https://www.astrobio.net/also-in-news/arctic-sea-ice-helps-remove-co2-atmosphere/

Photo: Søren Rysgaard

Soil, atmosphere, sea and lake ice

Outline

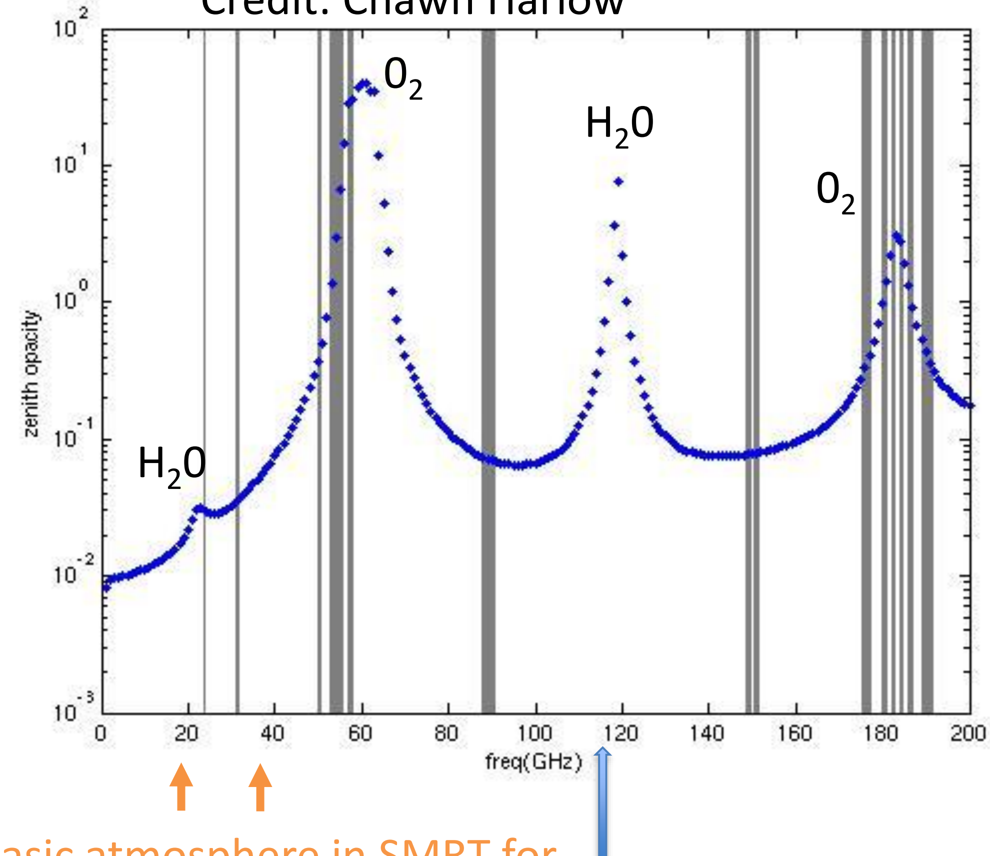
- Atmosphere
- Soil
- Sea ice
- Lake ice
- Ocean

Atmosphere

http://maxpixel.freegreatpicture.com/Blue-Atmosphere-Space-Clouds-Flying-Milky-Way-2295189

Atmospheric Opacity

Credit: Chawn Harlow

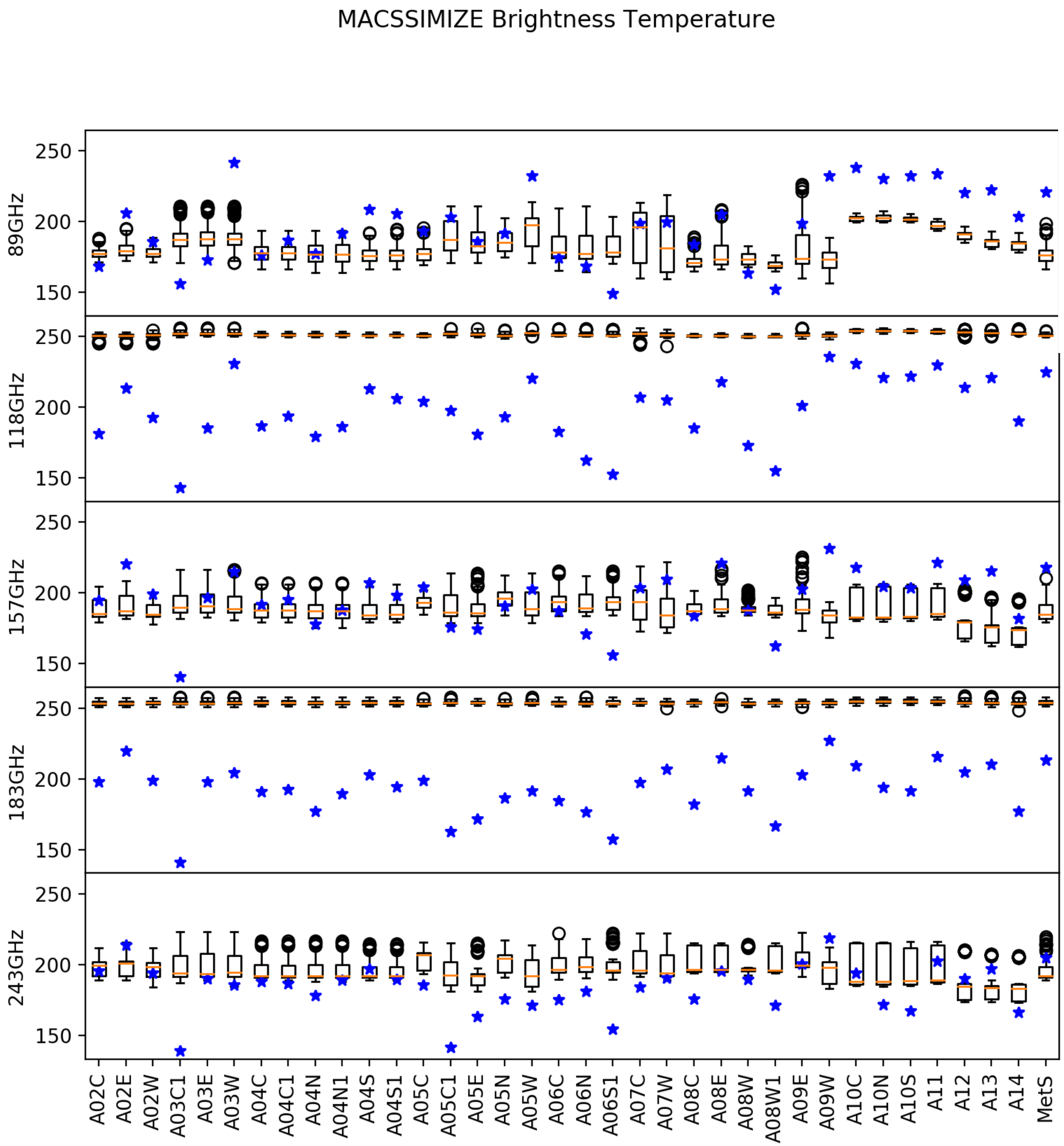


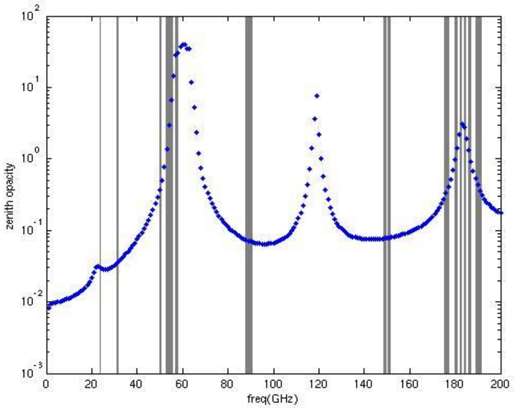
Basic atmosphere in SMRT for higher transparency frequencies

MetOp-SG

Grey bands: observation channels from AMSU-A and MHS

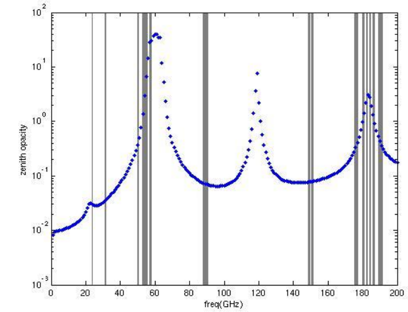
Blue: example atmospheric opacity



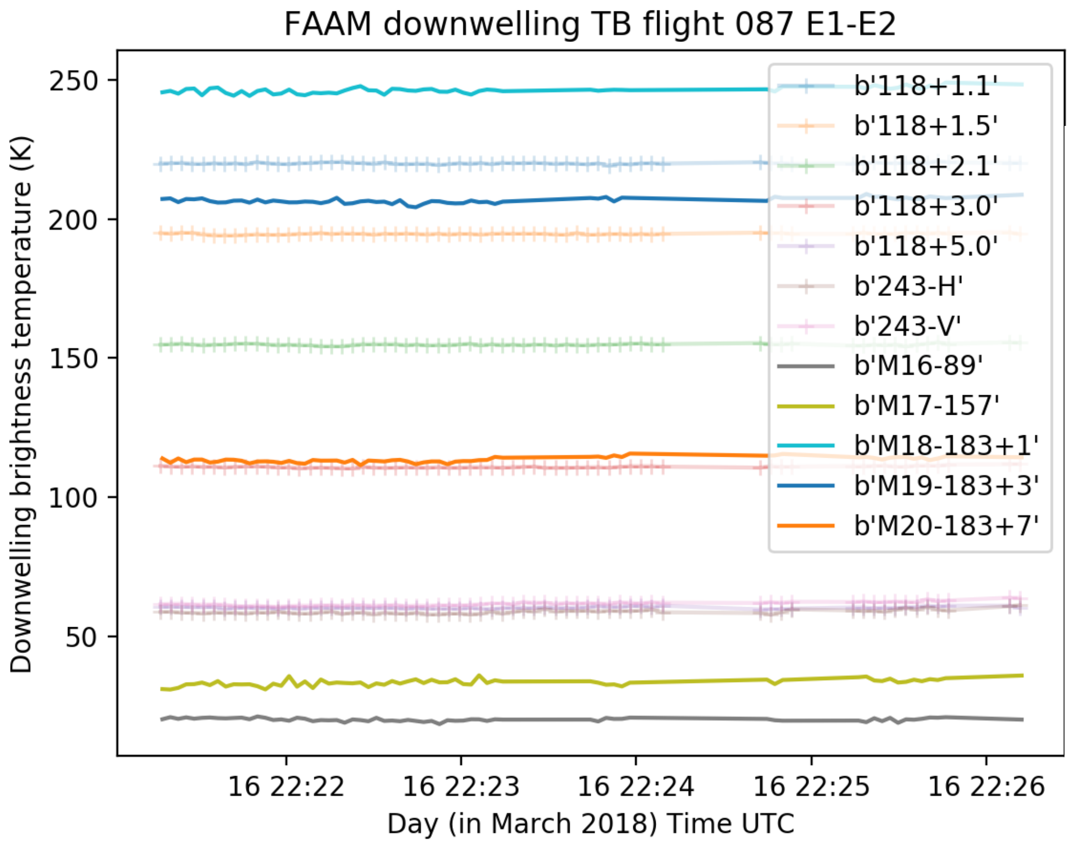


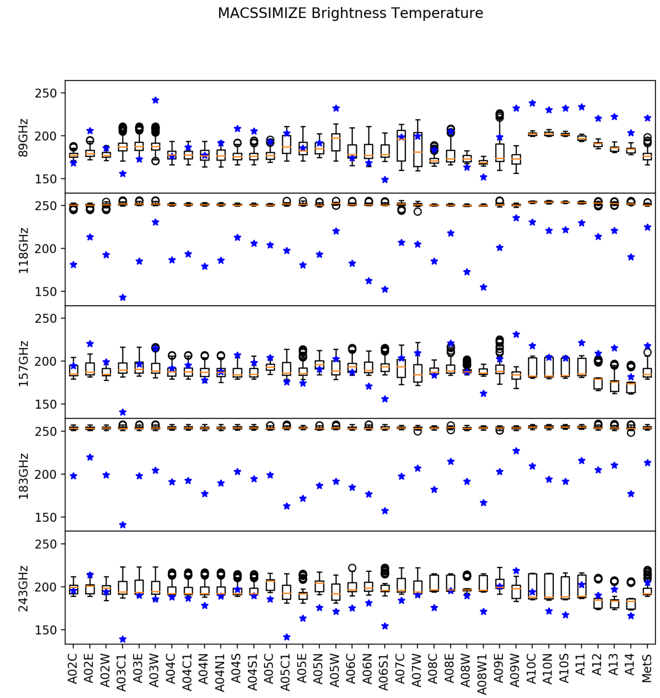
- MACSSIMIZE@TVC
- IBA_{EXP} ,DORT
- 3-layer
- SSA -> p_{ex}

Atmospheric Contribution



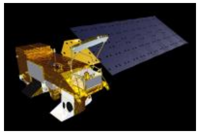
Nadir observations!

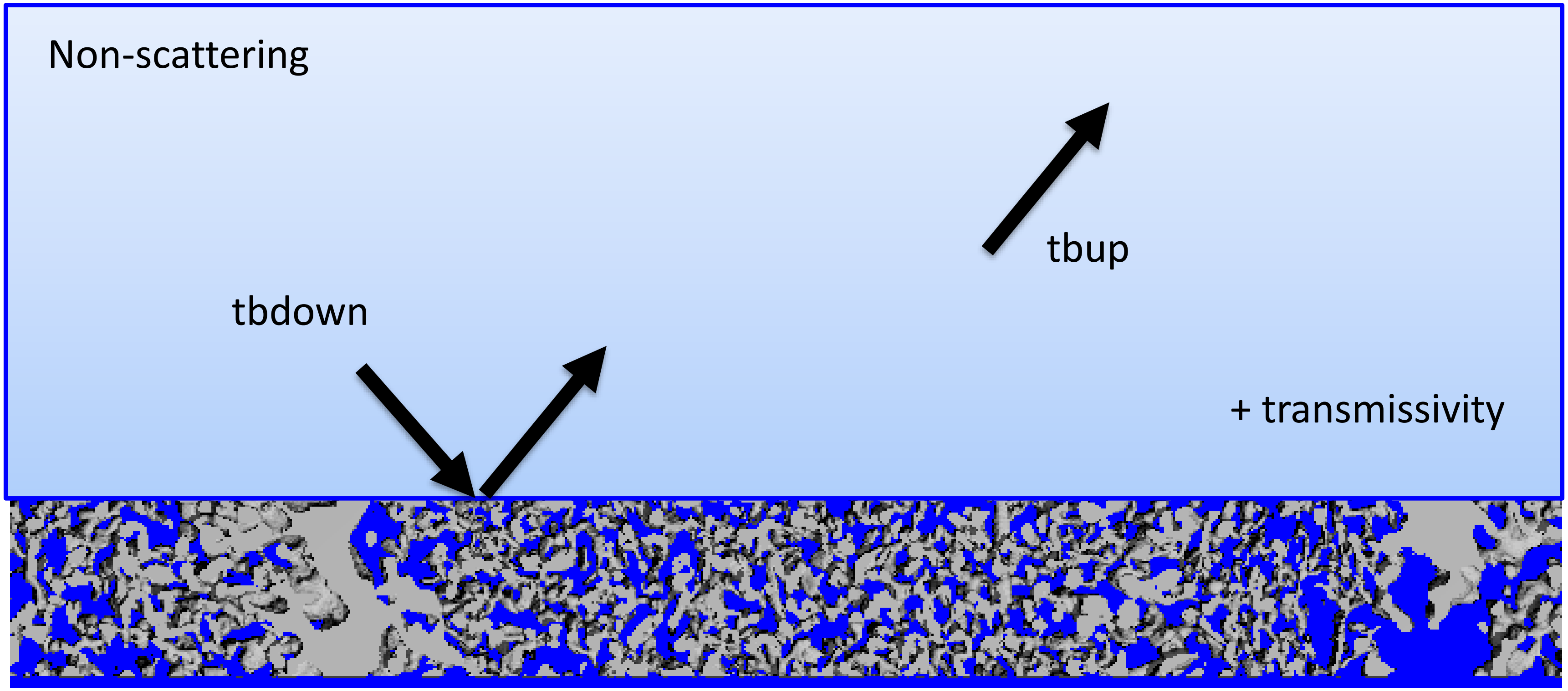




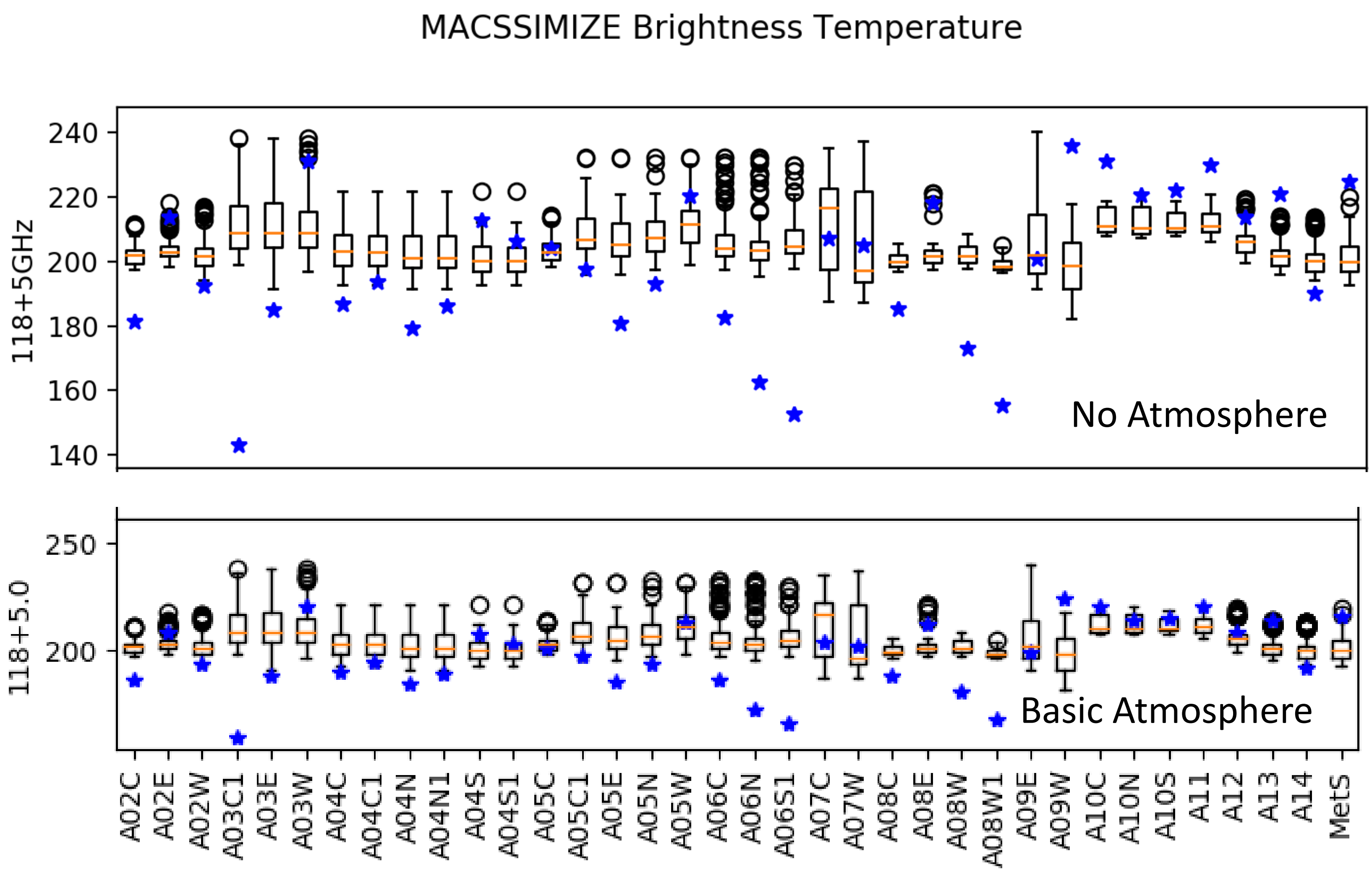
SMRT Basic Atmosphere

(NASA image by Marit Jentoft-Nilsen.)





Atmospheric Contribution



SMRT Basic Atmosphere

```
In [1]: from smrt import make_snowpack, make_model
        from smrt.inputs.sensor_list import passive
        from smrt.atmosphere.simple_isotropic_atmosphere import SimpleIsotropicAtmosphere

In [2]: # Create snowpack, sensor and model
        snowpack = make_snowpack([10], 'independent_sphere', temperature=260., density=320., radius=0.5e-3)
        rad = passive(21e9, 35)
        model = make_model('rayleigh', 'dort')

In [3]: atmos = SimpleIsotropicAtmosphere(tbdown=30., tbup=6., trans=0.90)
        model.run(rad, snowpack, atmosphere=atmos).TbV()

Out[3]: 149.5455044014671
```

If no atmosphere specified: tbup = 0, tbdown = 0, trans = 1

Idea:

medium = make_atmosphere(...) + make_snowpack(...) + make_ice(...) + make_ocean(...)
m.run(sensor, medium)

Types of substrate

A way to specify the lower boundary: what is underneath the lowest layer



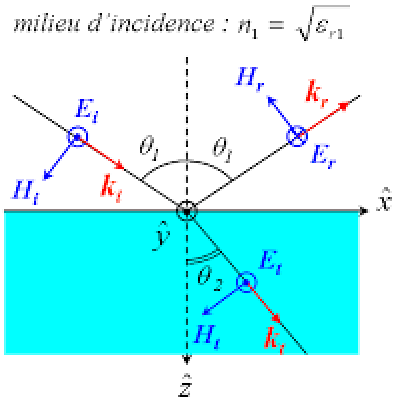


Specialist materials: reflector plate, absorber....

It's a choice!

$\frac{\partial T}{\partial z}$? λ?

Generic, flat surface (Fresnel)

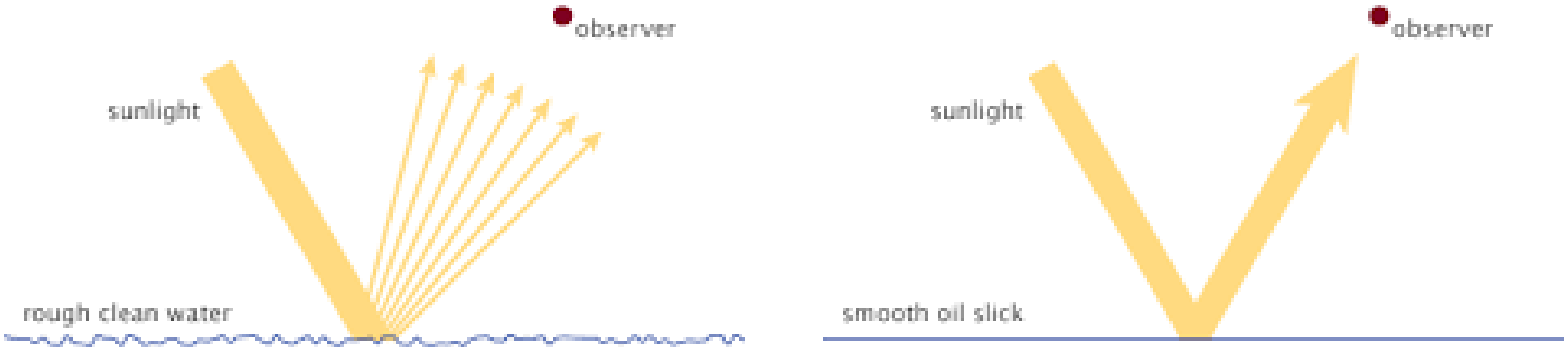


Use law of refraction: $n_1 \sin \theta_1 = n_2 \sin \theta_2$

$$\mathbf{R}^{bottom,(l),[specular]}(\mu) = \begin{bmatrix} \left(\frac{\epsilon_{eff}^{(l+1)} \cos\theta - \sqrt{\epsilon_{eff}^{(l)}} \sqrt{\epsilon_{eff}^{(l+1)} - \epsilon_{eff}^{(l)} \sin^2\theta}}{\epsilon_{eff}^{(l+1)} \cos\theta + \sqrt{\epsilon_{eff}^{(l)}} \sqrt{\epsilon_{eff}^{(l+1)} - \epsilon_{eff}^{(l)} \sin^2\theta}} \right)^2 & 0 \\ 0 & \left(\frac{\sqrt{\epsilon_{eff}^{(l)}} \cos\theta - \sqrt{\epsilon_{eff}^{(l+1)} - \epsilon_{eff}^{(l)} \sin^2\theta}}{\sqrt{\epsilon_{eff}^{(l)}} \cos\theta + \sqrt{\epsilon_{eff}^{(l+1)} - \epsilon_{eff}^{(l)} \sin^2\theta}} \right)^2 \end{bmatrix}$$

- SMRT layers are numbered from top = 0
- No cross-pol terms!!

Modification for effect of roughness



Modification for effect of roughness

Wegmüller and Mätzler, 1999

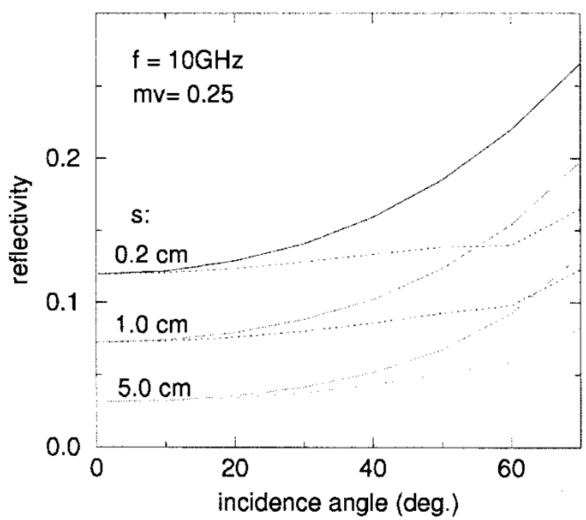


Fig. 1. Rough bare soil reflectivity model. Simulated incidence angle dependence of rough bare soil reflectivity at H- (solid) and V–polarization (dotted) for different standard deviations of the surface height s.

The new rough bare soil reflectivity model is defined by

$$r_{h,\,\mathrm{mod}}(mv, ks, \theta) = r_{h,\,\mathrm{Fresnel}} \cdot \exp\left\{-(ks)^{\sqrt{0.10\cos\theta}}\right\} \quad (12)$$

$\theta \leq 60°$:

$$r_{v,\,\mathrm{mod}}(mv, ks, \theta) = r_{h,\,\mathrm{mod}}(mv, ks, \theta) \cdot (\cos\theta)^{0.655} \quad (13a)$$

$60° \leq \theta \leq 70°$:

$$r_{v,\,\mathrm{mod}}(mv, ks, \theta) = r_{h,\,\mathrm{mod}}(mv, ks, \theta) \cdot (0.635 - 0.0014 \cdot (\theta - 60°)). \quad (13b)$$

The range of validity is restricted to the 1–100-GHz range at H- and V-polarization and incidence angles between 0° and 70°. The range of validity with respect to the standard

SMRT: Passive only

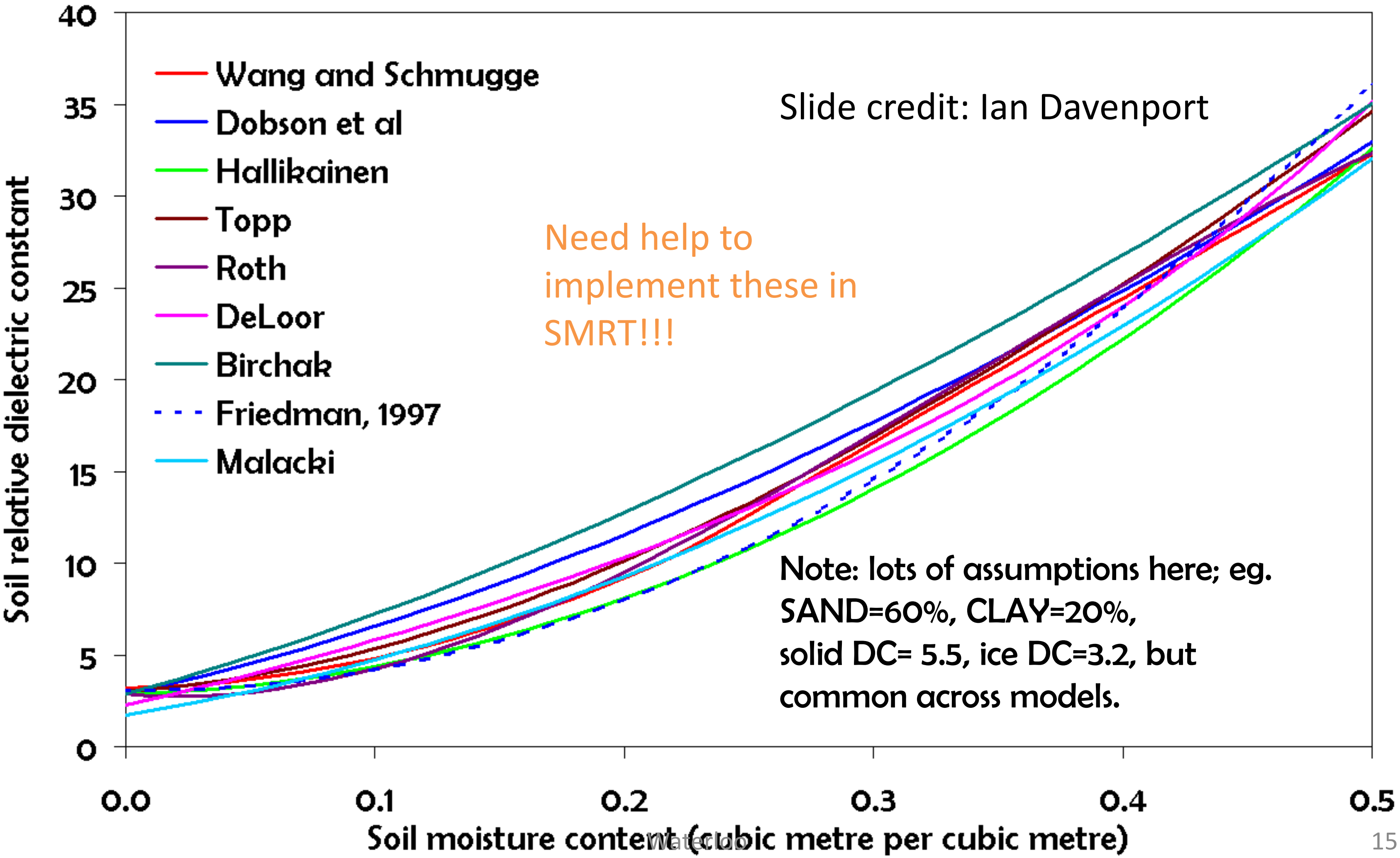
Generic, flat surface (Fresnel)

$$\mathbf{R}^{bottom,(l),[specular]}(\mu) = \begin{bmatrix} \left(\frac{\epsilon_{eff}^{(l+1)} \cos\theta - \sqrt{\epsilon_{eff}^{(l)}} \sqrt{\epsilon_{eff}^{(l+1)} - \epsilon_{eff}^{(l)} \sin^2\theta}}{\epsilon_{eff}^{(l+1)} \cos\theta + \sqrt{\epsilon_{eff}^{(l)}} \sqrt{\epsilon_{eff}^{(l+1)} - \epsilon_{eff}^{(l)} \sin^2\theta}} \right)^2 & 0 \\ 0 & \left(\frac{\sqrt{\epsilon_{eff}^{(l)}} \cos\theta - \sqrt{\epsilon_{eff}^{(l+1)} - \epsilon_{eff}^{(l)} \sin^2\theta}}{\sqrt{\epsilon_{eff}^{(l)}} \cos\theta + \sqrt{\epsilon_{eff}^{(l+1)} - \epsilon_{eff}^{(l)} \sin^2\theta}} \right)^2 \end{bmatrix}$$

Need to define soil permittivity. Depends on:

- Soil moisture
- Soil type
- Bulk soil density

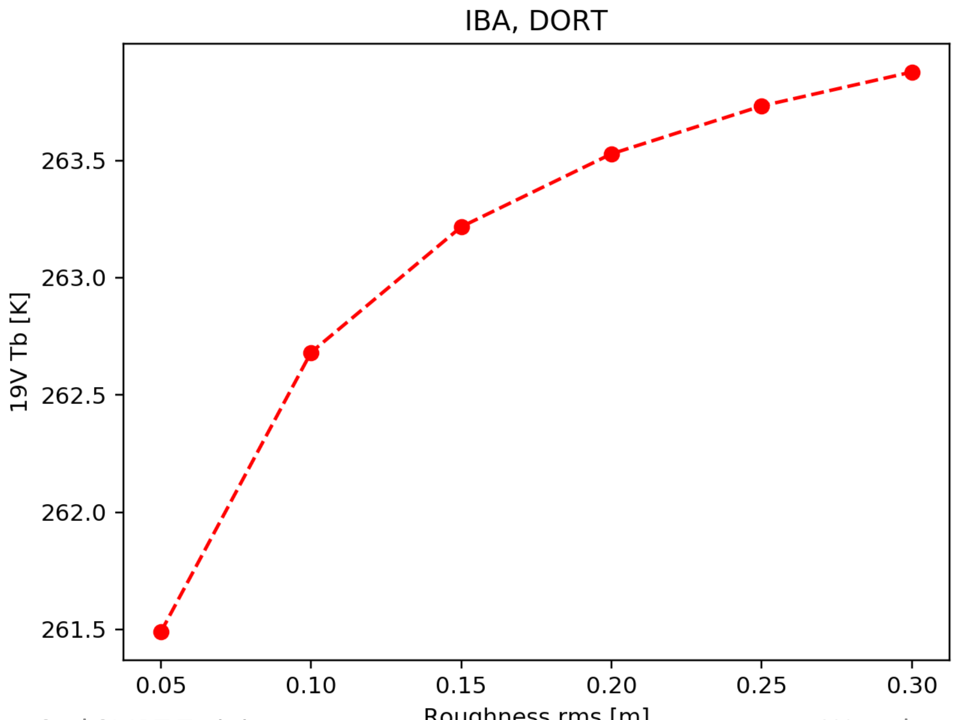
Empirical models of soil dielectric constant



SMRT substrate: soil

Make a soil substrate with Wegmüller and Mätzler (1999) model

```
In [1]: from smrt import make_soil, make_snowpack

In [2]: soil = make_soil('soil_wegmuller', 'dobson85', temperature=265, roughness_rms=0.25,
                         moisture=0.25, sand=0.01, clay=0.7, drymatter=1300)

In [3]: snow_with_soil = make_snowpack([1], "exponential", temperature=[265], density=[280], corr_length=[5e-5], substrate=soil)
```

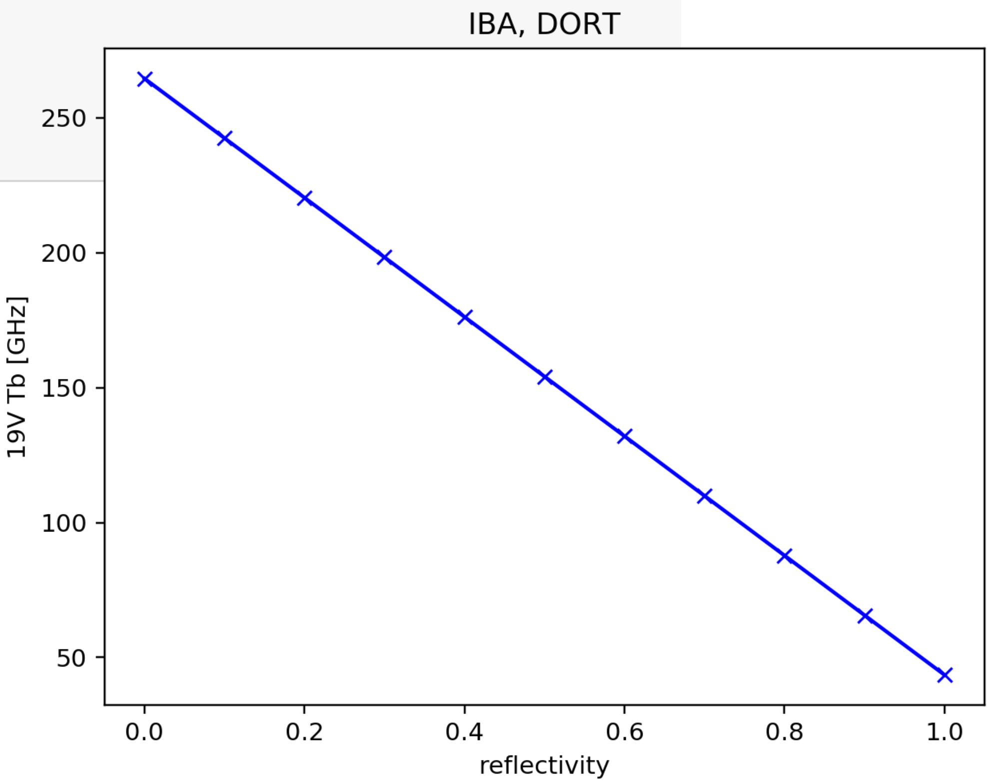


This way of specifying soil available in DMRT-ML

Reflector

SMRT substrate: reflector

```
from smrt.substrate.reflector import make_reflector
refl = np.arange(0, 1.1, 0.1)
reflector = [make_reflector(temperature=265, specular_reflection=r) for r in refl]
snowpacks = [make_snowpack([1], "exponential", temperature=[265], density=[280],
                            corr_length=[5e-5], substrate=r) for r in reflector]

# Run model on snowpacks
results = m.run(rad, snowpacks)

# Plot results
plt.plot(refl, results.TbV(),'bx-')
plt.xlabel('reflectivity')
plt.ylabel('19V Tb [GHz]')
plt.title('IBA, DORT')
```



MEMLS uses this approach

Useful for SMRT evaluation…

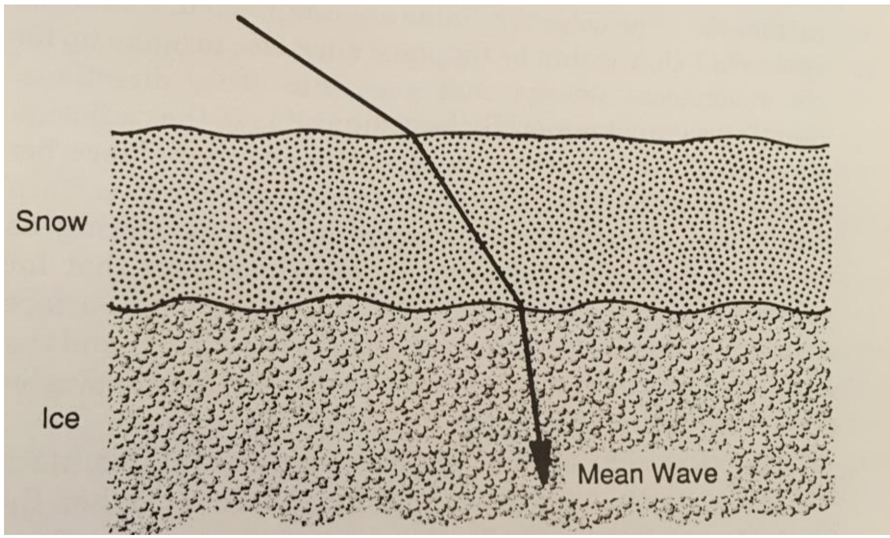
SMRT substrate: active reflector

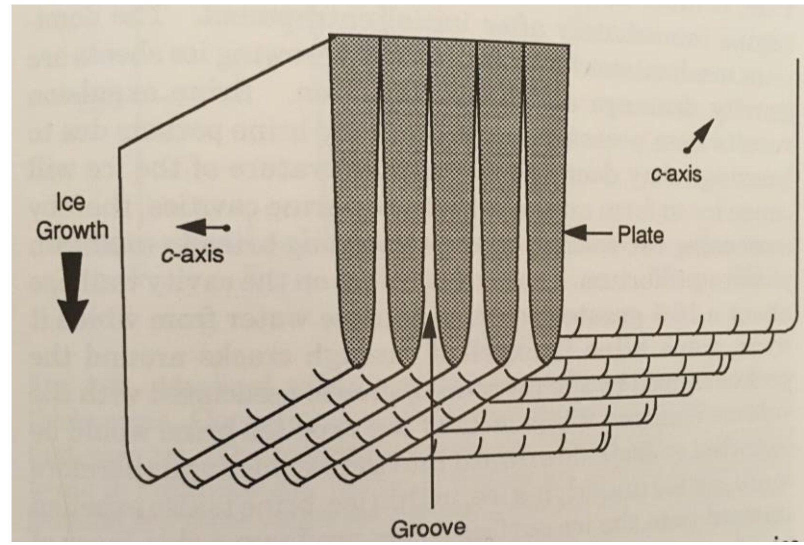
```
from smrt.inputs.sensor_list import active
from smrt.utils import dB
from smrt.substrate.reflector_backscatter import make_reflector

# Need to define an active sensor
scatt = active(13e9, 45)

# Make active reflector
reflector = make_reflector(temperature=265, specular_reflection=0.,
                           backscattering_coefficient={'VV': 0.1, 'HH': 0.1})

# Make snowpack
snow_active = make_snowpack([1], "exponential", temperature=[265],
                            density=[280], corr_length=[5e-5], substrate=reflector)
dB(m.run(scatt, snow_active).sigmaVV())
```

```
active model is not yet fully implemented, need modification for the third component

array([-12.23784209])
```

Sea Ice

Figures from
Tucker et al. 1992

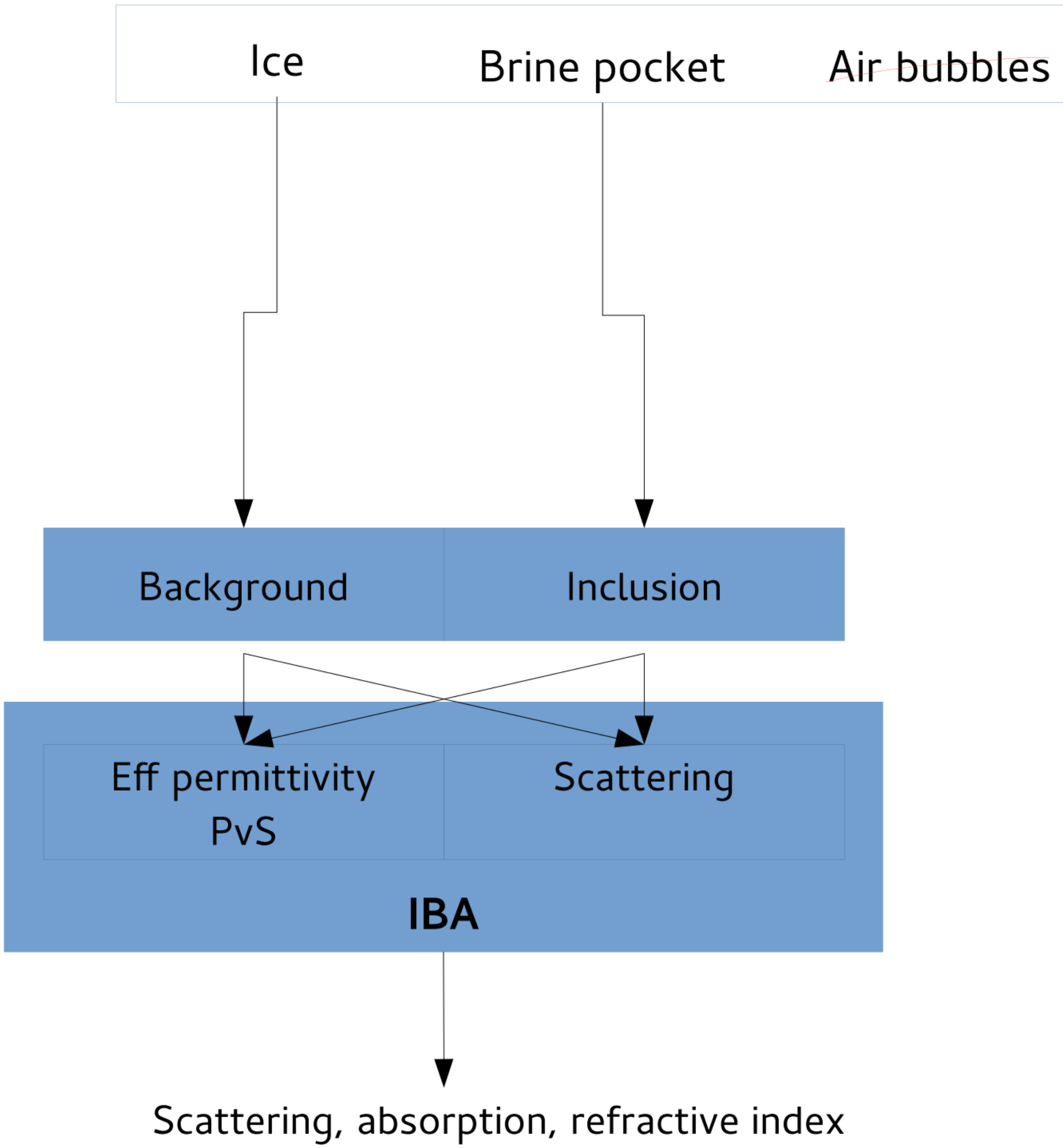


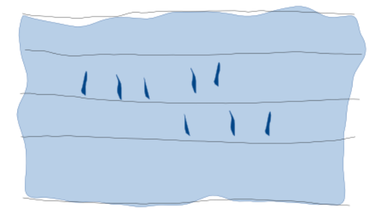


- 1st year ice: brine trapped in grooves of pure ice plates. Electromagnetically lossy
- Multi year ice: air bubbles in saline ice. Relatively transparent

Modeling sea–ice

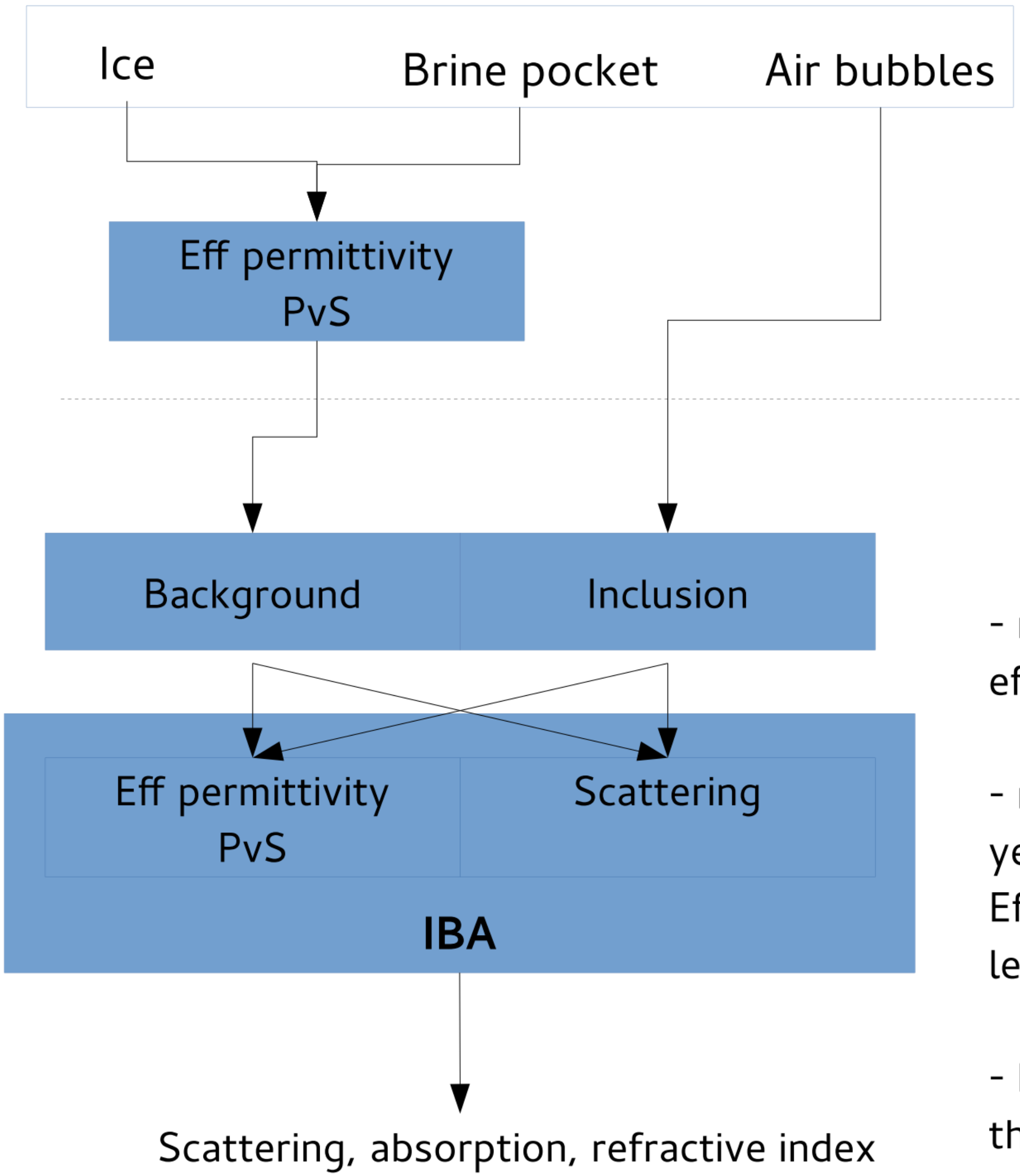
First year sea–ice

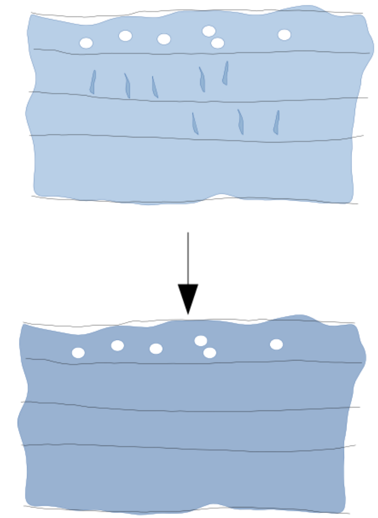




Modeling sea–ice

Multi year sea–ice





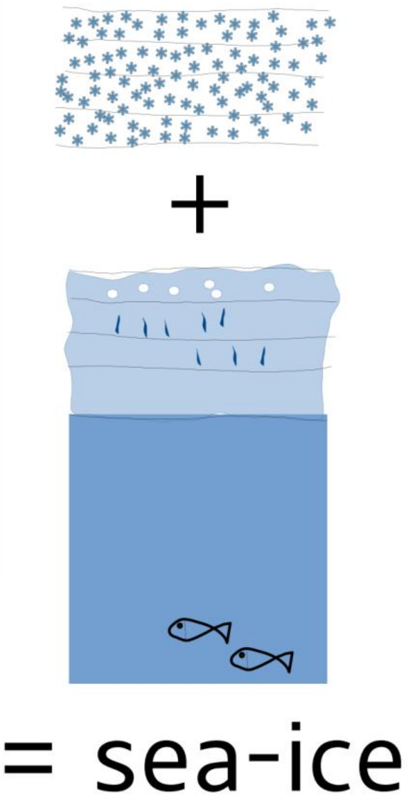
- no scattering by brine pockets, but their effect on the absorption is taken into account

- no 'continuity' between first year and multi year sea-ice, **except for low frequencies** (if Eff permittivy uses the same formula at both levels)

- but no better solution has been proposed in the past (to our knowledge)

Sea ice: implementation

```
# create the snowpack
snowpack = make_snowpack(thickness=thickness_s,
                         microstructure_model="exponential",
                         density=density_s,
                         temperature=temperature_s,
                         corr_length=p_ex_s)

# create the sea-ice
ice_column = make_ice_column(ice_type=ice_type, thickness=thickness,
                             temperature=temperature,
                             microstructure_model="exponential",
                             brine_inclusion_shape="spheres",
                             salinity=salinity,
                             porosity=porosity,
                             corr_length=p_ex,
                             add_water_substrate="ocean"
                             )

# add snowpack on top of ice column:
medium = snowpack + ice_column
```



Lake ice

Photo: N. Rutter

- Make ice column as for sea ice, with

  ice_type = ‘fresh’

- Medium is spherical air bubbles in pure ice background

NB current incompatibility: doc says ‘lake’, code says ‘fresh’

Water substrate

- Optional parameters: water temperature, salinity, permittivity model
- Default below lake ice: salinity = 0, T = 0^{o}C
- Default below sea ice: salinity = 0.032 kg/kg, T=-1.8^{o}C