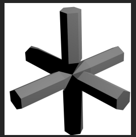
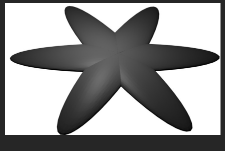
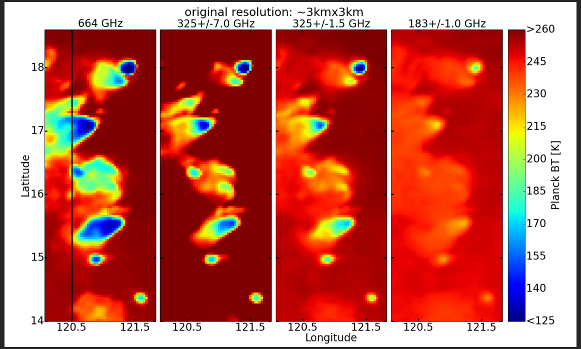
Scattering calculations in ARTS: Improvements and new features

Jana Mendrok, P. Eriksson
(Chalmers University of Technology)

and the ARTS team
(Hamburg, ...)







Contents

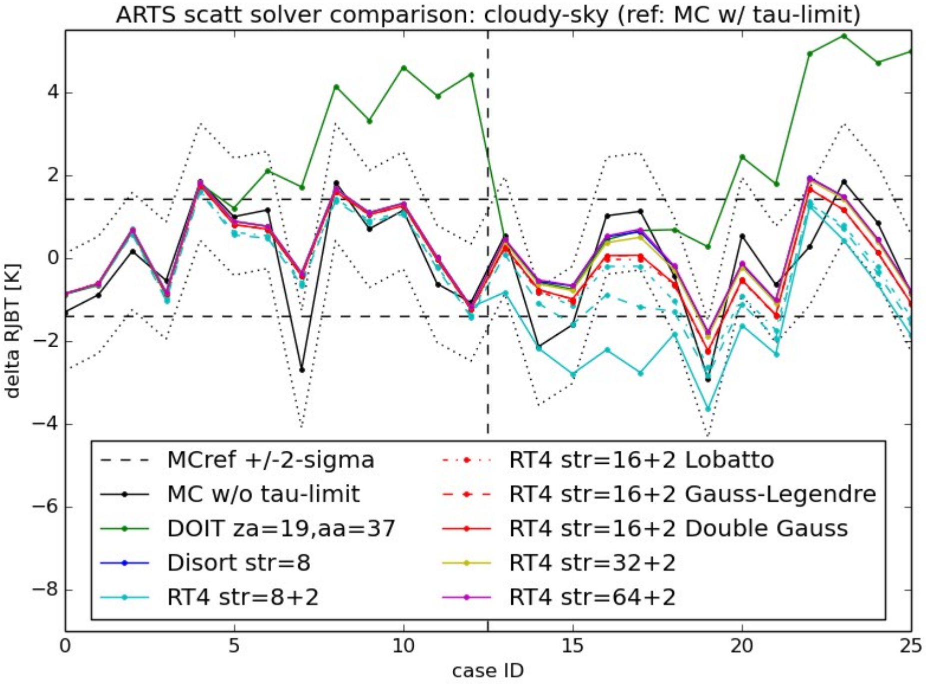
- Scattering solvers
  - for passive sensor simulations
- Handling of single scattering data
- Microphyiscs
  - aka deriving particle number (density) fields

Scattering solvers

	MC	DOIT	(FOS)	**(Hybrid)**	**Disort**	**RT4**
dimensionality	3D	1D (+3D)	3D	1D(?)	1D	1D
planet shape	sp+	sp+	sp+	sp	flat (pp)	flat (pp)
polarization	4	4	4	4*	1	2
orientation	any	any	any	any*	totally random	azimuth. random
speed	very slow	slow	medium - fast	medium*	fast	medium
output	**rad**	**field**	**rad**	**rad**	**field**	**field**
notes			scat in thin atmospheres	not stand-alone (background rad field) all-sky Jacobians	Lambertian; Stamnes	ARTS surf., auto-adaptive Nstreams; Evans

Scattering solvers

➢ Performance & Issues



- MC = reference

- DOIT:
  - setup sensitive (level spacing)
  - speed and accuracy(?) improved

- Disort (Legendre-decomp based):
  - very stable (energy conservation ensured)

- RT4 (Fourier-decomp based):
  - diff between solvers not very big
  - setup sensitive for strongly peaked Z (Nstreams & Naz => Z norm)

- Hybrid:
  - not stand-alone (primary focus: Jacobians)
  - accuracy & performance largly determined by background radfield “generator”

Scattering solvers

- Performance & Issues:
  - 1-thread only for FORTRAN-interfaced Disort & RT4
  - ARTS-side data management/prep needs time, too (my tests: 1:2)

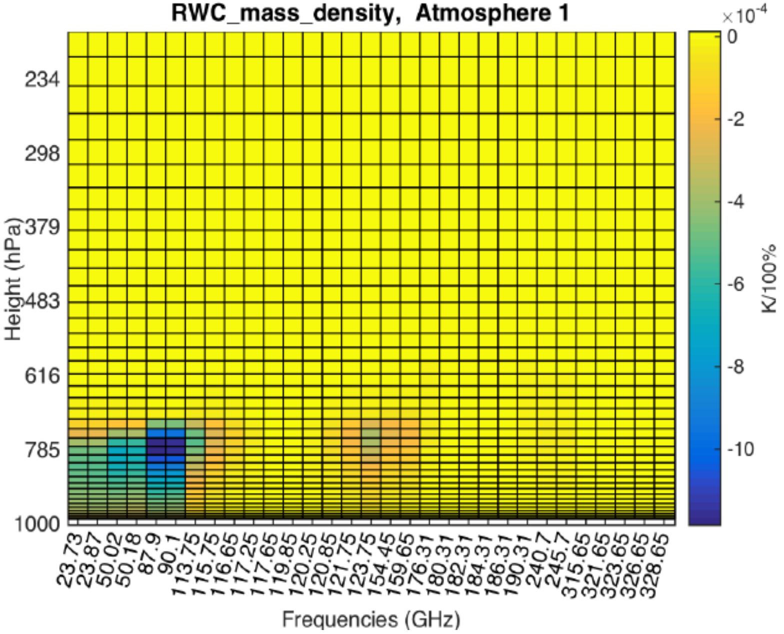
Scattering solvers

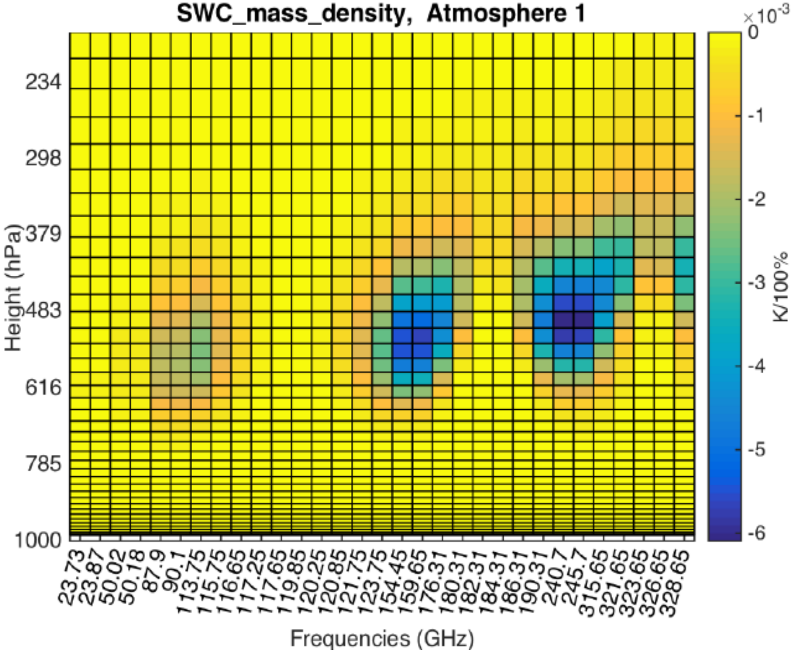
Notes to users (incl experts):

- For interfaced external solvers (Disort, RT4) cite the original sources, too.

- Read the documentation, please.
  - primarily the **built-in doc** (use the doc browser)
- Be a little careful when changing (degrading) default settings.
  - e.g. for speed improvements or to force non-failure
- Feel free to consult the users mailinglist.

- More new stuff: WSM `JacobianDOIT`
  - full-field perturbations using coordinated optimized DOIT setup





Scattering solvers

- “common” interface:
  - `scat_data`[SE](f,T) with Z, K, a
  - `pnd_field`(SE,p,lat,lon)
  - $K_{bulk} = \sum^{se} K_i * pnd_i$
- scattering element (SE)?
  - an individual particles (series representing, e.g., a size distribution; pnd=actual number of particles)
  - bulk representative (pnd=some weighting/rescaling factor)

Single scattering data

- each SE has its own f- & T-grids
- so far: Z/K/a for a specific **f**&T extracted (read: interpolated) just-in-time for each SE
  - ⇨ computationally expensive

- WSM `scat_dataCalc`:
  - "reduce" f-grid of all SE to RTE f-grid (or 1-f data; Z/K/a independently)
  - future standard (revising all solvers now; non-reduced: scat_data_raw)

Single scattering data

- ➢ each SE has its own f- & T-grids
- ➢ so far: Z/K/a for a specific f&**T** extracted (read: interpolated) just-in-time for each SE
  - ⇨ computationally expensive

- ➢ WSM `scat_dataReduceT` (? - to come...):
  - reduce T-dimension to 1 (on Z primarily; consistency?)

Microphysics

How to derive the `pnd_field`?

1) external:

- some functionality in atmlab (&typhon?)
- the users own methods

Microphysics

How to derive the `pnd_field`?

1) external

2) WSM `pnd_fieldCalcFromscat_speciesFields`

3) WSM `pnd_fieldCalcFromParticleBulkProps`

Microphysics

How to derive the `pnd_field`?

2) WSM `pnd_fieldCalcFromscat_speciesFields`

- ...

Microphysics

How to derive the `pnd_field`?

- ➢ scattering species (SS)?
  - a (scattering) entity characterised by one (or several) atmospheric fields
    - ➢ e.g. one hydrometeor species
  - consisting of a number of SE

Microphysics

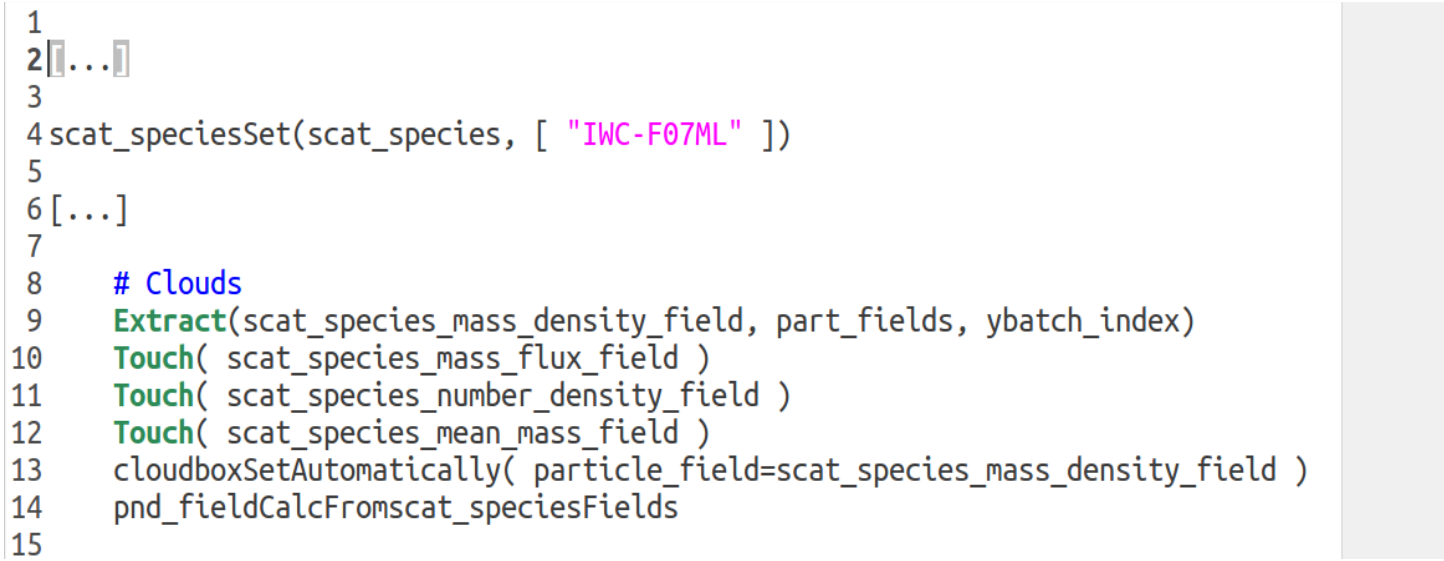
How to derive the `pnd_field`?

2) WSM `pnd_fieldCalcFromscat_speciesFields`

- series of atm fields to characterise the species (mass density, mass flux, number density, mean mass with one entry per SS)
- microphysical model specified by one tag per SS (`scat_speciesSet`)

How to derive the `pnd_field`?

2) WSM `pnd_fieldCalcFromscat_speciesFields`



```

[...]

scat_speciesSet(scat_species, [ "IWC-F07ML" ])

[...]

    # Clouds
    Extract(scat_species_mass_density_field, part_fields, ybatch_index)
    Touch( scat_species_mass_flux_field )
    Touch( scat_species_number_density_field )
    Touch( scat_species_mean_mass_field )
    cloudboxSetAutomatically( particle_field=scat_species_mass_density_field )
    pnd_fieldCalcFromscat_speciesFields

```

Microphysics

How to derive the `pnd_field`?

2) WSM `pnd_fieldCalcFromscat_speciesFields`

- series of atm fields to characterise the species (mass density, mass flux, number density, mean mass with one entry per SS)
- microphysical model specified by one tag per SS (`scat_speciesSet`)

- tested & applied
- fairly easy setup (a bit tedious for non-compact atms)
- but somewhat black-boxy and inflexible...

Microphysics

How to derive the `pnd_field`?

3) WSM `pnd_fieldCalcFromParticleBulkProps`

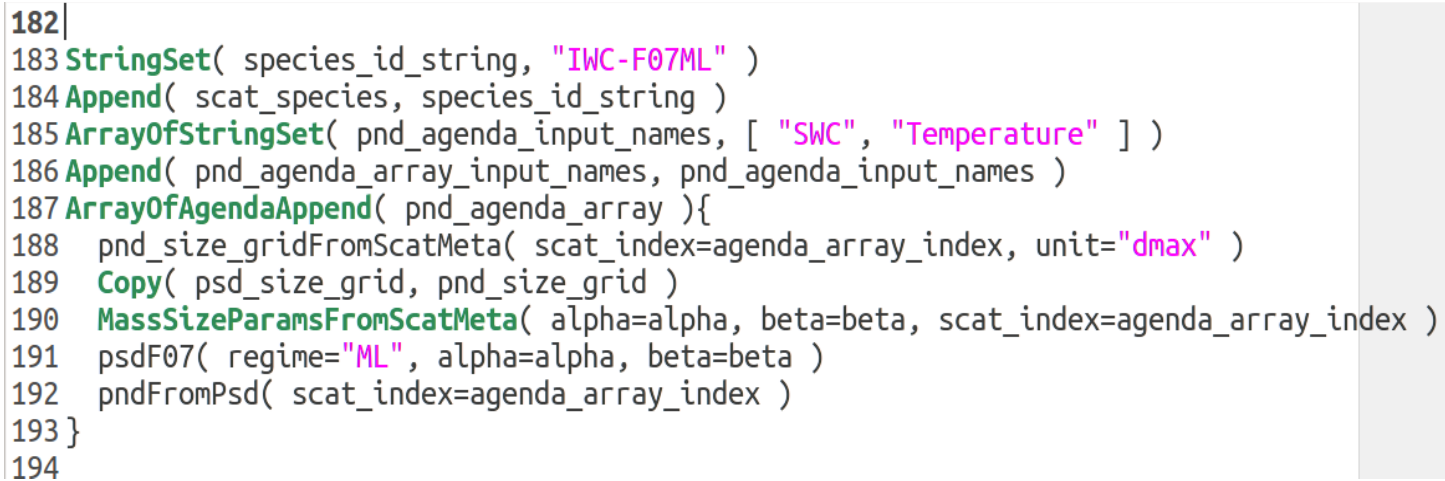
- all SS related atm fields in one WSV: `particle_bulkprop_field`
- a corresponding WSV of (semi-free) field names:

  `particle_bulkprop_names`
- one agenda per SS: `pnd_agenda_array`
- a corresponding WSV linking atm fields to each agenda/SS:

  `pnd_agenda_array_input_names`

How to derive the `pnd_field`?

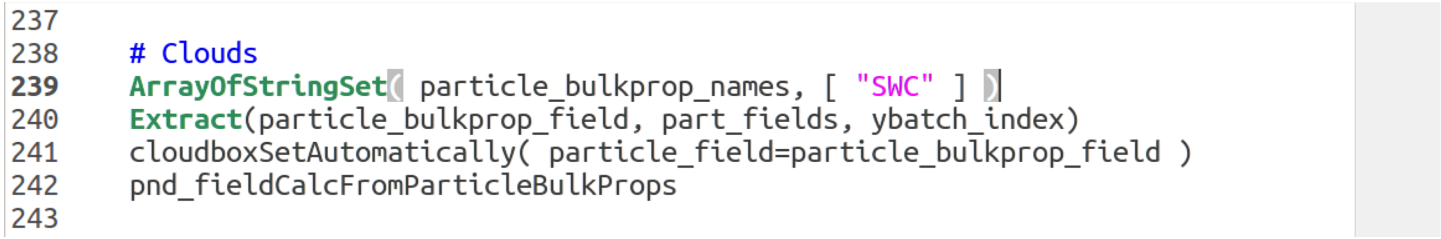
3) WSM `pnd_fieldCalcFromParticleBulkProps`



```

StringSet( species_id_string, "IWC-F07ML" )
Append( scat_species, species_id_string )
ArrayOfStringSet( pnd_agenda_input_names, [ "SWC", "Temperature" ] )
Append( pnd_agenda_array_input_names, pnd_agenda_input_names )
ArrayOfAgendaAppend( pnd_agenda_array ){
  pnd_size_gridFromScatMeta( scat_index=agenda_array_index, unit="dmax" )
  Copy( psd_size_grid, pnd_size_grid )
  MassSizeParamsFromScatMeta( alpha=alpha, beta=beta, scat_index=agenda_array_index )
  psdF07( regime="ML", alpha=alpha, beta=beta )
  pndFromPsd( scat_index=agenda_array_index )
}

```



```

    # Clouds
    ArrayOfStringSet( particle_bulkprop_names, [ "SWC" ] )
    Extract(particle_bulkprop_field, part_fields, ybatch_index)
    cloudboxSetAutomatically( particle_field=particle_bulkprop_field )
    pnd_fieldCalcFromParticleBulkProps

```

Microphysics

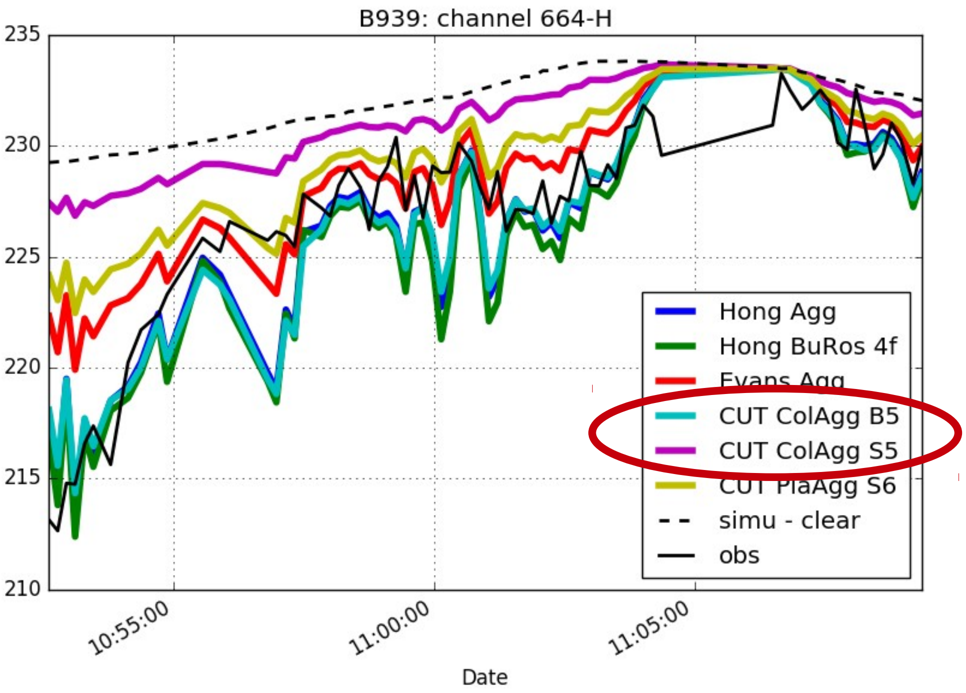
How to derive the `pnd_field`?

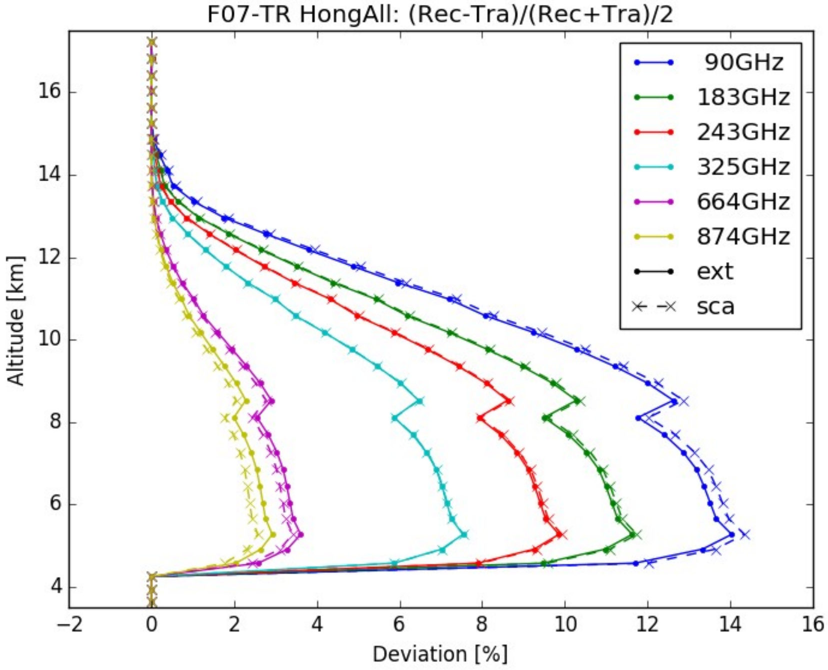
3) WSM `pnd_fieldCalcFromParticleBulkProps`

- more flexible reg. input atm fields parameters (any thinkable field possible, e.g. mean size, asphericity, ...)
- more control over the individual calculation steps and applied parametrisations (e.g., specify valid T-range for certain SS, rescale pnd to intended mass?, ...)
- **providing Jacobians** (`dpnd_field_dx`)

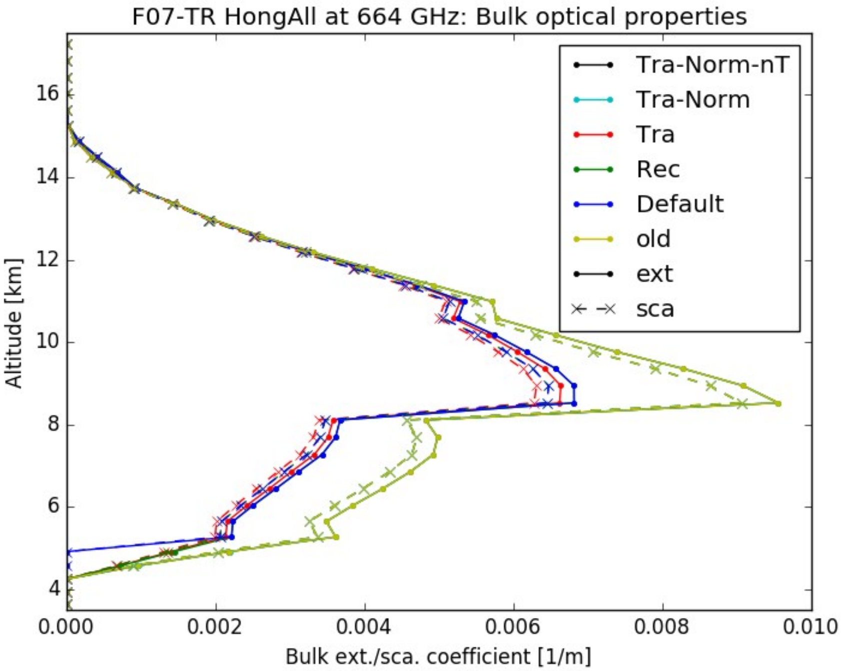
CHALMERS
UNIVERSITY OF TECHNOLOGY

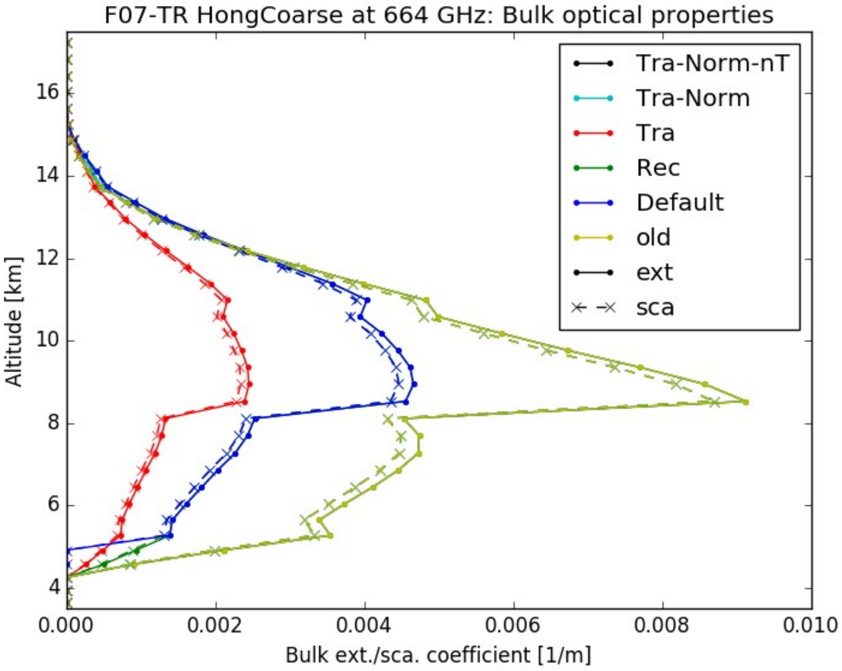
Example: Scaling issues





Example: Scaling issues





CHALMERS
UNIVERSITY OF TECHNOLOGY