A Path-tracing Monte Carlo Library for 3D Radiative Transfer in Highly Resolved Cloudy Atmospheres

Najda Villefranque[1,2], Richard Fournier[2], Fleur Couvreux[1], Stéphane Blanco[2], Céline Cornet[3], Vincent Eymet[4], Vincent Forest[4], Jean-Marc Tregan[2]

[1] Centre National de Recherches Météorologiques, Toulouse
[2] Laboratoire Plasma et Conversion d'Energie, Toulouse
[3] Laboratoire d'Optique Atmosphérique, Lille
[4] Meso-Star, Toulouse

June 28, 2019

Outline

Context: radiation in clouds, Monte Carlo

Why do we need new Monte Carlo tools?

Transfering expertise from computer graphics to atmospheric optics

Implementation and performance tests

Outlook: a parameterisation of 3D radiative effects of clouds

Outline

Context: radiation in clouds, Monte Carlo

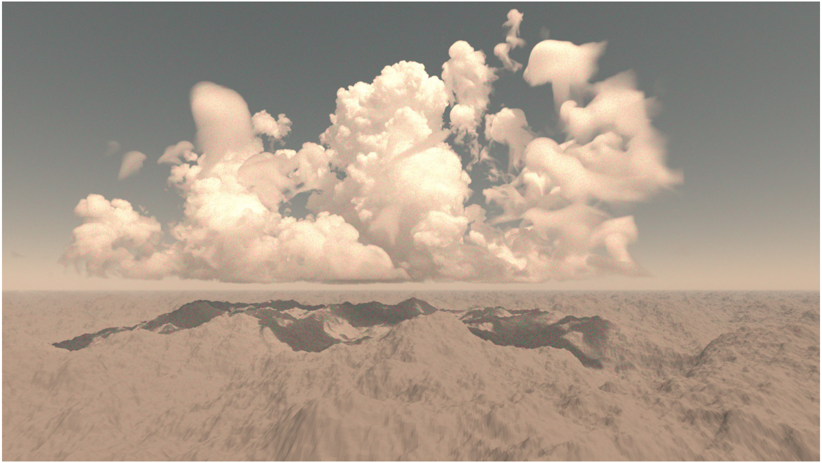
Why do we need new Monte Carlo tools?

Transfering expertise from computer graphics to atmospheric optics

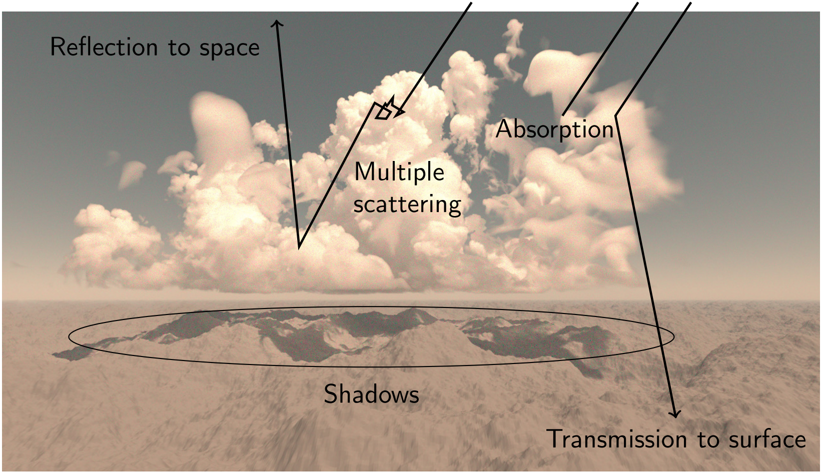
Implementation and performance tests

Outlook: a parameterisation of 3D radiative effects of clouds

How do clouds affect solar radiation?

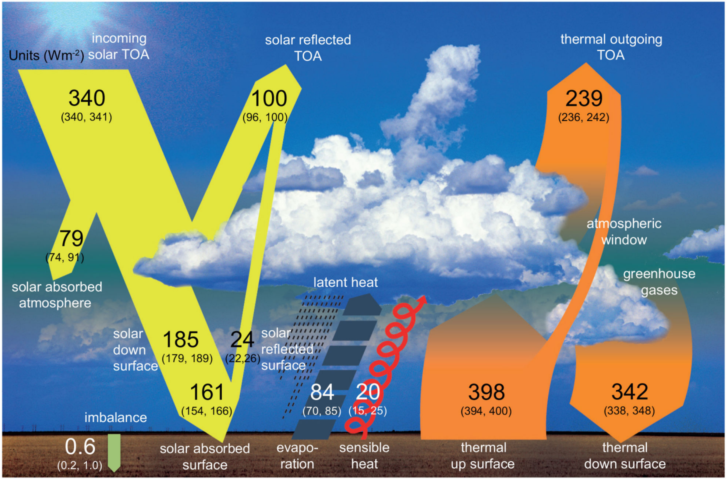


How do clouds affect solar radiation?



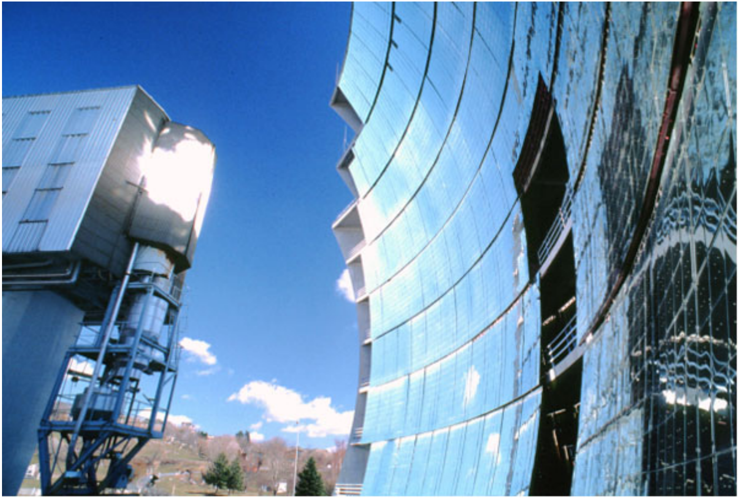
Clouds redistribute solar radiation through scattering and absorption.
Scattering leads to either reflection to space or transmission to surface.
Absorption heats the atmosphere. Clouds create shadows at the surface.

Why do we care about the impact of clouds on radiation?



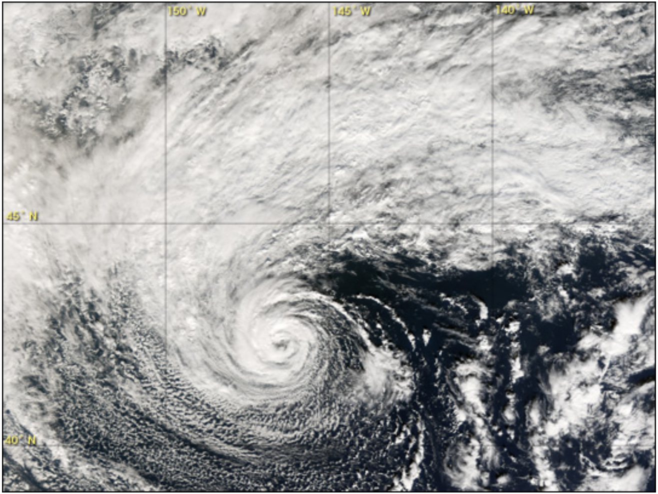
Weather & climate: understand and predict the Earth energy cycle & budget

Why do we care about the impact of clouds on radiation?



Weather & climate: understand and predict the Earth energy cycle & budget
Solar energy: predict the amount of solar radiation that will reach a solar furnace

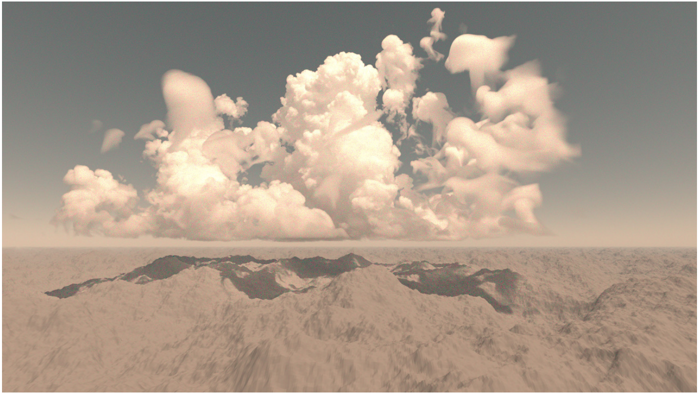
Why do we care about the impact of clouds on radiation?



Weather & climate: understand and predict the Earth energy cycle & budget
Solar energy: predict the amount of solar radiation that will reach a solar furnace
Atmospheric observation: use radiation observations to infer atmospheric state

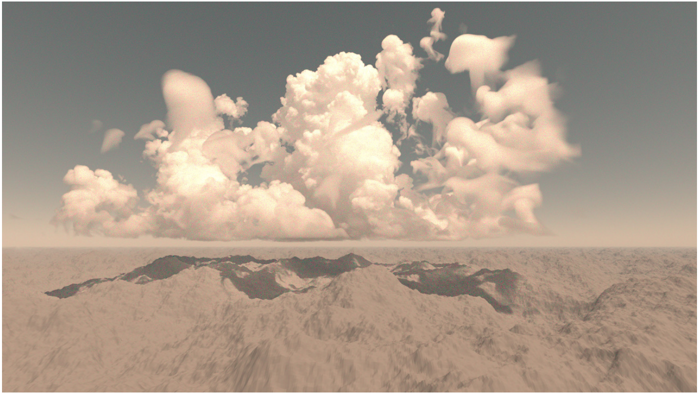
How can we measure the impact of clouds on radiation?

Observation is difficult.



How can we measure the impact of clouds on radiation?

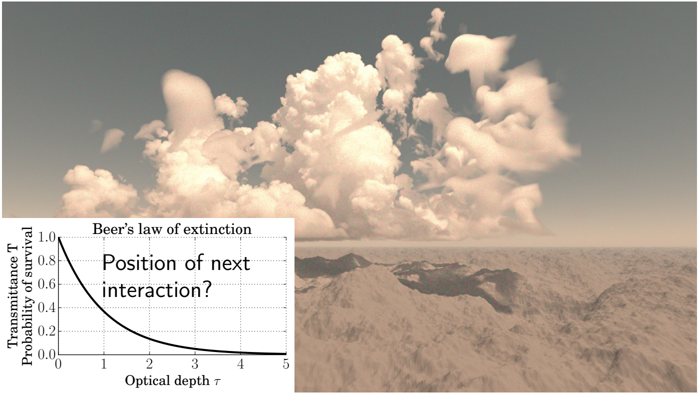
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)



How can we measure the impact of clouds on radiation?

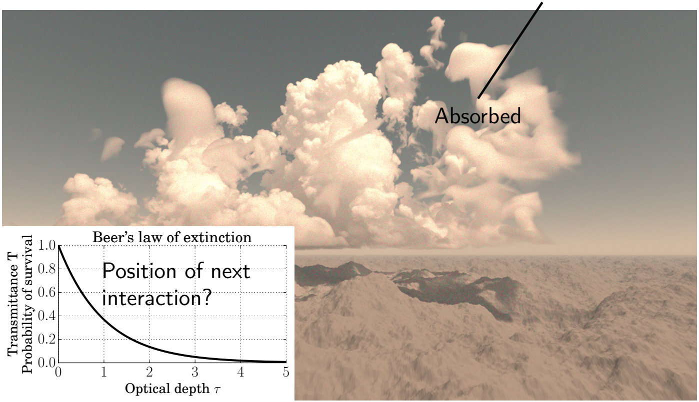
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)

Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

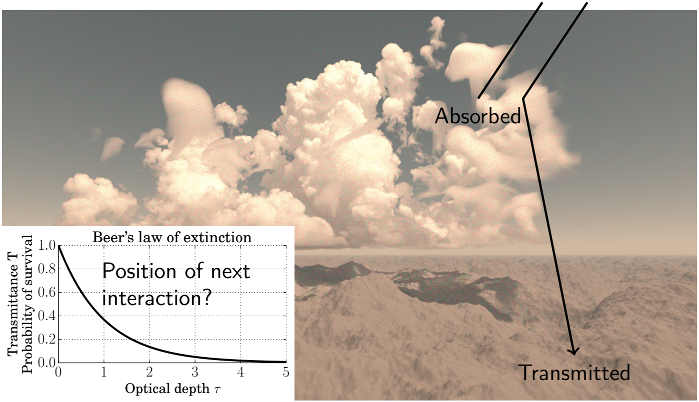
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)
Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

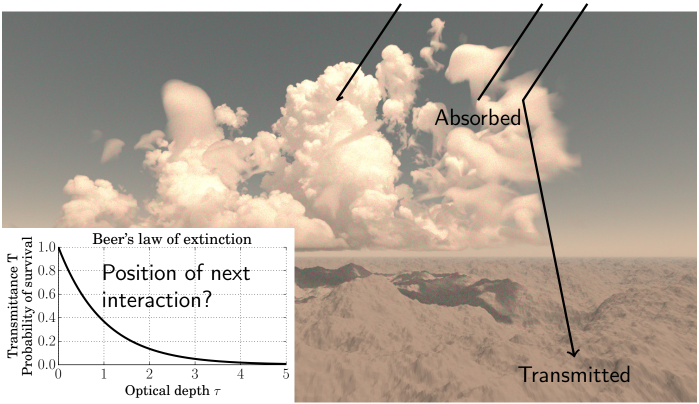
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)

Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

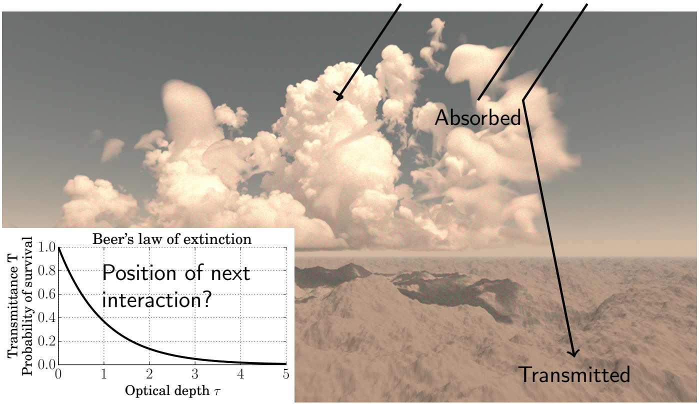
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)
Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

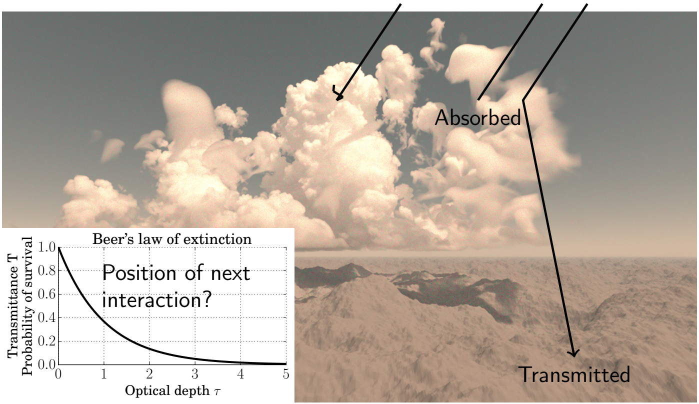
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)

Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

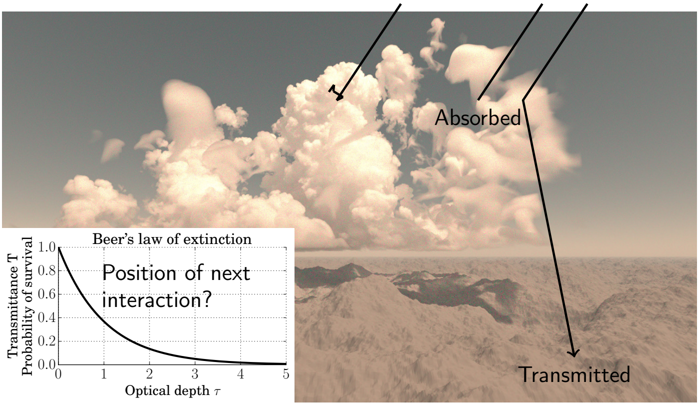
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)
Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

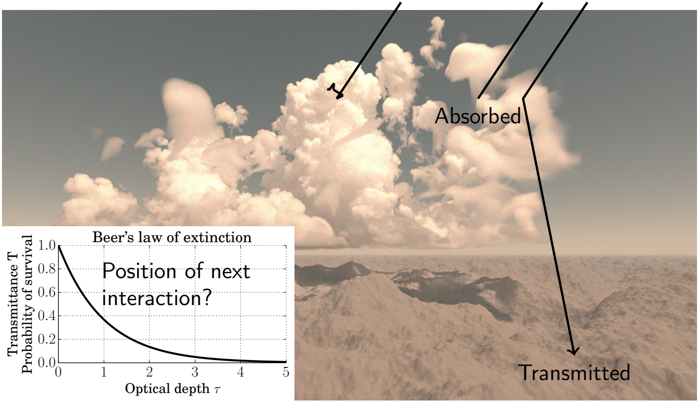
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)

Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



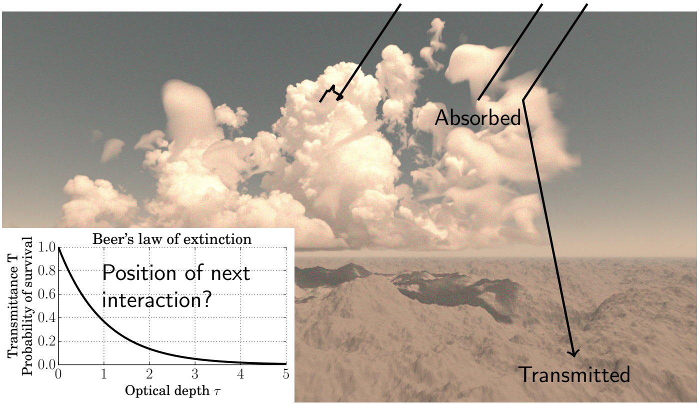
How can we measure the impact of clouds on radiation?

Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)
Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



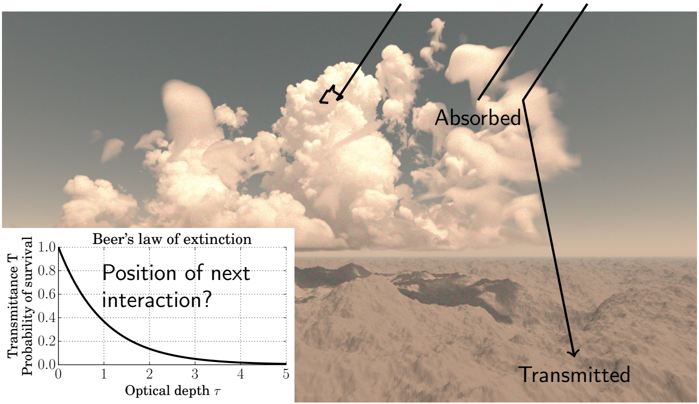
How can we measure the impact of clouds on radiation?

Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)
Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



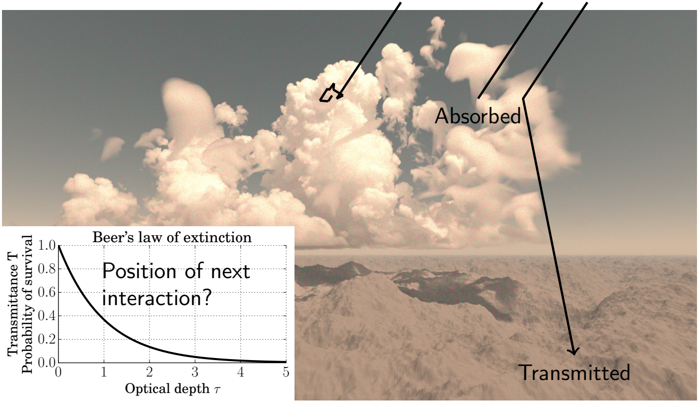
How can we measure the impact of clouds on radiation?

Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)
Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

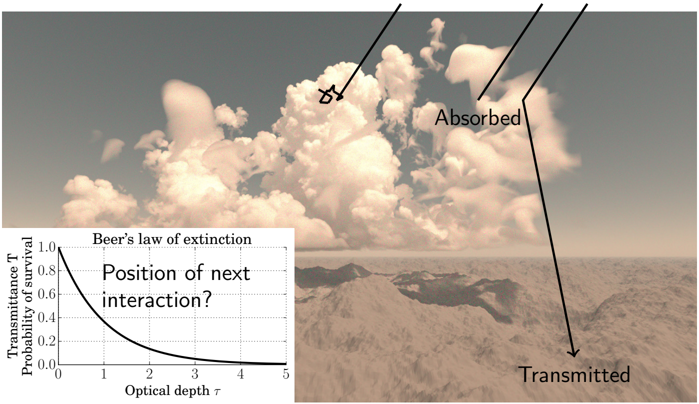
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)
Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

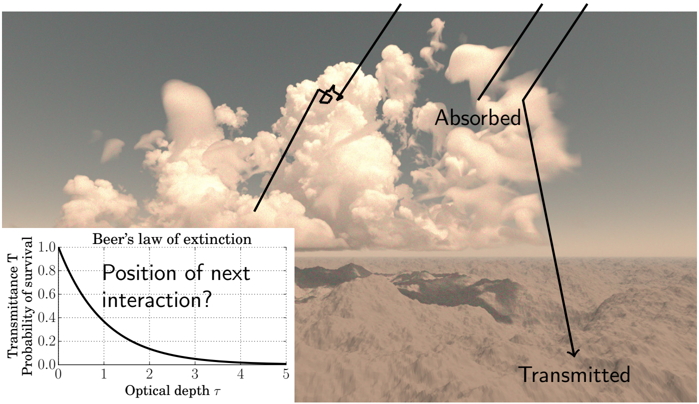
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)

Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

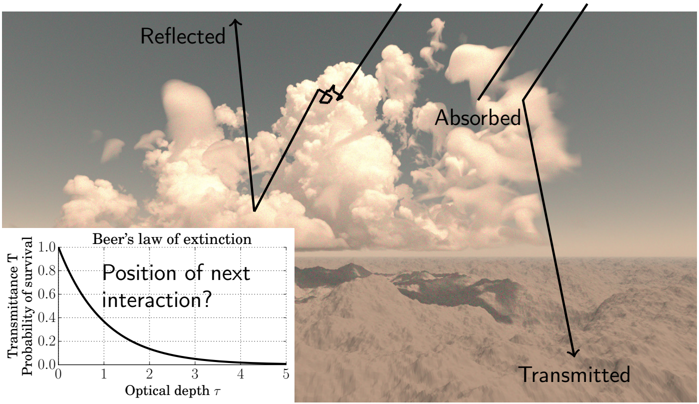
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)
Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

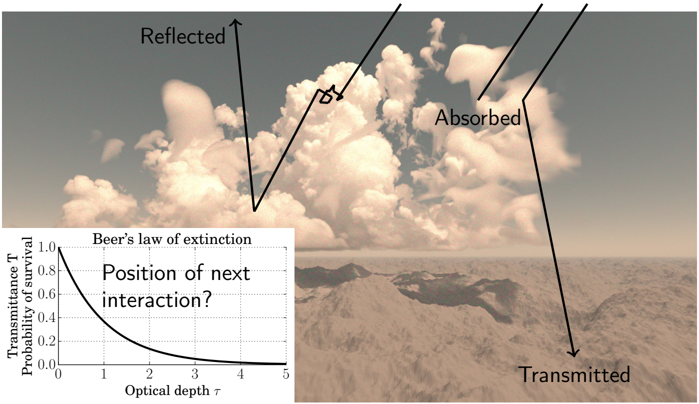
Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)

Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.



How can we measure the impact of clouds on radiation?

Observation is difficult. Reference models exist e.g. SHDOM (Evans, 1998), **Monte Carlo** (MC) (Marchuk et al., 1980b; Mayer, 2009)
Path-tracking MC methods are slow but handle infinite complexity, by tracking N paths by randomly sampling physical laws, interpreted as pdfs.

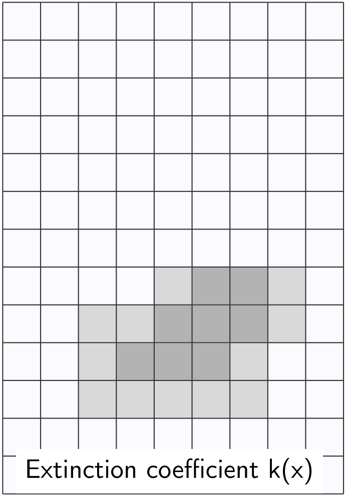


MC methods are accurate and reliable: the distribution of sampled paths converges towards the true density of paths. Convergence rate $\sim \sqrt{N}$

Outline

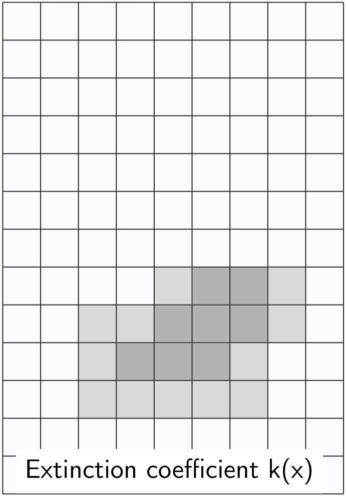
What is going on in the existing Monte Carlo codes?

The next position of interaction depends on optical depth
$\tau_s = \int_0^s k(\sigma) \mathrm{d}\sigma$ (integral of extinction along the ray)



What is going on in the existing Monte Carlo codes?

The next position of interaction depends on optical depth $\tau_s = \int_0^s k(\sigma)\mathrm{d}\sigma$ (integral of extinction along the ray)

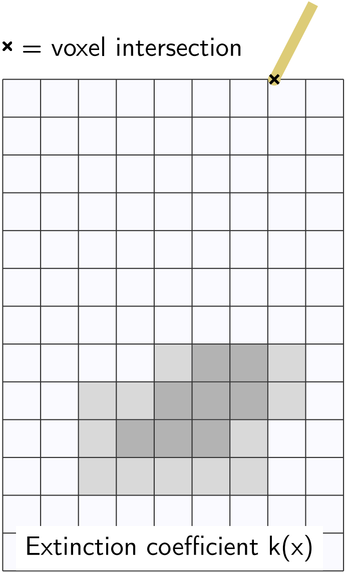


Basic steps of a MC algorithm:
a) Randomly sample the optical depth the ray can survive, $\tau_s \sim \mathcal{E}(1)$
b) Go to s, the location of interaction

What is going on in the existing Monte Carlo codes?

The next position of interaction depends on optical depth $\tau_s = \int_0^s k(\sigma)\mathrm{d}\sigma$ (integral of extinction along the ray)

× = voxel intersection

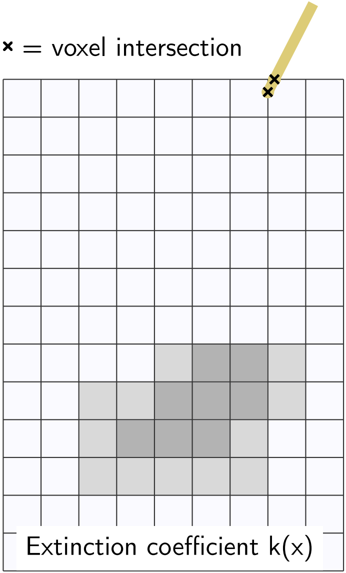


Basic steps of a MC algorithm:
a) Randomly sample the optical depth the ray can survive, $\tau_s \sim \mathcal{E}(1)$
b) Go to s, the location of interaction

- ▷ Problem: cannot jump directly to s because of the integral.

What is going on in the existing Monte Carlo codes?

The next position of interaction depends on optical depth $\tau_s = \int_0^s k(\sigma)\mathrm{d}\sigma$ (integral of extinction along the ray)



Basic steps of a MC algorithm:

a) Randomly sample the optical depth the ray can survive, $\tau_s \sim \mathcal{E}(1)$

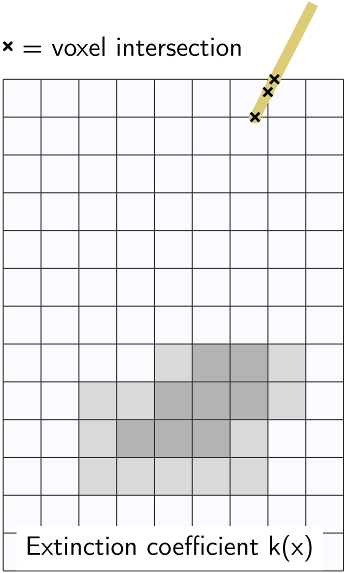
b) Go to s, the location of interaction

- Problem: cannot jump directly to s because of the integral.
- Solution: cross voxels, compute τ until s is found ($\tau = \tau_s$)

$$\tau = \sum_i k_i l_i$$

What is going on in the existing Monte Carlo codes?

The next position of interaction depends on optical depth $\tau_s = \int_0^s k(\sigma)\mathrm{d}\sigma$ (integral of extinction along the ray)



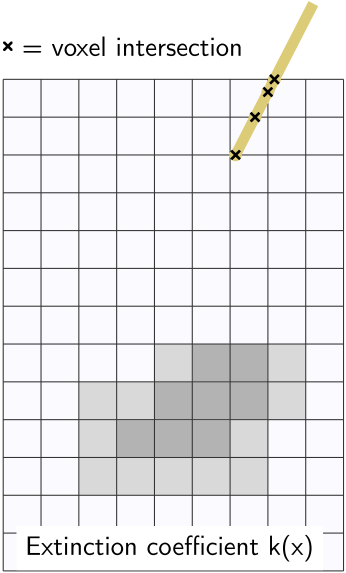
Basic steps of a MC algorithm:
a) Randomly sample the optical depth the ray can survive, $\tau_s \sim \mathcal{E}(1)$
b) Go to s, the location of interaction

- Problem: cannot jump directly to s because of the integral.
- Solution: cross voxels, compute τ until s is found ($\tau = \tau_s$)

$$\tau = \sum_i k_i l_i$$

What is going on in the existing Monte Carlo codes?

The next position of interaction depends on optical depth $\tau_s = \int_0^s k(\sigma)\mathrm{d}\sigma$ (integral of extinction along the ray)



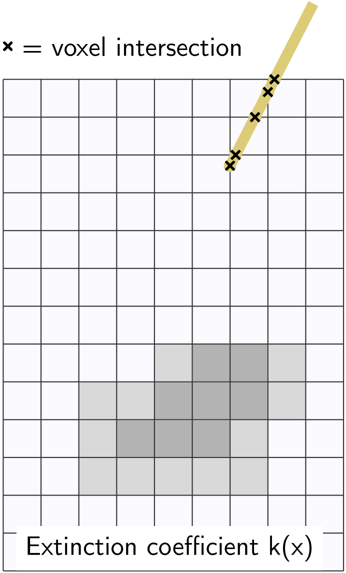
Basic steps of a MC algorithm:
a) Randomly sample the optical depth the ray can survive, $\tau_s \sim \mathcal{E}(1)$
b) Go to s, the location of interaction

- Problem: cannot jump directly to s because of the integral.
- Solution: cross voxels, compute τ until s is found ($\tau = \tau_s$)

$$\tau = \sum_i k_i l_i$$

What is going on in the existing Monte Carlo codes?

The next position of interaction depends on optical depth
$\tau_s = \int_0^s k(\sigma)\mathrm{d}\sigma$ (integral of extinction along the ray)



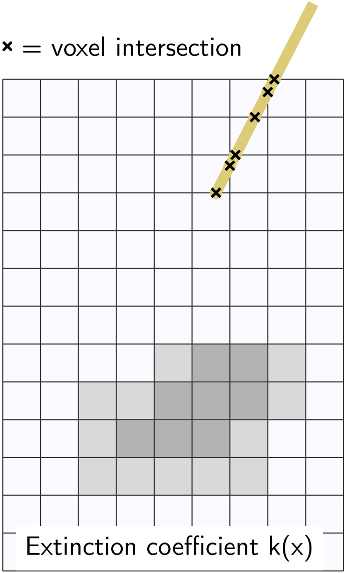
Basic steps of a MC algorithm:
a) Randomly sample the optical depth the ray can survive, $\tau_s \sim \mathcal{E}(1)$
b) Go to s, the location of interaction

- Problem: cannot jump directly to s because of the integral.
- Solution: cross voxels, compute τ until s is found ($\tau = \tau_s$)

$$\tau = \sum_i k_i l_i$$

What is going on in the existing Monte Carlo codes?

The next position of interaction depends on optical depth
$\tau_s = \int_0^s k(\sigma)\mathrm{d}\sigma$ (integral of extinction along the ray)



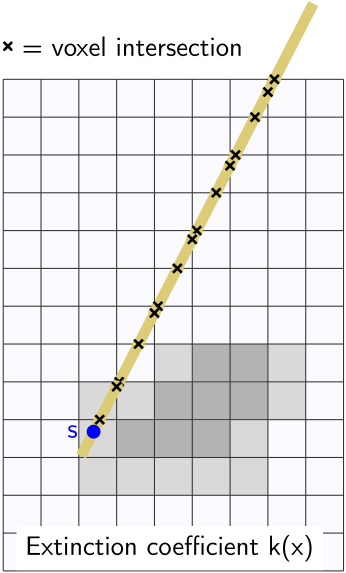
Basic steps of a MC algorithm:
a) Randomly sample the optical depth the ray can survive, $\tau_s \sim \mathcal{E}(1)$
b) Go to s, the location of interaction

- Problem: cannot jump directly to s because of the integral.
- Solution: cross voxels, compute τ until s is found ($\tau = \tau_s$)

$$\tau = \sum_i k_i l_i$$

What is going on in the existing Monte Carlo codes?

The next position of interaction depends on optical depth $\tau_s = \int_0^s k(\sigma)\mathrm{d}\sigma$ (integral of extinction along the ray)



Basic steps of a MC algorithm:
a) Randomly sample the optical depth the ray can survive, $\tau_s \sim \mathcal{E}(1)$
b) Go to s, the location of interaction

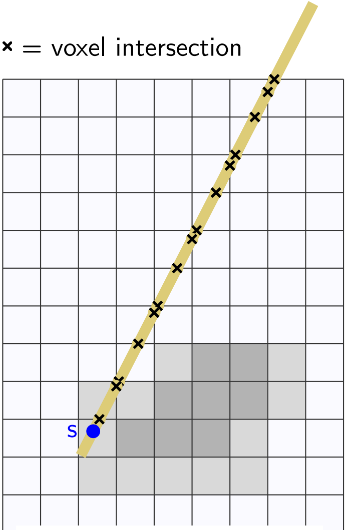
- Problem: cannot jump directly to s because of the integral.
- Solution: cross voxels, compute τ until s is found ($\tau = \tau_s$)

$$\tau = \sum_i k_i l_i$$

What is going on in the existing Monte Carlo codes?

The next position of interaction depends on optical depth
$\tau_s = \int_0^s k(\sigma)\mathrm{d}\sigma$ (integral of extinction along the ray)

× = voxel intersection



Extinction coefficient k(x)

Basic steps of a MC algorithm:
a) Randomly sample the optical depth the ray can survive, $\tau_s \sim \mathcal{E}(1)$
b) Go to s, the location of interaction

- Problem: cannot jump directly to s because of the integral.
- Solution: cross voxels, compute τ until s is found ($\tau = \tau_s$)

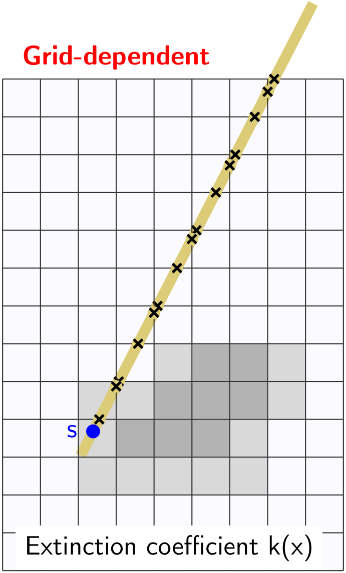
$$\tau = \sum_i k_i l_i$$

Ray tracing is grid-dependent

Computing time will ↗ with resolution

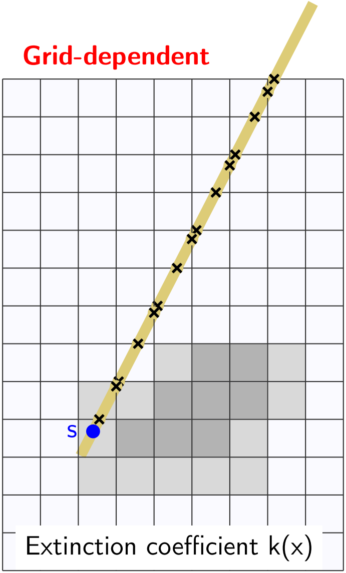
How can we avoid grid-dependent ray-tracing?

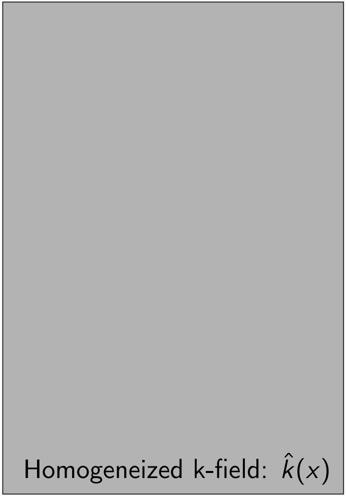
Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))



How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))



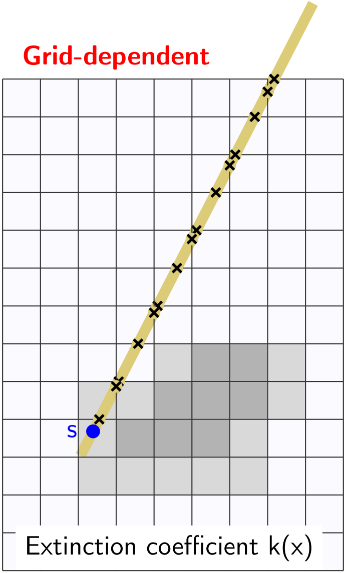


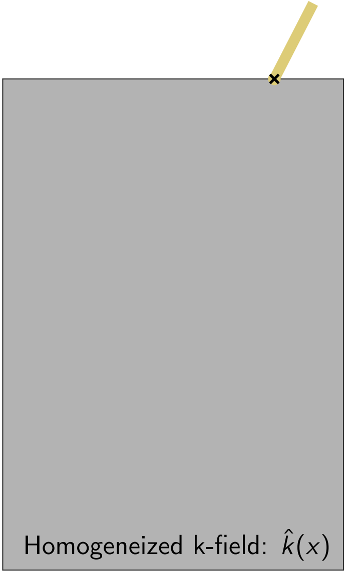
Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))

Grid-dependent



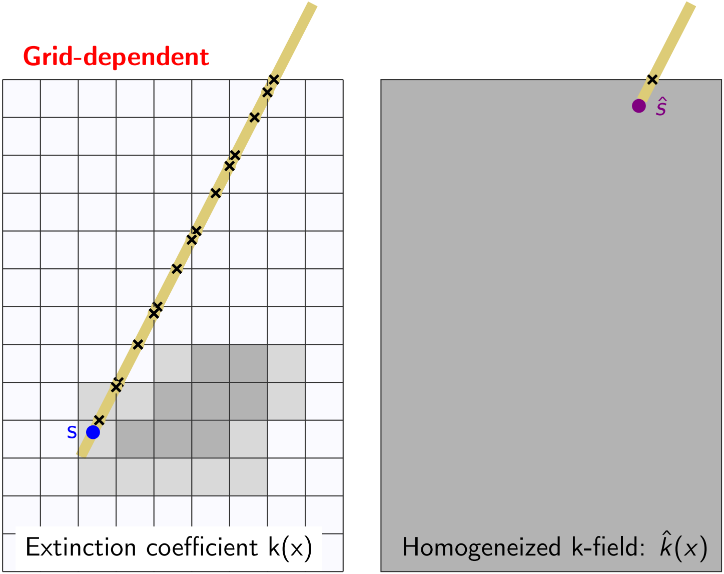


Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))



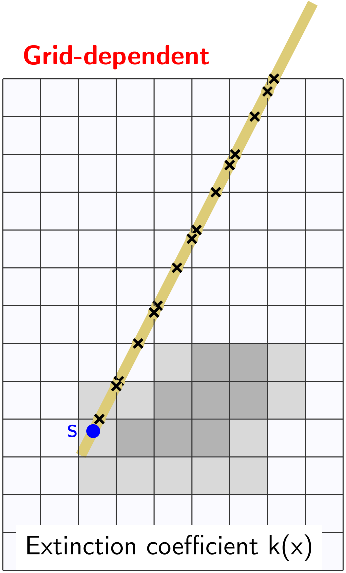
Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

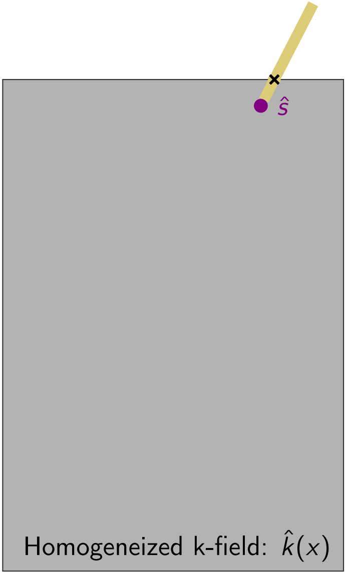
Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))





Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

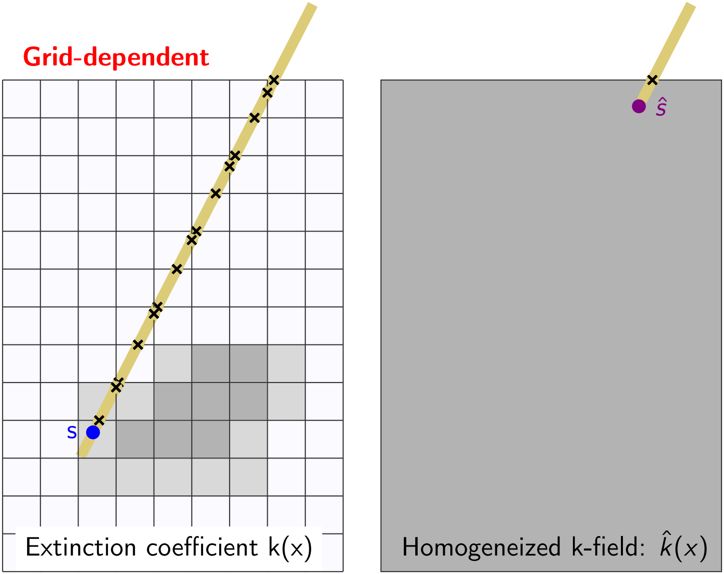
Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

Sample nature:
$\mathbb{P}_{true} = k/\hat{k}$
$\mathbb{P}_{null} = k_n/\hat{k}$

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))



Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

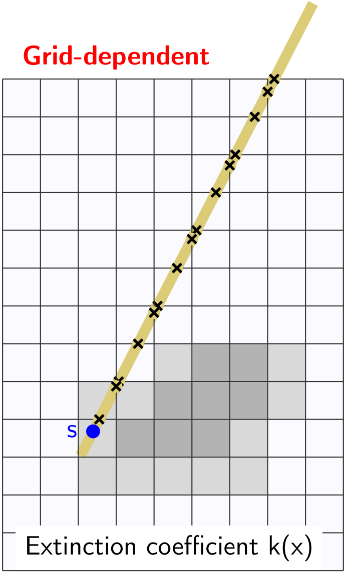
Sample nature:
$\mathbb{P}_{true} = k/\hat{k}$
$\mathbb{P}_{null} = k_n/\hat{k}$

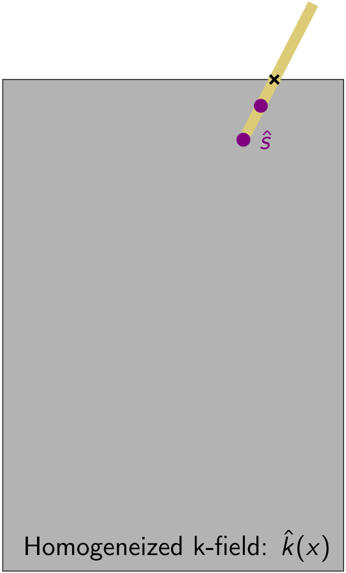
While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))

Grid-dependent





Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

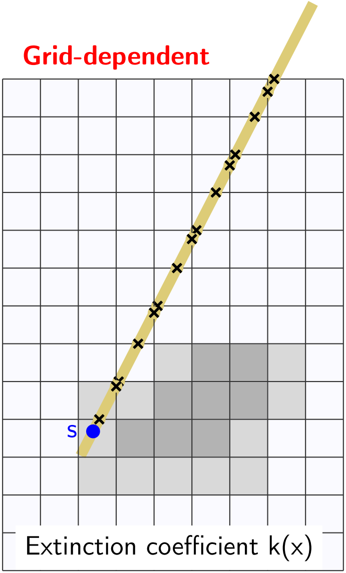
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

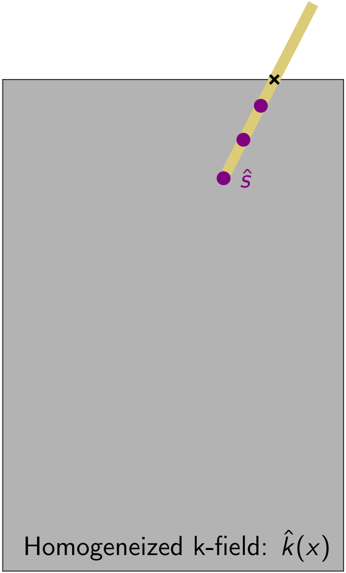
Sample nature:
$\mathbb{P}_{true} = k/\hat{k}$
$\mathbb{P}_{null} = k_n/\hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))





Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

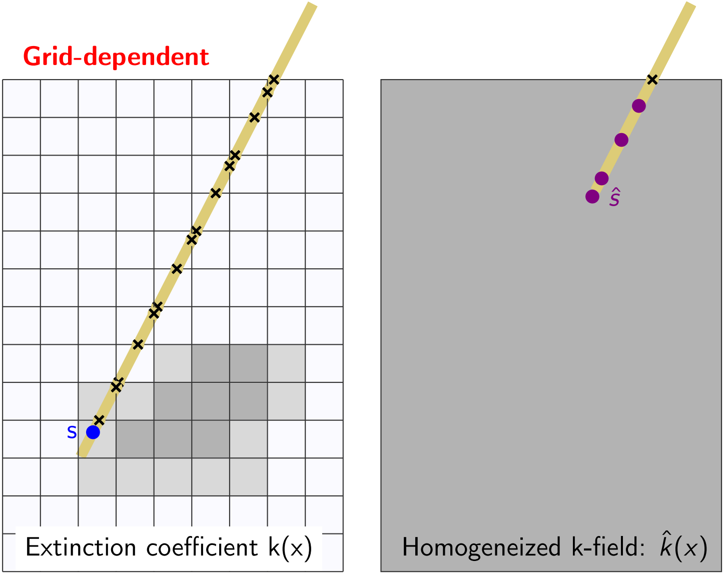
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

Sample nature:
$\mathbb{P}_{true} = k/\hat{k}$
$\mathbb{P}_{null} = k_n/\hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))



Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

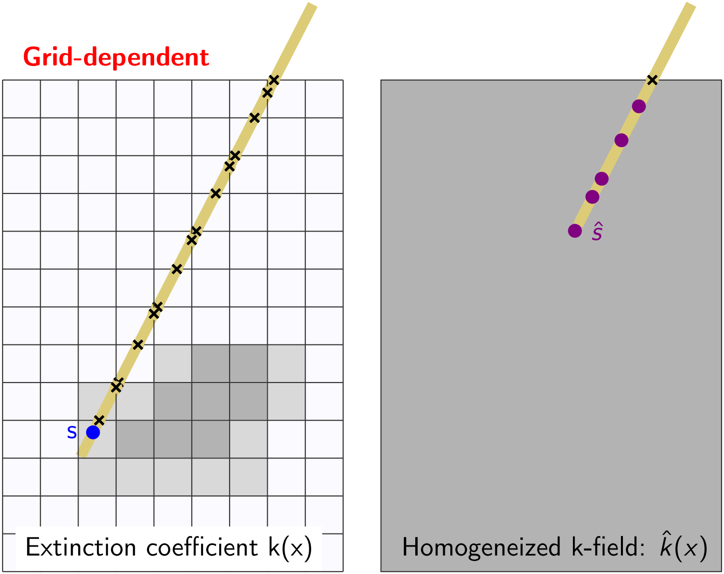
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

Sample nature:
$\mathbb{P}_{true} = k/\hat{k}$
$\mathbb{P}_{null} = k_n/\hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))



Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

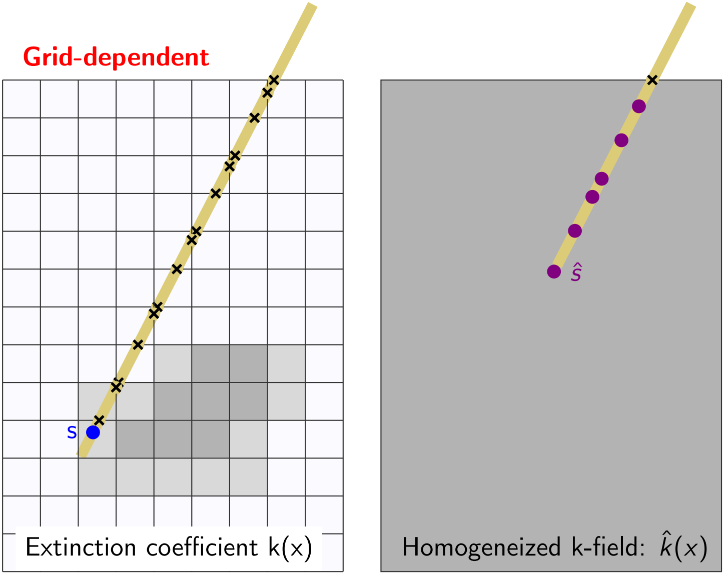
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

Sample nature:
$\mathbb{P}_{true} = k/\hat{k}$
$\mathbb{P}_{null} = k_n/\hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))



Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

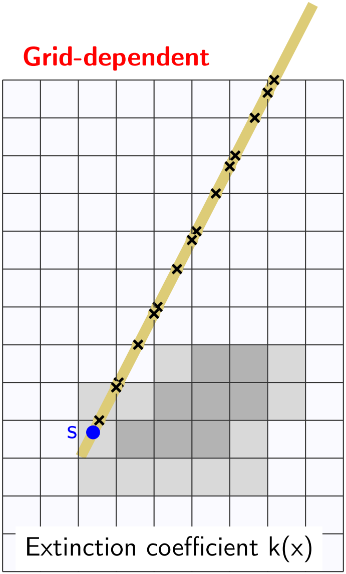
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

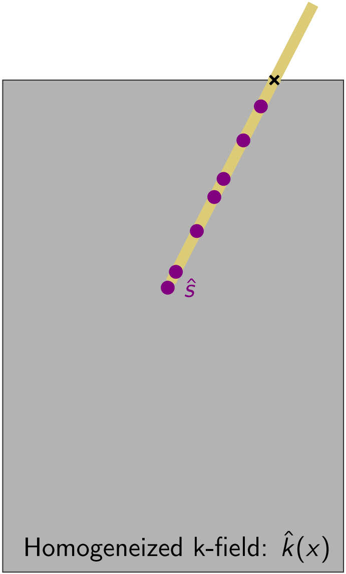
Sample nature:
$\mathbb{P}_{true} = k/\hat{k}$
$\mathbb{P}_{null} = k_n/\hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))





Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

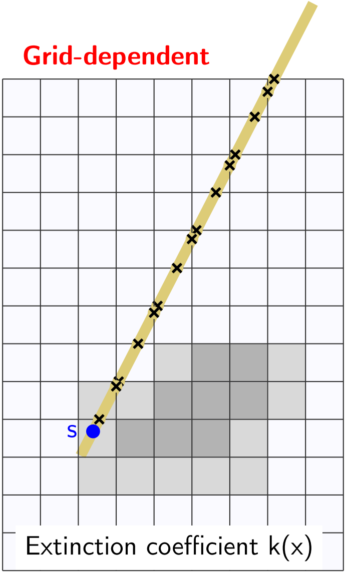
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

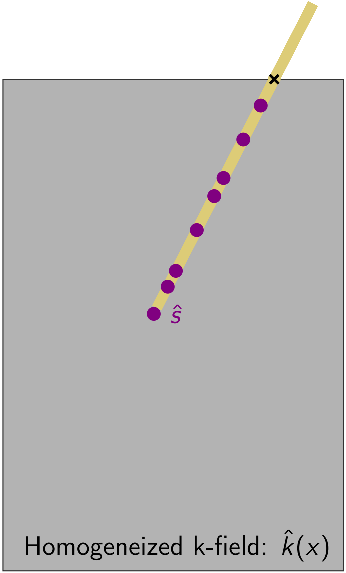
Sample nature:
$\mathbb{P}_{true} = k/\hat{k}$
$\mathbb{P}_{null} = k_n/\hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))





Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

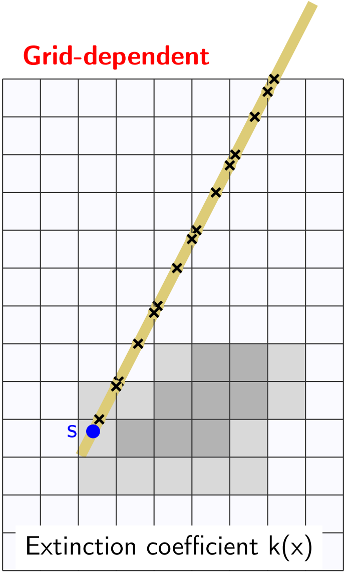
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

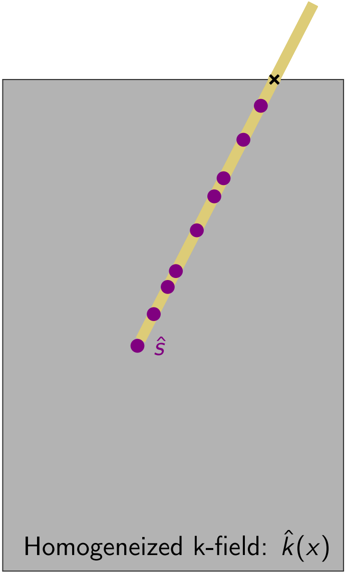
Sample nature:
$\mathbb{P}_{true} = k / \hat{k}$
$\mathbb{P}_{null} = k_n / \hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))





Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

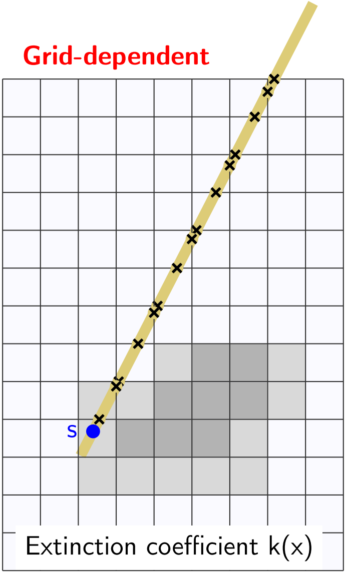
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

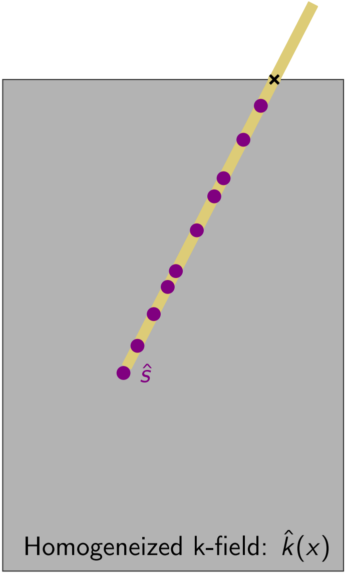
Sample nature:
$\mathbb{P}_{true} = k/\hat{k}$
$\mathbb{P}_{null} = k_n/\hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))





Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

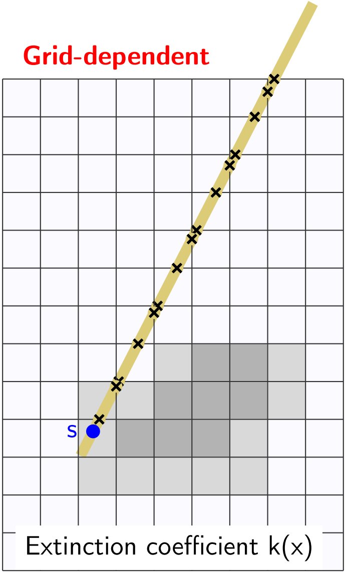
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

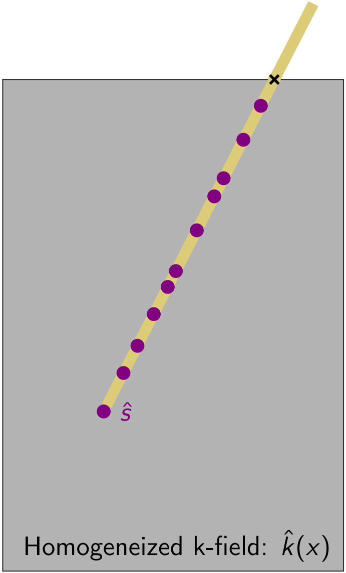
Sample nature:
$\mathbb{P}_{true} = k/\hat{k}$
$\mathbb{P}_{null} = k_n/\hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))





Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

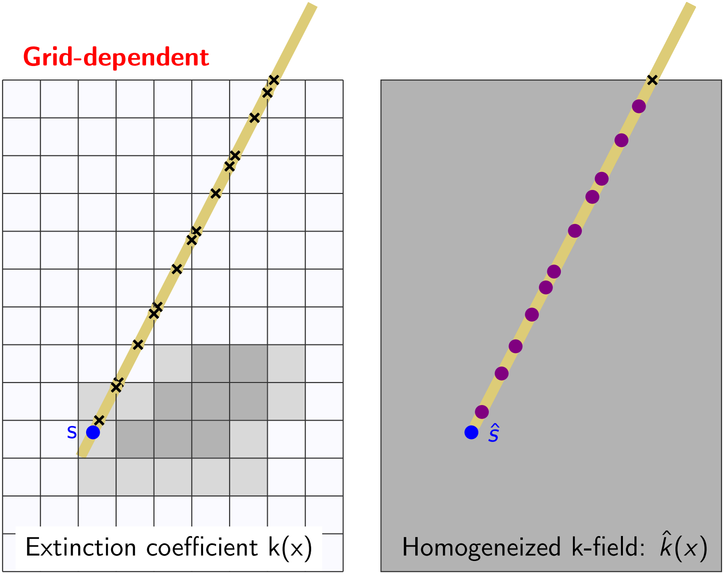
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

Sample nature:
$\mathbb{P}_{true} = k / \hat{k}$
$\mathbb{P}_{null} = k_n / \hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))



Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

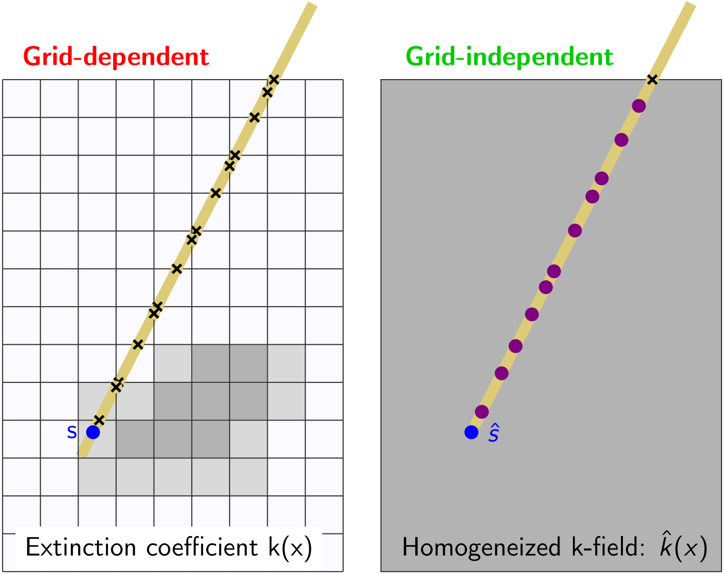
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

Sample nature:
$\mathbb{P}_{true} = k / \hat{k}$
$\mathbb{P}_{null} = k_n / \hat{k}$

While null: repeat

How can we avoid grid-dependent ray-tracing?

Using Null-Collision Algorithms (NCA) (Coleman, 1968; Galtier et al., 2013)
(= the Maximum Cross-Section method, Marchuk et al. (1980a))



Add *null* particles
$\hat{k} = k + k_n$
$\Rightarrow \hat{\tau}_{\hat{s}} = \hat{k} \times \hat{s}$

Sample $\hat{\tau}_{\hat{s}} \sim \mathcal{E}(1)$

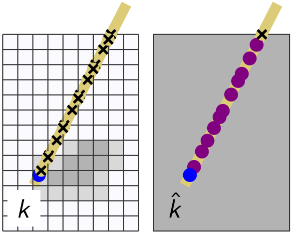
Go to $\hat{s} = \hat{\tau}_{\hat{s}} / \hat{k}$

Sample nature:
$\mathbb{P}_{true} = k / \hat{k}$
$\mathbb{P}_{null} = k_n / \hat{k}$

While null: repeat

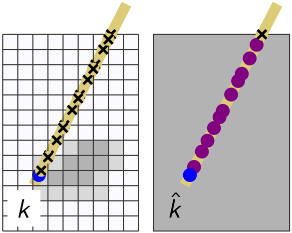
What should we retain so far?

- We want to know how clouds impact radiation



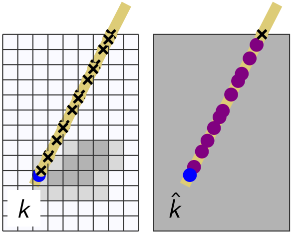
What should we retain so far?

- We want to know how clouds impact radiation
- **Monte Carlo methods** are slow but accurate



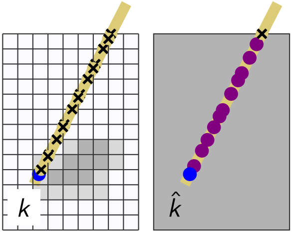
What should we retain so far?

- We want to know how clouds impact radiation
- **Monte Carlo methods** are slow but accurate
- They are able to integrate any level of detail and complexity



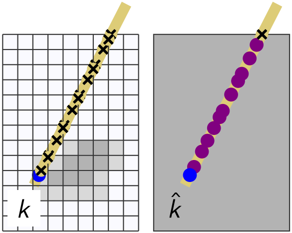
What should we retain so far?

- We want to know how clouds impact radiation
- **Monte Carlo methods** are slow but accurate
- They are able to integrate any level of detail and complexity
- But classic ray tracing is fully dependent on data grid



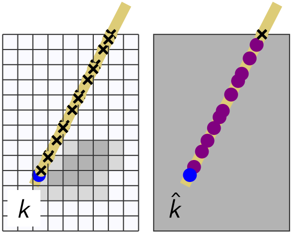
What should we retain so far?

- We want to know how clouds impact radiation
- **Monte Carlo methods** are slow but accurate
- They are able to integrate any level of detail and complexity
- But classic ray tracing is fully dependent on data grid
- This cannot prevail when model resolution is ever increasing



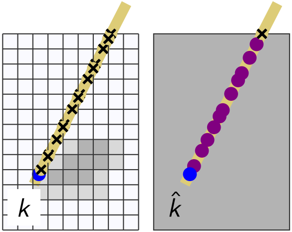
What should we retain so far?

- We want to know how clouds impact radiation
- **Monte Carlo methods** are slow but accurate
- They are able to integrate any level of detail and complexity
- But classic ray tracing is fully dependent on data grid
- This cannot prevail when model resolution is ever increasing
- **Null-collision algorithms** are a way to get rid of the grid



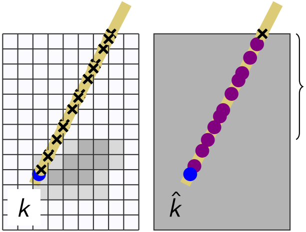
What should we retain so far?

- We want to know how clouds impact radiation
- **Monte Carlo methods** are slow but accurate
- They are able to integrate any level of detail and complexity
- But classic ray tracing is fully dependent on data grid
- This cannot prevail when model resolution is ever increasing
- **Null-collision algorithms** are a way to get rid of the grid
- But they are not very efficient on their own...



What should we retain so far?

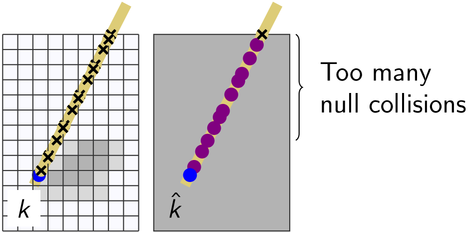
- We want to know how clouds impact radiation
- **Monte Carlo methods** are slow but accurate
- They are able to integrate any level of detail and complexity
- But classic ray tracing is fully dependent on data grid
- This cannot prevail when model resolution is ever increasing
- **Null-collision algorithms** are a way to get rid of the grid
- But they are not very efficient on their own...
- Because cloudy atmospheres are highly heterogeneous, with large cloud-free portions where $\mathbb{P}_{null} \gg \mathbb{P}_{true}$



Too many null collisions

What should we retain so far?

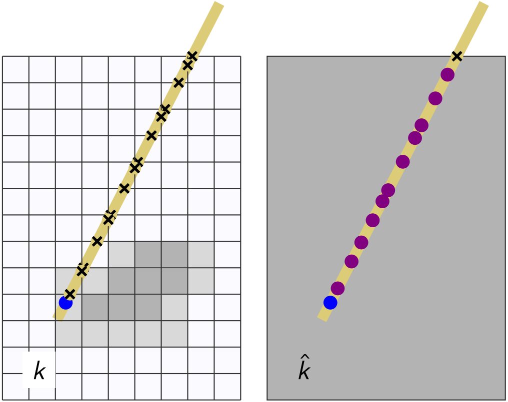
- We want to know how clouds impact radiation
- **Monte Carlo methods** are slow but accurate
- They are able to integrate any level of detail and complexity
- But classic ray tracing is fully dependent on data grid
- This cannot prevail when model resolution is ever increasing
- **Null-collision algorithms** are a way to get rid of the grid
- But they are not very efficient on their own...
- Because cloudy atmospheres are highly heterogeneous, with large cloud-free portions where $\mathbb{P}_{null} \gg \mathbb{P}_{true}$



But who said $\hat{k}$ is necessary uniform?

Introducing acceleration grids for ray tracing in NCA

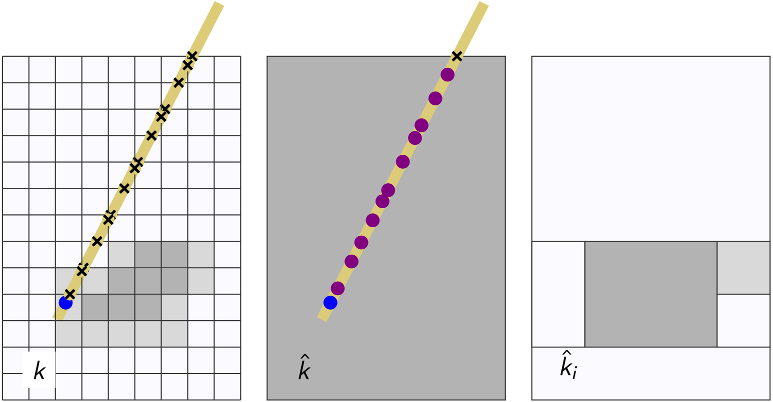
Objective: find balance between voxel intersections and null collisions



Introducing acceleration grids for ray tracing in NCA

Objective: find balance between voxel intersections and null collisions

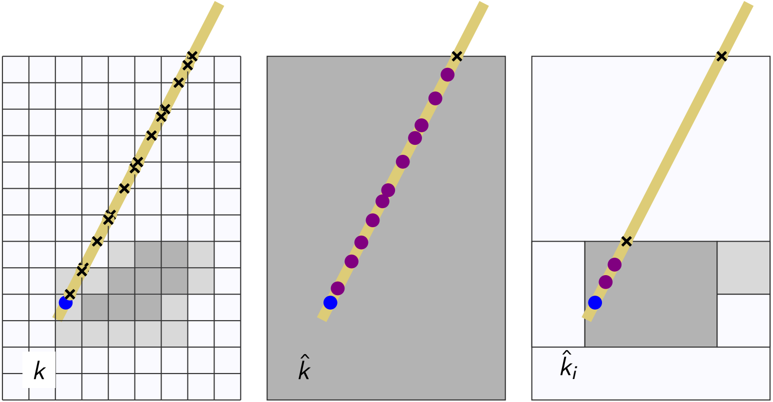
Idea: $\hat{k}$ homogeneous by regions, efficiently capturing k-field variations



Introducing acceleration grids for ray tracing in NCA

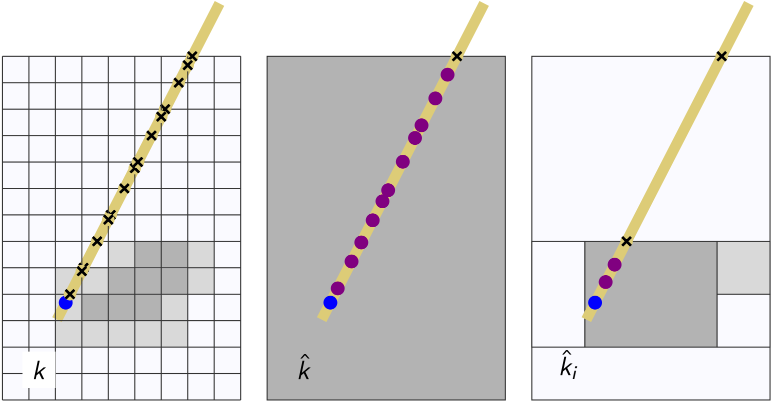
Objective: find balance between voxel intersections and null collisions

Idea: $\hat{k}$ homogeneous by regions, efficiently capturing k-field variations



Introducing acceleration grids for ray tracing in NCA

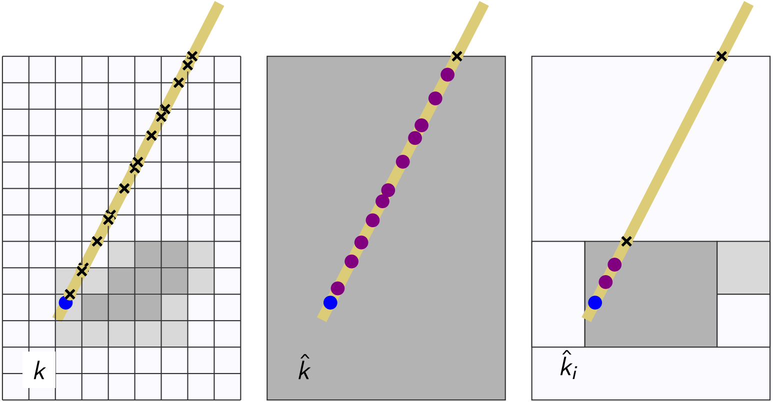
Objective: find balance between voxel intersections and null collisions
Idea: $\hat{k}$ homogeneous by regions, efficiently capturing k-field variations



Similar idea in Iwabuchi and Okamura (2017) (refined/coarse $\hat{k}$-grids in cloudy/clear layers).

Introducing acceleration grids for ray tracing in NCA

Objective: find balance between voxel intersections and null collisions
Idea: $\hat{k}$ homogeneous by regions, efficiently capturing k-field variations



Similar idea in Iwabuchi and Okamura (2017) (refined/coarse $\hat{k}$-grids in cloudy/clear layers). But outside atmospheric science, a community has been using very sophisticated grids to accelerate MC ray tracing...

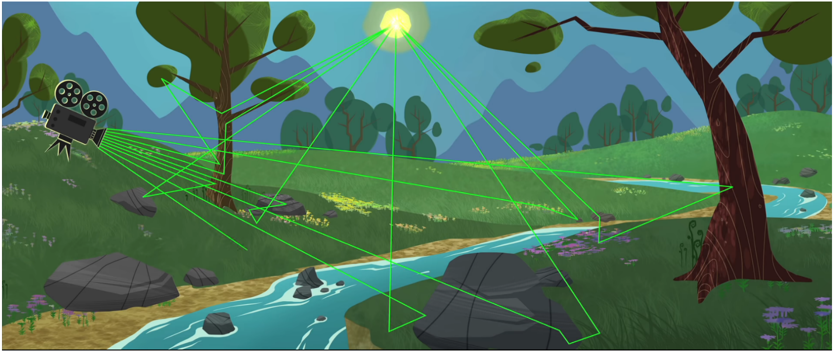
The community of physically based image rendering

Have been using path tracing Monte Carlo to render numerical scenes made of billions of triangles (in this example 3.1 billions).

Cover of the *Physically Based Rendering* book (Pharr and Humphreys, 2018)

The community of physically based image rendering

Have been using path tracing Monte Carlo to render numerical scenes made of billions of triangles (in this example 3.1 billions).

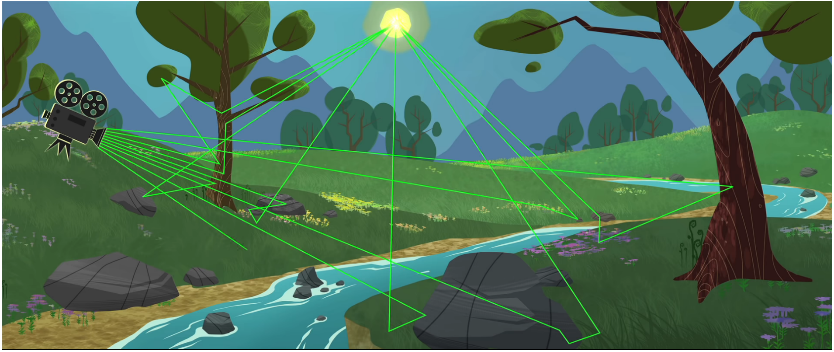


Disney's Practical Guide to Path Tracing (Walt Disney Animation Studios, 2016)

Problem: looking for next ray-surface interactions.
Testing if a given surface intersects a ray is a simple geometry problem.
Efficiently testing billions of surfaces... is a computer problem.

The community of physically based image rendering

Have been using path tracing Monte Carlo to render numerical scenes made of billions of triangles (in this example 3.1 billions).

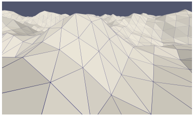


Disney's Practical Guide to Path Tracing (Walt Disney Animation Studios, 2016)

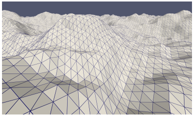
Problem: looking for next ray-surface interactions.
Testing if a given surface intersects a ray is a simple geometry problem.
Efficiently testing billions of surfaces... is a computer problem.
Solutions were developed: hierarchical structures that organize the data.

Using hierarchical structures makes rendering time insensitive to the complexity of the surface

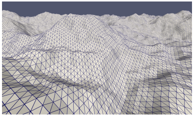
a) Ground geometries representing orography



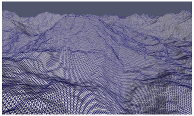
2×64×64 triangles



2×256×256 triangles



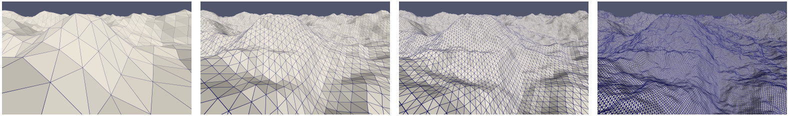
2×512×512 triangles



2×2048×2048 triangles

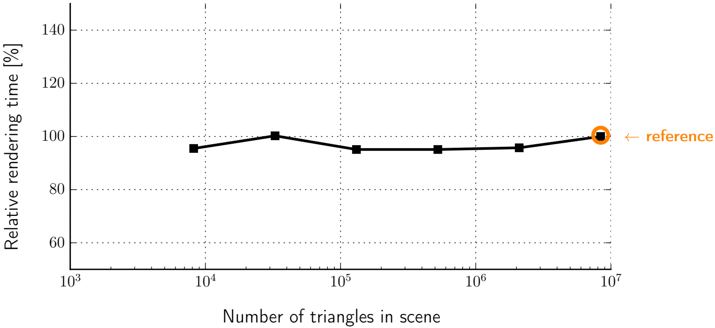
Using hierarchical structures makes rendering time insensitive to the complexity of the surface

a) Ground geometries representing orography



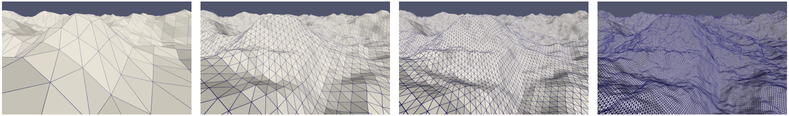
2×64×64 triangles | 2×256×256 triangles | 2×512×512 triangles | 2×2048×2048 triangles

b) Relative rendering time of scenes of increasing complexity



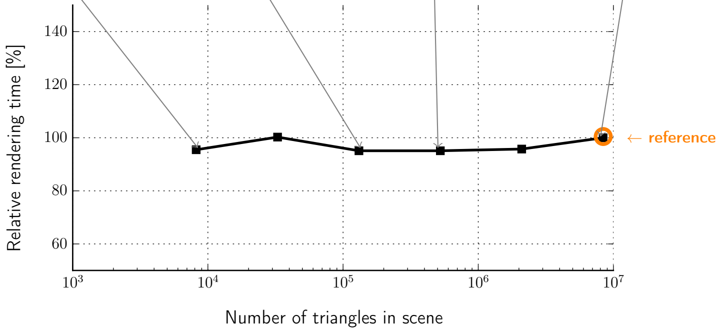
Using hierarchical structures makes rendering time insensitive to the complexity of the surface

a) Ground geometries representing orography



$2\times64\times64$ triangles — $2\times256\times256$ triangles — $2\times512\times512$ triangles — $2\times2048\times2048$ triangles

b) Relative rendering time of scenes of increasing complexity



Outline

Context: radiation in clouds, Monte Carlo

Why do we need new Monte Carlo tools?

Transfering expertise from computer graphics to atmospheric optics

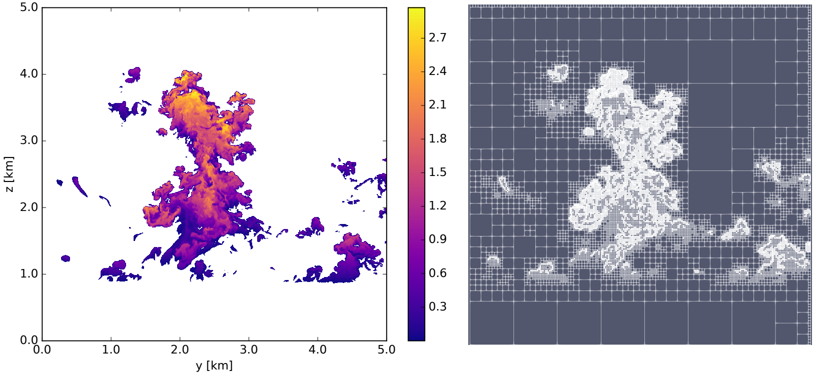
Implementation and performance tests

Outlook: a parameterisation of 3D radiative effects of clouds

Flexible tools to implement efficient Monte Carlo codes

A collection of independent modules, distributed as a **free library**, designed for **Monte Carlo specialists**, dedicated to **fast ray-tracing** in **highly-resolved volumes** and **complex surfaces**.

a) Liquid water mixing ratio [g/kg]　　b) Hierarchical grid



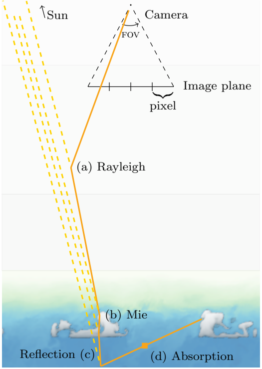
Hierarchical grid: recursively merged groups of voxels (until τ_i reaches $\tilde{\tau}$)

Learn more in the paper! `https://arxiv.org/abs/1902.01137` (in rev. for JAMES)

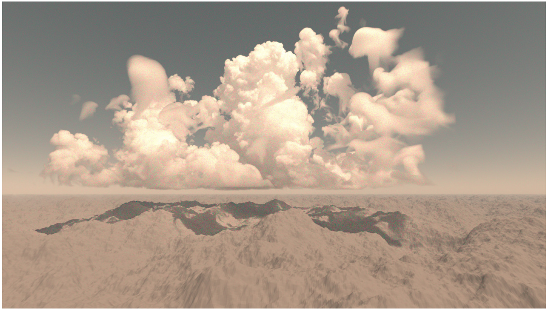
Outline

The RenDeRer (htrdr)

A rendering application to test performances and serve as a tutorial
Computes radiance fields of 3D clouds from Large-Eddy Simulations



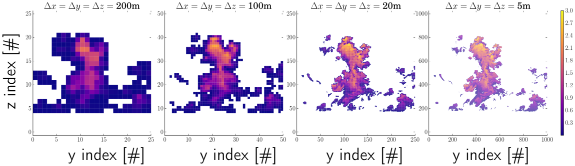
← a) Schematic illustrating the rendering algorithm



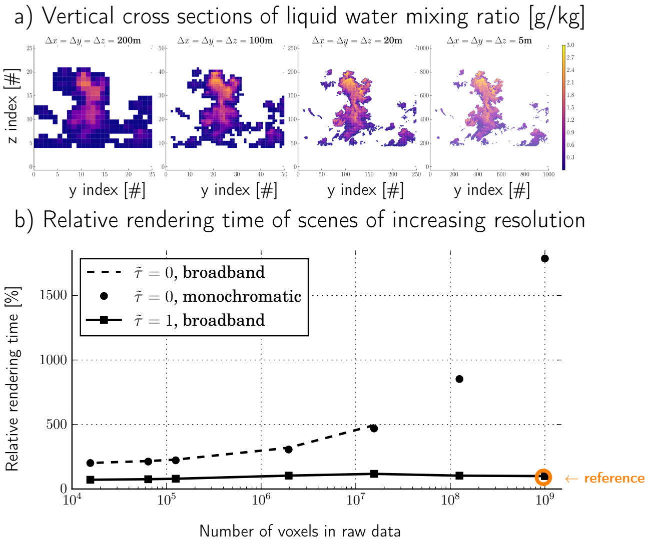
b) Example of a radiance field produced by the renderer

Insensitivity of computing time to resolution of cloud data

a) Vertical cross sections of liquid water mixing ratio [g/kg]

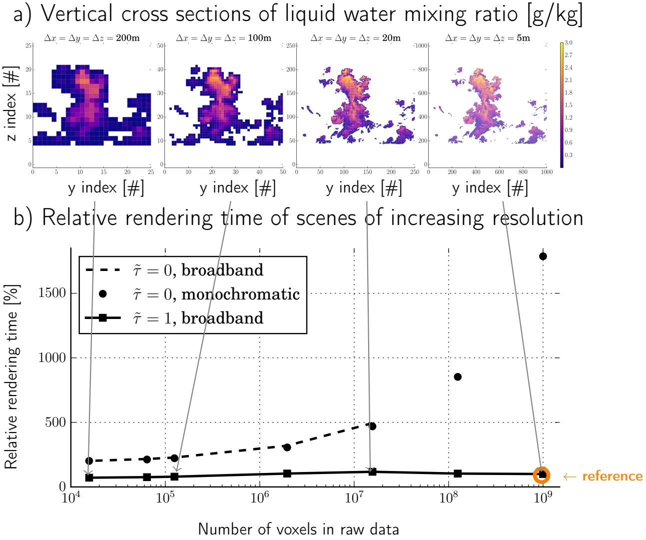


Insensitivity of computing time to resolution of cloud data



Thanks to hierarchical grids, rendering time is independent of cloud field resolution

Insensitivity of computing time to resolution of cloud data

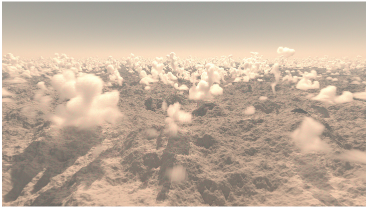


Thanks to hierarchical grids, rendering time is independent of cloud field resolution

Comparative tests for typical boundary-layer clouds (1)

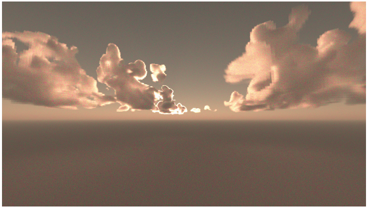
Tests on marine (a) and continental (b,c) cumulus + stratocumulus (d)

What explains difference of rendering speed? Not complexity!

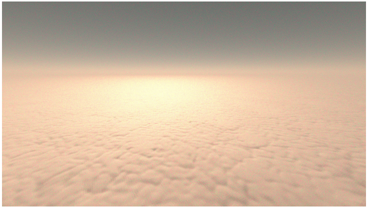


a) BOMEX, 2h59, 1 054 433 paths.s^{-1}

b) ARMCu 1, 8h22, 375 983 paths.s^{-1}



c) ARMCu 2, 4h59, 631 249 paths.s^{-1}



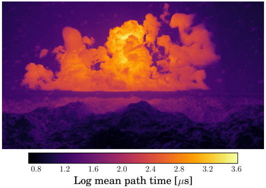
d) FIRE, 10h01, 314 049 paths.s^{-1}

$$(\overbrace{3}^{\text{channels}} \times \overbrace{1280 \times 720}^{\text{pixels}} \times \overbrace{4096}^{\text{paths per pixel}} = 11\ 324\ 620\ 800 \text{ paths per image})$$

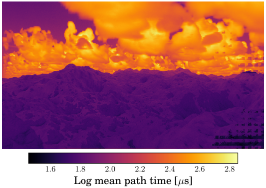
Comparative tests for typical boundary-layer clouds (2)

Mean path rendering time, averaged over pixel (map) or image ($\overline{t}$)

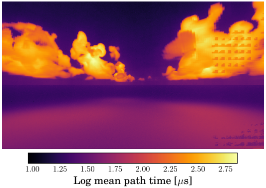
a) Congestus 5m, $\overline{t} = 110\mu s$



b) ARMCu 1, $\overline{t} = 105\mu s$



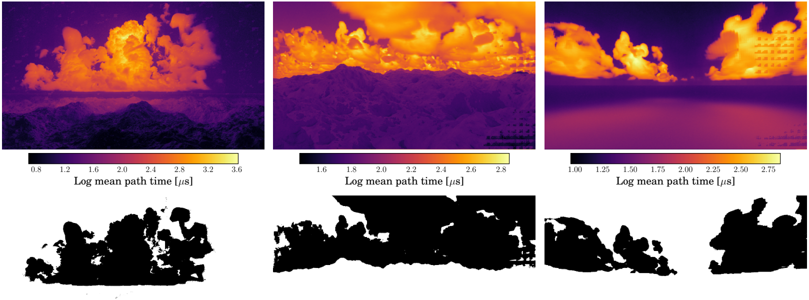
c) ARMCu 2, $\overline{t} = 60\mu s$



Comparative tests for typical boundary-layer clouds (2)

Mean path rendering time, averaged over pixel (map) or image ($\bar{t}$)

a) Congestus 5m, $\bar{t} = 110\mu s$ b) ARMCu 1, $\bar{t} = 105\mu s$ c) ARMCu 2, $\bar{t} = 60\mu s$



22.9% cloudy 39.8% cloudy 24.9% cloudy

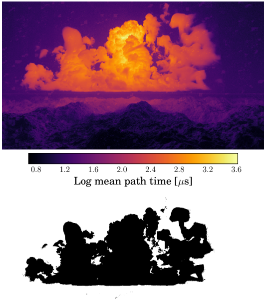
Cloud mask (black pixels): mean pixel rendering time > mean image rendering time

Strong contrast between clear & cloudy pixels,

Comparative tests for typical boundary-layer clouds (2)

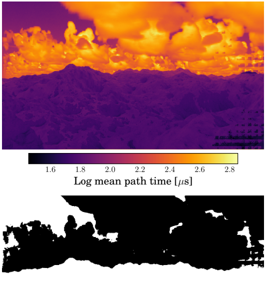
Mean path rendering time, averaged over pixel (map) or image ($\overline{t}$)

a) Congestus 5m, $\overline{t} = 110\mu s$



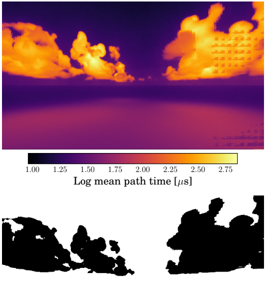
22.9% cloudy

b) ARMCu 1, $\overline{t} = 105\mu s$



39.8% cloudy

c) ARMCu 2, $\overline{t} = 60\mu s$



24.9% cloudy

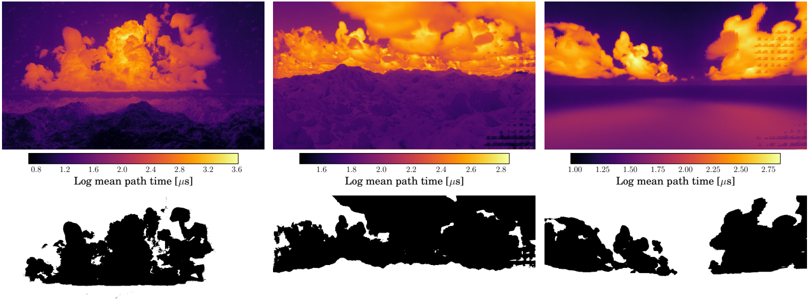
Cloud mask (black pixels): mean pixel rendering time > mean image rendering time

Strong contrast between clear & cloudy pixels, thick & thin clouds

Comparative tests for typical boundary-layer clouds (2)

Mean path rendering time, averaged over pixel (map) or image ($\overline{t}$)

a) Congestus 5m, $\overline{t} = 110\mu s$ b) ARMCu 1, $\overline{t} = 105\mu s$ c) ARMCu 2, $\overline{t} = 60\mu s$

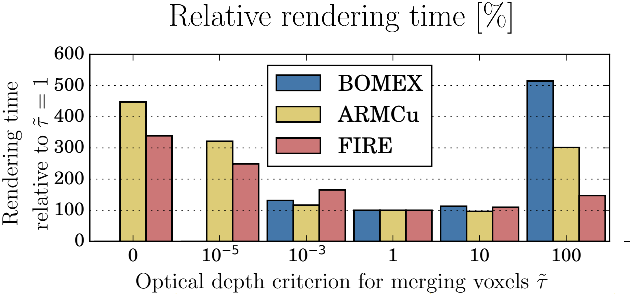


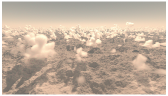
22.9% cloudy 39.8% cloudy 24.9% cloudy

Cloud mask (black pixels): mean pixel rendering time > mean image rendering time

Strong contrast between clear & cloudy pixels, thick & thin clouds
Path rendering time is a function of the order of scattering

Refinement of hierarchical grid

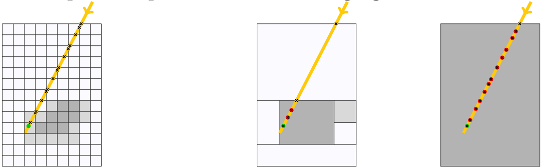




BOMEX

ARMCu

FIRE



Outline

Context: radiation in clouds, Monte Carlo

Why do we need new Monte Carlo tools?

Transfering expertise from computer graphics to atmospheric optics

Implementation and performance tests

Outlook: a parameterisation of 3D radiative effects of clouds

Cloud-radiation interactions in large-scale models

Recent developments e.g.

- cloud heterogeneity: McICA (Cahalan et al., 1994; Pincus et al., 2003); Tripleclouds (Shonk and Hogan, 2008)
- horizontal transport: SPARTACUS (Hogan and Shonk, 2013; Schäfer et al., 2016; Hogan et al., 2016)

Cloud-radiation interactions in large-scale models

Recent developments e.g.

- cloud heterogeneity: McICA (Cahalan et al., 1994; Pincus et al., 2003); Tripleclouds (Shonk and Hogan, 2008)
- horizontal transport: SPARTACUS (Hogan and Shonk, 2013; Schäfer et al., 2016; Hogan et al., 2016)

Mechanism	Net surface
Add horizontal structure	+3.8 (±2)
Add 3D effects	+2.1

← adapted from Hogan (2018)
Global effect of adding complexity to radiative scheme

Cloud-radiation interactions in large-scale models

Recent developments e.g.

- cloud heterogeneity: McICA (Cahalan et al., 1994; Pincus et al., 2003); Tripleclouds (Shonk and Hogan, 2008)
- horizontal transport: SPARTACUS (Hogan and Shonk, 2013; Schäfer et al., 2016; Hogan et al., 2016)

Mechanism	Net surface
Add horizontal structure	+3.8 (±2)
Add 3D effects	+2.1

← adapted from Hogan (2018)
Global effect of adding complexity to radiative scheme

Questions remain:

- Evaluate more complete diagnosis e.g. direct / diffuse partitionning
- How best to constrain cloud parameters in radiative scheme? Parameterisations to convey available info from host model? Tuning strategies?

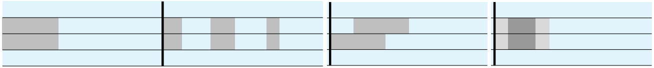
Evaluation of a large-scale radiative scheme

Direct / diffuse partition important to e.g. solar energy, vegetation.
Domain-average broadband direct-to-total ground flux, MC vs ecRad.

Evaluation of a large-scale radiative scheme

Direct / diffuse partition important to e.g. solar energy, vegetation.
Domain-average broadband direct-to-total ground flux, MC vs ecRad.

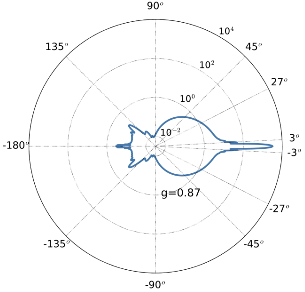
- ecRad solvers: TripleClouds (2-stream) or SPARTACUS (+ param 3D)
- Cloud characteristics diagnosed in LES field and given to ecRad.
- Overlap needs to be scaled otherwise cloud cover is overestimated.



Evaluation of a large-scale radiative scheme

Direct / diffuse partition important to e.g. solar energy, vegetation.
Domain-average broadband direct-to-total ground flux, MC vs ecRad.

- ecRad solvers: TripleClouds (2-stream) or SPARTACUS (+ param 3D)
- Cloud characteristics diagnosed in LES field and given to ecRad.
- Overlap needs to be scaled otherwise cloud cover is overestimated.
- Delta-scaling to correct total fluxes ⇒ overestimates direct flux.

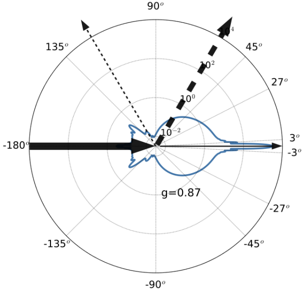


Mie phase function

Evaluation of a large-scale radiative scheme

Direct / diffuse partition important to e.g. solar energy, vegetation.
Domain-average broadband direct-to-total ground flux, MC vs ecRad.

– ecRad solvers: TripleClouds (2-stream) or SPARTACUS (+ param 3D)
– Cloud characteristics diagnosed in LES field and given to ecRad.
– Overlap needs to be scaled otherwise cloud cover is overestimated.
– Delta-scaling to correct total fluxes ⇒ overestimates direct flux.

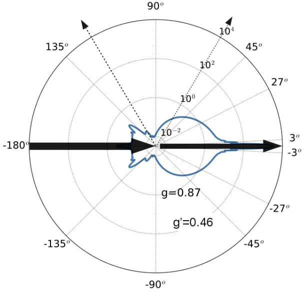


2-stream approximation (– direct, - - diffuse)

Evaluation of a large-scale radiative scheme

Direct / diffuse partition important to e.g. solar energy, vegetation.
Domain-average broadband direct-to-total ground flux, MC vs ecRad.

- ecRad solvers: TripleClouds (2-stream) or SPARTACUS (+ param 3D)
- Cloud characteristics diagnosed in LES field and given to ecRad.
- Overlap needs to be scaled otherwise cloud cover is overestimated.
- Delta-scaling to correct total fluxes ⇒ overestimates direct flux.

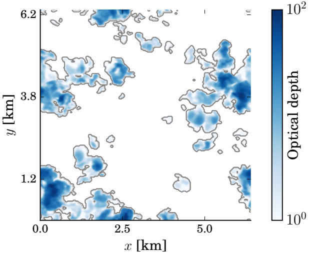


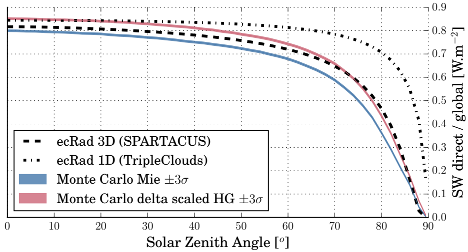
Delta-scaled 2-stream (– direct, - - diffuse) (**?**)

Evaluation of a large-scale radiative scheme

Direct / diffuse partition important to e.g. solar energy, vegetation.
Domain-average broadband direct-to-total ground flux, MC vs ecRad.

– ecRad solvers: TripleClouds (2-stream) or SPARTACUS (+ param 3D)
– Cloud characteristics diagnosed in LES field and given to ecRad.
– Overlap needs to be scaled otherwise cloud cover is overestimated.
– Delta-scaling to correct total fluxes ⇒ overestimates direct flux.
MC Mie vs MC delta-scaled HG vs SPARTACUS (- -) vs TripleClouds (-.)

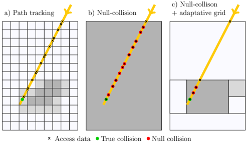


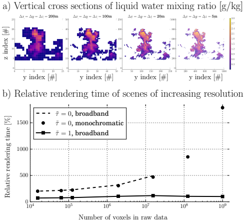


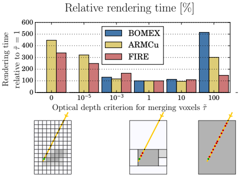
Subgrid horizontal transport crucial for direct / diffuse (side illumination)

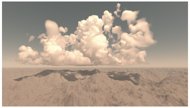
Summary

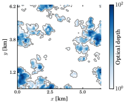
- ACN + hierarchical grids = computing time $\perp\!\!\!\perp$ to data complexity
- Online distributed library to facilitate MC implementation
- Renderer to test performances, optimum hierarchical grid with $\tilde{\tau} \approx 1$
- Reference MC to study 3D effects
  - 3D effects crucial for direct / diffuse
  - SPARTACUS + cloud description from LES = close to MC

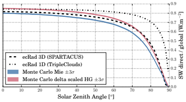












References I

Cahalan, R. F., Ridgway, W., Wiscombe, W. J., Gollmer, S., and Harshvardhan (1994). Independent Pixel and Monte Carlo Estimates of Stratocumulus Albedo. *Journal of the Atmospheric Sciences*, 51(24):3776–3790.

Coleman, W. A. (1968). Mathematical verification of a certain monte carlo sampling technique and applications of the technique to radiation transport problems. *Nuclear Science and Engineering*, 32(1):76–81.

Evans, K. F. (1998). The Spherical Harmonics Discrete Ordinate Method for Three-Dimensional Atmospheric Radiative Transfer. *Journal of the Atmospheric Sciences*, 55(3):429–446.

Galtier, M., Blanco, S., Caliot, C., Coustet, C., Dauchet, J., El Hafi, M., Eymet, V., Fournier, R., Gautrais, J., Khuong, A., Piaud, B., and Terrée, G. (2013). Integral formulation of null-collision Monte Carlo algorithms. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 125:57–68.

Hogan, R. (2018). Challenges for radiation in NWP models.

Hogan, R. J., Fielding, M. D., Barker, H. W., Villefranque, N., and Schäfer, S. A. K. (2019). Entrapment: An important mechanism to explain the shortwave 3d radiative effect of clouds. *Journal of the Atmospheric Sciences*.

Hogan, R. J., Schäfer, S. A. K., Klinger, C., Chiu, J. C., and Mayer, B. (2016). Representing 3-D cloud radiation effects in two-stream schemes: 2. Matrix formulation and broadband evaluation. *Journal of Geophysical Research: Atmospheres*, 121(14):2016JD024875.

References II

Hogan, R. J. and Shonk, J. K. P. (2013). Incorporating the Effects of 3d Radiative Transfer in the Presence of Clouds into Two-Stream Multilayer Radiation Schemes. *Journal of the Atmospheric Sciences*, 70(2):708–724.

Iwabuchi, H. and Okamura, R. (2017). Multispectral monte carlo radiative transfer simulation by the maximum cross-section method. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 193:40–46.

Joseph, J. H., Wiscombe, W. J., and Weinman, J. A. (1976). The Delta-Eddington Approximation for Radiative Flux Transfer. *Journal of the Atmospheric Sciences*, 33(12):2452–2459.

Marchuk, G. I., Mikhailov, G. A., Nazaraliev, M. A., Darbinjan, R. A., Kargin, B. A., and Elepov, B. S. (1980a). Elements of Radiative-Transfer Theory Used in the Monte Carlo Methods. In *The Monte Carlo Methods in Atmospheric Optics*, Springer Series in Optical Sciences, pages 5–17. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-540-35237-2_2.

Marchuk, G. I., Mikhailov, G. A., Nazareliev, M. A., Darbinjan, R. A., Kargin, B. A., and Elepov, B. S. (1980b). *The Monte Carlo Methods in Atmospheric Optics*. Springer Series in Optical Sciences. Springer-Verlag, Berlin Heidelberg.

Mayer, B. (2009). Radiative transfer in the cloudy atmosphere. *The European Physical Journal Conferences*, 1:75–99.

Pharr, M. and Humphreys, G. (2018). *Physically Based Rendering, Third Edition: From Theory To Implementation*. 3rd edition.

References III

Pincus, R., Barker, H. W., and Morcrette, J.-J. (2003). A fast, flexible, approximate technique for computing radiative transfer in inhomogeneous cloud fields: FAST, FLEXIBLE, APPROXIMATE RADIATIVE TRANSFER. *Journal of Geophysical Research: Atmospheres*, 108(D13):n/a–n/a.

Schäfer, S. A. K., Hogan, R. J., Klinger, C., Chiu, J. C., and Mayer, B. (2016). Representing 3-D cloud radiation effects in two-stream schemes: 1. Longwave considerations and effective cloud edge length. *Journal of Geophysical Research: Atmospheres*, 121(14):2016JD024876.

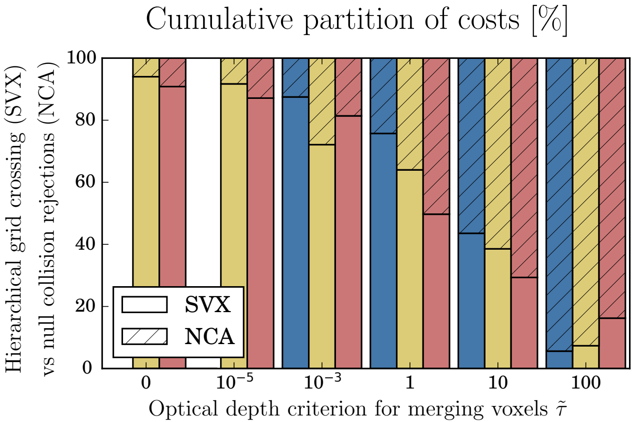
Shonk, J. K. P. and Hogan, R. J. (2008). Tripleclouds: An Efficient Method for Representing Horizontal Cloud Inhomogeneity in 1d Radiation Schemes by Using Three Regions at Each Height. *Journal of Climate*, 21(11):2352–2370.

Walt Disney Animation Studios (2016). Disney's practical guide to path tracing. `https://youtu.be/frLwRLS_ZR0?t=301`.

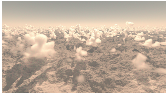
Other results

- Refinement of hierarchical grid
- HIGH-TUNE paradigm
- Horizontal Distances for Entrapment
- Direct diffuse Monte Carlo
- Absorptivity bias in ICA MC
- Transmissivity vs reflectivity bias in ICA MC
- Direct vs diffuse surface bias in ICA MC
- Integrated ICA error and role of cloud size

Refinement of hierarchical grid ◁

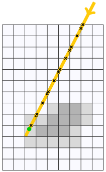


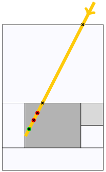
BOMEX

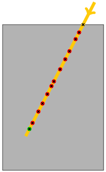


ARMCu

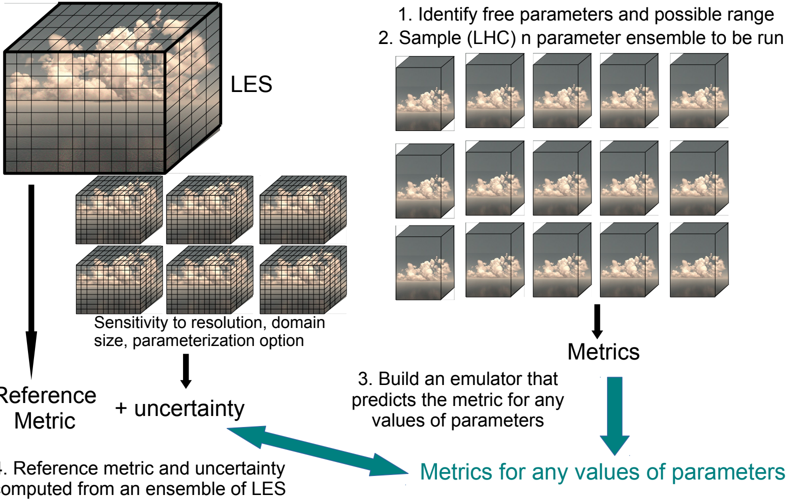
FIRE







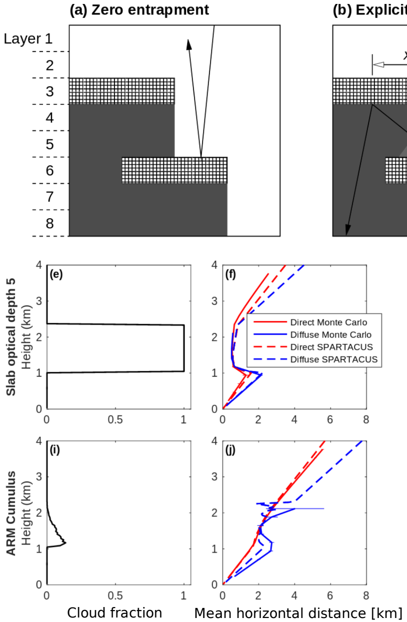
HIGH-TUNE paradigm ◁



5. Compute Implausibility, a normalised distance, used to rull out impossible values of parameters

$$I=\frac{\left(M_{LES}-M_{Emulator=f(p1,\ldots,pn)}\right)^2}{\varepsilon^2_{LES}+\varepsilon^2_{Structural}+\varepsilon^2_{Emulator}}$$

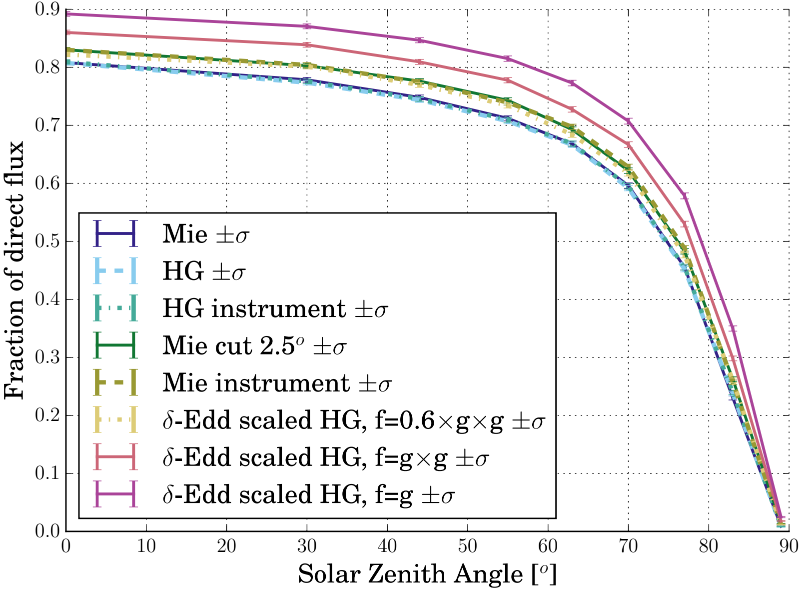
Horizontal Distances for Entrapment ◁



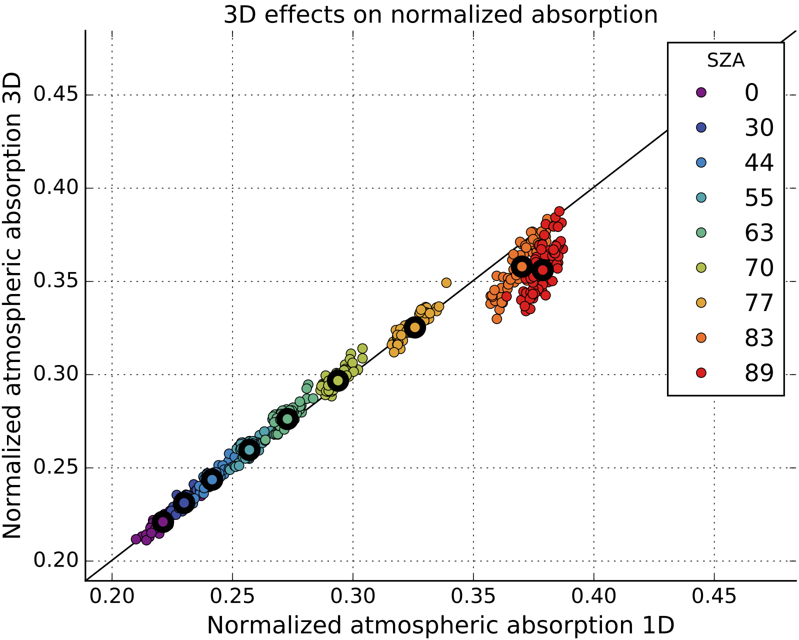
↑ Entrapment decreases reflectance (Hogan et al., 2019)

← Parameterized in SPARTACUS as a function of mean horizontal distance traveled by photons before being reflected back to a given level

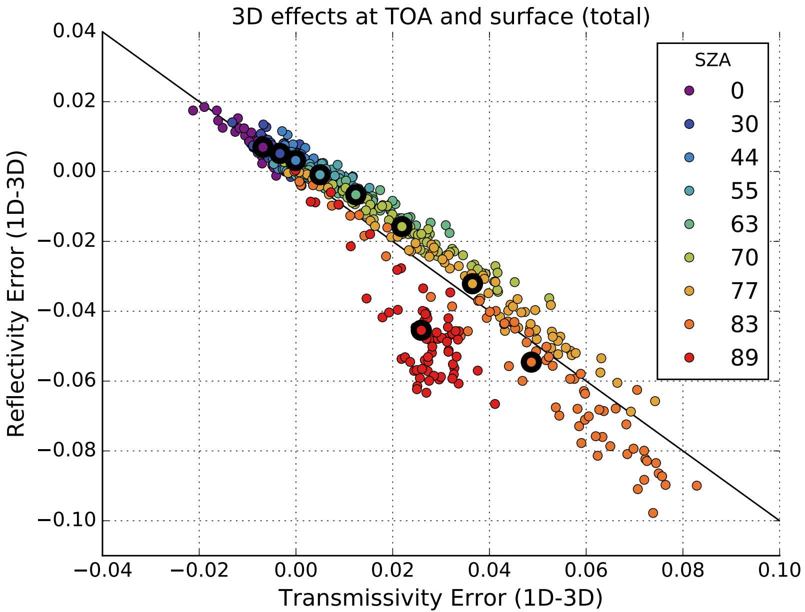
Direct diffuse Monte Carlo ◁



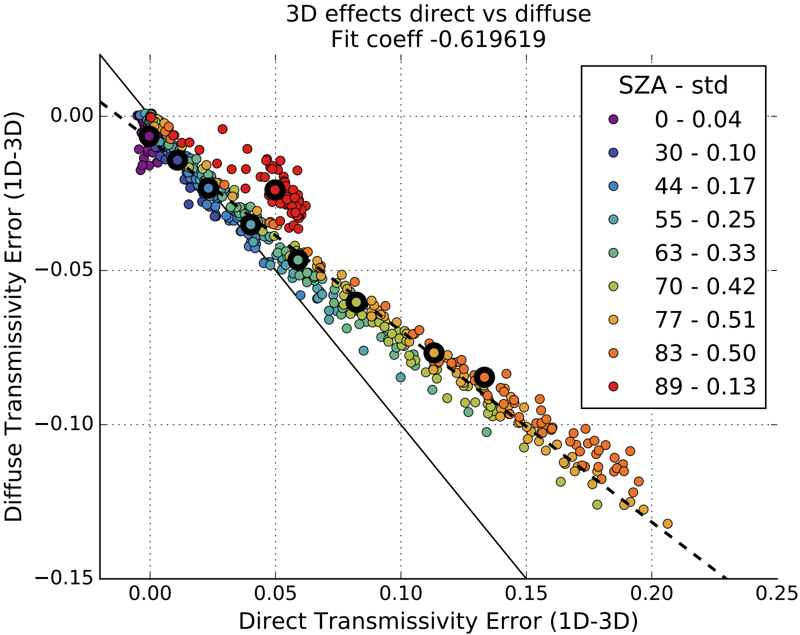
Absorptivity bias in ICA MC ◁



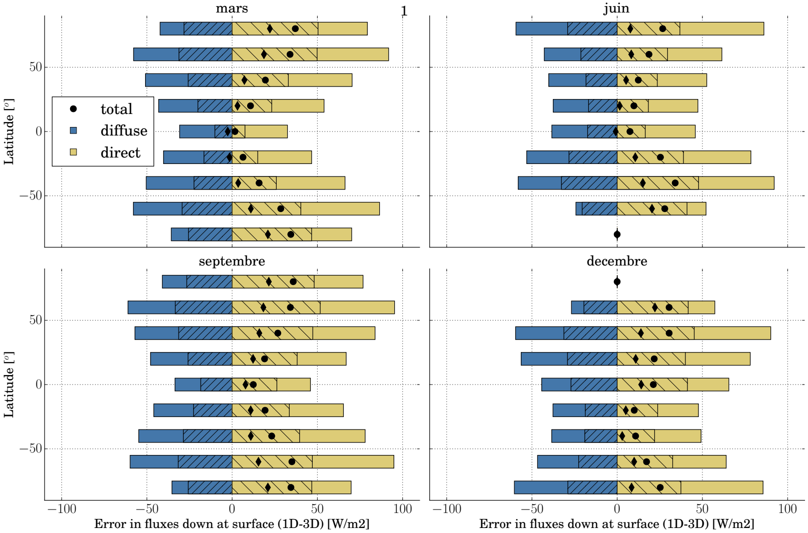
Transmissivity vs reflectivity bias in ICA MC ◁



Direct vs diffuse surface bias in ICA MC ◁



Integrated ICA error and role of cloud size ◁



Hashed / diamonds = artificially doubled gridcells horizontal size