**Important annotations for running SNOWPACK**

**General annotations**

* SNOWPACK is written in C++. The most important sources files are located in:
  + *snowpack/trunk/applications/snowpack/* 🡪 main file (*Main.cc*)
  + *snowpack/trunk/snowpack/* 🡪 e.g. physical parameterisations and data classes
  + *snowpack/trunk/snowpack/snowpackCore/* 🡪 e.g. phase changes and water transport
* Framework around SNOWPACK consist of Python and Shell scripts
* Python scripts can be executed locally and on ECMWF (🡪 relevant paths have to be defined at the beginning of the scripts and are then automatically set depending on the environment)
* Python scripts *SP\_in\_meteo\_RACMO.py* and *SP\_in\_meteo\_RACMO\_trans.py* process all atmospheric parameters needed to force SNOWPACK with either mass balance (*MASSBAL*) or energy balance components (*ATMOS*).
* Use SNOWPACK revision number r48
* Installation of SNOWPACK on ECMWF 🡪 replace files *CMakeLists.txt* and *FindMeteoIO.cmake* with modified versions
* Install MeteoIO (also available in SNOWPACK repository) prior to SNOWPACK. Selected path in *CMAKE\_INSTALL\_PREFIX* must be identical for MeteoIO and SNOWPACK.

**Applying SNOWPACK for Antarctica 🡪 open questions**

* Densification: test performance of densification scheme for various locations (particularly for locations with very cold surface climate) 🡪 adjust parameters *Q\_fac* and *criticalExp* in function *SnLaws::snowViscosityTemperatureTerm()*
* Fresh snow density: which parameterisation should be used? Switch on/off *enhanced\_wind\_slab* in *Snowpack.cc* (implemented for *ANTARCTICA* variant) depending on selected parametrisation
* Spin-up method: What’s the best method to derive the necessary number of spin-up iterations for a certain location? Use steady-state density profiles (analogue to IMAU-FDM)? Or derive number of iterations from amount of solid precipitation (🡪 *SP\_prepare\_ECMWF\_run.py*)?
* SNOWPACK forcing method (*MASSBAL* or *ATMOS*): Which method should be used?
* Minor issues:
  + Function *Snowpack::sn\_ElementKtMatrix()*: The effective thermal conductivity is computed differently for low- and high-density snow/firn. The transition (at a volumetric ice content of 0.55) is not smooth and step-like.
  + *DataClasses.cc* 🡪 function *ElementData::heatCapacity()* 🡪 heat capacity of ice is constant and not a function of temperature (in contrast to IMAU-FDM). This could potentially be problematic for very cold locations on Antarctica…