Radiative transfer solver(s) in SMRT

Sponsored by :

European Space Agency

General equations

RT solver:

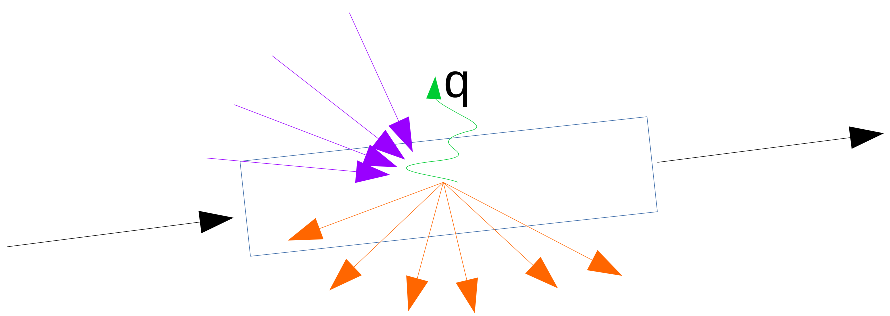
Inputs:
All the constants required in the RT equation and boundary conditions, inc. incident radiance

Output:
The radiance at the top of the snowpack

General equations

The radiative transfer equation

$$\mu \frac{\partial \mathbf{I}(\mu,\phi,z)}{\partial z} = -\boldsymbol{\kappa}_{\mathrm{e}}(\mu,\phi,z)\mathbf{I}(\mu,\phi,z) + \frac{1}{4\pi}\iint_{4\pi} \mathbf{P}(\mu,\phi;\mu',\phi',z)\mathbf{I}(\mu',\phi',z)\, d\Omega' + \boldsymbol{\kappa}_{\mathrm{a}}(\mu,\phi,z)\,\alpha T(z)\mathbf{1}$$



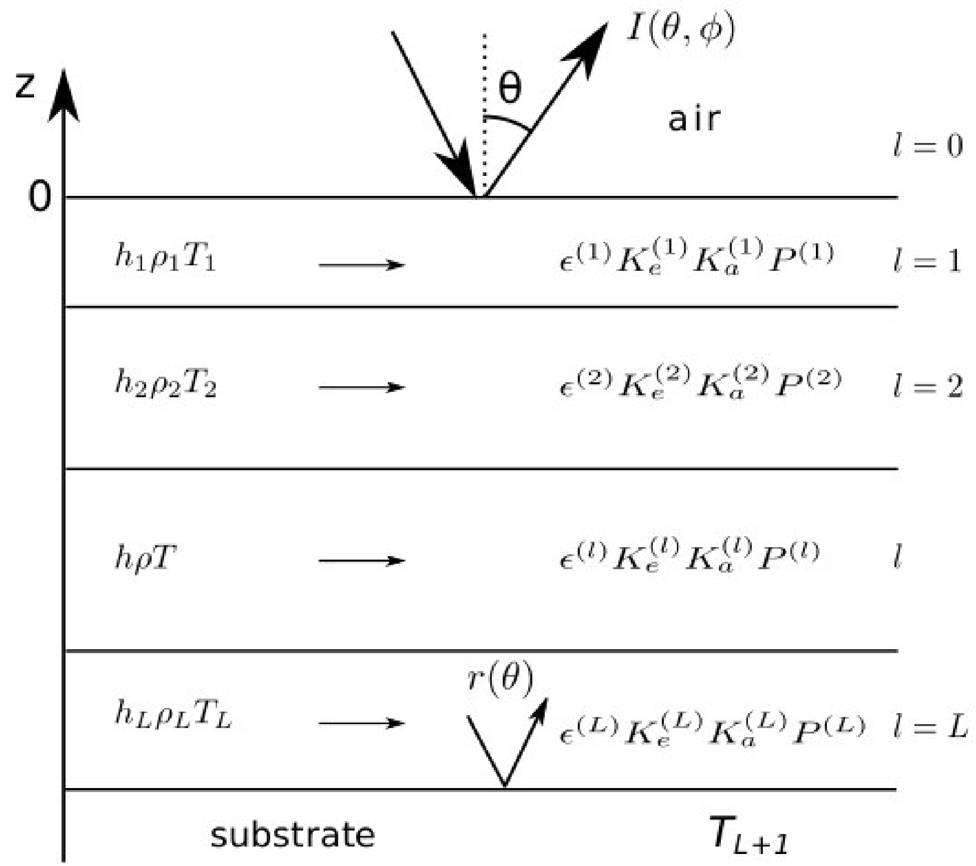
<u>Rmq:</u> stationary equation → no time dependence / wave travel is not resolved.

Time-depend RT exists and is needed for altimetry. Such a code is being developed in SMRT (available from 2020).

<u>Rmq:</u> vector radiative transfer equation → full polarizations
- Radiance I is a 4-component vector
- Phase matrix P is a 4x4 matrix

General equations

In SMRT, we consider plane-parallel layers:



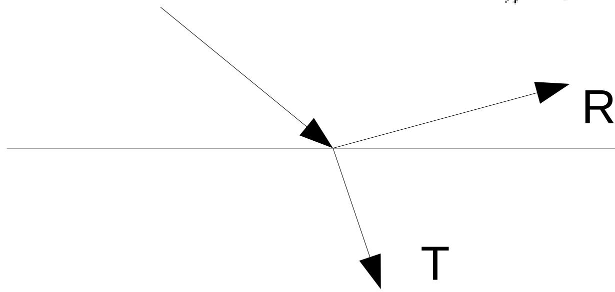
Layers are not necessarily « smooth » from the EM point of view, they can be rough, but for the propagation of energy (RT) perspective, they are parallel

General equations

Boundary conditions for **top, bottom and inter-layer** interfaces:

$$\mathbf{I}^{(l)}\left(\mu<0,\phi,z_{l-1}\right)=\mathbf{R}^{\mathrm{spec,top},(l)}(\mu)\mathbf{I}^{(l)}\left(-\mu,\phi,z_{l-1}\right)+\frac{1}{2\pi}\iint_{2\pi,\mu'>0}\mathbf{R}^{\mathrm{diff,top},(l)}(\mu,\mu',\phi-\phi')\mathbf{I}^{(l)}\left(\mu',\phi',z_{l-1}\right)d\Omega'$$
$$+\mathbf{T}^{\mathrm{spec,bottom},(l-1)}(\mu)\mathbf{I}^{(l-1)}\left(\mu,\phi,z_{l-1}\right)+\frac{1}{2\pi}\iint_{2\pi,\mu'<0}\mathbf{T}^{\mathrm{diff,bottom},(l-1)}(\mu,\mu',\phi-\phi')\mathbf{I}^{(l-1)}\left(\mu',\phi',z_{l-1}\right)d\Omega'$$

$$\mathbf{I}^{(l)}\left(\mu>0,\phi,z_{l}\right)=\mathbf{R}^{\mathrm{spec,bottom},(l)}(\mu)\mathbf{I}^{(l)}\left(-\mu,\phi,z_{l}\right)+\frac{1}{2\pi}\iint_{2\pi,\mu'<0}\mathbf{R}^{\mathrm{diff,bottom},(l)}(\mu,\mu',\phi-\phi')\mathbf{I}^{(l)}\left(\mu',\phi',z_{l}\right)d\Omega'$$
$$+\mathbf{T}^{\mathrm{spec,top},(l+1)}(\mu)\mathbf{I}^{(l+1)}\left(\mu,\phi,z_{l}\right)+\frac{1}{2\pi}\iint_{2\pi,\mu'<0}\mathbf{T}^{\mathrm{diff,top},(l+1)}(\mu,\mu',\phi-\phi')\mathbf{I}^{(l+1)}\left(\mu',\phi',z_{l}\right)d\Omega'$$



<u>Rmq:</u>
Here we distinguish the specular and diffuse components, for the numerical implementation.

General equations

Many methods have been propose to solve these equations:

- Iterative solutions: assume weak single scattering albedo (=Ks/Ke) and interface reflexions. Analytical solution for first order (simple) and second order (much less fancy)… nearly untracktable for higher orders. For snow, this applies in the low frequency limit, and has been widely used for radar.

- 1 flux (HUT)
- 2 streams and 6-flux (used in MEMLS, 2S). Account for multiple scattering. Computionally efficient. Poor angular resolution (only "forward" and "backward" phase matrix).

- discrete ordinate methods often called DISORT or DORT (used in DMRT-QMS, DMRT-ML). Reference method in most RT studies. Multiple scattering. Suitable for very thick media. Slow. Complex implementation. Many variants!!

- adding / doubling methods. Fast but more suitable for thin media (to my knowledge)

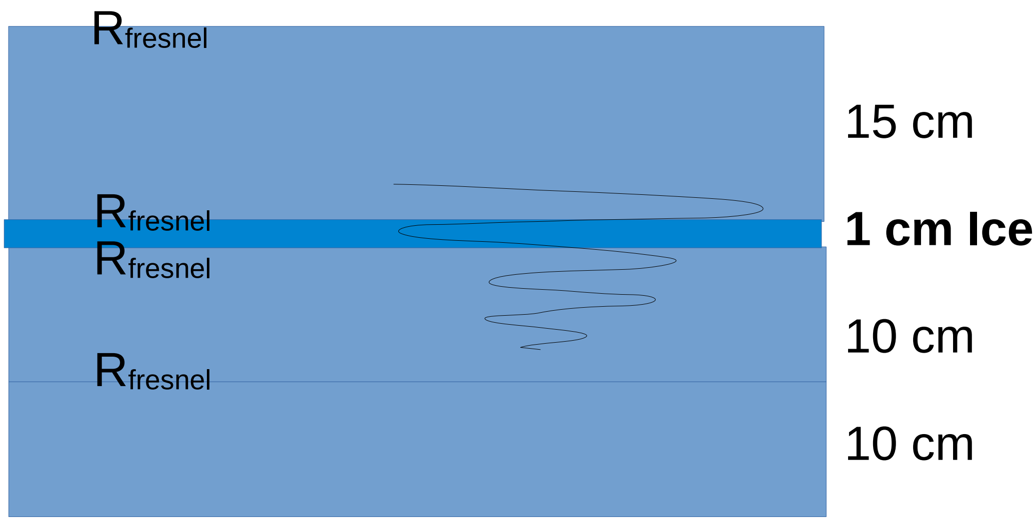
- monte-carlo methods. Very slow. Easy to implement. Versatile for none plane-parallel geometries

- Neural-network (in dev for SMRT). Iteratively solve the RT equation along with boundary conditions and possibly observations (inverse problem). Easy to implement but can be slow, can diverge unexpectedly.

Short digression on coherent layers

“Coherent layers »

MEMLS solution :



6 GHz (5 cm)

6 GHz (5 cm)

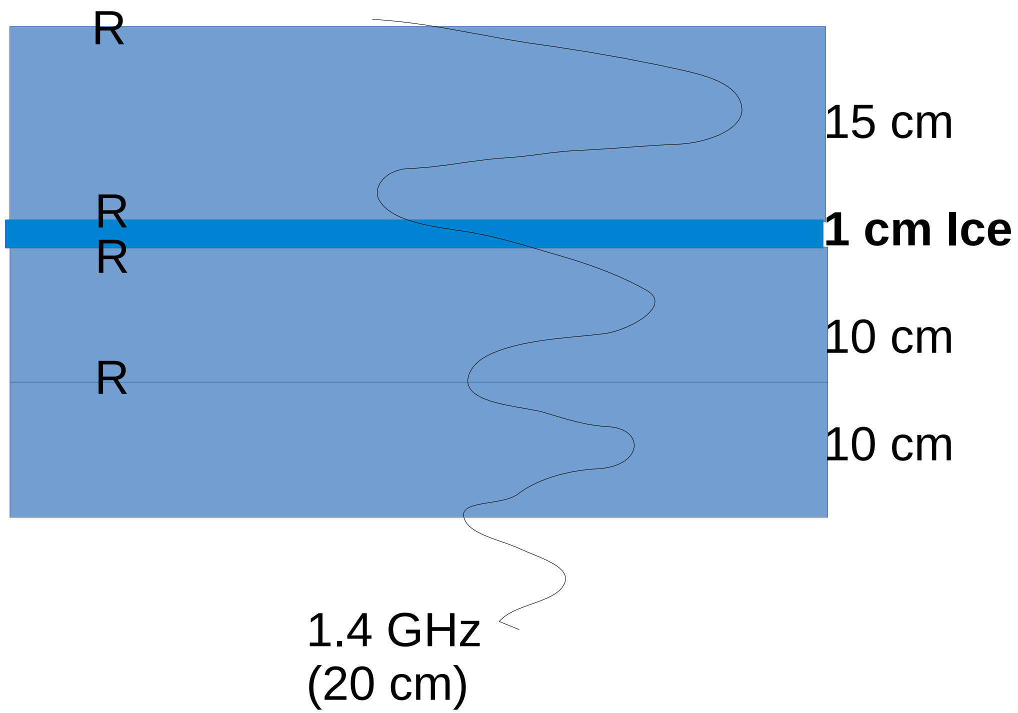
- assume the ice layer is non-scattering
- wave theory → R effective

For SMRT : dort_coherent_layer could be (easily) implemented

Short disgression on coherent layers

The problem with this solution:

It can't handle consecutive small layers.



MEMLS solution does not work

→ no RT solution

General equations

SMRT is equipped with :

- robust Discrete Ordinate Method.

- time-resolved first-order solver for altimetry

We'd like to see more options in the future:

- 6-flux as in MEMLS (under implementation)

- variants of DORT → blend iterative/DORT methods, ...

- adding-doubling for shallow snowpack (e.g. seasonal snow at radar frequencies)

- first-order for fast radar computation and (easy) extension to bi-refringent media (snow anisotropy).

- neural-network (NN) based solvers

DORT in SMRT

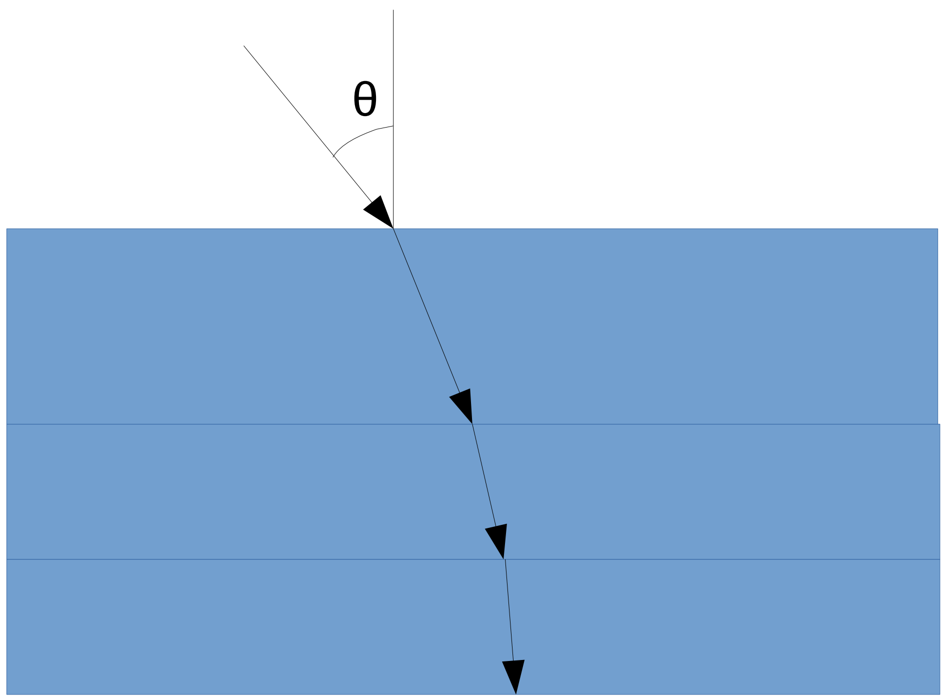
The DORT in SMRT is a new implemention based on ideas from:

- DORT for radar backscatter on forest (Picard et al. 2004) → sparse medium
  Background refractive index = 1

- DORT for passive microwave in snow in DMRT-ML (Picard et al. 2013) inspired from Jin 1994 for single layer
  Varying background refractive index

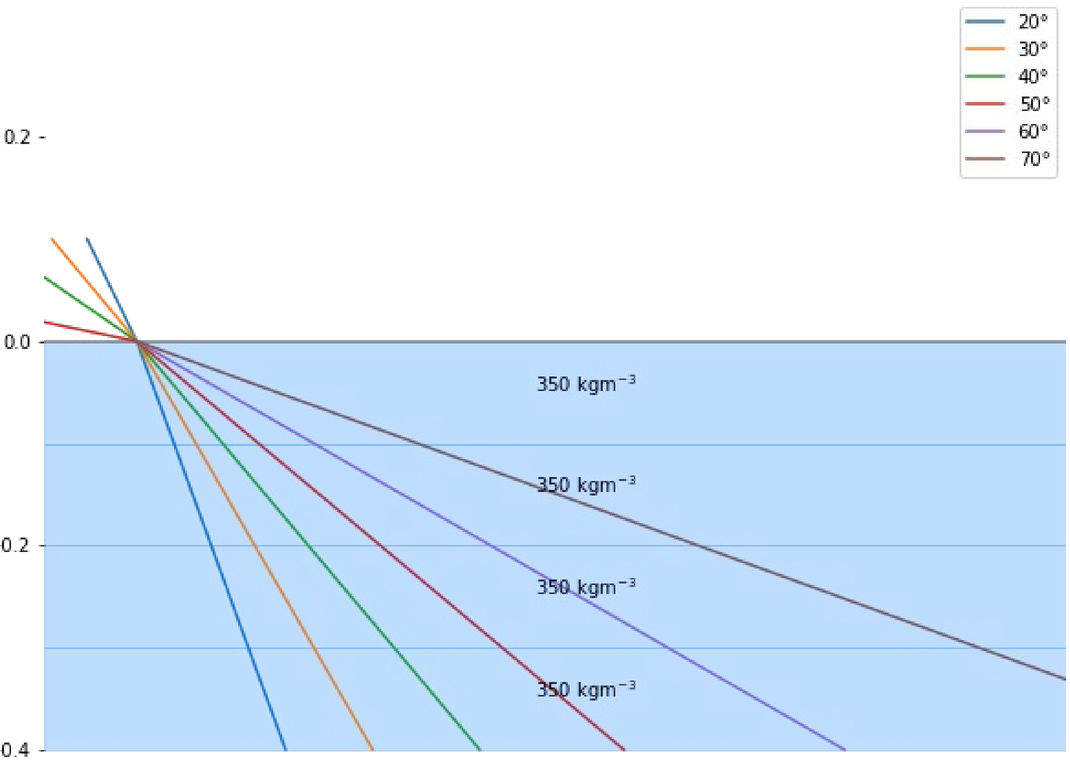
Snell's law: $n \sin \theta = cst$

For snow: n is driven by density



DORT in SMRT

Done with plot_snowpack in smrt.utils.mpl_plots

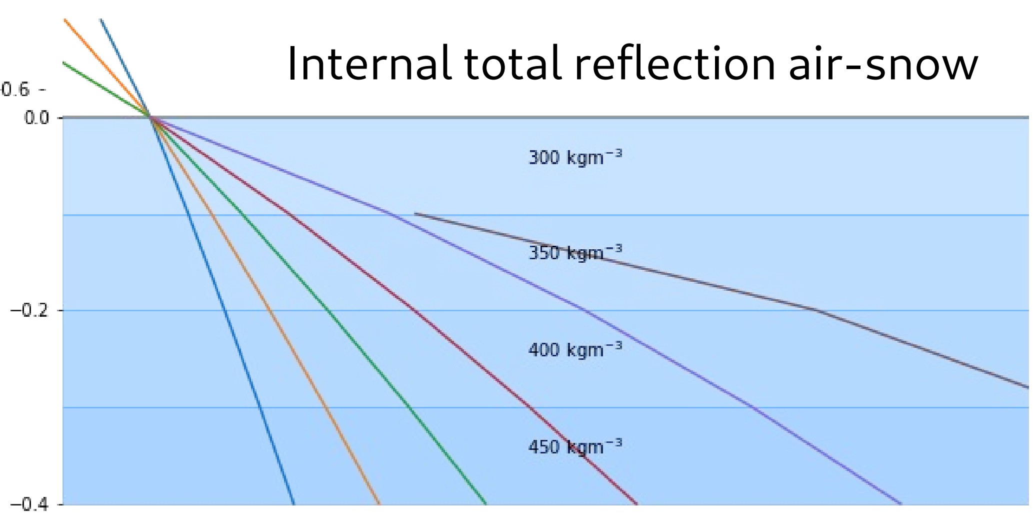


Total reflection air-snow

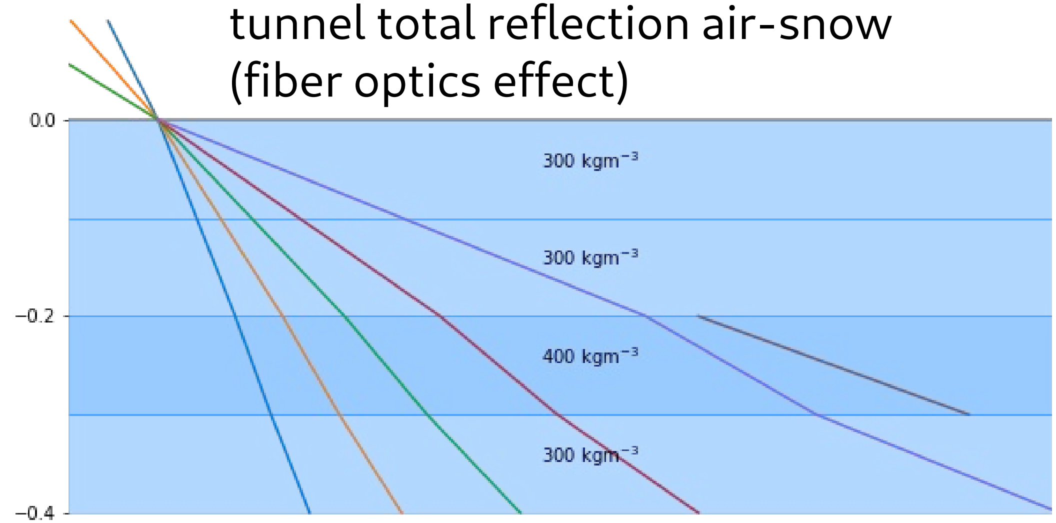
Total radiation is a significant cause of radiation trapping (= lower emissivity)

→ the number of streams varies depending on the refractive index of the layers

Internal total reflection air-snow



tunnel total reflection air-snow (fiber optics effect)



DORT in SMRT

DORT method works by discretizing the zenith and azimuth angles dependencies in the RT equation. There are many many variants.

Users must understand the specificities of the implementation in SMRT to understand some behavior when running simulations

Azimuthal angle dependency :

- Assume isotropic medium (=no aligned structures, no sastrugi)
- Treated with cosine series decomposition (Fourier series):

$$\mathbf{I}(\mu, \phi, z) = \sum_{m=0}^{\infty} \mathbf{I}^{\mathrm{c},m}(\mu, z)\cos(m\phi) + \mathbf{I}^{\mathrm{s},m}(\mu, z)\sin(m\phi)$$

$$\mathbf{P}^{(l)}(\mu, \phi, \mu', \phi') = \sum_{m=0}^{\infty} \mathbf{P}^{\mathrm{c},(l),m}(\mu, \mu')\cos[m(\phi - \phi')] + \mathbf{P}^{\mathrm{s},(l),m}(\mu, \mu')\sin[m(\phi - \phi')]$$

Pros: Very common approach.
Cons: Cause « big number – big number = not precise small number » in active mode, see later

DORT in SMRT

<u>Azimuthal angle dependency in practice:</u>

The truncation of the series is controlled by *m_max* parameter

make_model(« …. », « dort », rtsolver_options=dict(m_max=10))

- for PM, m_max is automatically forced to zero because emission, atmosphere and snow are azimuthally isotropic → mode 0 is sufficient

- for AM, m_max =2 by default which is ~ok for very smooth phase functions.
Have not explored the impact. I recommend m_max to be even (not odd).

DORT in SMRT

<u>Zenith angle dependency:</u>
- Treated with weighted non-uniform discretization.

$$\int_{-1}^{1} d\mu' \mathbf{P}^{(l),m}(\mu,\mu')\mathbf{I}^m(\mu',z) \approx$$

$$\sum_{i=1}^{N(l)} w_i^{(l)} \left[\mathbf{P}^{(l),m}(\mu,\mu_i^{(l)})\mathbf{I}^m(\mu_i^{(l)},z) + \mathbf{P}^{(l),m}(\mu,-\mu_i^{(l)})\mathbf{I}^m(-\mu_i^{(l)},z)\right]$$

In DMRT-QMS, discretization = Gaussian quadrature in all layers.

<u>Pros:</u> the integral is optimal in all layers, number of streams is the same in all layers (=angular resolution)
<u>Cons:</u> streams are not connected between the layers → boundary conditions are complex because interpolation is needed to apply Snell law. Risk of « leaks » of energy.

In SMRT-DORT, Gaussian quadrature is used for the most refringent layer (=the highest density) only and Snell's law is applied to obtain stream zenith angles in other layers.

<u>Pros:</u> the boundary conditions follow the physics (Snell law) and conserve the energy.
<u>Cons:</u> the integral discretization is sub-optimal except in the most refringent layer(s). The number of streams varies between layers due to total reflections.

DORT in SMRT

Zenith angle dependency in practice:

- The number of zenith angles of outgoing streams in the air is (much) lower than the parameter 'n_max_stream' which controls the number of streams in the most refringent (densest) layer.
- This number and the zenith angle of each stream depend on the max density of the snowpack.

Warning: sensitivity analysis where the max density varies can result in discontinuous curves when the number of streams in the air (or other light layers) increases/decreases.

To moderate this effect, DORT uses **linear interpolation** to convert the radiance computed at zenith angles enforced by Gaussian+Snell's law into the user requested zenith angles.

Advice:

- always work with 64 (default) or 128 streams or more if max density is high. Computation increases in **cubic power of #stream** (and layers).
- make twin simulations with n_max = n and n_max = n/2 to see the impact of #stream

Update: I have implemented a new approach where n_max_stream = #stream in the air.
Not fully tested yet, but could become the default in the future because it is more intuitive.

DORT in SMRT

After the discretization in azimuth angles, got coupled equations (in each layer)

$$\mu \frac{d\mathbf{I}^{c,m}(\mu,z)}{dz} = -\boldsymbol{\kappa}_e^{(l)}(\mu)\mathbf{I}^{c,m}(\mu,z) + \int_{-1}^{1} d\mu' \left[\mathbf{P}^{c,(l),m}(\mu,\mu')\mathbf{I}^{c,m}(\mu',z) - \mathbf{P}^{s,(l),m}(\mu,\mu')\mathbf{I}^{s,m}(\mu',z)\right] + \delta_m \boldsymbol{\kappa}_a^{(l)}(\mu)T^{(l)}\mathbf{1}$$

$$\mu \frac{d\mathbf{I}^{s,m}(\mu,z)}{dz} = -\boldsymbol{\kappa}_e^{(l)}(\mu)\mathbf{I}^{s,m}(\mu,z) + \int_{-1}^{1} d\mu' \left[\mathbf{P}^{s,(l),m}(\mu,\mu')\mathbf{I}^{c,m}(\mu',z) + \mathbf{P}^{c,(l),m}(\mu,\mu')\mathbf{I}^{s,m}(\mu',z)\right] + \delta_m \boldsymbol{\kappa}_a^{(l)}(\mu)T^{(l)}\mathbf{1}$$

$$\mathbf{P}^{c,m} = \begin{bmatrix} P_{11}^{c,m} & P_{12}^{c,m} & 0 & 0 \\ P_{21}^{c,m} & P_{22}^{c,m} & 0 & 0 \\ 0 & 0 & P_{33}^{c,m} & P_{34}^{c,m} \\ 0 & 0 & P_{43}^{c,m} & P_{44}^{c,m} \end{bmatrix} \qquad \mathbf{P}^{s,m} = \begin{bmatrix} 0 & 0 & P_{13}^{s,m} & P_{14}^{s,m} \\ 0 & 0 & P_{23}^{s,m} & P_{24}^{s,m} \\ P_{31}^{s,m} & P_{32}^{s,m} & 0 & 0 \\ P_{41}^{s,m} & P_{42}^{s,m} & 0 & 0 \end{bmatrix}$$

DORT in SMRT

Which can be assembled into one (using azimuthal isotropy):

$$\mu \frac{d\mathbf{I}^{\mathrm{e},m}(\mu,z)}{dz} = -\,\boldsymbol{\kappa}_e^{(l)}(\mu)\mathbf{I}^{\mathrm{e},m}(\mu,z) \quad + \int_{-1}^{1} d\mu' \left[\mathbf{P}^{\mathrm{e},(l),m}(\mu,\mu')\mathbf{I}^{\mathrm{e},m}(\mu',z)\right] + \delta_m \boldsymbol{\kappa}_a^{(l)}(\mu) T^{(l)}\mathbf{1}.$$

$$\mathbf{P}^{\mathrm{e},(l),m} = \begin{bmatrix} P_{11}^{\mathrm{c},(l),m} & P_{12}^{\mathrm{c},(l),m} & -P_{13}^{\mathrm{s},(l),m} & -P_{14}^{\mathrm{s},(l),m} \\ P_{21}^{\mathrm{c},(l),m} & P_{22}^{\mathrm{c},(l),m} & -P_{23}^{\mathrm{s},(l),m} & -P_{24}^{\mathrm{s},(l),m} \\ P_{31}^{\mathrm{s},(l),m} & P_{32}^{\mathrm{s},(l),m} & P_{33}^{\mathrm{c},(l),m} & P_{34}^{\mathrm{c},(l),m} \\ P_{41}^{\mathrm{s},(l),m} & P_{42}^{\mathrm{s},(l),m} & P_{43}^{\mathrm{c},(l),m} & P_{44}^{\mathrm{c},(l),m} \end{bmatrix}$$

DORT in SMRT

After discretization in zenith angles, the integral and -Ke I terms are merge in to A matrix:

$$\frac{d\boldsymbol{\mathcal{I}}^{(l),m}(z)}{dz} = -\boldsymbol{\mathcal{A}}^{(l),m}\boldsymbol{\mathcal{I}}^{(l),m}(z) + \delta_m {\mu^{(l)}}^{-1}\boldsymbol{\kappa}_a^{(l)} T^{(l)}\mathbf{1}$$

where

$$\boldsymbol{\mathcal{A}}^{(l),m} = [{\mu^{(l)}}^{-1}\boldsymbol{\kappa}_e^{(l)} - {\mu^{(l)}}^{-1}\boldsymbol{\mathcal{P}}^{(l),m}\mathbf{w}]$$

This equation is a first order ordinary differential equation.

→ general solution is easy I(z) = X exp(-A z) where X are unknowns
→ particular solution is easy because the non-I term is constant

Once the solution is known for each layer, the unkowns X are determined by applying the boundary conditions.
→ big linear system to solve: size is ~ 3 x 2 x sum(N(l)).
    N(l) number of streams in layer
    sum(N(l)) is the total number of streams. It increases with L, the number of layers
    3 = pola
    2 = up and down
This is usually the computational bottleneck.

Conclusion & perspectives

- DORT is robust for PM, a bit less for AM, you can trust it in most cases.

In the future:

- 2-flux and 6-flux. For checking how inaccurate they are, not recommended.
- 1st order solver for radar. Fast. Easy to extend.
- other DORT variants, ...
- Neural network solver for efficiency and inverse problems
- birefringent solver (→ snow anisotropy), starting from 1st order code