Prague, 30.08.2024

Report on

ADDING SNOW FROM TOWN TO TOTAL SNOW IN AN ATMOSPHERIC MODEL AND INITIALIZATION OF EXPLICIT SNOW SCHEME IN COUPLED ALARO-SURFEX

Author: Gabriel Stachura, IMGW
Supervisor: Ján Mašek, CHMI

1. Introduction

The report sums up work which has been done during a research stay in Prague in August 2024 (2 weeks), as well as the follow up work done locally. The stay was a part of the SU3.3 task in the RWP 2024.

Lately, more and more SURFEX components have become adopted to work correctly within ALARO framework. However, inconsistencies are still being discovered, particularly connected with the Town Energy Balance (TEB) model, and there are still several untested options there. Some time ago it was found out by Ján Mašek (and later confirmed by SURFEX team in Météo-France) that in a fully coupled run with an atmospheric model, total snow reservoir in an atmospheric file (SURFRESERV.NEIGE) is supplied only by snow from NATURE tile in SURFEX. To make it more realistic, we intended to include also snow from the TOWN tile, particularly from roads, roofs, and eventually gardens (at the moment we use option LGARDEN=F, which treats gardens within nature tile).

Apart of this issue with snow in town, we also wanted to continue the topic from the previous stay, which concerned the initialization of snow in the Explicit Snow (ES) scheme. Last year, we didn't manage to produce the surface initial file for this snow scheme by running FULLPOS-PREP configuration. This was because of a lack of tool for splitting integral snow cover characteristics into multiple layers in the model, which is necessary in case of a multi-layer snow scheme. This time, our plan was to create a dedicated external tool for splitting ISBA snow fields from an atmospheric file and to add them to the existing surface initial file. Only after the stay we learned from Patrick Samuelsson that this functionality already exists in PREP tool, which is able to produce ISBA-ES multi-later snow fields from ECMWF single-layer snow scheme, reading surface GRIB fields from TESSEL scheme as input. During 2024 autumn ACCORD Surface Working Week the issue was discussed, and it is recommended that snow redistribution should be accomplished by standard SURFEX tools like (FULLPOS-)PREP, rather than by external tools whose maintenance is likely to become problematic in longer term. Extension of FULLPOS-PREP to transform snow between D95/EBA and ISBA-ES snow schemes will be a topic of future stay.

Therefore, the main part of the report is split into two sections – the first one considering inclusion of snow from TEB into the total snow in an atmospheric file and the second one – regarding initialization of snow in ES.

2. Inclusion of snow from TEB

a. Snow in TEB – dataflow

In order to better understand a process of passing information from SURFEX to ALARO, a first thing we did was to figure out the dataflow under a routine aplpar.F90. An essential part of it is shown in Fig. 1. In the first part, after a routine ARO_GROUND_PARAM is called, surface physics is calculated, separately for every tile (routines COUPLING_{tile}_n). For TEB, this is a routine COUPLING_TEB_n. It is important to notice that in this routine a loop over patches occurs. Within this loop, a state of prognostic variables at every timestep is calculated.

```
APLPAR
(...)
 ARO_GROUND_PARAM
   COUPLING_SURF_ATM_N
     TREAT_SURF
       COUPLING_SEA_n
       COUPLING_INLAND_WATER_n
       COUPLING_NATURE_n
       COUPLING_TOWN_n
         COUPLING_TEB_n
           TEB_GARDEN                              ┐
             URBAN_SOLAR_ABS                       │
             URBAN_LW_COEF                         │
             TEB_GARDEN2                           │
               GARDEN                              │
             TEB_GARDEN3                           │
               TEB                                 │
                 URBAN_DRAG                        │
                 URBAN_SNOW_EVOL                   ├ loop over patches
                   SNOW_COVER_1LAYER (for roofs)   │
                   SNOW_COVER_1LAYER (for roads)   │
                 ROOF_LAYER_E_BUDGET               │
                 ROAD_LAYER_E_BUDGET               │
                 FACADE_E_BUDGET                   │
                 URBAN_FLUXES                      │
               AVG_URBAN_FLUXES                    │
           DIAG_MISC_TEB_n                         ┘
           AVERAGE_RAD
           DIAG_INLINE_TEB_n
     AVERAGE_FLUX
     AVERAGE_RAD
     AVERAGE_PHY
     DIAG_INLINE_SURF_ATM_n
 ARO_GROUND_DIAG(...,ZTWSNOW,...)
   DIAG_SURF_ATM_N(...,YSURFEX_ARO_CUR%TM%DGT,...)
   GET_SURF_VAR_N(...,YSURFEX_ARO_CUR%TM%DGMT, PTWSNOW,...)
     GET_VAR_SEA_n
     GET_VAR_NATURE_n
     GET_VEG_n
     GET_VAR_TOWN_n(...,PTWSNOW...,)
   GET_FLUX_N
(...)
```

Fig. 1 Dataflow of snow in TEB. Only selected subroutines are listed in the diagram.

Snow in TEB can occur on roads and building roofs and - if garden option is activated - in town gardens. Snow in town gardens is calculated similarly to how it is done for nature – by running ISBA. This is done in a subroutine GARDEN. Snow on roads and roofs is calculated subsequently (subroutine URBAN_SNOW_EVOL). So far, only a simple 1-layer snow scheme is used for it. Calculation is done separately for roads and for roofs. Results are saved to a SURFEX data structure (e.g., snow water equivalent on roofs is under YSC%TM%CUR%TSNOW_ROOF%WSNOW). After that, fluxes and miscellaneous diagnostics for each patch are calculated. Finally, fluxes are tile-averaged (AVERAGE_FLUX) and subroutine ARO_GROUND_DIAG is called, which main task is to fetch surface diagnostics for an atmospheric model. This routine starts with an empty array PTWSNOW, which is a total snow water reservoir needed by an atmospheric model (SURFRESERV.NEIGE). It is then passed to the subroutine GET_SURF_VAR_n, which calls among others a subroutine GET_VAR_NATURE_n. There, the empty array is filled with snow water reservoir from nature tile. When calling a subroutine GET_VAR_TOWN_n, no snow is imported from there. This is then passed to the atmospheric file and saved under the name SURFRESERV.NEIGE.

b. Description of code modifications

Our modification involves adding to snow coming from nature tile also snow from town, which is a sum of snow from roads and roofs weighted with their fractions:

$$TWSNOW_{town} = TWSNOW_{roof} \cdot f_{roof} + TWSNOW_{road} \cdot f_{road}$$

Prior to this calculation, $TWSNOW_{roof}$ and $TWSNOW_{road}$ are calculated by vertical summation of SWE of each snow layer, although at present it is not necessary since there is only 1 snow layer allowed in TEB. The calculation is done in a subroutine DIAG_MISC_TEB_n and the field is added to miscellaneous diagnostics in TEB, analogously as it is in ISBA. The whole modification may be disactivated by setting LSURF_MISC_BUDGET=F in a namelist NAM_DIAG_TEBn. In total, 15 subroutines and modules have been modified.

c. Experiment setup

A 72-h forecast was produced, starting from 12th January 2021 00 UTC. This date was deliberately selected to capture significant snowfall that occurred in most of the CHMI domain. The experiment forecast (EXP) is a forecast that includes snow from TEB, while the reference (REF) – only from ISBA (nature tile). Land cover characteristics were derived from ECOCLIMAP 2.6.

d. Results

Our initial results were flawed by an erroneous accumulation of snow in places with little town fraction (mostly lower than 5%). In Fig. 2a we can see that these values occupy large areas e.g. in Poland, Germany, Pannonian Basin and Romanian Plain. Therefore, the PGD file was modified by transforming town fraction lower than 5% to nature tile. It is important to be noticed than the 5% threshold has been set subjectively by visually assessing spatial distribution of urban areas. Fig. 2b shows that the threshold seems to be a reasonable compromise between reducing excessive extent of small town fraction and misrepresenting land cover.

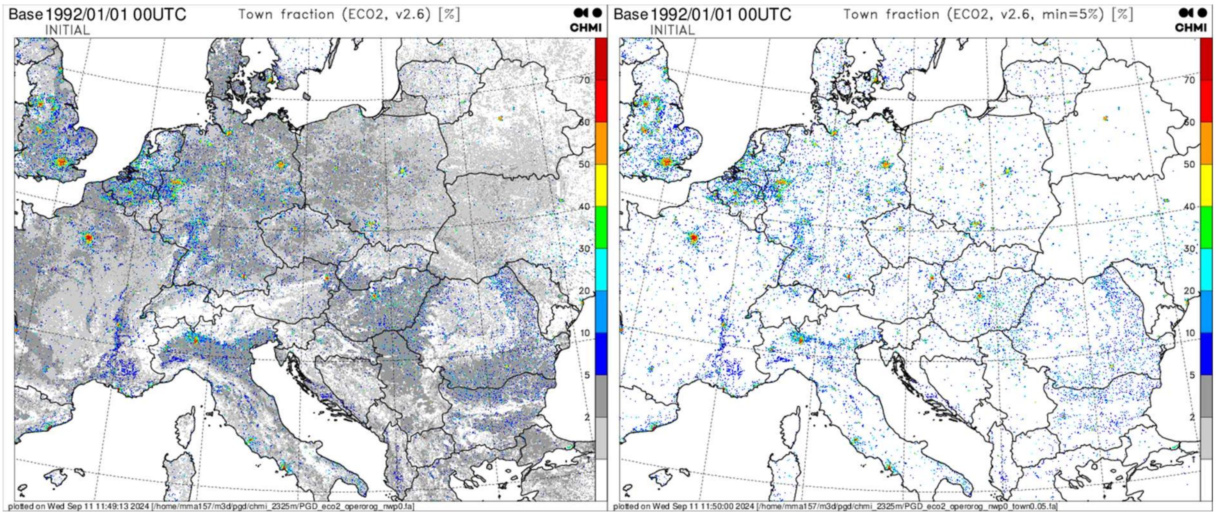

Fig. 2 Left: Initial fraction of town from ECOCLIMAP 2.6. Right:Fraction of town from ECOCLIMAP 2.6 with values lower than 5% filtered out. .

Using the modified PGD file, a forecast was calculated. In Fig.3 snow water equivalent from an atmospheric file (SURFRESERV.NEIGE) at the end of the forecast range (+72h) is shown. We can see especially on a difference plot that the amount of snow increased in towns, while it remained the same in the rest of the area, which is a desired effect. Fig. 4 depicts spatial distribution of SWE on roofs. Both maps are exactly the same – this is because snow in urban areas exists both in reference and the experimental run – the only difference is that in the experiment it is included in the total amount of snow passed to the atmospheric file. Notice that spatial distribution of snow on roofs is very similar to the difference plot of SURFRESERV.NEIGE (Fig. 3c).

Nevertheless, a comparison of total snow water equivalent in SURFEX and atmospheric files raises suspicion that arrays WSN_RF1 and WSN_RD1 carry SWE per town area, not per gridbox area as we supposed. This will be subject to re-checking and if confirmed, their multiplication by fraction of town will have to be applied before adding them to field SURFRESERV.NEIGE in the atmospheric file.

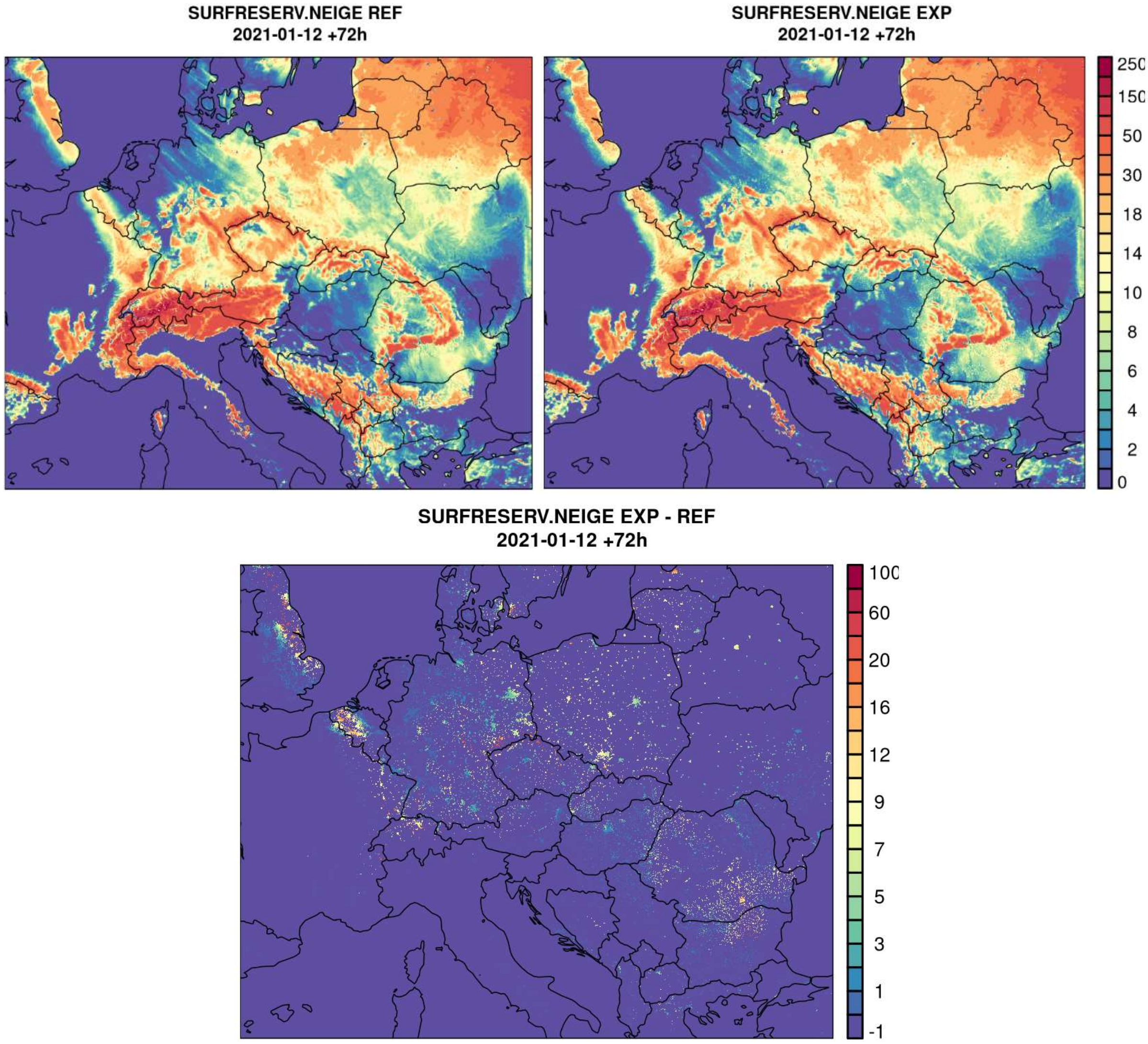

Fig. 3 Top: spatial distribution of snow water equivalent [kg/m^2] in a reference and experimental run. Bottom: difference between them.

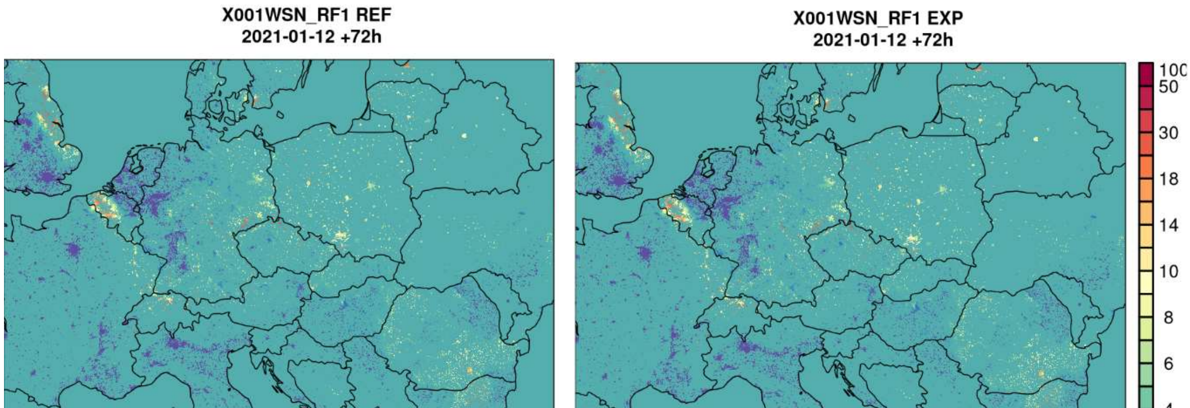

Fig. 4 Spatial distribution of snow water equivalent on roofs [kg/m^2] in a reference and experimental run.

3. Initialization of snow in ES

a. Description of a tool

In the second part of the stay, an effort was made to create an external tool to process snow fields from initial conditions coming from a global ARPEGE model so that initialization of a multi-layer snow scheme could be possible. We focused in particular on Explicit Snow (ES) scheme.

Based on an already existing tool for manipulating FA records, we created a program which reads snow fields from an input atmospheric file, processes them based on options set in a namelist (which is a second command line argument) and adds them to the output file, which is a SURFEX initial file. In the namelist, several parameters are declared:

- NLAYERS – desired number of snow layers (integer)
- LHOMODZ – vertically uniform layer thickness (logical)
- LHOMORSN – vertically uniform density (logical)
- XSAGVAL – uniform value of snow age [days] (real)

After reading the namelist setting, thickness of snow layers is computed at first. This is done based on two parameters: NLAYERS and LHOMODZ. If vertically uniform distribution of snow layers is asked, the thickness of each layers is simply a total snow depth divided by the number of layers. If LHOMODZ=F, layering from ES is applied (from a module MODE_SNOW3L). Once layering is completed, density of each layer is calculated. If vertically uniform density is asked, density of each layer is simply the snow density from an atmospheric file. Otherwise, it is kept uniform only for total snow depth lower than 20 cm. For greater snow depth, density of the top layer is assumed to be 100 kg/m3 and for every n-th layer below (n=2,…,NLAYERS) it increases with depth according to a modified formula proposed by Sexstone and Fassnacht (2014):

$$\rho_{sn,n}\left[\frac{kg}{m^3}\right] = 54.96h_{sn,n} + 145[m]$$

where h is assumed to be a middle point of a snow layer. The top layer of snow is assumed to have 100 kg/m^3 density. After calculation of density of the last layer, density for bottom layers are adjusted to ensure mass conservation. If an adjusted layer density violates the thresholds of minimum or maximum density for ES (which are here the same as in the model), it is set to this threshold value and upper layer is also adjusted. The algorithm runs iteratively up to the top layer if needed.

Once densities are obtained, snow water equivalent of each layer is calculated by simply multiplying snow thickness and snow density. After that, snow age for every layer is assigned to a value given in the namelist under XSAGVAL. Finally, snow heat content (Hsn) is determined according to a formula 3.1 but with two simplifications:

- snow temperature in every layer equals a minimum between 0°C and surface temperature from an atmospheric model
- snow layer liquid water content is equal to zero

$$Hsn_i\left[\frac{J}{m^2}\right] = c_{si}D_i(Tsn_i - T_{tt}) - L_f\rho_w(w|si - w_{sli})$$

(3.1)

where: i is a layer number, $c_{si} = k \cdot \rho_{sni}$ is a snow heat capacity $\left[\frac{J}{K \cdot m^3}\right]$, D_i – snow layer thickness [m], Tsn_i – snow layer temperature [K], T_{tt} – triple point temperature [K], L_f – latent heat of fusion $[\frac{J}{kg}]$, ρ_w – water density $[\frac{kg}{m^3}]$, w_{si} – snow layer total water equivalent depth [m], w_{sli} – snow layer liquid water content [m].

Snow albedo is passed to the output file without any modifications.

After all snow fields are created, they are written to an existing output file and named appropriately: X001{parameter abbreviation}_VEG{layer number}. In order to be internally consistent, output file must be created already for ISBA-ES scheme, initialized with idealized snow profile (no snow):

```
&NAM_PREP_ISBA_SNOW
  CSNOW='3-L',
  NSNOW_LAYER=6,  ! desired number of snow layers
  LSNOW_IDEAL=.T.,
  LSNOW_PREP_PERM=.F.,
  XWSNOW(1:6)=6*0.,  ! initial value of SWE in layers
/
```

This is because the initial .sfx file contains records specifying the setup of snow scheme that will be used in SURFEX integration.

b. Experiment setup and results

Initially it was intended to run experiments with 6 snow layers and different vertical distribution of snow density. The reference run was a forecast based on a surface initial file with 6 snow layers and other snow fields set to 0. However, the experiment run crashed due to a bug in a calculation of heat snow content in our tool, from which surface temperature is later retrieved. Additionally, as we were later recommended by ACCORD Surface Team to work on this subject within SURFEX environment, the external tool will not be developed.

4. Conclusions and further work

Although the introduced modification does not affect model physics, it makes the amount of snow being written to the output field more realistic. Further tests are also necessary to check weather the modification is correct also for more than 1 patch in nature as well as for TEB with garden option on. Although for great majority of the domain area snow from nature and town tiles comprises 100% of total snow reservoir, on a local scale adding snow from other tiles may also be meaningful (e.g., snow on an ice-covered lake in FLAKE). Moreover, correct handling of fractions has to be rechecked, so that total SWE stored in an atmospheric file is relative to the gridbox area.

As far as initialization of snow in ES is considered, a tool to transfer snow from an atmospheric file with a 1-layer snow scheme to a surface file with a multiple snow scheme has been drafted. Unfortunately, we didn't manage to master the external tool so that the produced file could be used in a forecast run. We suspect there is at least one bug in computation of snow heat content. As it was agreed to move the functionality from external tool to FULLPOS-PREP, the future work will be reoriented accordingly.

5. References

Sexstone G.A., Fassnacht S.R., 2014, *What drives basin scale spatial variability of snowpack properties in northern Colorado*?, in: *The Cryosphere* 8(2), DOI: 10.5194/tc-8-329-2014