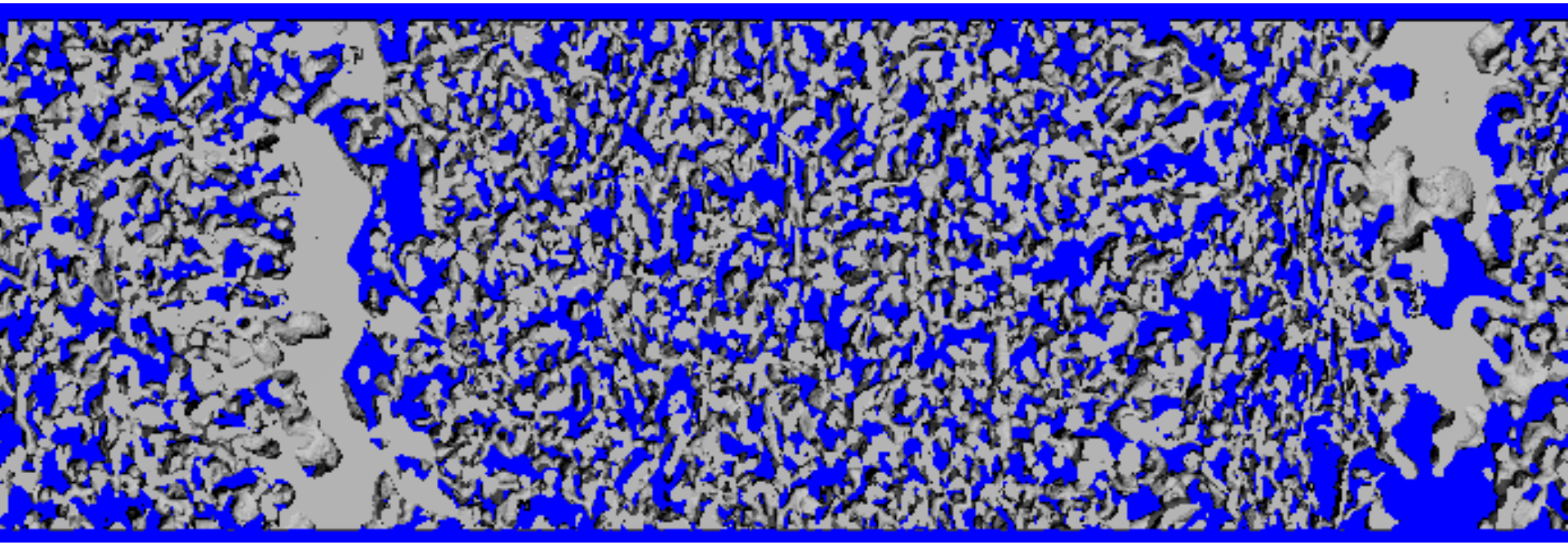
SMRT: A new, modular snow microwave radiative transfer model



Outline

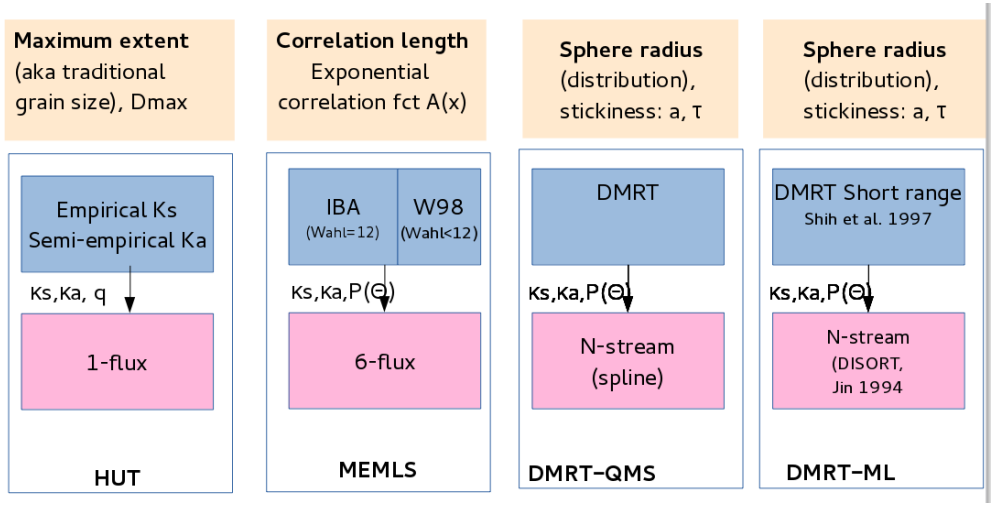
- Overview
- Implementation and structure
- Microstructure
- Model intercomparison via wrappers
- Github / website / help / documentation
- Summary

SMRT: Context

~~Are~~ existing snow microwave emission models so different ?

aren't

Löwe and Picard (TC, 2015)
Pan et al. (TGRS, 2016)



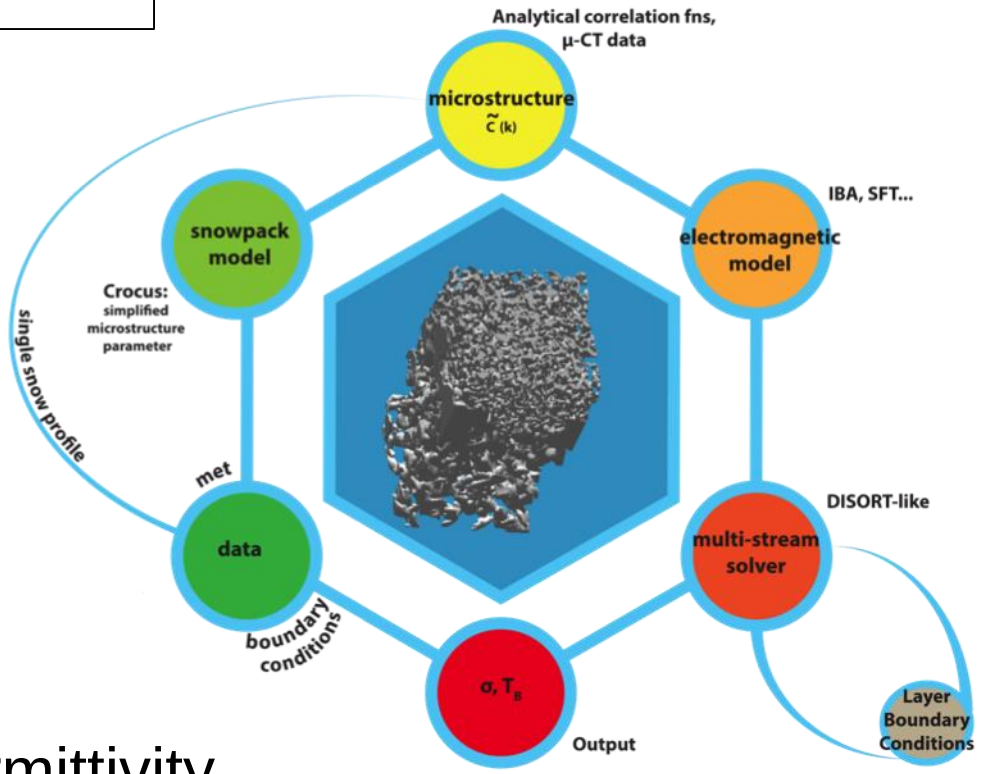
→ the different **electromagnetic theories**
→ the different **micro-structure representation** used by these models
→ the different **solution** of the radiative transfer equation

SMRT: Overview

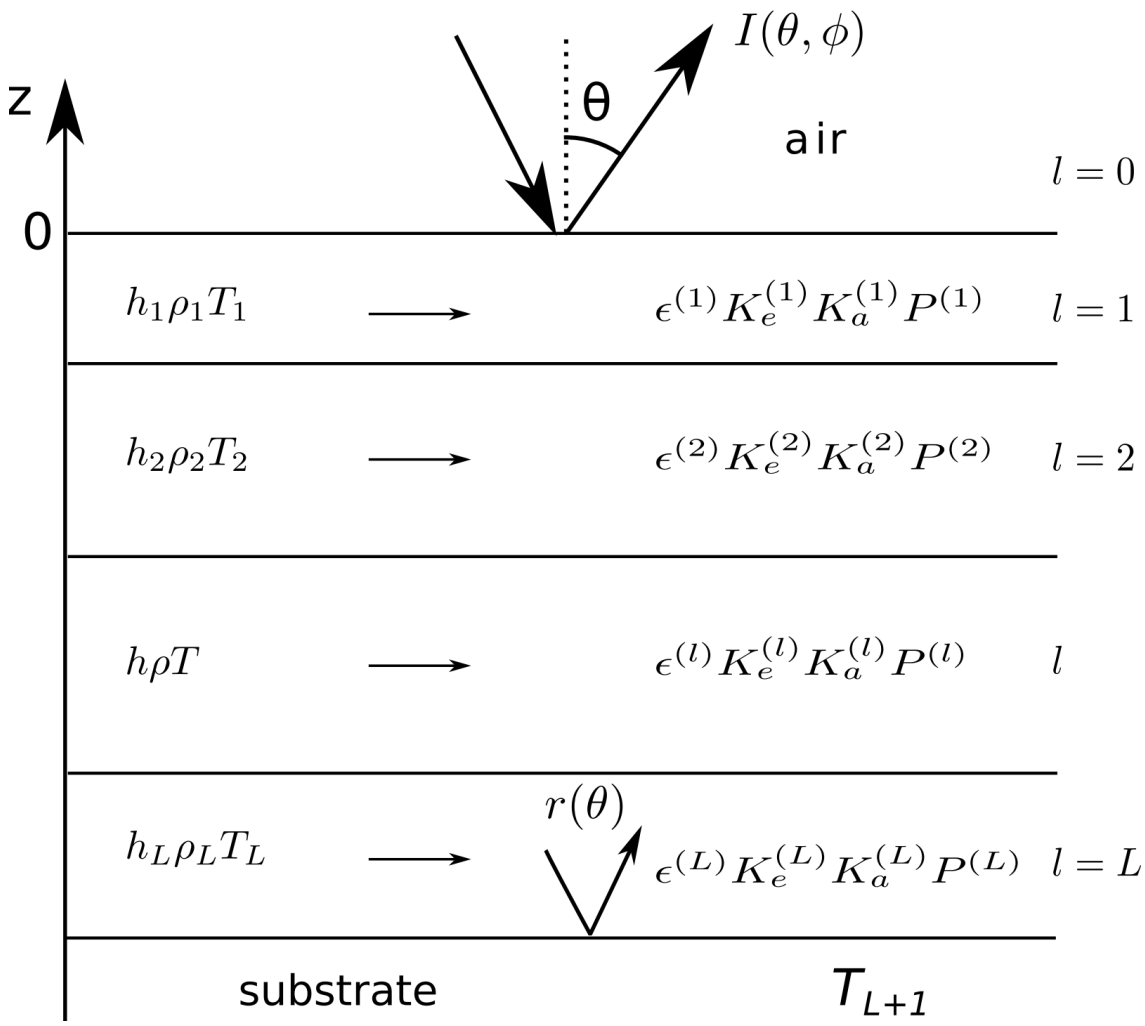
MICROSTRUCTURAL ORIGIN OF ELECTROMAGNETIC SIGNATURES IN MICROWAVE REMOTE SENSING OF SNOW

Main physics:

- Active / passive
- Multilayer
- Layer microstructure:
  ACF
- EM Theory:
  κ_s,κ_a, phase function, ε
- RT Solver
- Interface: Fresnel
- Substrate / atmosphere / permittivity



SMRT: Overview



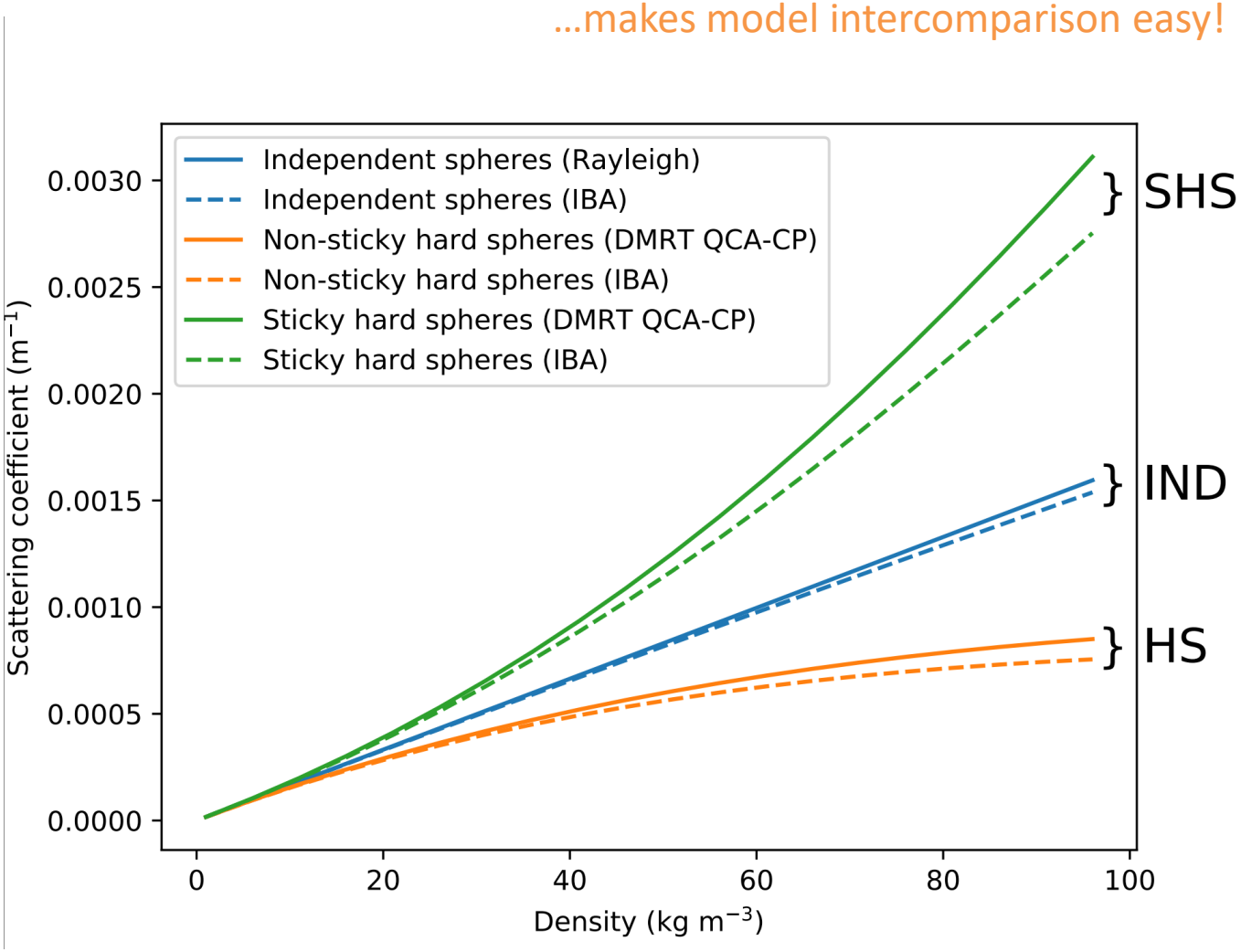
Switching modules….

- Microstructure
- EM model
- Interfaces
- Substrate
- Atmosphere
- RT Solver

Picard et al. (2018), GMD

SMRT: Overview

…makes model intercomparison easy!



Picard et al. (2018) GMD

SMRT: Implementation

Why python?

- open, object oriented (to realize "easy to use")

Imported modules hide computational complexity

Example of a model run:

Inputs

Create snowpack

Choose sensor config

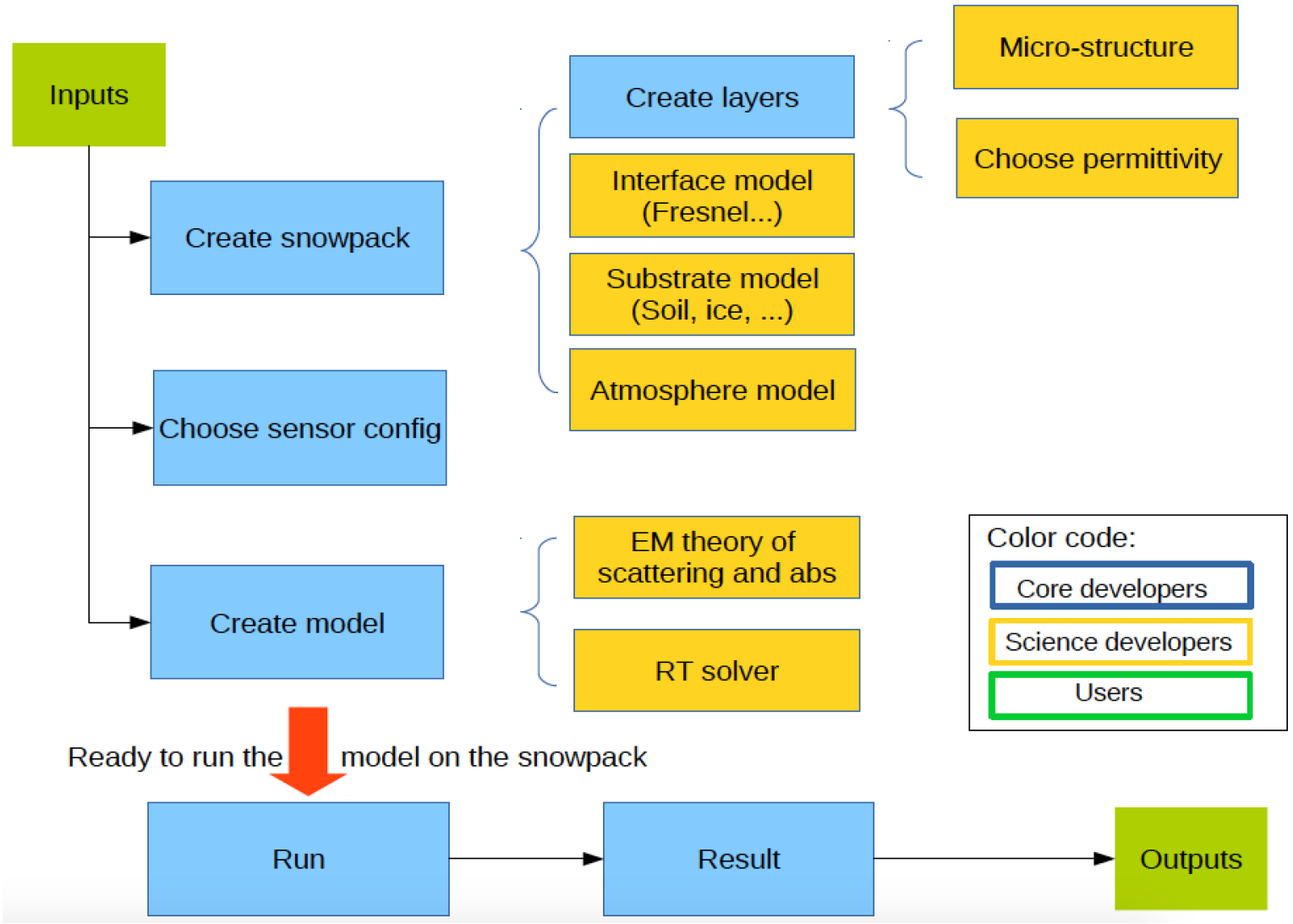
Create model

Run

Outputs

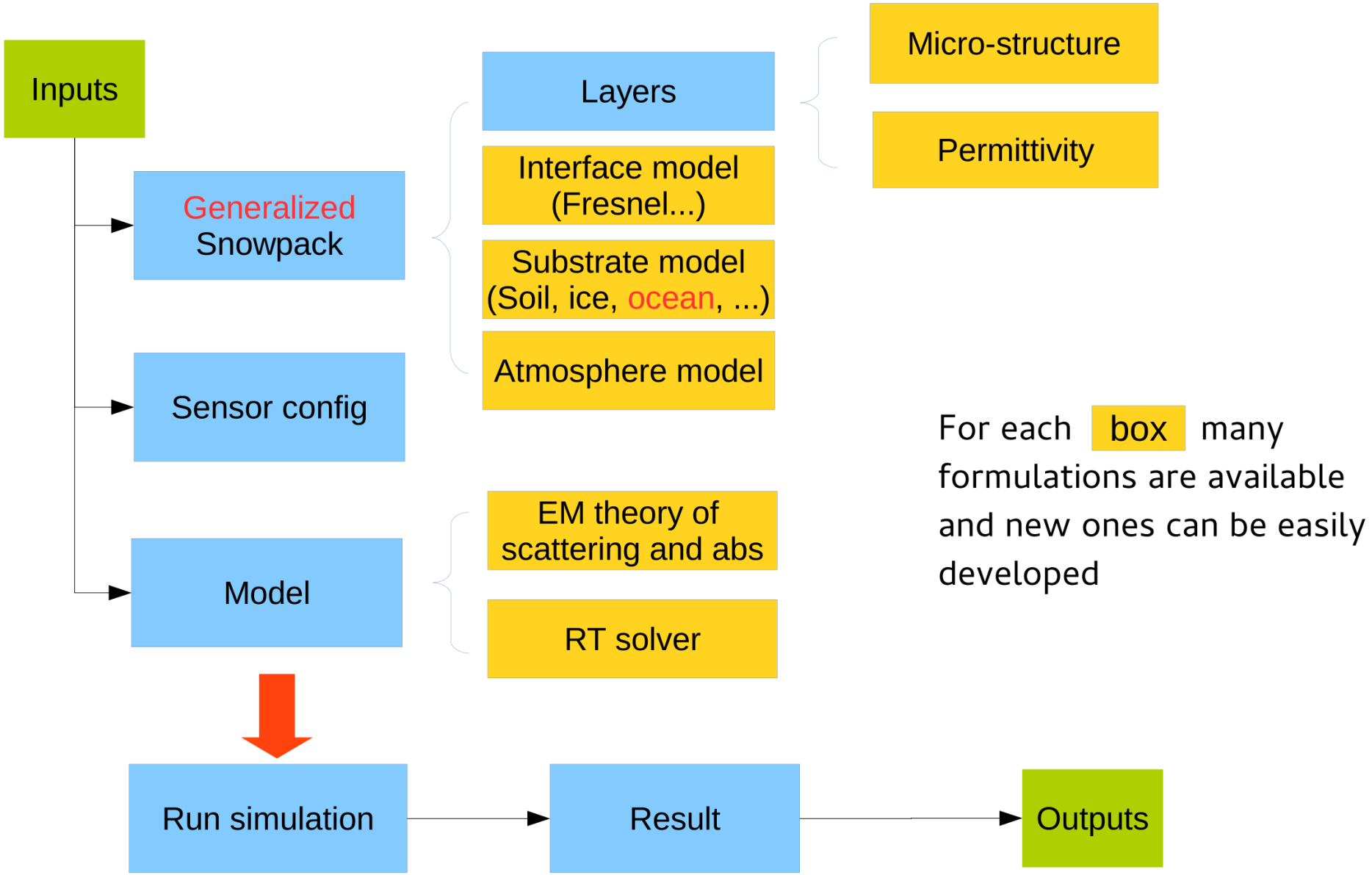
```
from smrt import make_snowpack, make_model, sensor

# prepare inputs
thickness = [100]
corr_length = [5e-5]
temperature = [270]
density = [320]

# create the snowpack
snowpack = make_snowpack(thickness=thickness,
                         microstructure_model="exponential",
                         density=density,
                         temperature=temperature,
                         corr_length=corr_length)

# create the sensor
radiometer = sensor.amsre('37V')

# create the model
m = make_model("iba", "dort")

# run the model
result = m.run(radiometer, snowpack)

# outputs
print(result.TbV(), result.TbH())
```

SMRT: Structure



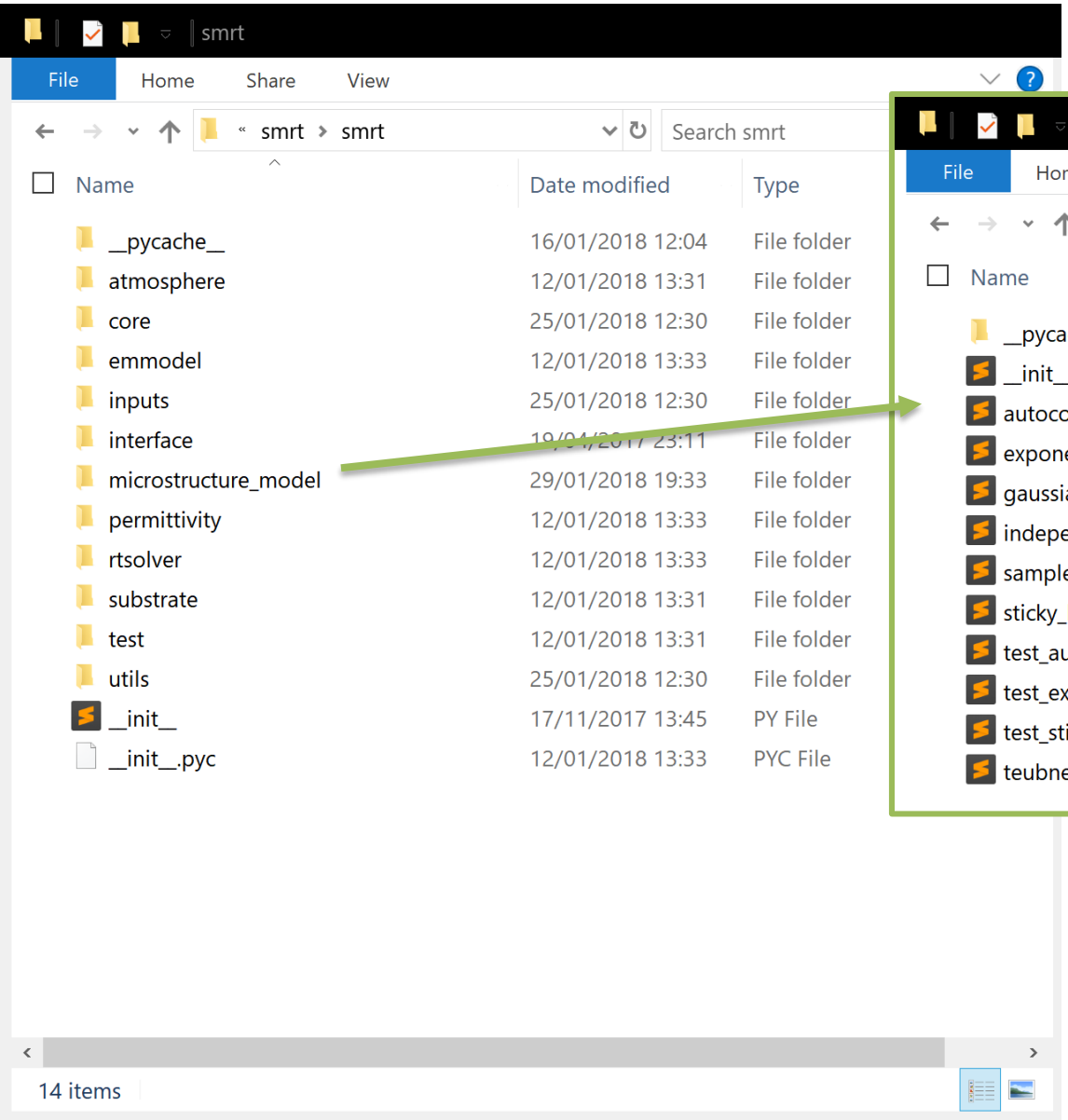
Implementation

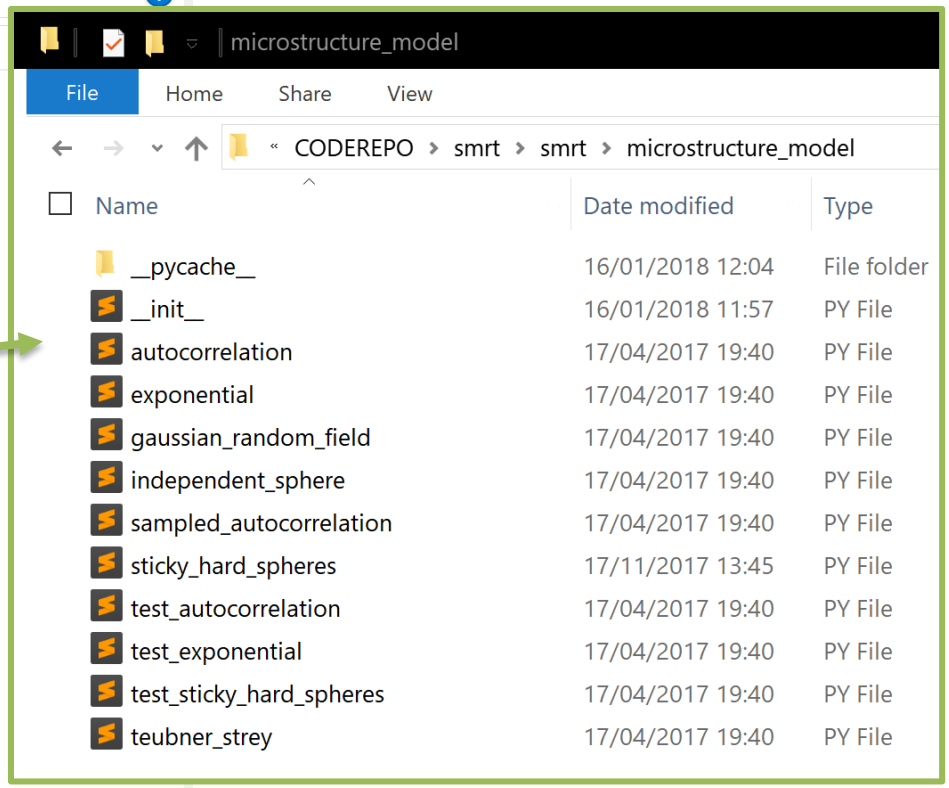
Structure of SMRT **after sea-ice**:



For each box many formulations are available and new ones can be easily developed

SMRT: Structure





Easy to add modules

\+ Tests!

SMRT: Microstructure

Layer microstructure:

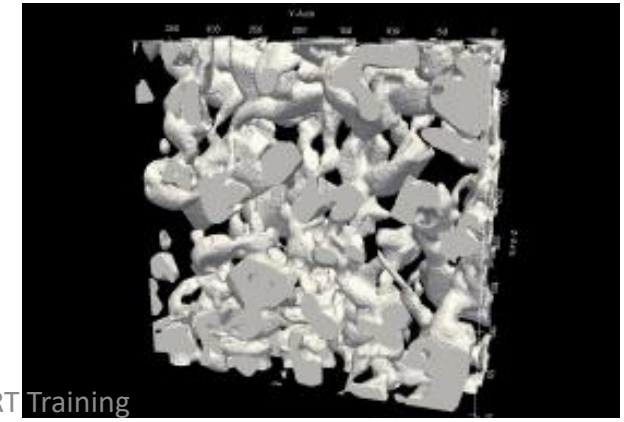
- represented by a two-point correlation *function*

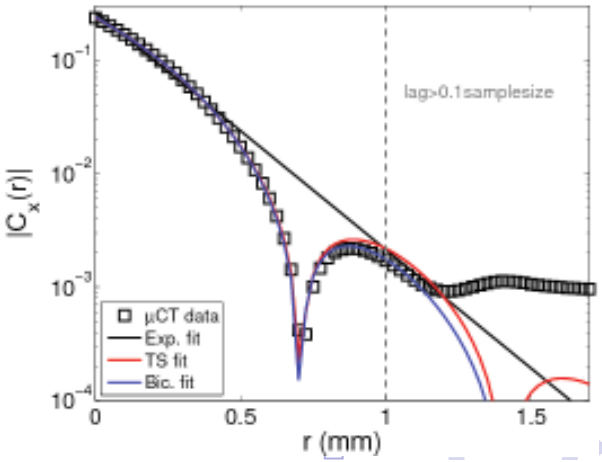
Implemented models: (and reasons for them)

- Exponential model (MEMLS)
- Sticky hard spheres (DMRT-ML, DMRT-QMS)
- Independent sphere (classic)
- Teubner–Strey (empirical evidence)
- (Level-cut) Gaussian random fields (full-field methods)

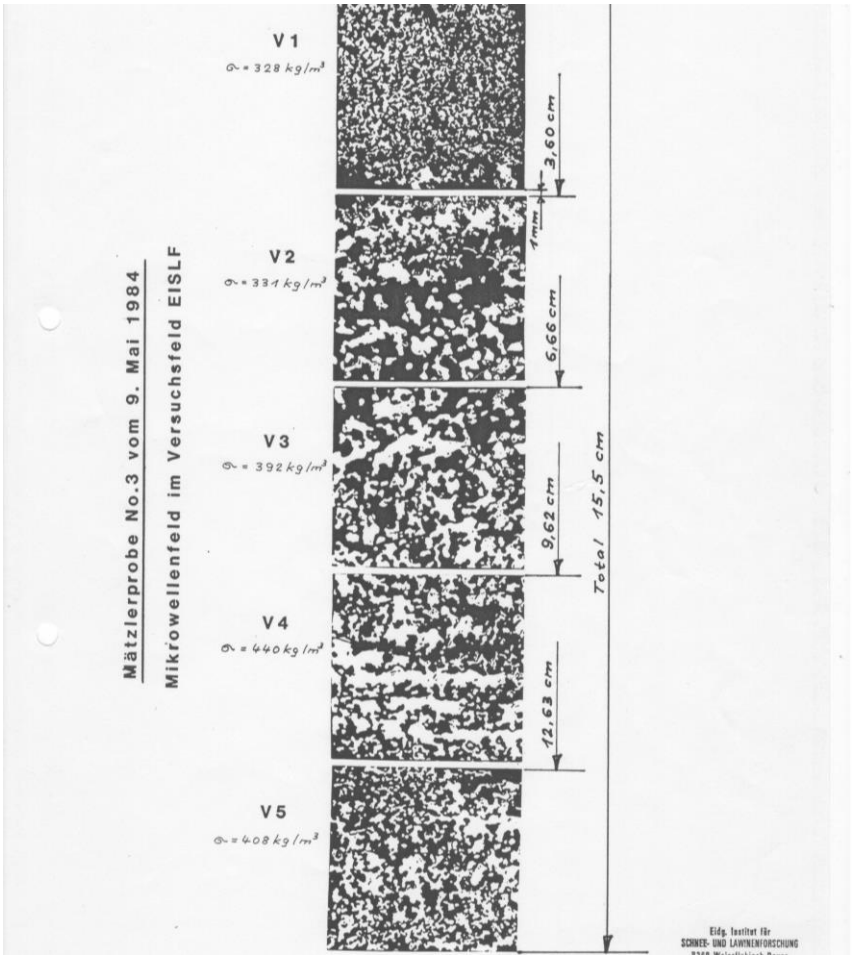
Why different choices?

- To be flexible with non-obvious correlation functions:





Micro-CT is not the only way!



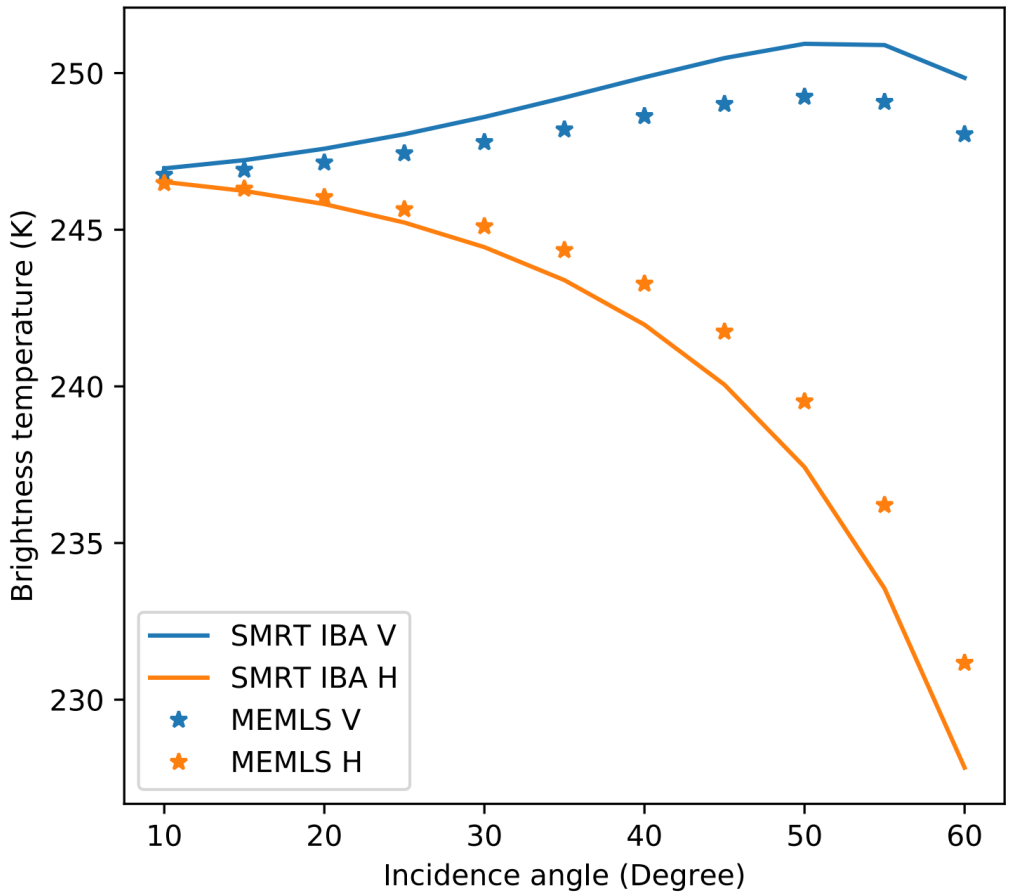
SMRT: Legacy

To facilitate model inter-comparison:

- Shallow wrappers for MEMLS, HUT, DMRT-QMS (no code)

```
# general import for smrt
from smrt import make_snowpack, make_model, sensor

# import for memls
from smrt.utils import memls_legacy

# prepare snowpack
pc = 0.2e-3
snowpack = make_snowpack(thickness=[10], microstructure_model="exponential",
                         density=[300], temperature=[265], corr_length=pc)

# create the sensor
theta = range(10, 80, 5)
radiometer = sensor.passive(37e9, theta)

# create the EM Model
m = make_model("iba", "dort")

# run the model
sresult = m.run(radiometer, snowpack)

# run MEMLS matlab code
mresult = memls_legacy.run(radiometer, snowpack)


# outputs
plt.plot(theta, sresult.TbV(), 'r-', label='SMRT V')
plt.plot(theta, sresult.TbH(), 'r--', label='SMRT H')
```

SMRT: intercomparison



Picard et al. (2018), GMD

https://github.com/smrt-model/smrt

This repository Search Pull requests Issues Marketplace Explore

smrt-model / smrt

Unwatch 4 Unstar 4 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Snow Microwave Radiative Transfert model to compute thermal emission and backscatter from snowpack

Edit

modeling snow microwave Manage topics

59 commits 1 branch 0 releases 2 contributors LGPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

ghislainp committed Jan 25, 2018 adapt test to the commit on absorption 808f257 Latest commit d418d57 Jan 25, 2018

doc	Change to documentation theme	12 days ago
examples	correction of the import in the examples	9 months ago
smrt	adapt test to the commit on absorption 808f257	4 days ago
.gitignore	Configuration and files for readthedocs autocompile	
LICENSE	Disclaimer and copyright added	
README.md	reduce the level of warning in the Disclamer	2 months ago
requirements.txt	Added pandas, scipy and xarray to requirements file	10 months ago
setup.py	ajout de find_package	

git pull

https://www.smrt-model.science

SMRT Home Getting Started Documentation Contribute

SMRT: Snow Microwave Radiative Transfer model

SMRT is an **active / passive microwave radiative transfer model for multilayer snow** written in Python. It was developed with European Space Agency support in order to investigate the representation of the snow microstructure, the main driver of scattering.

SMRT is modular, so allows easy intercomparisons between different modelling approaches in a plug-and-play way. SMRT proposes different electromagnetic theories to compute scattering (DMRT, IBA, Rayleigh independent, ...). In the case of IBA, different microstructure representations can be used (Sticky Hard Spheres, Exponential, Gaussian random field, ...). The current version proposes only one radiative transfer solver (DORT) but this can be extended. Last but not least, wrappers are included to run MEMLS, HUT and DMRT-QMS models (in their original matlab code) from within SMRT. Whilst there is plenty to get started with, there are more theoretical advances that can be made. SMRT is intended to be a community model - all are welcome to use it, and to contribute to its development!

SMRT uses the lastest python version (e.g. 3.6) but also works with earlier versions (2.7, 3.4 and higher). The code is open source and is hosted on github.

Getting started with SMRT

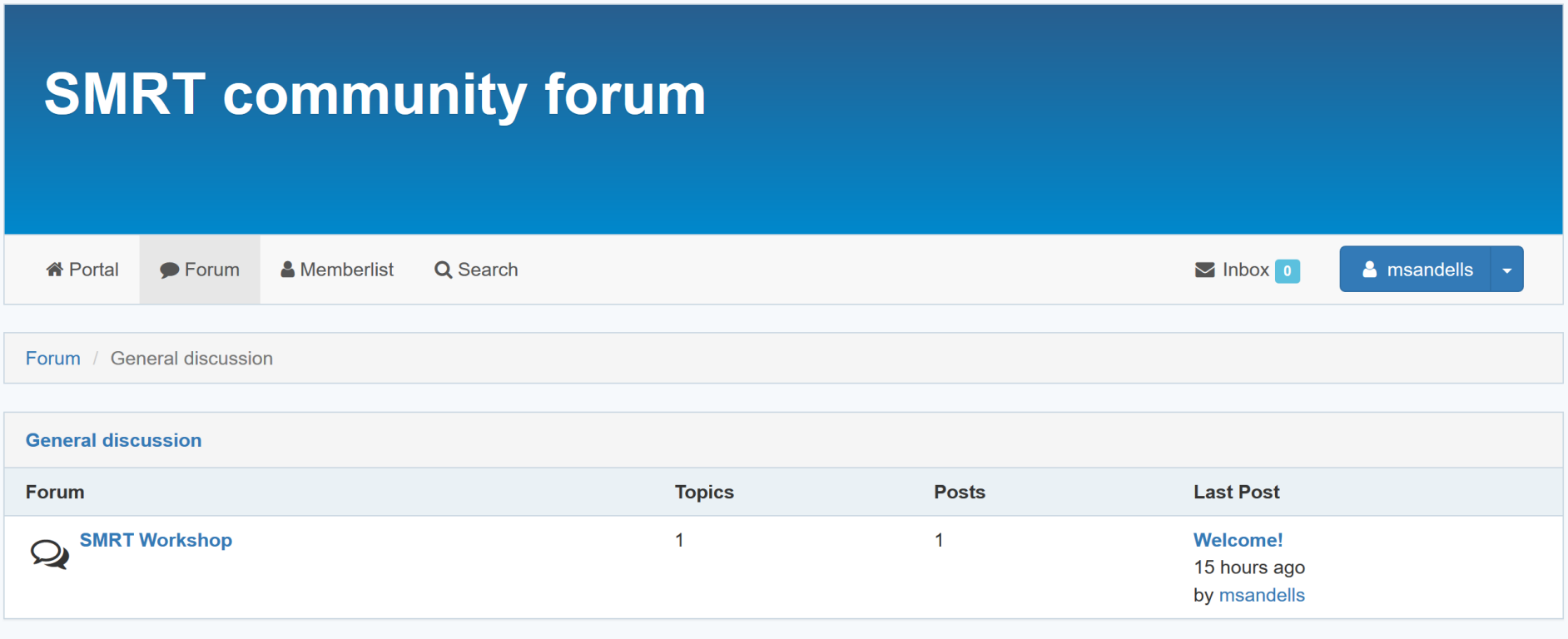
Using SMRT is easy

You must write a (short) driver code, SMRT is a library that your driver code call to perform a calculation. This is simple, there are four main steps in a typical driver code:

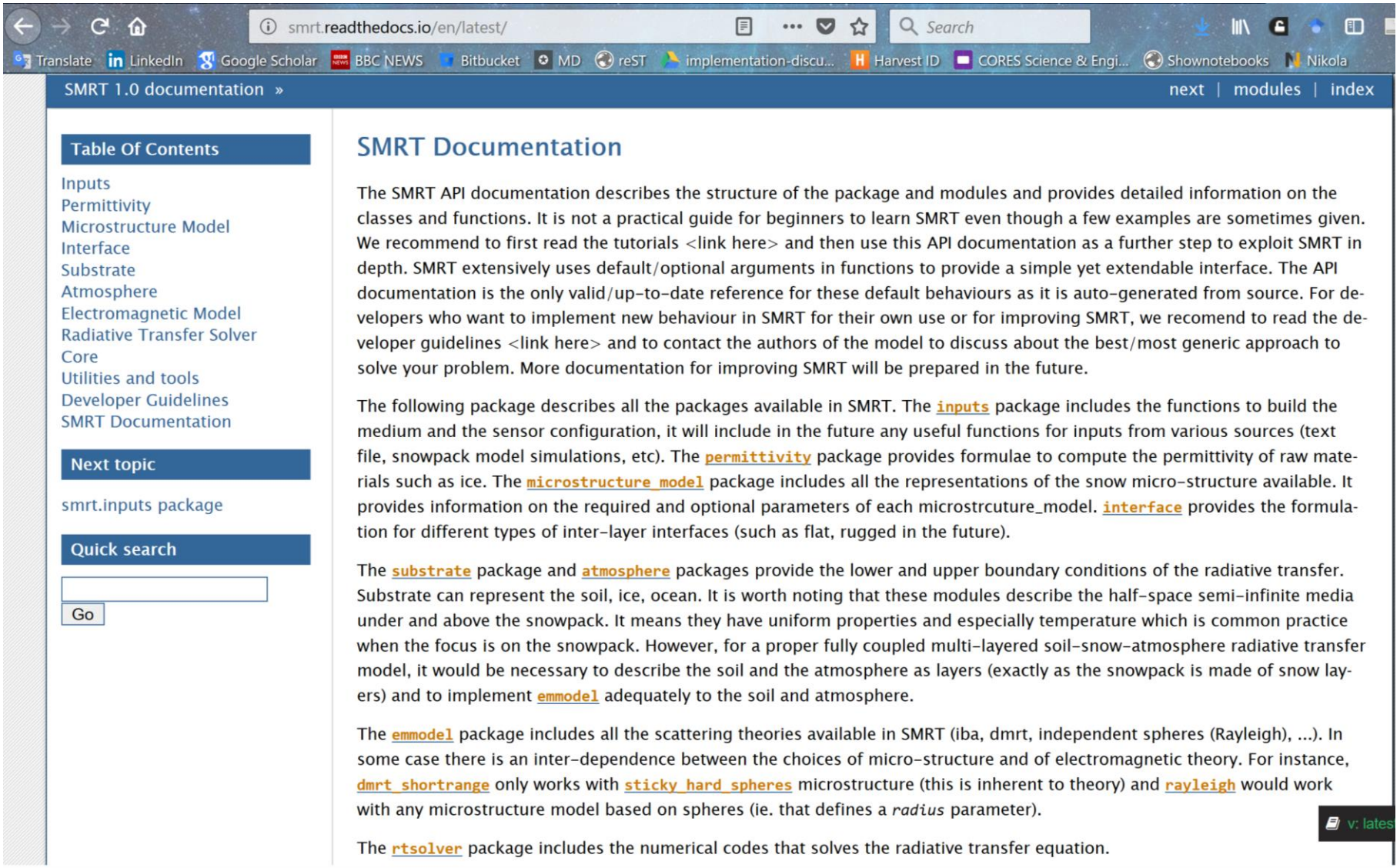
1. construct a snowpack (either from field data or snowpack model).

```
snowpack = make_snowpack(thickness=10.,   # 10 m deep snowpack
                         microstructure_model="sticky_hard_spheres",
                         density=320.0,   # 320 kg/m3
                         temperature=260, # 260 K
                         radius=100e-6)   # 100 microns
```

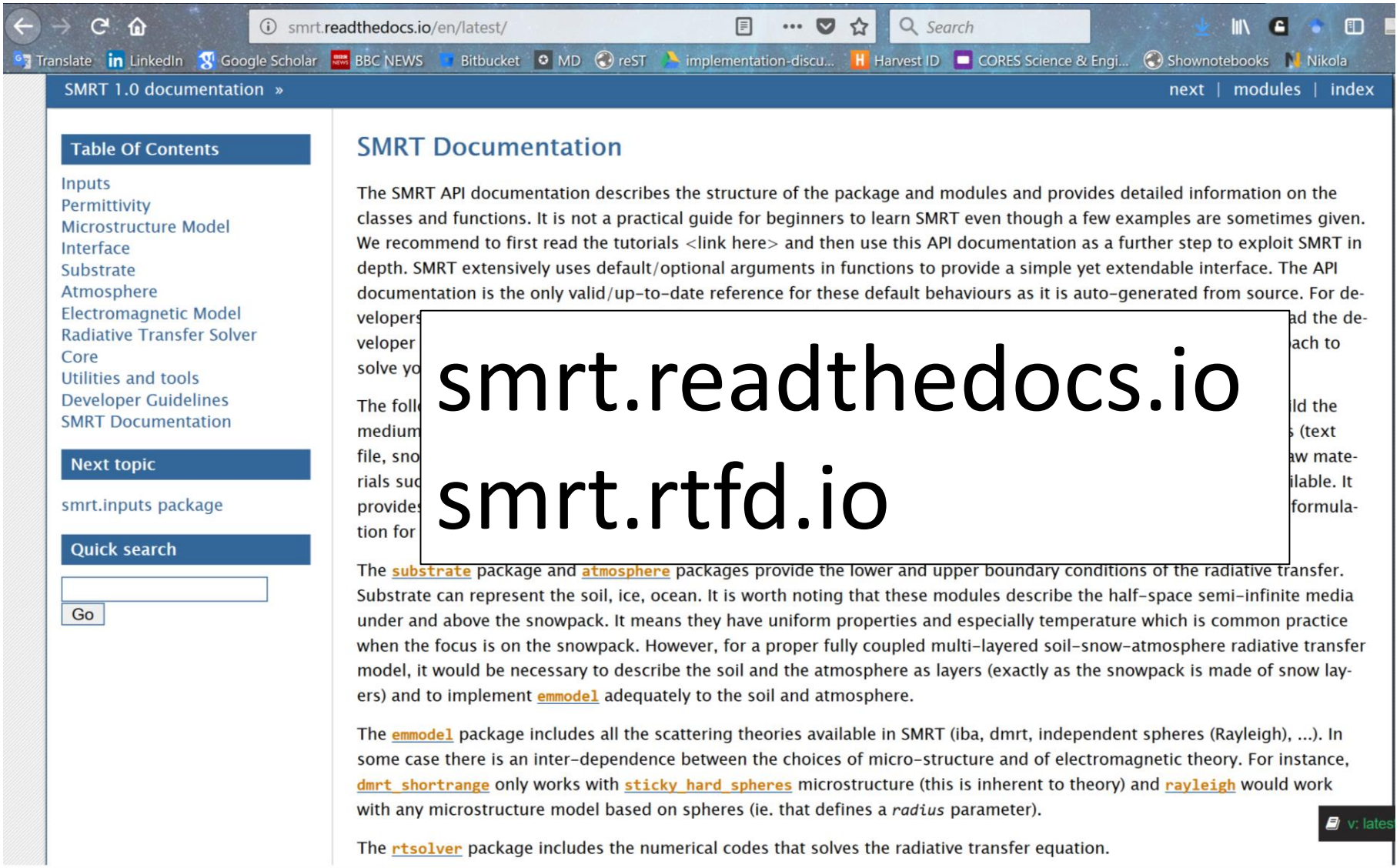
http://community.smrt-model.science



Online documentation



Online documentation



SMRT

- Why a new model ?
We don't need a new model (yet) but we need:
a repository of microwave community knowledge
= merge all RT models / theories in one code base, one framework
with extended capabilities to explore the micro-structure
with multi mode capabilities (passive, radar, altimeter)
with easier access for beginners and non-specialists
using modern and more efficient languages and programming techniques

SMRT: flexibility, python, git, forum, documentation

We hope SMRT will be taken up by the community as a **joint** endeavour

https://www.smrt-model.science
https://github.com/smrt-model/smrt
http://community.smrt-model.science
smrt.readthedocs.io