Surface assimilation using extended Kalman filter

Report from stay at OMSZ Budapest
05–16/10/2015, 02–13/11/2015

Viktor Tarjáni

Slovak Hydrometeorological Institute, Bratislava

Supervisors: Máté Mile and Helga Tóth

Hungarian Meteorological Service, Budapest

May 14, 2016

1 Introduction

It's generally agreed that deviations of soil moisture content WG and soil temperature TG in NWP models from their actual values can lead to significant forecast errors of screen-level atmospheric variables e.g., 2 m air temperature T2M and 2m relative humidity HU2M [1, 2]. For accurate screen-level forecasts it is therefore very important to ensure accurate estimation of the soil variables in the NWP model. Several methods have been developed to minimize errors in soil variables. In this study we tested one proposed by Mahfouf [2] which is based on Extended Kalman filter (EKF) making use of the offline SURFEX platform [3]. It allows assimilation of both conventional (screen-level) and non-conventional (satellite) observations to produce surface analysis. In this study however only screen-level measurements of T2M and HU2M were used.

Basic idea behind assimilation of T2M and HU2M measurements in the surface analysis is that close physical relation exist between screen-level atmospheric variables T2M, HU2M and soil variables WG, TG. Primarily it is turbulent transport of heat, momentum and humidity which mediates this relation. It can be utilized to correct soil variables WG and TG in the model with the computed analysis increments ΔWG and ΔTG. If measurements of T2M and HU2M are available with enough high spatial density they can substitute direct measurements of WG and TG which are usually not available or are available sparsely.

2 Method

AROME-HU is a main operational NWP LAM model at OMSZ. Its grid has 490 x 310 points with horizontal resolution 2.5 x 2.5 km and 60 vertical levels. Currently it uses downscaled ALARO 8km surface analysis and OI_MAIN is also experimentally used. It was decided to make preliminary test study to check feasibility of the EKF method to assimilate both conventional and non-conventional (satellite) measurements to produce accurate surface analysis. Surface parametrization in AROME model is implicitly treated with SURFEX. This is the so called inline (or coupled) SURFEX as it is integrated parallel with integration of atmospheric model. EKF assimilation used in this study is based on offline (or de-coupled) SURFEX. It is integrated independently from AROME model but it requires to be forced by atmospheric forcing. Atmospheric forcing consist of time series of forcing variables (there are 11) covering the whole time-interval of SURFEX run which was in our case 6 hour long assimilation window. Forcing files were generated with AROME in-line full-pos to 17 m level height above surface (`RFP3H`=17 in `&NAMFPC` namelist). This approximately corresponds to the first model level of ALADIN. In principle surface assimilation is possible also with inline SURFEX however using the offline SURFEX variant is much more time-efficient while the results are comparable [4], thats why offline SURFEX variant is used preferably.

The soil physics inside the SURFEX is treated by ISBA model (Interaction Soil–Biosphere–Atmosphere). It is responsible for soil and vegetation evolution within the nature tiles of SURFEX, and their interaction with lowest level of atmospheric model through momentum, heat, and water fluxes. Other three SURFEX tiles – town, inland water and sea/ocean – are treated with other numerical schemes e.g., TEB, WATFLX, SEAFLX. Within our present study ISBA is the most important part of the SURFEX because EKF method [2] performs surface analysis only for the nature tiles. In original EKF setup 2 layer ISBA scheme is considered. In our study 3-layer version was used. Four main prognostic variables of ISBA model were used in EKF analysis: TG1 – surface temperature, TG2 – deep temperature, WG1 – superficial soil moisture (approx. 1cm surface layer) and WG2 – root zone soil moisture. Detailed description of the ISBA model within SURFEX can be found in [6].

The EKF assimilation is driven by the bash control scripts. Original EKF source package contains its own control bash script `'run_ekf.sh'`. Control scripts and namelist used at OMSZ were originally written at Belgian RMI for ALADIN. Here two other scripts were added which are responsible mostly for preparation of atmospheric forcing and initial state for offline SURFEX run. At OMSZ they were adapted for AROME model [5], originally on top of cy36t1 and during the stay they were modified also for cy38t1. Belgian colleagues also provided us a FORTRAN program which prepares ASCII forcing files from AROME full-pos files.

3 Extended Kalman filter (EKF) setup

EKF is recursive state estimation method. In this study it is used to estimate main state variables of ISBA soil model (e.g. soil analysis). Each cycle of EKF consists of *time-update* part followed by *measurement-update* part.

In time-update, state vector estimate (analysis) and analysis error covariance matrix from previous time step, $\mathbf{x}_a^{k-1}$ and $\mathbf{A}^{k-1}$ respectively, are pushed forward in time by single time step giving the prediction on them (guess, forecast) at actual time step $\mathbf{x}_b^k$ and $\mathbf{B}^k$ (often referred to as *background state vector* and *background error covariance matrix*):

$$\mathbf{x}_b^k = \mathcal{M}(\mathbf{x}_a^{k-1}) \tag{1}$$

$$\mathbf{B}^k = \mathbf{M}\mathbf{A}^{k-1}\mathbf{M}^T + \mathbf{Q} \tag{2}$$

$\mathcal{M}(\mathbf{x}; k-1, k)$ is state transition operator which maps state vector from time step $k-1$ to k. In our case it is ISBA model. In covariance matrix time-update nonlinear state transition operator $\mathcal{M}$ is replaced with its Jacobian matrix $\mathbf{M} \equiv \mathbf{M}(k-1, k)$ calculated around $\mathbf{x} = \mathbf{x}_a^{k-1}$ (also called linear tangential operator). Matrix $\mathbf{Q}$ is covariance matrix of model error.

In measurement update $\mathbf{x}_b^k$ and $\mathbf{B}^k$ are further updated using the actual observation vector $\mathbf{y}_k$ giving the best estimate of state vector $\mathbf{x}_a^k$ (new analysis) and its covariance matrix $\mathbf{A}^k$:

$$\mathbf{x}_a^k = \mathbf{x}_b^k + \mathbf{K}^k \left[\mathbf{y}^k - \mathcal{H}(\mathbf{x}_b^k)\right] \tag{3}$$

$$\mathbf{A}^k = (\mathbf{I} - \mathbf{K}^k\mathbf{H})\mathbf{B}^k \tag{4}$$

where

$$\mathbf{K}^k = \mathbf{B}^k\mathbf{H}^T(\mathbf{H}\mathbf{B}^k\mathbf{H}^T + \mathbf{R})^{-1} \tag{5}$$

is Kalman gain matrix, $\mathcal{H}(\mathbf{x})$ is observation operator which maps from state space to observation space, $\mathbf{H}$ is Jacobian matrix of nonlinear observation operator $\mathcal{H}$ calculated around $\mathbf{x}_b^k$ and $\mathbf{R}$ is covariance matrix of measurement error.

After measurement-update $\mathbf{x}_a^k$ and $\mathbf{A}^k$ can be substituted back to equations (1) resp. (2) and procedure can be repeated giving the $\mathbf{x}_a^{k+1}$ and $\mathbf{A}^{k+1}$. In this way EKF can be cycled arbitrary number of times which is why EKF is termed as recursive state estimator. Starting EKF requires providing it with initial state estimate $\mathbf{x}_b^0$ and initial covariance matrix of background error $\mathbf{B}^0$.

Actual EKF program for soil assimilation is written in FORTRAN and can be configured with three namelists: `&NAM_IO_VARASSIM` for switching between different EKF execution modes, `&NAM_VAR` for setting control vector related variables and `&NAM_OBS` for setting observation related variables. Actual values of all EKF related namelists used at OMSZ are shown in appendix A together with short description of each variable. Most of them are default values. In the following actual configuration of EKF used at OMSZ for running test cases is described.

3.1 Control and observation vectors

In an actual EKF design, not all ISBA prognostic variables are included in the state vector so that dimension of the state space is reduced. In this case it is better to speak about *control space, control vector and control variables* [7, p.8]. In actual EKF setup four main ISBA prognostic variables were included in the control vector:

$$\mathbf{x} = (WG2, WG1, TG2, TG1)^T \tag{6}$$

Number of control variables used in analysis can be set with `NVAR` in `&NAM_VAR` namelist and i-th control variable is included/excluded by setting `INCV(i)=1/0` in the same namelist, see A.

Three observation types are allowed in current EKF assimilation: 2m air temperature T2M and 2m relative humidity HU2M station observations, and satellite measurements of superficial soil moisture WG1. The later one wasn't used in actual setup resulting in two component observation vector:

$$\mathbf{y} = (T2M, HU2M)^T \tag{7}$$

Both observation types are first interpolated with CANARI from stations onto numerical grid - resulting in *gridded observations* which enter to assimilation. Number of used observation types is set with `NOBSTYPE` in `&NAM_OBS` namelist and j-th observation type is included/excluded in the analysis by setting `INCO(j)=1/0` in the same namelist, see A.

3.2 Error covariance matrices

Covariance matrix of observation error R is time-invariant and diagonal (e.g. errors of individual observation types are assumed to be uncorrelated at each grid point). It is same for each grid point and has a following form:

$$\mathbf{R} = \begin{bmatrix} \sigma_{o_1}^2 & 0 \\ 0 & \sigma_{o_2}^2 \end{bmatrix} \tag{8}$$

where $\sigma_{o_1}^2 \equiv \mathrm{var}(\varepsilon_{o_1})$ is variance of T2M observation error and $\sigma_{o_2}^2 \equiv \mathrm{var}(\varepsilon_{o_2})$ is variance of HU2M observation error. Standard deviation of j-th observation-type error σ_{o_j} is set with `YERROBS(j)` in `&NAM_OBS` namelist, see A.

Initial covariance matrix of background error $\mathbf{B}_0$ is:

$$\mathbf{B}_0 = \begin{bmatrix} \sigma_{b_1}^2 & 0 & 0 & 0 \\ 0 & \sigma_{b_2}^2 & 0 & 0 \\ 0 & 0 & \sigma_{b_3}^2 & 0 \\ 0 & 0 & 0 & \sigma_{b_4}^2 \end{bmatrix} \tag{9}$$

where diagonal elements are background error variances corresponding to each control variable. Standard deviation of background error for i-th control variable σ_{b_i} is set with `XSIGMA_M(i)` in `&NAM_VAR` namelist (see A).

Covariance matrix of model error Q is diagonal and constant in time and space. It is given as a q_s^2 fraction of the initial covariance matrix of background error $\mathbf{B}_0$:

$$\mathbf{Q} = q_s^2 \mathbf{B}_0 \tag{10}$$

Value of q_s is set with `SCALE_Q` in `&NAM_VAR` namelist (see A).

3.3 Calculation of Jacobian matrices M and H

Time-update of background error covariance matrix (2) requires calculating Jacobian matrix $\mathbf{M}$ of model operator $\mathcal{M}(\mathbf{x})$ in the vicinity of the control vector at previous time step $\mathbf{x}^{k-1}$:

$$\mathbf{M} = \left. \frac{\partial \mathcal{M}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}^{k-1}} = \frac{\partial \mathbf{x}^k}{\partial \mathbf{x}^{k-1}} \tag{11}$$

In addition to initial reference ISBA surface state it also requires additional 4 perturbed initial surface states to be generated by perturbing initial reference state each time for different ISBA prognostic variable (EKF control variable) according to:

$$x_i' = x_i + \delta x_i \tag{12}$$

where $\delta x_i = \varepsilon_i x_i$ is initial perturbation of i-th control variable, ε_i is dimensionless perturbation amplitude of i-th control variable. It is set by `TPRT_M(i)` in `&NAM_VAR` namelist. Default values were used: `TPRT_M(TG1)` = `TPRT_M(TG2)` = 10^{-5}, `TPRT_M(WG1)` = `TPRT_M(WG2)` = 10^{-4}.

Initially (time step $k-1$) thus we had 5 different '`PREP.lfi`' files, one representing reference SURFEX state and remaining four perturbed SURFEX states. From each of them one offline

SURFEX run is started and at the end (time step k) there are 5 different new SURFEX states each saved to separate 'PREP.lfi' file. Elements of matrix $\mathbf{M}$ are calculated as a part of $\mathbf{B}$ matrix time-update (see 2. in section 4.1):

$$M_{ij} = \frac{\delta x_i^{(k)}}{\delta x_j^{(k-1)}} = \frac{x_i'^{(k)} - x_i^{(k)}}{\delta x_j^{(k-1)}} \tag{13}$$

where $x_i'^{(k)}$ is i-th control variable after j-th perturbed offline SURFEX run, $x_i^{(k)}$ is i-th control variable after reference offline SURFEX run, and $\delta x_j^{(k-1)}$ is initial perturbation of j-th control variable $x_j^{(k-1)}$. Each element of $\mathbf{M}$ matrix is stored in separate ASCII file 'LTM_del[XVAR(i)]_del[XVAR(j)]' where XVAR(i) resp. XVAR(j) is name of i-th resp. j-th control variable (set in the &NAM_VAR namelist), $x_i'^{(k)}$ is read from 'MDSIMU_PERT_[i]' ASCII file, $x_i^{(k)}$ is read from 'MDSIMU_REFR_' ASCII file, and $x_j^{(k-1)}$ is read from initial reference file 'PREP.lfi'.

Jacobian matrix $\mathbf{H}$ is required in measurement-update part of EKF. It enters both calculation of new analyzed state $\mathbf{x}_a^k$ through Kalman gain matrix $\mathbf{K}^k$ (equation (3)) and also analysis error covariance matrix $\mathbf{A}^k$ (equation (4)). Jacobian matrix $\mathbf{H}$ has 8 elements in 2 rows and 4 columns:

$$\mathbf{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \end{bmatrix} = \begin{bmatrix} \frac{\partial T2M}{\partial WG2} & \frac{\partial T2M}{\partial WG1} & \frac{\partial T2M}{\partial TG2} & \frac{\partial T2M}{\partial TG1} \\ \frac{\partial HU2M}{\partial WG2} & \frac{\partial HU2M}{\partial WG1} & \frac{\partial HU2M}{\partial TG2} & \frac{\partial HU2M}{\partial TG1} \end{bmatrix} \tag{14}$$

where differentials are replaced with finite differences in actual calculation ($\partial \to \Delta$). Calculation of joint Jacobian is very similar to calculation of Jacobian $\mathbf{M}$ described in section 3.3 with only difference that x in numerator of (13) is replaced with y e.g., control variables are replaced with simulated observations stored in OBSIMU ASCII files.

4 Assimilation execution sequence

Assimilation sequence is executed from scripts directory which should contain at least following 5 files:

- dayfile
- include.in
- run_EE927.sh
- run_001.sh
- run_ekf.sh

from which dayfile is ASCII file and other four are bash scripts. File dayfile contains only the date and production hour HPROD used by all scripts. Script include.in is called from remaining three scripts and it sets all the stuff to run whole assimilation sequence, like input and output paths, path to executables, paths to constant files as PGD.lfi, clim file, and so on. Remaining three bash scripts will be described shortly.

Surface analysis for the date 2015/10/05 12:00 for example, is executed with following sequence from the scripts directory:

1. Set '2015 10 05 06' in dayfile
2. Execute command: 'qsub -v EXPERIMENT=EX1,LCOLD=1 run_EE927.sh'

3. Execute command: `'qsub -v EXPERIMENT=EX1,LCOLD=1 run_001.sh'`

4. Set `'2015 10 05 12'` in `dayfile`

5. Execute command: `'qsub -v EXPERIMENT=EX1,LCOLD=0 run_EE927.sh'`

6. Execute command: `'qsub -v EXPERIMENT=EX1 run_ekf.sh'`

where `EX1` is arbitrary name of the experiment. In step 1. time in dayfile is set to beginning of the assimilation window e.g., 6 hours prior to analysis time. In step 2. script `run_EE927.sh` is submitted as a job and setting logical switch `LCOLD`=1. In this step three LBC files are prepared for HPROD, HPROD+3h and HPROD+6h and one SURFEX file TEST.lfi. These four files will be used in the next step - AROME model integration. Three LBC files are prepared using the AROME configuration EE927 (Lancelot) using three ECMWF LBC files as input. TEST.lfi file is prepared with SURFEX PREP executable initializing surface fields from ALADIN grib file. This step is shown in flowchart diagram in figure 7 with LCOLD=1 branch relevant in this case, which forces TEST.lfi to be prepared with PREP. In step 3. AROME is integrated 6 hors ahead using LBC files and SURFEX file from step 2 (Morgane configuration). At output we get: ICMSH file (not used in EKF analysis), new SURFEX file AROMOUT_0360.lfi and 7 in-line full-pos files which are used in EKF analysis (see flowchart 8). In step 4. time for which analysis is required is set in dayfile. In step 5. `run_EE927.sh` script is executed again but now with LCOLD=0. In this case LBC files are prepared in exactly same way as in step 2, but SURFEX file is now prepared also in Lancelot and not PREP. Last, 6-th step is a main analysis. It is shown in flowchart 9. Prior to analysis CANARI gridded observations of T2M and HU2M are prepared. Then ASCII forcing files are prepared from AROME full-pos files using the `forcing` program. Forcing files and initial TEST.lfi enter into integration of the OFFLINE SURFEX 6 hours ahead, resulting in a new SURFEX file SURFOUT_REF.lfi - this represents a reference SURFEX state. From this file simulated observations are extracted and stored in OBSIMU_REF and evolved control variables are stored in MDSIMU_REF ASCII files. For this `varassim` is executed with LSIM=T in namelist file. Next, loop through NVAR control variables is entered in which initial TEST.lfi file is perturbed for control variable IVAR and it is stored in corresponding perturbed PREP.lfi file which is used as initial state for offline surfex integration 6 hours forward, after which new state is stored into SURFOUT_PER[IVAR].lfi file and subsequently all control variables are extracted and stored in MDSIMU_PER_[IVAR] file and simulated observations in OBSIMU_PER_[IVAR] ASCII file again by executing varassim with LSIM=T. After last variable (IVAR=NVAR) loop exits and varassim is executed in analysis mode using file AROMOUT_.0360.lfi as guess and CANARI file as observations. After performing analysis new PREP.lfi is generated which represents actual surface analysis.

4.1 VARASSIM execution control

For the proper working of the EKF analysis settings of the EKF execution control switches are crucial. There are three logical switches `LPRT, LSIM, LBEV` which control how VARASSIM program is executed. They could be set in `&NAM_IO_VARASSIM` namelist in `OPTIONS.nam` file for VARASSIM program. Following 4 execution modes are possible:

1. Preparation of (initial) perturbed ISBA soil state from reference one

LPRT	TRUE
LSIM	arbitrary
LBEV	arbitrary

This mode is normally executed at initial time prior to SURFEX offline runs, and it must be executed as many times as many control variables are in control vector (set by NVAR in the &NAM_VAR namelist). It adds perturbation to i-th control variable of control vector required in calculation of $\mathbf{M}$ and $\mathbf{H}$ Jacobians (see section 3.3). Control variable which should be perturbed is specified by IVAR in the &NAM_VAR namelist. First original (unperturbed) value x_i of the i-th control variable is read from (initial) reference 'PREP.lfi' file. Then new (perturbed) value x'_i of i-th control variable is calculated using equation (12) and saved to output 'PREP.lfi' file (beware of the same name as input file, this overwrites input file without warning). Finally background error covariance matrix $\mathbf{B}$ is initialized and stored in ASCII file 'BGROUNDin0' if it already doesn't exist.

2. Store simulated observations and evolved prognostic variables

LPRT	FALSE
LSIM	TRUE
LBEV	arbitrary

This mode is normally executed after the all offline SURFEX runs (reference + each perturbed) have successfully finished. It should be executed separately after each offline SURFEX run e.g., NVAR+1 times (NVAR perturbed runs + one reference run). In fact this mode just reads final SURFEX file and reads from it all simulated observations (model equivalents of observations) and stores them in ASCII file 'OBSIMU'. It also reads all (NVAR) control variables from final SURFEX LFI file and stores them in ASCII file 'MDSIMU'. 'OBSIMU' file contains as many columns as many observation types are considered in assimilation (specified by NOBSTYPE in &NAM_OBS namelist) and 'MDSIMU' file contains one column per each control variable (NVAR columns in file).

3. Update background error covariance matrix with M matrix

LPRT	FALSE
LSIM	FALSE
LBEV	TRUE

This updates background error covariance matrix $\mathbf{B}$ according to model time-evolution:

$$\mathbf{B}^{(k)} = \mathbf{M}_{k-1\to k}\mathbf{B}^{(k-1)}\mathbf{M}^T_{k-1\to k} \tag{15}$$

where $\mathbf{M}$ is Jacobian matrix of model operator $\mathcal{M}$. Note that this is in fact standard EKF time-update equation (2) with only difference that addition of $\mathbf{Q}$ matrix is omitted here, because it is added later in analysis step. Input $\mathbf{B}^{(k-1)}$ is read from file 'BGROUNDin' and output $\mathbf{B}^{(k)}$ is written to file 'BGROUNDout' (both ASCII). Calculation of jacobian matrix $\mathbf{M}$ is described in section 3.3.

4. EKF soil analysis

LPRT	FALSE
LSIM	FALSE
LBEV	FALSE

EKF soil analysis is performed by setting all three execution-control switches to FALSE. It first reads gridded observations 'CLSTEMPERATURE' and 'CLSHUMI.RELATIVE' from CANARI file to observation vectors YO. Then simulated observations are read from reference and all perturbed runs from ASCII files 'OBSIMU_REF' and 'OBSIMU_PRT_[i]' into vectors YF. Then covariance matrices **R**, **B** and **Q** are set according to section 3.2. If LBFIXED logical variable in &NAM_IO_VARASSIM namelist is set to TRUE (which was our case) then **B** matrix is each time initialized to fixed value specified in &NAM_VAR namelist with XSIGMA_M(i) variables ('i' corresponds to i-th control variable). If LBFIXED is set to FALSE then previous value of **B** matrix is read from 'BGROUNDin' ASCII file and is updated by fixed **Q** matrix ($\mathbf{B} = \mathbf{B} + \mathbf{Q}$) each time-step. After that Jacobian matrix **H** and innovation vector ZB = YO - YF is calculated. After that analysis increment is calculated which is added to control vector guess to get control vector analysis XI. This is then written to 'PREP.lfi' file.

5. LBFIXED switch Fourth logical variable LBFIXED in &NAM_IO_VARASSIM namelist has meaning only in analysis mode (paragraph 4.) in other modes its value doesn't play any role. If it's value is FALSE then **B** is read from 'BGROUNDin' file and is updated by adding model error covariance matrix, $\mathbf{B} = \mathbf{B} + \mathbf{Q}$. Otherwise, if it is TRUE then at the beginning of the analysis mode it is always reinitialized to its default initial value $\mathbf{B} = \mathbf{B}_0$ according to (9) and is not updated with **Q** matrix nor with Kalman gain **K** in measurement-update equation (4). In any case at the end of EKF analysis **B** is saved in 'BGROUNDin' file. If this file is renamed outside the VARASSIM program to 'BGROUNDin' it acts as initial value in the next assimilation cycle.

5 Test cases

Following main SURFEX and EKF settings were used throughout the all case studies:

- Offline SURFEX v6.0 was used for the first few test cases. Later it was upgraded to SURFEX v7.2 from cy38t1 pack.
- 3-layer ISBA scheme was used which is default option for SURFEX v7.2 while Mahfouf used 2-layer ISBA scheme (could be set with CISBA option in &NAM_ISBA namelist).
- Only single patch was used for NATURE tiles in ISBA scheme (default for SURFEX v7.2) e.g., no tile subdivision to patches only tile aggregated parameters are considered (set with NPATCH=1 in &NAM_ISBA namelist).
- Over NATURE tiles canopy schema was turned ON (LISBA_CANOPY=TRUE in &NAM_PREP_ISBA namelist). This computes screen-level variables prognosticaly. For the TOWN tiles cannopy schema was switched OFF (LTEB_CANOPY=FALSE in &NAM_PREP_TEB) and T2M, HU2M values were computed diagnostically over TOWN tiles using the Geleyn vertical interpolation method (N2M=2 in &NAM_DIAG_SURFn namelist).
- Time step of offline SURFEX integration was set to 300 seconds (5 min) (XTSTEP_SURF=300 in &NAM_IO_OFFLINE namelist).

- VARASSIM control switch LBEV was set to FALSE meaning that background error covariance matrix **B** is not updated by **M**.

- VARASSIM control switch LBFIXED was set to TRUE in analysis mode of EKF meaning that at each analysis cycle **B** matrix is reset to its default initial value $\mathbf{B_0}$ (set in the namelist) before calculating analysis and it is also not updated according to (4).

5.1 Test run with offline SURFEX v6.0 and AROME cy36t1

First test case was run for analysis time 5/10/2015 12:00 UTC using the offline SURFEX v6.0 and AROME cy36t1 (the original configuration). It contains probably the old inline SURFEX v5.0. Running this configuration we've found that PREP.lfi files after EKF analysis but also before it (just after offline steps) all had unrealistic high T2M and HU2M values, up to 420 K (approx. 150 deg.C) resp. 1.4 (140%). This is shown in figure 1 which show output SURFOUT.lfi after the very first offline step (5/10/2015 06:05 UTC). This issue occur only for those grid cells which

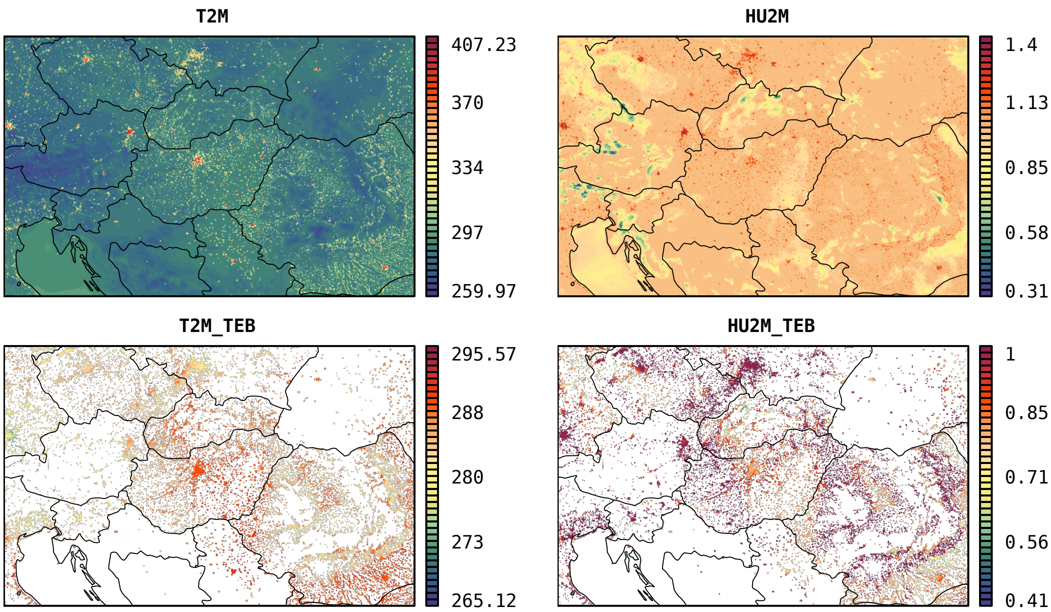

Figure 1: Tile aggregated T2M and HU2M (top) and T2M_TEB and HU2M_TEB (bottom) specific for TOWN tiles taken from the original offline SURFEX v6.0 after the first time step (5 minutes) after initialization e.g. for the time 05/10/2015 06:05.

contain any fraction of the town tile. There was evident relationship between T2M resp. HU2M on one side and FRAC_TOWN on the other e.g., pixels with largest fractions of town tiles, as occur in large cities like Budapest, Vienna, Prague, has largest deviations from physically reasonable values. This can be seen if one compares T2M map in figure 1 with maps of FRAC_TOWN at the top in figure 2. It was also checked that initial SURFEX file TEST.lfi has properly initialized TEB prognostic variables (T_ROAD1, T_ROAD2, T_ROAD3, T_ROOF1, T_WALL1...) from aladin.ALA grib file. We have made another run with exactly same configuration but with LTOWN_TO_ROCK in NAM_PGD_ARRANGE_COVER namelist set to TRUE while originally this was FALSE and remaking PGD.lfi file. This switch effectively eliminates use of the TEB scheme because all town tiles are replaced with nature tiles for which ISBA scheme is used. By this, unrealistic T2M and HU2M values were eliminated for all forcing ranges. This suggested some problems with TEB scheme,

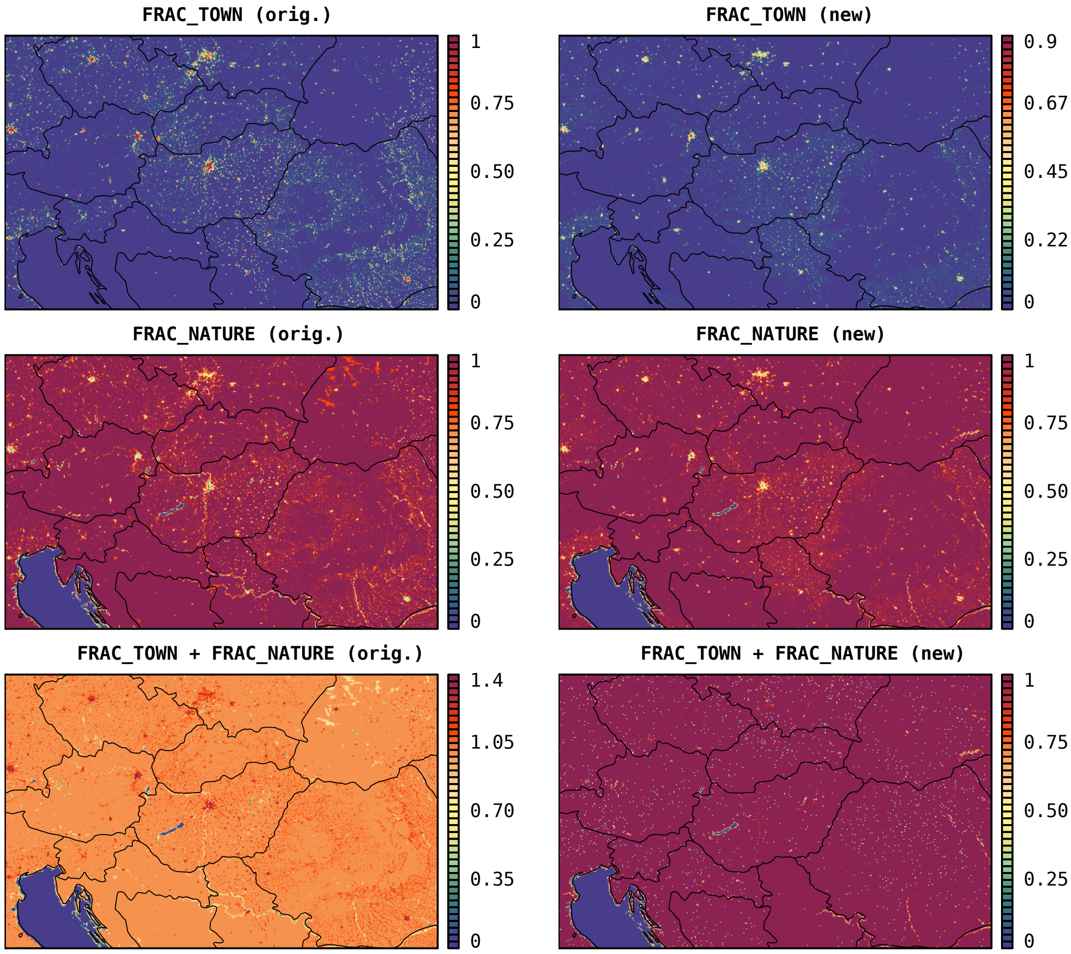

Figure 2: Cover fractions for NATURE and TOWN tiles and their sum for original setup based on AROME cy36t1 and SURFEX v6.0 (left) and updated setup based on AROME cy38t1 and SURFEX v7.2 (right).

however `T2M_TEB` (2m temperature over the town tiles) in offline SURFEX output had realistic values. Same was true also for `T2M_ISBA`, which is important because this field is used in EKF as simulated observation (`MDSIMU`). This is reason why EKF analysis increments doesn't show ugly values even if T2M does. Those facts indicate that problem could be in wrong aggregation over tiles because value of lets say T2M over grid box should be aggregated in some way from per-tile values `T2M_ISBA, T2M_TEB, T2M_SEA, T2M_WAT` using the tile fractions `FRAC_XX` as weighting factors. The sum of `FRAC_TOWN` and `FRAC_NATURE` really show some mismatch in tile cover fractions since sum through all fraction must give exactly 1, but we see that we had greater values with only TOWN and NATURE added (fractions are positive numbers), see botom of figure 2. We thus concluded that wrong initialization of tile cover fractions probably causes the mentioned problem. Why initialization is wrong has to be further investigated (it could be PGD related issue), but in the following we show runs with new configuration which is free of this problem and is based on more recent SURFEX v7.2 thus we suggest using rather new configuration instead of original SURFEX v6.0.

As both offline and inline SURFEX versions used in first cases were quite old we decided to set-up a new assimilation suite based on offline SURFEX v7.2 and AROME cy38t1 which

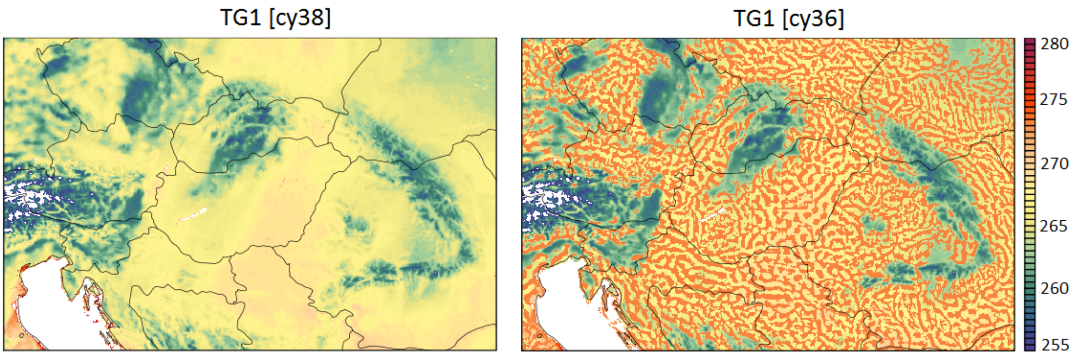

Figure 3: Comparison of TG1 control variable from initial `TEST.lfi` (18/01/2016 06:00) between SURFEX v7.2 (cy38t1) and v6.0 (cy36t1).

contains inline SURFEX v7.2. Equal version number minimizes potential incompatibilities. We checked for the same analysis time whether problem with high T2m values disappears or not. We also turned `LTOWN_TO_ROCK=T` switch on and off. In figure 3 TG1 from initial `'TEST.lfi'` (e.g. one which is used as a offline run initial condition at 18/01/2016 06:00) is compared between cy36t1 and cy38t1 runs for same analysis date (18/01/2016 12:00). In both runs identical forcing were used. T2M and HU2M can't be compared in initial file, because those diagnostic filed are calculated only in the course of offline run. TG1 however shows quite large differences between cy36t1 and cy38t1 both in value and also in spatial structure. Same is true for remaining surface fields although we doesn't show that. Initialization in cy38t1 seems to be physically more reasonable than in cy36t1,which show patch-like structures over whole domain. TG2 field has similar spatial structure and is not shown. Reason for that can be in different PGD files used in surface initialisation (PREP).

5.2 Test run with offline SURFEX v7.2 and AROME cy38t1

This case was run on AROME cy38t1 (coupled with SURFEX v7.2) and offline SURFEX v7.2 for 05/02/2016 12:00. In figure 4 maps of Jacobian matrix **H** elements are shown. Note that units used in figures are: (K) for temperatures TG, T2M, (1) for air relative humidity HU2M and (m^3/m^3) for soil moisture WG. These represents sensitivity of simulated observations (OBSIMU) to perturbation of control variables. Z-axis scale is same for pair of elements for better comparison of sensitivity to surface layer variables (WG1, TG1) with root layer ones (WG2, TG2). Obtained results are in some sense similar to Mahfouf results [2] in characteristic behavior, but our values are by order lower then their which can be attributed mainly to very different weather conditions. At the beginning of February, which was our case, turbulent exchange is much lower than at the beginning of the July due to reduced solar radiation forcing. Also, they considered two daytime assimilation windows: [06-12] and [12-18] UTC while we have tested only [06-12] UTC windows in this study.

Though sign of the mean value of each **H** element over the domain is same as in [2]. It can be seen also on maps in figure 4 where elements $H_{11}, H_{12}, H_{23}, H_{24}$ are negatively biased while remaining four are positively biased. It means that increasing soil moisture WG1, WG2 decreases 2m air temperature T2M and increasing soil temperature TG1, TG2 decreases 2m humidity HU2M. On the other hand increasing soil moisture increases 2m humidity and increasing soil temperature increases 2m temperature. This is true on average for both soil layers for this

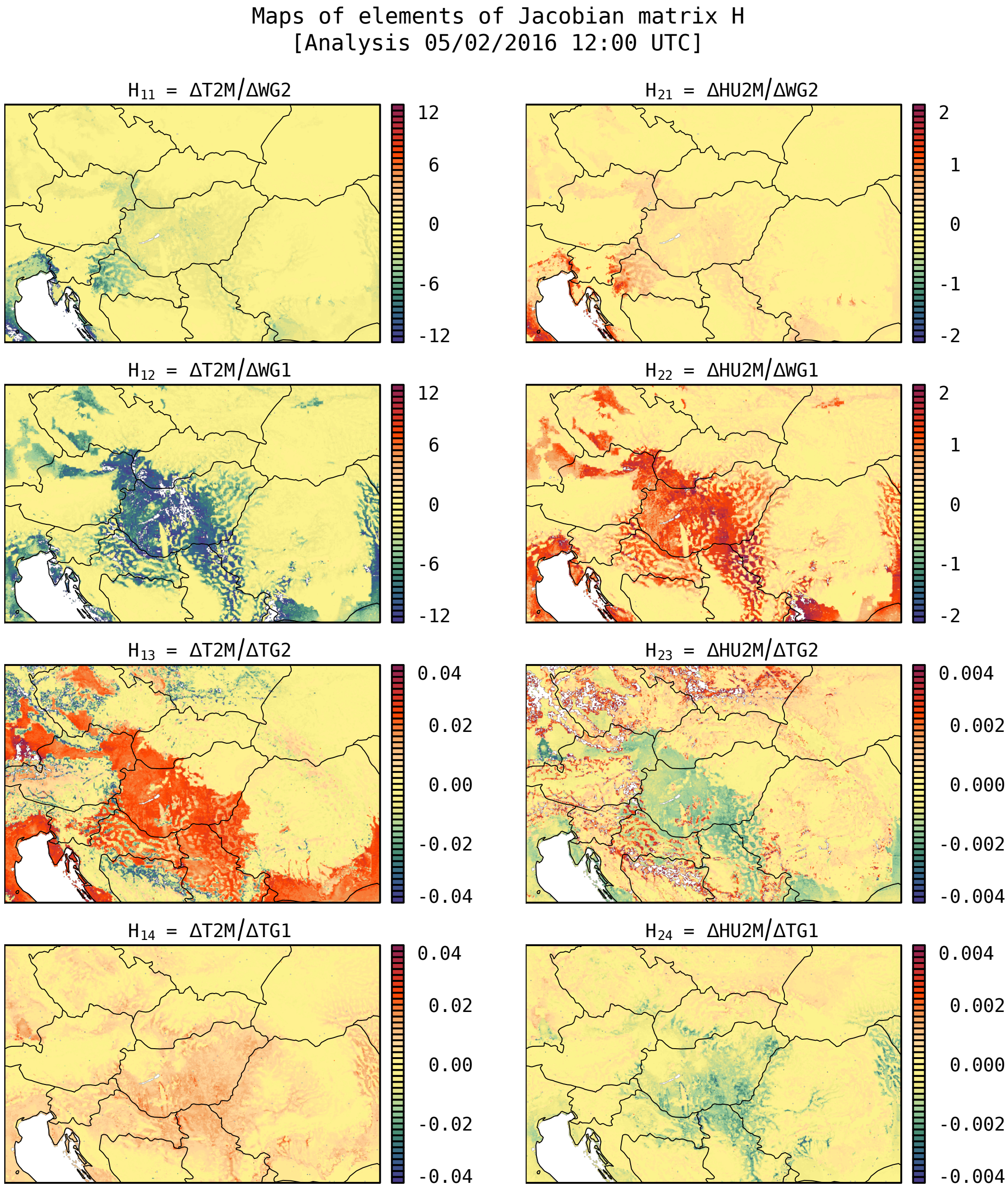

Figure 4: Maps of observation operator Jacobian matrix elements H_{ij} for 05/02/2016 12:00 test run after switching to AROME cy38t1 and offline SURFEX v7.2. Layout of maps in figure corresponds to matrix $\mathbf{H}^T$ (14).

specific case.

If one want to be precise the linearity of simulated T2M and HU2M response to soil variables perturbation should be verified. We used default values for perturbation amplitudes `TPRT_M(i)`, same as in [2] with only exception that perturbation amplitudes for TG1 and TG2 were reduced by a factor of 10. Although in this study Jacobian matrix $\mathbf{M}$ was not used, but same question about size of perturbation amplitudes have to be considered also for it.

We can see that in this case T2M is more sensitive to perturbations of volumetric water content in 1-st soil layer (superficial soil moisture) near the surface then to 2-nd (root) layer. At the same time it is more sensitive to perturbations of soil temperature TG2 in 2-nd ISBA layer then to perturbations of surface temperature TG1. Similar relations in the absolute value sense are valid for relative humidity in 2m but with opposite sign.

Analysis increments for each control variable are shown in figure 5. Kalmain gain matrix **K** elements are in figure 6. They represent ratio of analysis increments to observation innovations for specific pair of control variable – measured variable.

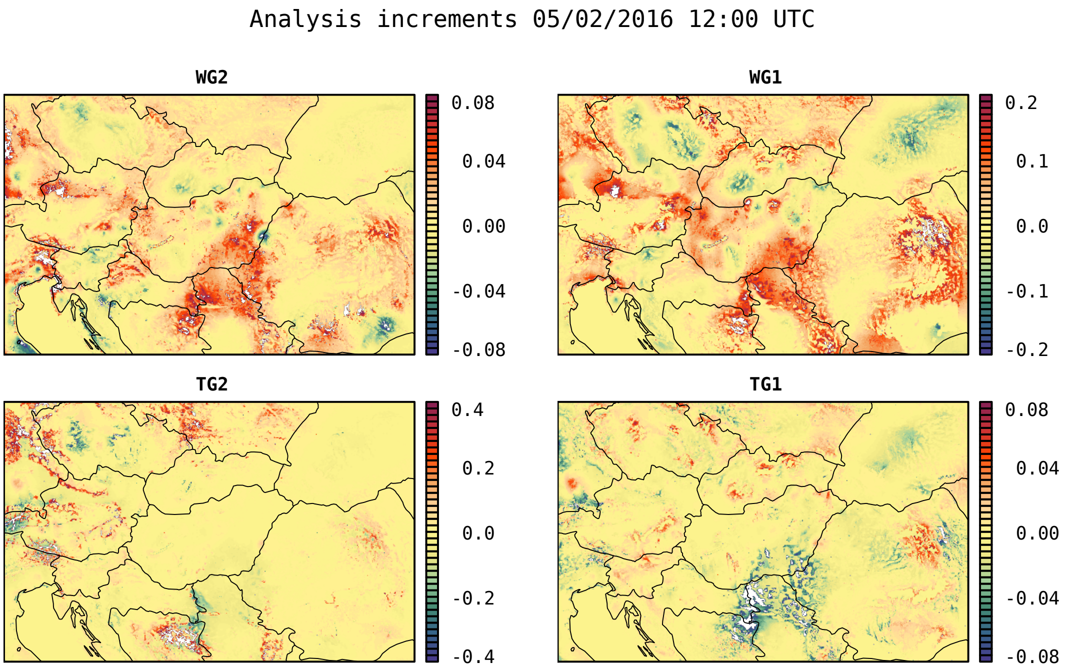

Figure 5: Analysis increment maps for each EKF control variable from the 05/02/2016 12:00 test run after switching to AROME cy38t1 and offline SURFEX v7.2

6 Suggestions and future plans

Several possibilities for improvement in current EKF assimilation and plans for future could be considered. To mention some of them:

- Do case study with cycling more EKF assimilation steps (6 hour window length) in a row.
- Perform several 1-column analysis case studies for particular surface points where measuring station is located. By this we can eliminate use of CANARI needed for producing gridded observations and we can directly use the T2M and HU2M observations from the measuring station at that point. Try runs for different weather conditions as dry vs. wet, clear-sky vs. overcast, mountains vs. plains, summer vs. winter, etc.
- Optimization: Since EKF consumes large amount of computing time it should be optimized as much as possible. One of the ways how to achieve this goal is to execute reference and 4 perturbed runs parallel on different resources (nodes, processors) as separate jobs since they are independent on each other (see flowchart 9). Then wait to finish all runs and go to analysis step. Presently those 5 runs are executed sequentially one after other. This can be seen from the intermediate and output file timestamps. For example most recent

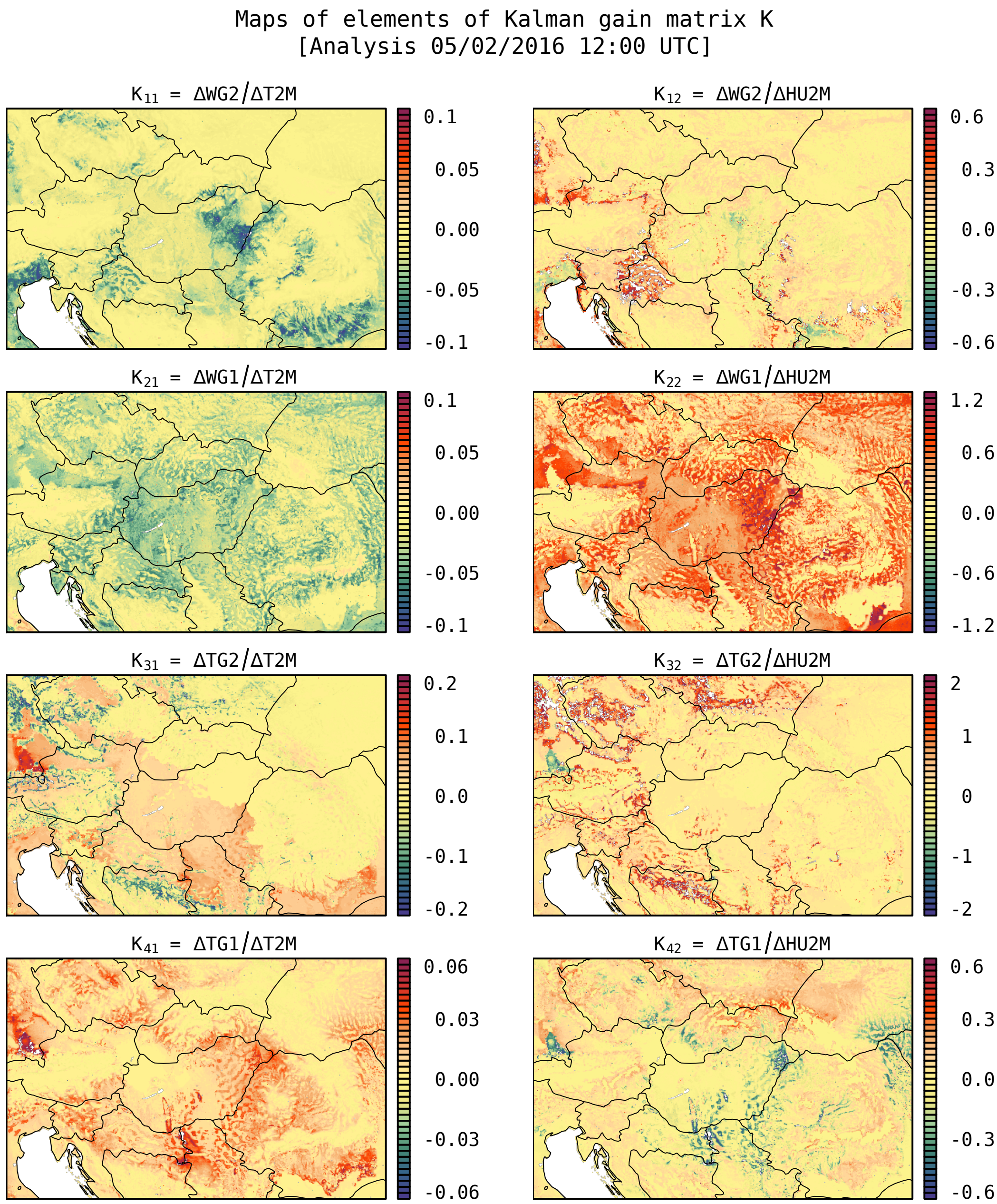

Figure 6: Maps of Kalman gain matrix elements K_{ji} for 05/02/2016 12:00 test run after switching to AROME cy38t1 and offline SURFEX v7.2. Note the different z-axis scale for each map.

file in REF subfolder is older then oldest file in PER1 subfolder. Most recent file in PER1 subfolder is older then oldest file in PER2 subfolder and so on.

- Computing time could be probably significantly reduced using the NETCDF forcing instead of ASCII forcing. Parsing 11 ASCII forcing files (for each forcing step) is considerably slower than parsing single NETCDF file. It would require modification in '`forcing.f90`' program.
- Consider if more reasonable specification of **R** matrix could bring any advantage. Now it is static, diagonal matrix independent on geographical location. But we use CANARI

analysis for screen-level observations and its error vary from point to point being larger for grid points far from observing stations or in complex terrain. If the CANARI analysis error can be estimated it can be taken into account in EKF by modifying the variances of **R** matrix so that they can differ for each grid box or even could change with time.

- Comparison of results obtained with EKF and OI_MAIN methods.
- Include satellite observation of WG1 into EKF assimilation.
- Consider to use the SODA (SURFEX offline data assimilation) for surface assimilation. It includes also here presented EKF method for assimilation in SURFEX nature tiles, but it allows assimilation in wider context in one common framework based on offline SURFEX.

7 Summary

Extended Kalman Filter (EKF) surface assimilation study was continued at OMSZ during the RC LACE stay in Budapest. Validation was originally started with offline SURFEX version 6.0 and AROME cy36t1. Unrealistic high values of 2m air temperature and relative humidity were detected over town tiles in offline SURFEX outputs. Problem could be eliminated by enabling `LTOWN_TO_ROCK` switch in `NAM_PGD_ARRANGE_COVER` SURFEX namelist, however this effectively eliminates TEB scheme for town tiles. Initial SURFEX file showed no indication of mentioned issue. Interestingly per-tile fields, T2M_TEB and HU2M_TEB were correct in the same SURFEX output files, indicating that problem could be with aggregation over town tiles. We've found that this could be due to mismatch in cover fraction values in SURFEX which are initialized in PGD from ecoclimap database. This would require further validation of used PGD. Nevertheless it was decided to use new offline SURFEX release and newer AROME common cycle. Two latest offline SURFEX releases were tested (7.2 and 7.3) with AROME cy38t1 but 001 configuration didn't worked with offline SURFEX v7.3 outputs only with v7.2 outputs. Necessary namelist upgrades for 927 and 001 configurations were also done. Preliminary tests showed more consistent results then with original configuration with no indication of mentioned issues. Results presented in the case studies should be understand as only preliminary. In the future more extensive case studies are planned with well controlled conditions. This would also require optimization of the execution sequence for example with NETCDF input or further parallelization. Switching to SODA may be a next step as it represents common interface for SURFEX based assimilations, part of which is also here studied EKF. Validation of the gridded observation procedure (CANARI) and the EKF analysis has been started as well.

8 Acknowledgements

I would like to acknowledge following people for their help and assistance related with my Budapest stay: *Mile Maté, Tóth Helga, Szintai Balázs, David Lancz, Kullmann László, Zsebeházi Gabriella, Jozef Vivoda, and Roman Zehnal.*

References

[1] R. Hess. “Assimilation of screen-level observations by variational soil moisture analysis”. English. In: *Meteorology and Atmospheric Physics* 77.1-4 (2001), pp. 145–154. ISSN: 0177-7971. DOI: 10.1007/s007030170023. URL: http://dx.doi.org/10.1007/s007030170023.

[2] J.-F. Mahfouf et al. “A comparison of two off-line soil analysis schemes for assimilation of screen level observations”. In: *Journal of Geophysical Research: Atmospheres* 114.D8 (2009), n/a–n/a. ISSN: 2156-2202. DOI: 10.1029/2008JD011077. URL: http://dx.doi.org/10.1029/2008JD011077.

[3] V. Masson et al. “The SURFEXv7.2 land and ocean surface platform for coupled or offline simulation of earth surface variables and fluxes”. In: *Geoscientific Model Development* 6.4 (2013), pp. 929–960. DOI: 10.5194/gmd-6-929-2013. URL: http://www.geosci-model-dev.net/6/929/2013/.

[4] A. Duerinckx et al. “Study of the Jacobian of an Extended Kalman Filter for soil analysis in SURFEXv5”. In: *Geoscientific Model Development Discussions* 7.5 (2014), pp. 7151–7196. DOI: 10.5194/gmdd-7-7151-2014. URL: http://www.geosci-model-dev-discuss.net/7/7151/2014/.

[5] H. Tóth. *Report from stay at Royal Meteorological Institute, Brussels, 24/03/2014–28/03/2014*. 2014.

[6] P. Le Moigne et al. “SURFEX scientific documentation”. In: *Note de centre (CNRM/GMME), Météo-France, Toulouse, France* (2012).

[7] F Bouttier and P Courtier. “Data assimilation concepts and methods March 1999”. In: *Meteorological Training Course Lecture Series. ECMWF* (2002).

A EKF related namelists

```
&NAM_IO_VARASSIM
  LPRT       = F/T, <Set from control script run_ekf.sh>
  LSIM       = F/T, <Set from control script run_ekf.sh>
  LBEV       = F,
  LBFIXED    = T,
/
&NAM_OBS
  NOBSTYPE   = 2,        !! Number of observation types
  INCO(1)    = 1,        !! Include obs. type 1 (T2M) in analysis
  INCO(2)    = 1,        !! Include obs. type 2 (HU2M) in analysis
  INCO(3)    = 0,        !! Include obs. type 3 (WG1) in analysis
  YERROBS(1) = 1.0,      !! STDEV of T2M obs. error in K
  YERROBS(2) = 0.1,      !! STDEV of HU2M obs. error in 1 (not %)
  YERROBS(3) = 0.4,      !! STDEV of WG1 obs. error in fraction of SWI
/
&NAM_VAR
  IVAR       = <Set from control script run_ekf.sh>, !! Control variable
  NVAR       = 4,        !! Number of control variables
  INCV(1)    = 1,        !! Include control variable 1 in analysis
  INCV(2)    = 1,        !! Include control variable 2 in analysis
  INCV(3)    = 1,        !! Include control variable 3 in analysis
  INCV(4)    = 1,        !! Include control variable 4 in analysis
  XVAR_M(1)  = 'WG2',    !! Name of control variable 1 in PREP.lfi
  XVAR_M(2)  = 'WG1',    !! Name of control variable 2 in PREP.lfi
  XVAR_M(3)  = 'TG2',    !! Name of control variable 3 in PREP.lfi
  XVAR_M(4)  = 'TG1',    !! Name of control variable 4 in PREP.lfi
  PREFIX_M(1) = 'X_Y_WG2 (m3/m3)', !! Prefix of control variable 1 in PREP.txt
  PREFIX_M(2) = 'X_Y_WG1 (m3/m3)', !! Prefix of control variable 2 in PREP.txt
  PREFIX_M(3) = 'X_Y_TG2 (m3/m3)', !! Prefix of control variable 3 in PREP.txt
  PREFIX_M(4) = 'X_Y_TG1 (m3/m3)', !! Prefix of control variable 4 in PREP.txt
  XSIGMA_M(1) = 0.1,     !! STDEV of control variable 1 background error
  XSIGMA_M(2) = 0.1,     !! STDEV of control variable 2 background error
  XSIGMA_M(3) = 2.0,     !! STDEV of control variable 3 background error
  XSIGMA_M(4) = 2.0,     !! STDEV of control variable 4 background error
  TPRT_M(1)  = 0.0001,   !! Perturbation amplitude of control variable 1
  TPRT_M(2)  = 0.0001,   !! Perturbation amplitude of control variable 2
  TPRT_M(3)  = 0.00001,  !! Perturbation amplitude of control variable 3
  TPRT_M(4)  = 0.00001,  !! Perturbation amplitude of control variable 4
  SCALE_Q    = 0.125,    !! Scale factor for Q matrix: Q = SCALE_Q^2 * B0
/
```

B Flow-charts

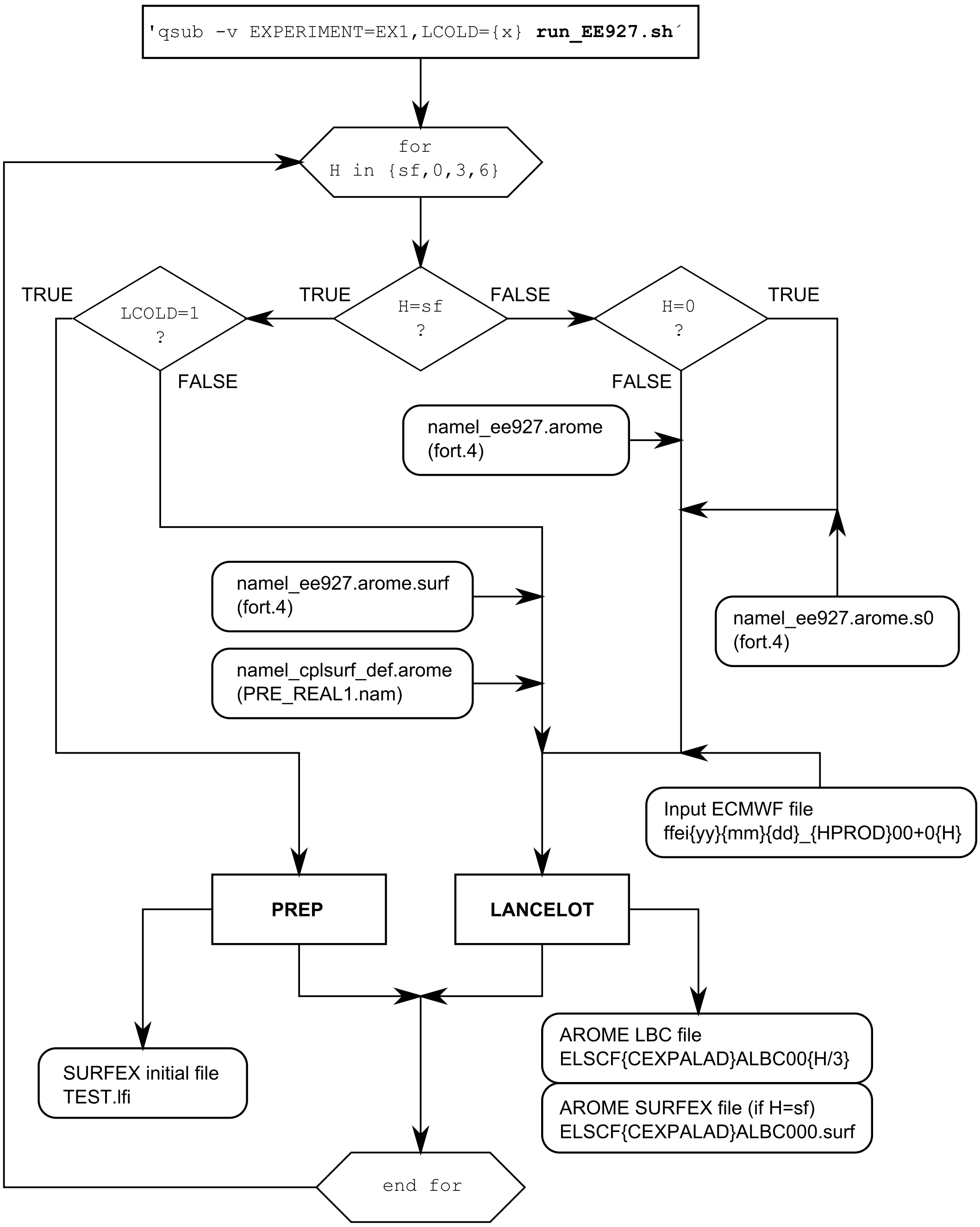

Figure 7: Flowchart of the `run_EE927.sh` bash script execution.

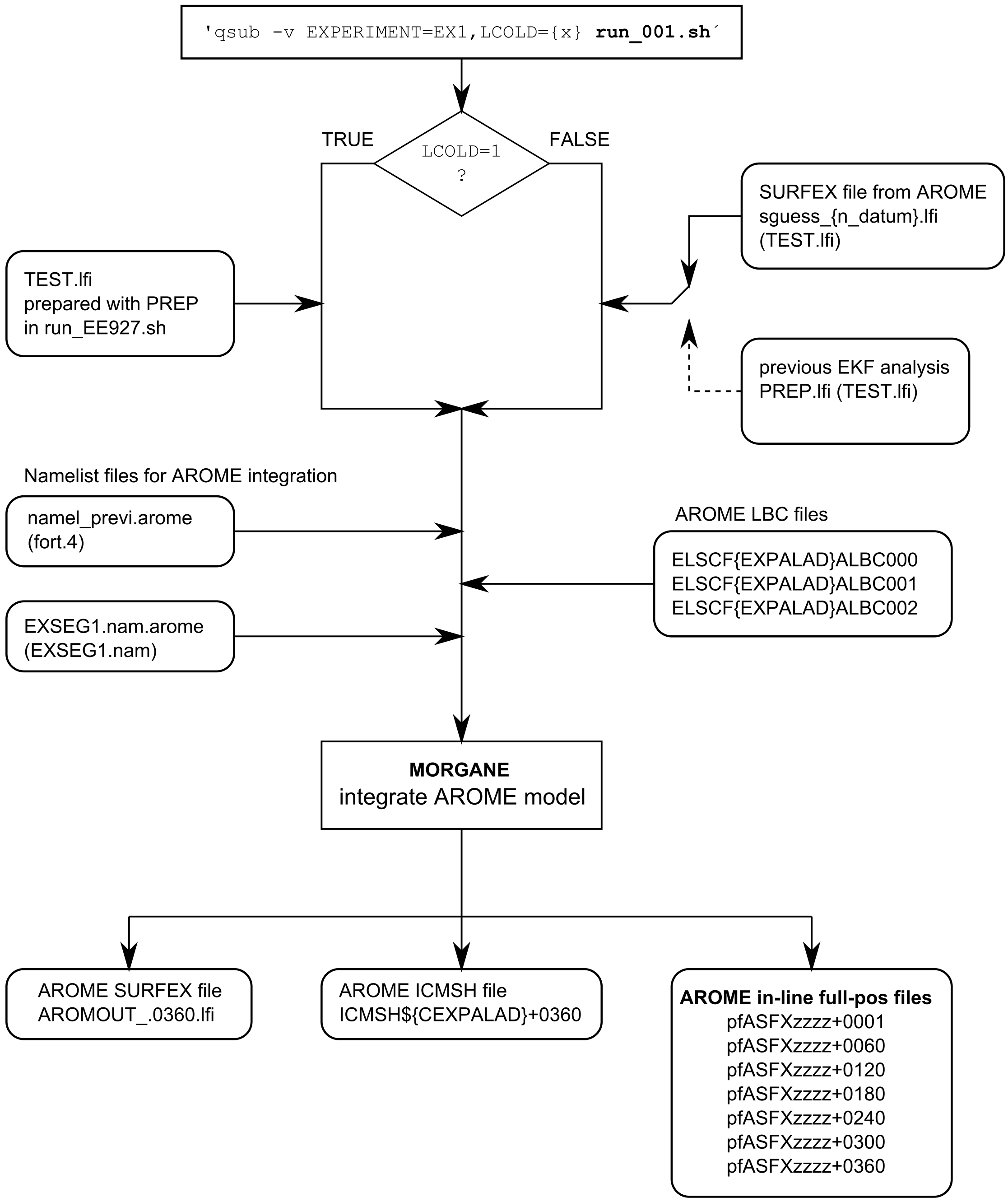

Figure 8: Flowchart of the `run_001.sh` bash script execution.

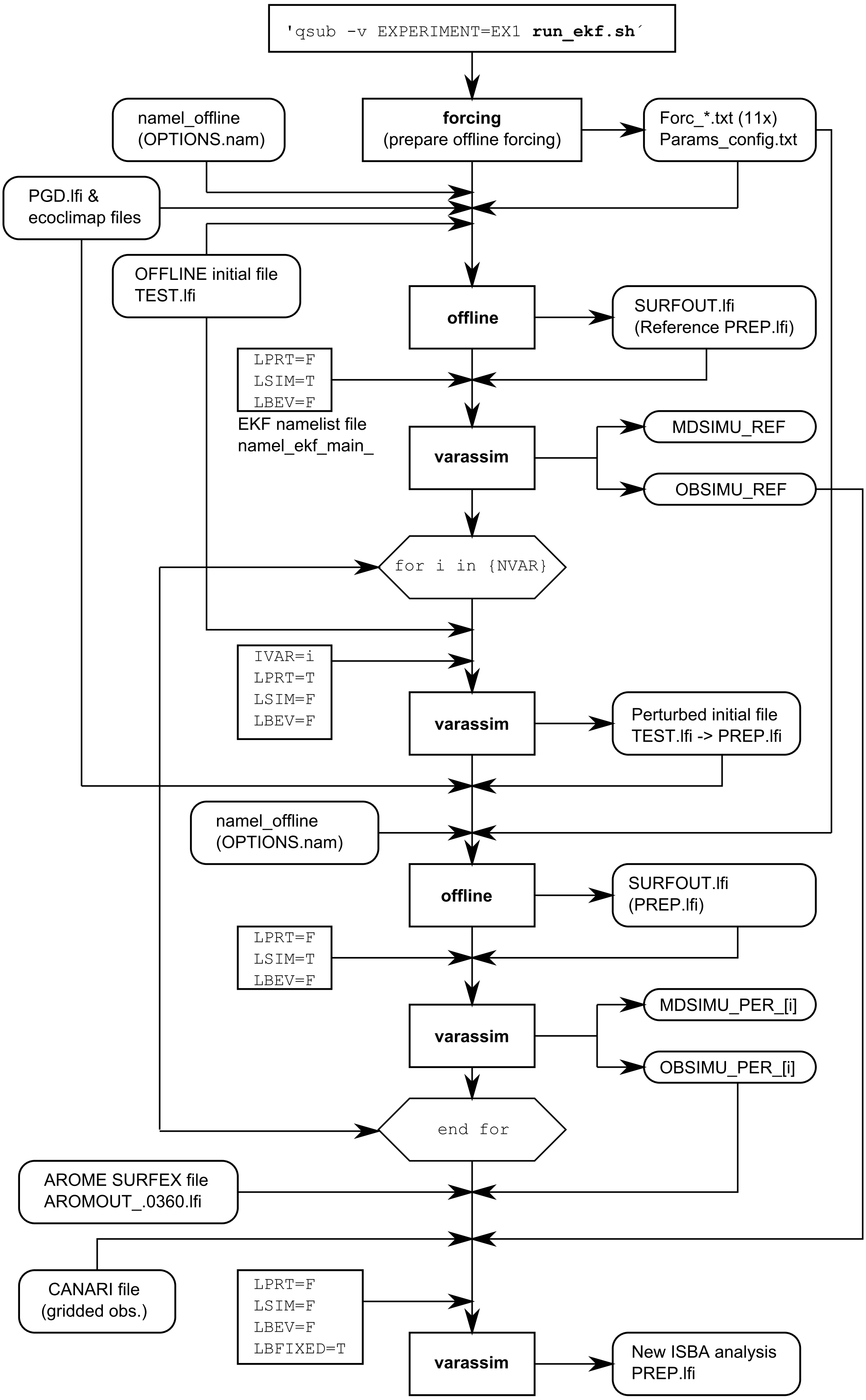

Figure 9: Flowchart of the `run_ekf.sh` bash script execution.