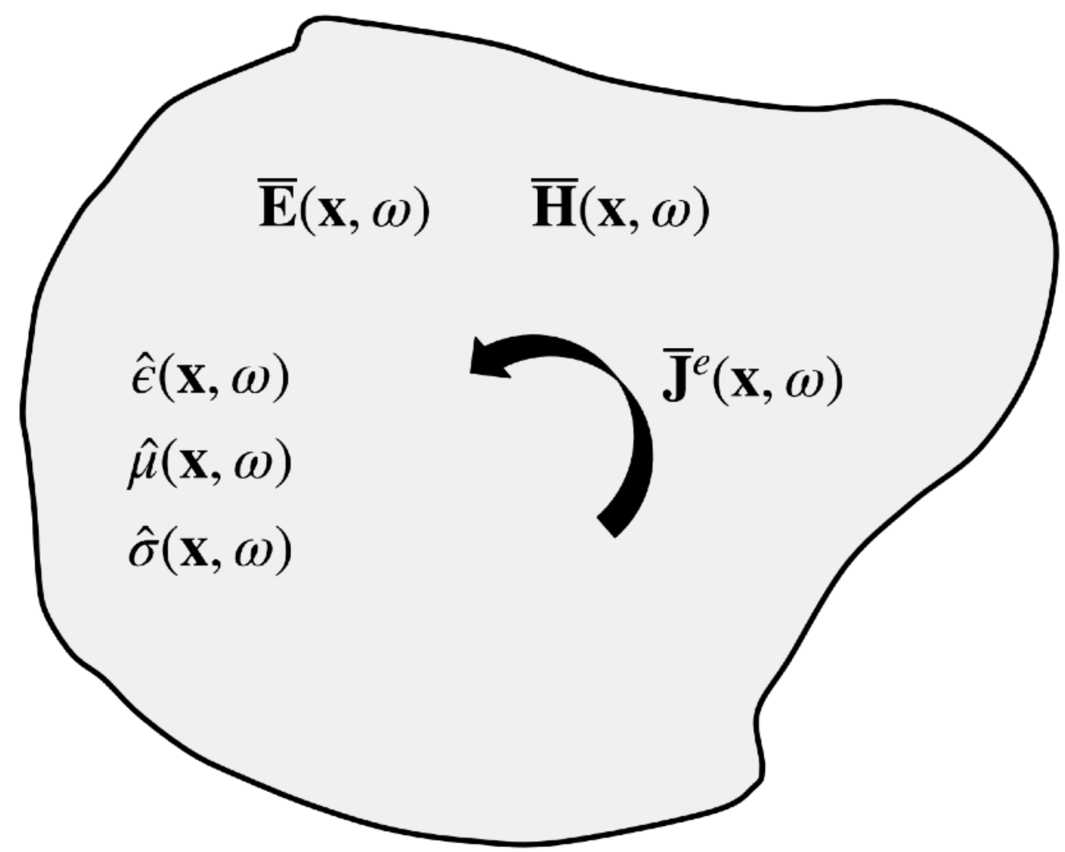
No “real” dependencies that are not already required for C8

Lorentz reciprocity

Classical electrodynamics has a built-in method
to relate two different situations *(with identical geometry)*

General, linear material distribution: $\epsilon(\mathbf{x}),\ \mu(\mathbf{x}),\ \sigma(\mathbf{x})$





$$\mathbf{J}^e \xrightarrow{\text{Maxwell's eqns.}} \mathbf{E},\ \mathbf{H}$$

External current distribution — Field distributions

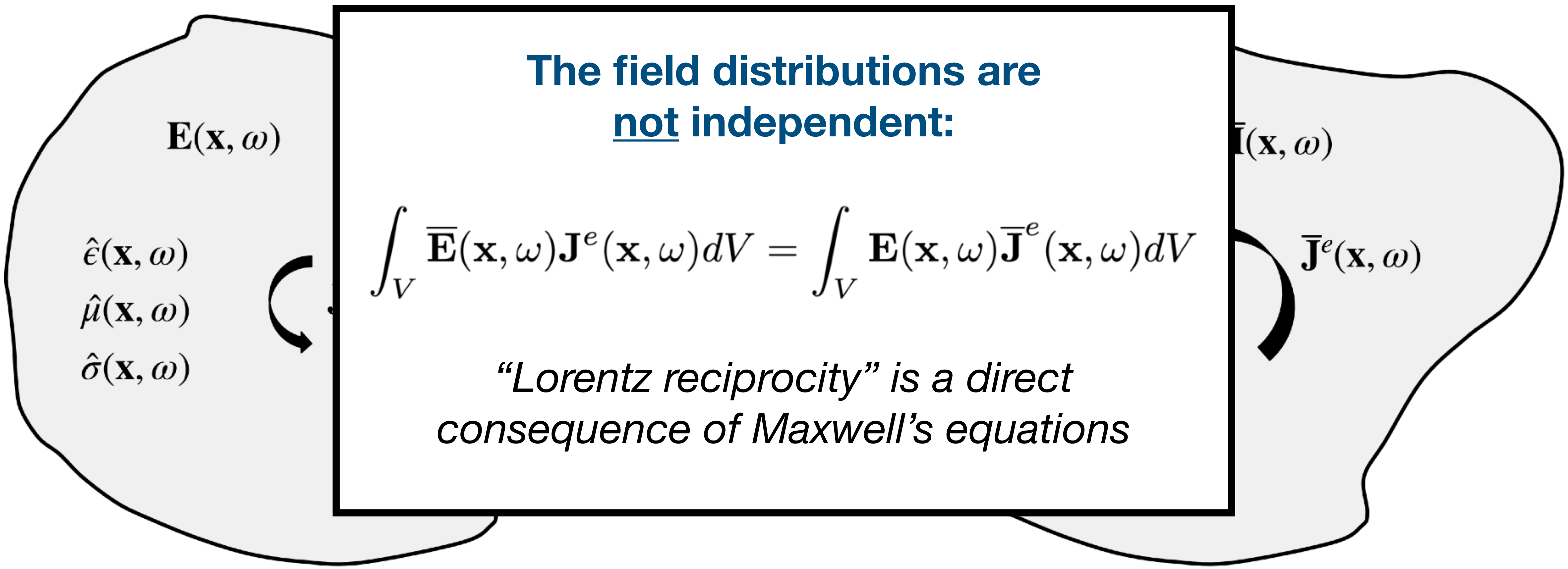
$$\bar{\mathbf{J}}^e \xrightarrow{\text{Maxwell's eqns.}} \bar{\mathbf{E}},\ \bar{\mathbf{H}}$$

External current distribution — Field distributions

Lorentz reciprocity

Classical electrodynamics has a built-in method to relate two different situations *(with identical geometry)*

General, linear material distribution: $\epsilon(\mathbf{x}),\ \mu(\mathbf{x}),\ \sigma(\mathbf{x})$



The field distributions are <u>not</u> independent:

$$\int_V \overline{\mathbf{E}}(\mathbf{x},\omega)\mathbf{J}^e(\mathbf{x},\omega)dV = \int_V \mathbf{E}(\mathbf{x},\omega)\overline{\mathbf{J}}^e(\mathbf{x},\omega)dV$$

"Lorentz reciprocity" is a direct consequence of Maxwell's equations

$\mathbf{J}^e \xrightarrow{\text{Maxwell's eqns.}} \mathbf{E},\ \mathbf{H}$

External current distribution — Field distributions

$\bar{\mathbf{J}}^e \xrightarrow{\text{Maxwell's eqns.}} \bar{\mathbf{E}},\ \bar{\mathbf{H}}$

External current distribution — Field distributions

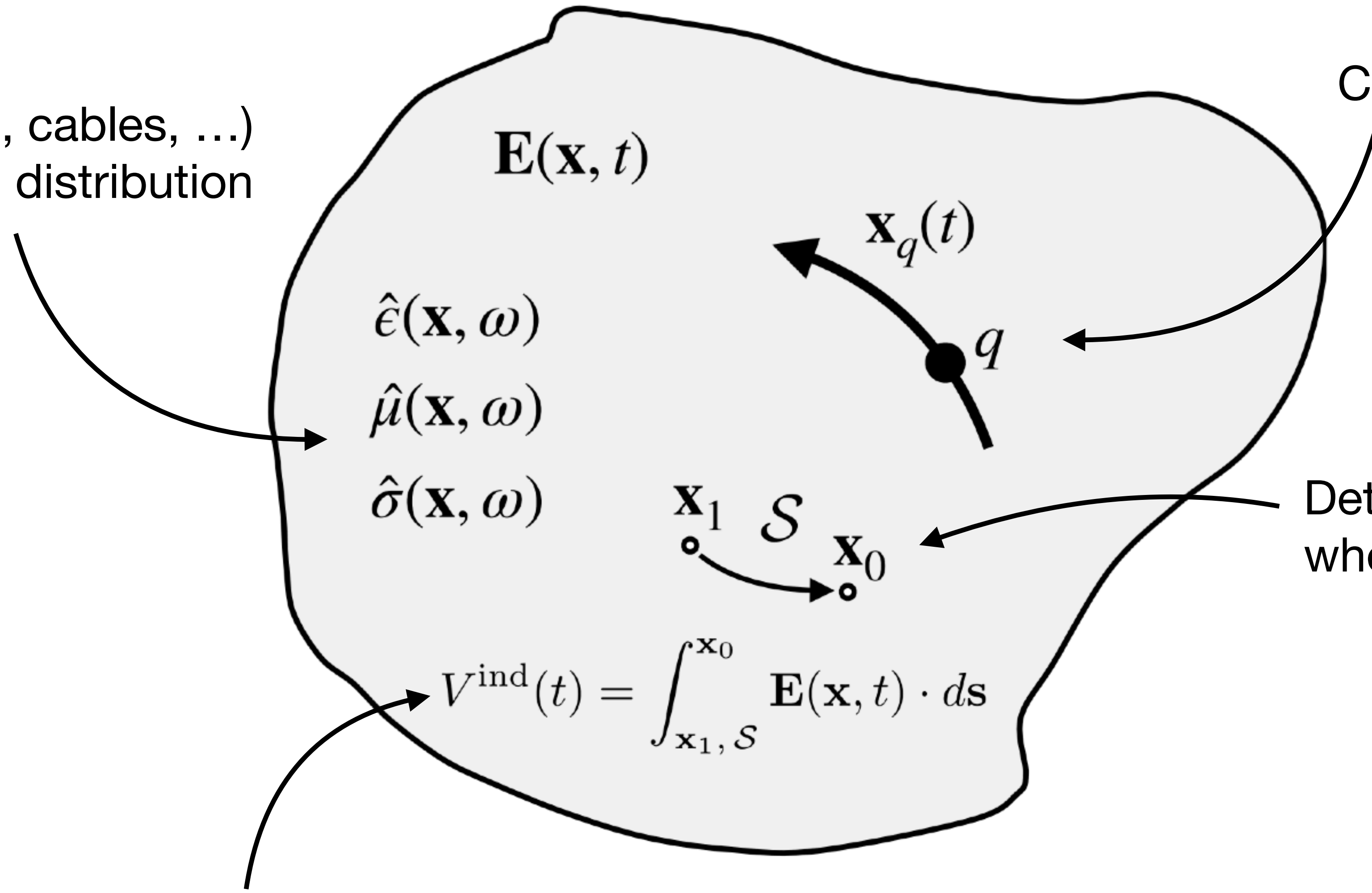
Towards a general signal theorem

Use this “duality” to compute signal induced in detector

The “primal” situation:

Detector (antennas, cables, …) encoded in material distribution

Charged particle



Detector “terminals” where signal is available

Detector “signal” is voltage difference across terminals
(measured along a specific path, $\nabla \times \mathbf{E} \neq 0$ *in general!)*

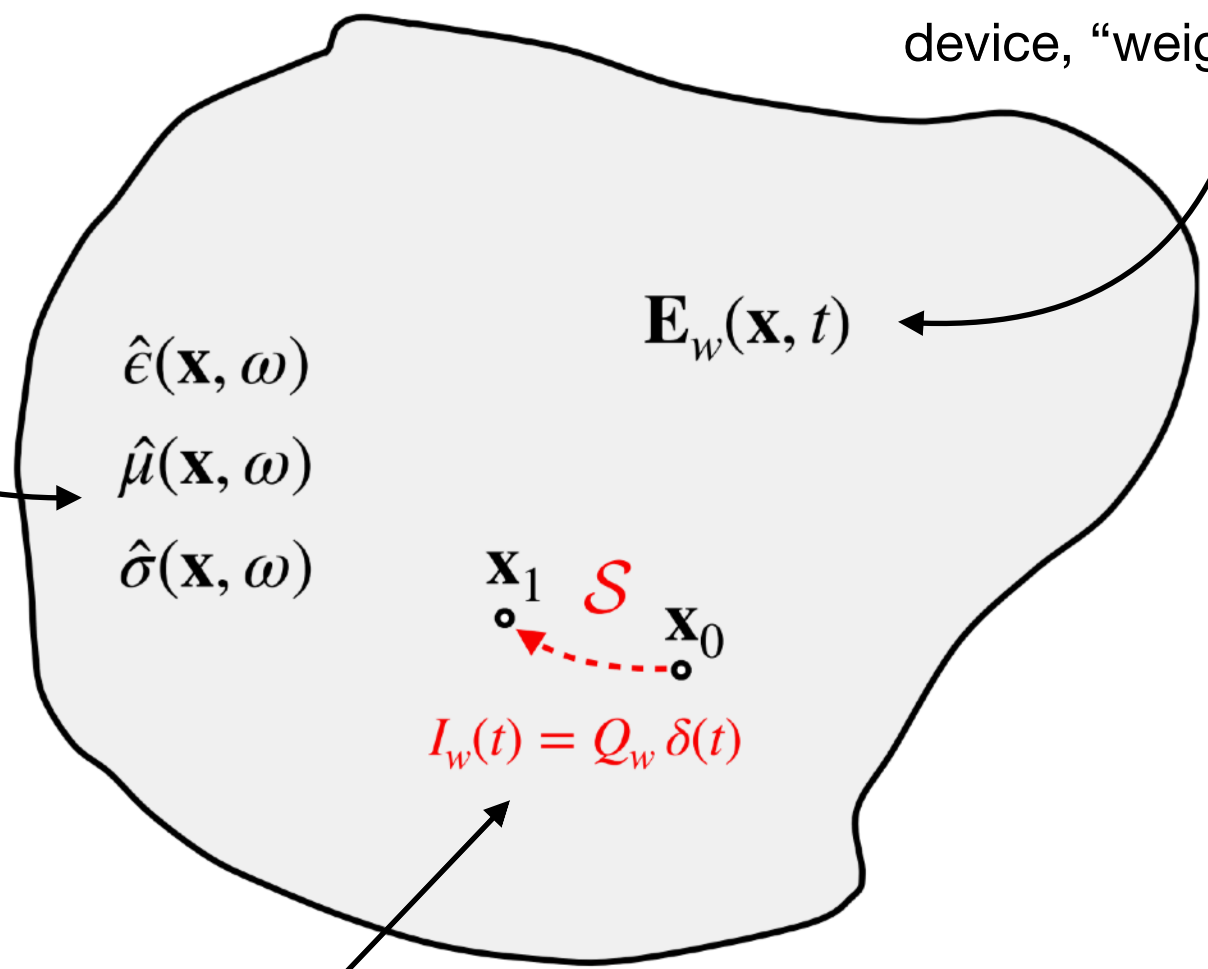
Towards a general signal theorem

Use this "duality" to compute signal induced in detector

The "dual" situation:

The same material distribution
(in general: the transposed response matrices)

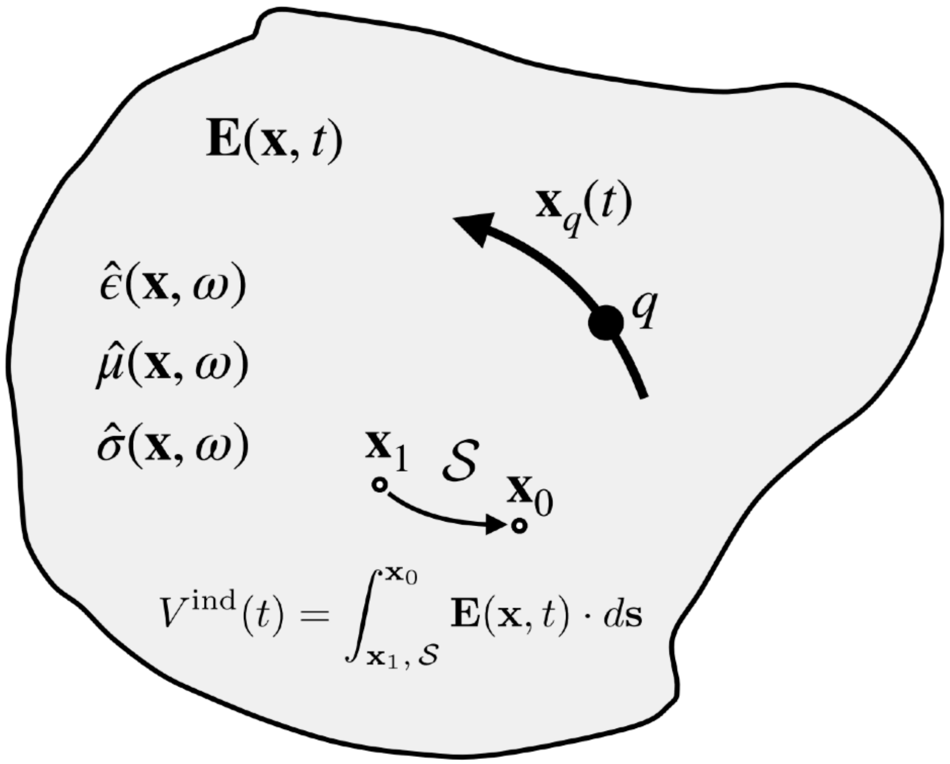
Detector acts as transmitting device, "weighting field distribution"

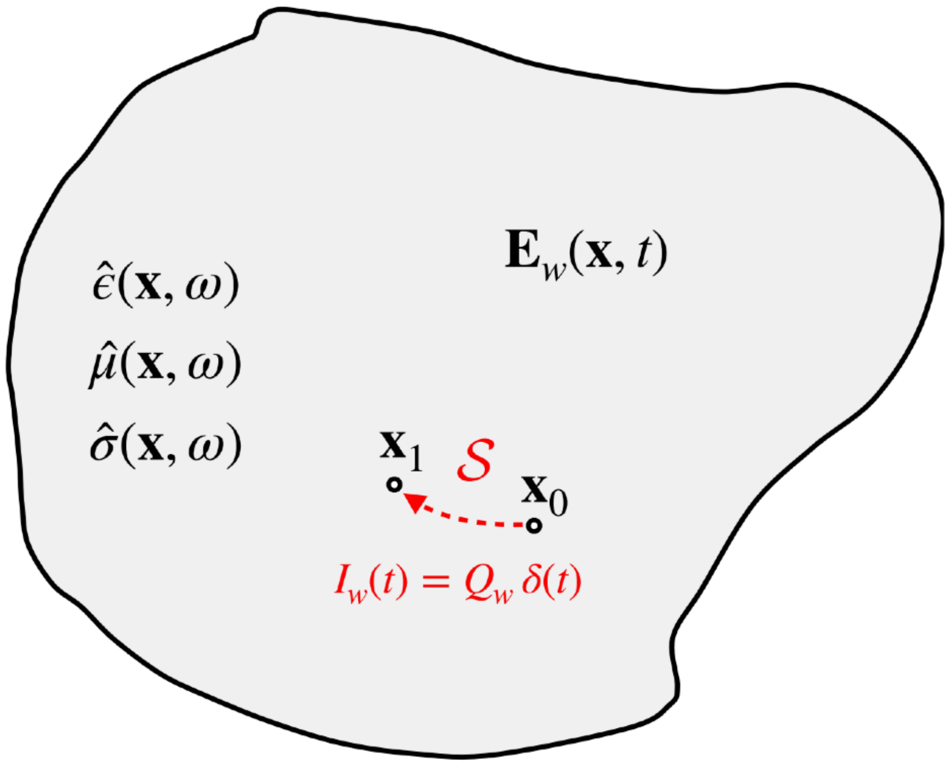


Current source attached to detector terminals:
delta-like current applied along the signal-defining path

Towards a general signal theorem

Use this "duality" to compute signal induced in detector





$$\int_V \overline{\mathbf{E}}(\mathbf{x}, \omega) \mathbf{J}^e(\mathbf{x}, \omega) dV = \int_V \mathbf{E}(\mathbf{x}, \omega) \overline{\mathbf{J}}^e(\mathbf{x}, \omega) dV$$

$$V^{\mathrm{ind}}(\omega) = \int_{\mathbf{x}_1, \mathcal{S}}^{\mathbf{x}_0} \mathbf{E}(\mathbf{x}, \omega) d\mathbf{s} = -\frac{1}{I_w(\omega)} \int_V \mathbf{E}_w(\mathbf{x}, \omega) \mathbf{J}^e(\mathbf{x}, \omega) dV$$

A fully general signal theorem

In the time-domain, this is

$$V^{\text{ind}}(t) = -\frac{q}{Q_w}\int_{-\infty}^{\infty} \mathbf{E}_w(\mathbf{x}_q(t'), t - t')\dot{\mathbf{x}}_q(t')dt'$$

Normalising constant — Weighting field — Particle trajectory

"Weighting field": Green's function for detector signal

Encodes information about detector geometry & environment;
reciprocity defines concrete algorithm to compute it

Fully general, no approximations

holds exactly for all linear, anisotropic materials;
approximately for nonlinear, anisotropic materials

The idea: reciprocity

Alice

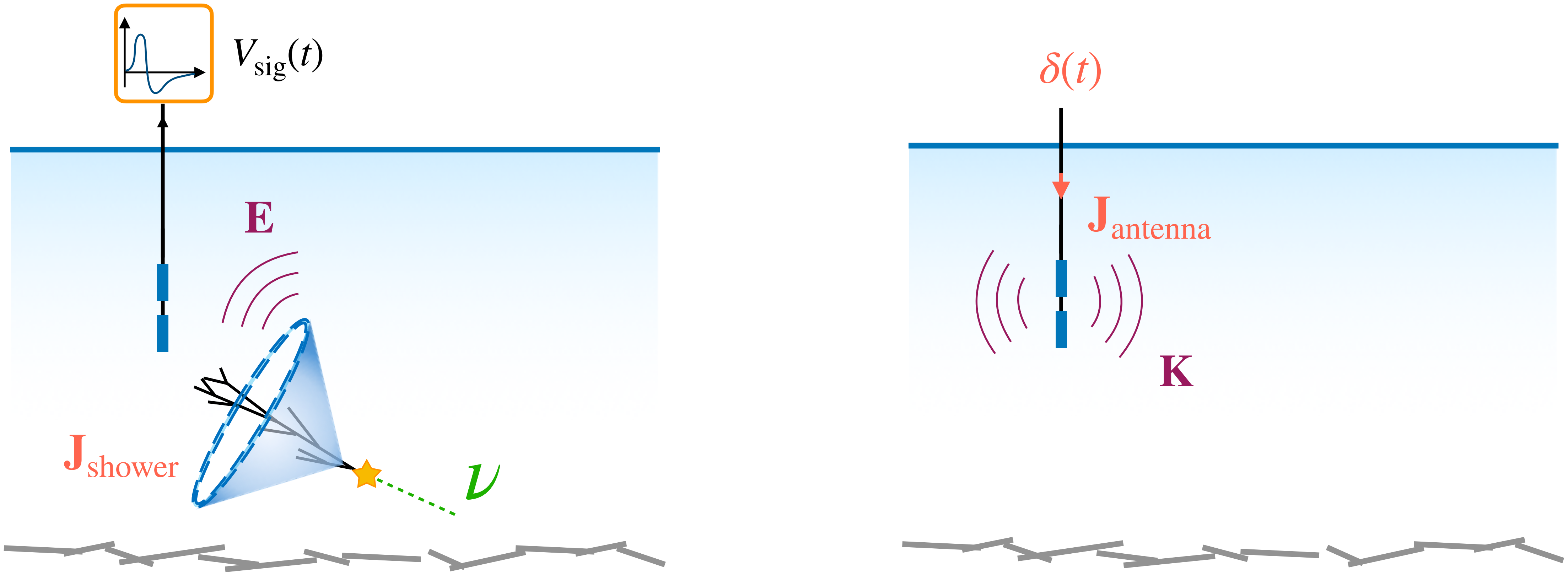
Alice can see Bob …

… if and only if
Bob can see Alice

Bob

The idea: reciprocity

Electromagnetic “communication channels” are symmetric



Antenna is receiver, produces voltage signal

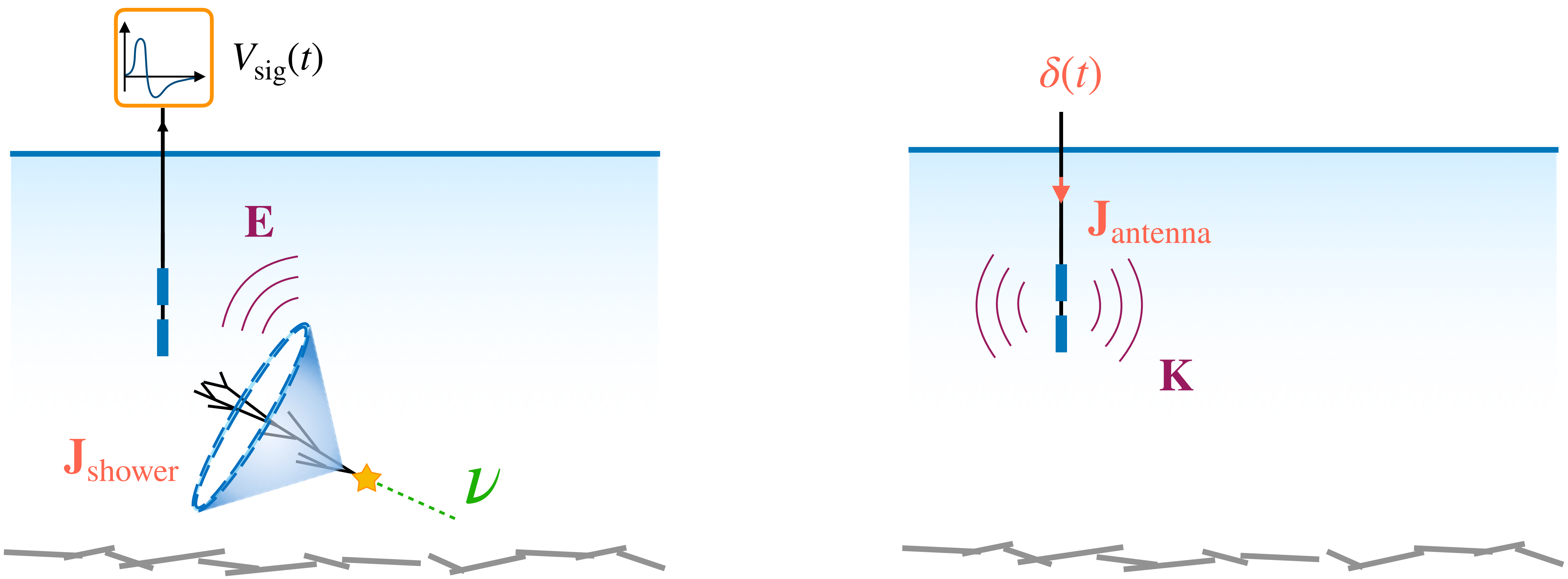
Feed delta-like signal into antenna, is transmitter

Antenna “sees” shower ⟷ *Shower “sees” antenna*

Reciprocity

The idea: reciprocity

Electromagnetic "communication channels" are symmetric



$$V_{\mathrm{sig}}(t) = \int dt' d^3x' \; \mathbf{K}(\mathbf{x}', t - t') \cdot \mathbf{J}_{\mathrm{shower}}(\mathbf{x}', t')$$

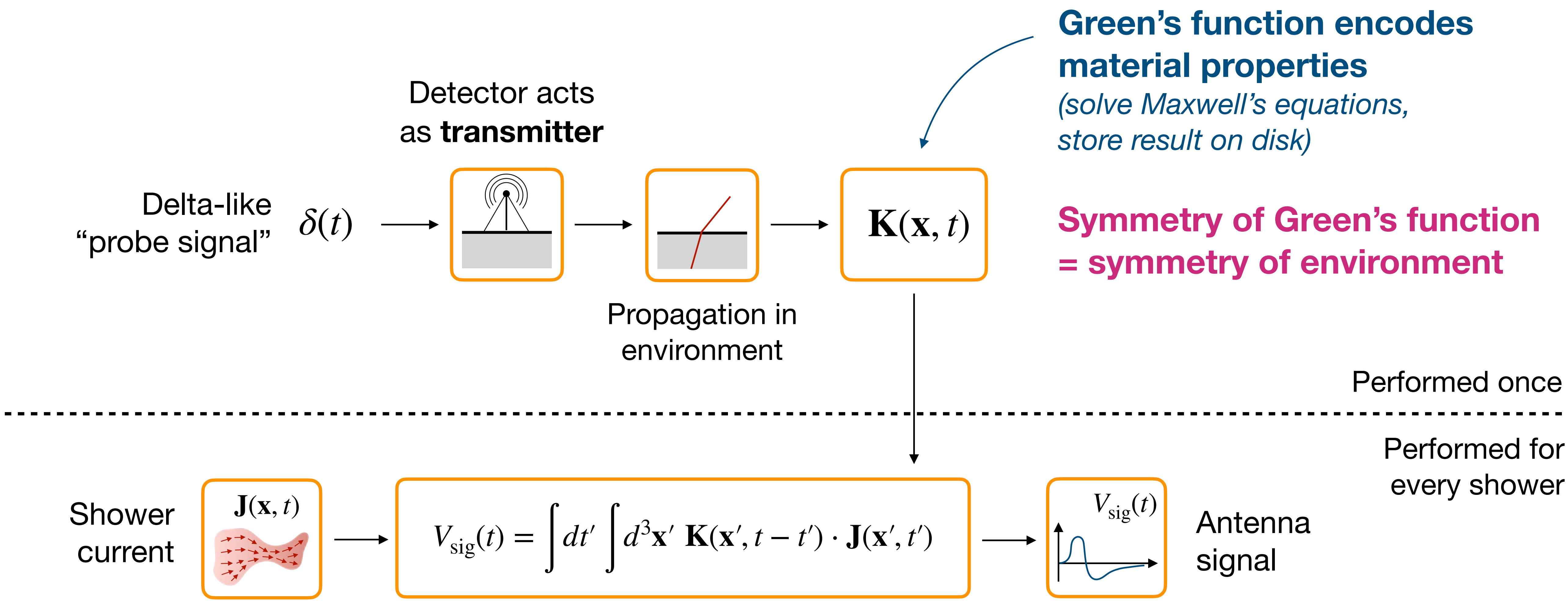
The electric field transmitted by the antenna is a Green's function for the received signal …

*… for **all linear media***
(even anisotropic, frequency-dependent ones)

W. Riegler, **PW**, NIM A, 980, 164471 (2020)

A reciprocal simulator

This makes fully-electrodynamic signal calculations possible!

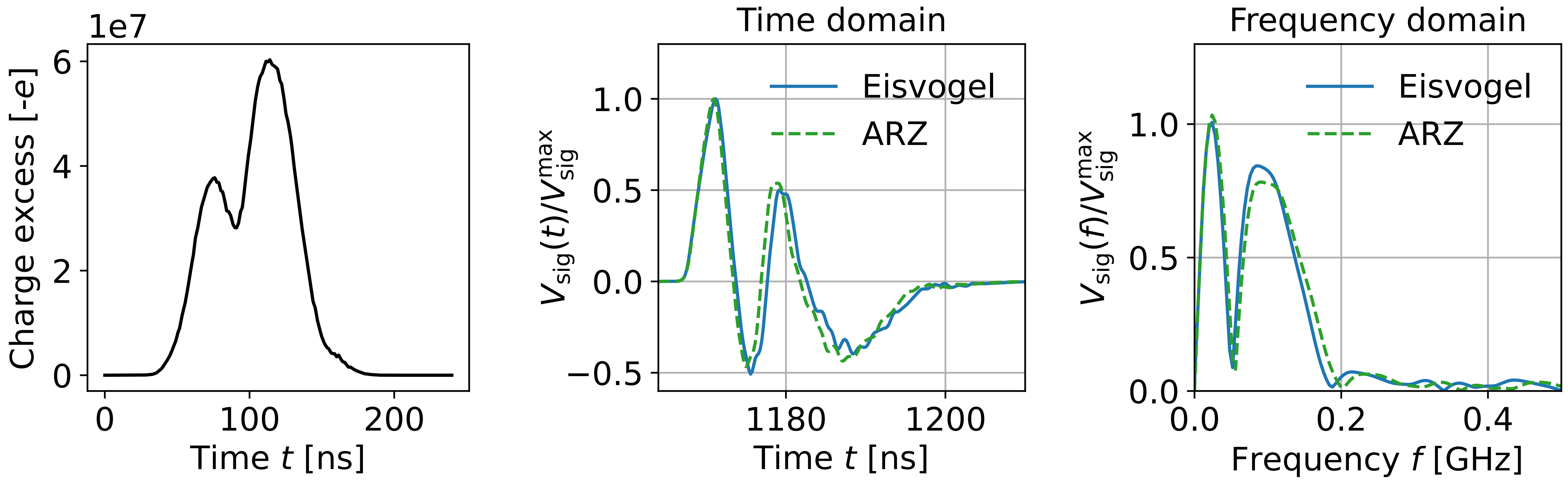


Convolution with shower is *fast*!

Calculation of complicated radiation propagation amortized into Green's function

Comparison with ARZ
(as implemented in NuRadioMC)

1-dim profile of electromagnetic shower developing in homogeneous medium with $n = 1.78$



Good agreement!