CALMO - Progress Report

P. Khain, I. Carmona, A. Voudouri, E. Avgoustoglou, J.-M. Bettems, F. Grazzini, P. Kaufmann

February 2017

Contents

1. Introduction – CALMO project

The CALMO project is based on the objective calibration method which was developed and implemented in regional climate model by Omar Bellprat and Christoph Schär (ETH). The purpose of the CALMO project is to implement the calibration method of [1] to NWP model COSMO. Briefly, the calibration method is the following:

a. First, define the parameters for tuning and their allowed ranges. The selected parameters have to be the most significant for the verified fields. For example, for the maximum 2m-temperature we may need to focus on soil and radiation schemes parameters, while for precipitation, we may need to consider also microphysical parameters, etc.
b. Define the forecast fields to be verified. It is important to select many meteorologically important fields in order to better reflect the weather conditions. Otherwise there is a danger that the calibration procedure will improve specific fields while degrading the overall skill of the forecast.
c. Define the time periods and geographical regions for calibration. The time periods and the regions should be chosen to represent a meaningful forecast.
d. Define the parameters (combinations) values for performing the COSMO simulations. The minimum required number of simulations to be performed is $2N + 0.5N(N - 1) + 1$, where N is the number of calibrated parameters.
e. Define the method to perform the COSMO simulations, i.e. initial and boundary conditions and the forecasts time ranges. For soil-related parameters, long term "spin-up" simulations of the COSMO soil scheme are needed for preparing proper initial conditions.
f. After the simulations are performed, the Meta-Models are constructed, i.e. the forecasted fields are interpolated in parameters space via N-dimensional quadratic polynomial (for each field, for each region and each day, separately). These interpolation formulas (the Meta-Models) allow estimating the forecasted field value for arbitrary parameter values (for each region and each day) without performing real COSMO simulation.
g. At the next stage, the parameters space is filled by a large number of parameters combinations. For each parameter combination, a forecast field time series is produced (using the Meta-Models), compared with the observations, and evaluated using a performance score.
h. Finally, the parameters combination which obtained the best score is selected.
i. In principle, it is reasonable to perform a real COSMO simulation with the selected parameters combination, and verify whether the forecasts are indeed better (than with the default parameters combination).

2. Overview: different stages of the CALMO project

The CALMO project included three stages. At CALMO-stage-1 we have performed a preliminary calibration of COSMO-7km. The detailed description of that stage is presented in COSMO technical report [2]. At CALMO-stages 2 and 3 we have performed several improvements of the calibration process. The main characteristics of the 3 stages are summarized in table 1:

	Stage-1	**Stage-2**	**Stage-3**
Resolution	7km	2.2km	1.1km
Simulations domain	~[17W-22E, 35N-57N]	~[0E-17E,42N-50N]	~[0E-17E, 42N-50N]
Calibration area	Switzerland	Switzerland and north Italy	Switzerland and north Italy
Calibrated atmospheric fields	T2m-max; T2m-min; 24h-precipitation	T2m-max; T2m-min; 24h-precipitation; sounding profiles diagnostics: CAPE; CIN; total column water vapor; vector wind shears between the levels 500-700mb/700-850mb/850-1000mb; temperature, relative humidity and wind components at 850, 700 and 500mb	T2m-max; T2m-min; 24h-precipitation; sounding profiles diagnostics: CAPE; CIN; total column water vapor; vector wind shears between the levels 500-700mb/700-850mb/850-1000mb; temperature, relative humidity and wind components at 850, 700 and 500mb
Simulations period	1-20/1/2008, 1-20/6/2008	1/1/2013-31/12/2013	~ 1/1/2013-1/2/2013
Tuning parameters	rlam_heat, tkhmin, tur_len	rlam_heat, tkhmin, tur_len, entr_sc, c_soil, v0snow	tkhmin, tur_len, entr_sc, c_soil, crsmin

Table 1: Overview: different stages of the CALMO project

3. Short overview of the tuned parameters

In table 2 we present the parameters which were tuned during the different stages of the CALMO project, and briefly describe their physical meaning.

	Used at stages:	**Brief physical meaning**	**Min**	**Default**	**Max**
rlam_heat	1,2	rlam_heat [no units] is the parameter which linearly determines the heat resistance length of laminar layer; so that the higher is rlam_heat the higher is the resistance of laminar layer for heat transfer, and consequently, the lower is the heat transfer between the surface and the lower atmosphere	0.1	1	2
tkhmin	1,2,3	tkhmin [m^2/s] and tkmmin [m^2/s] determine the minimum limits for the turbulence coefficients. tkhmin presence is evident when the turbulent diffusion coefficients (then the mixing) are small, which occurs in stable conditions, mainly at night near the surface	0.1	0.4	1

tur_len	1,2,3	tur_len [m] is l_∞ in Blackadar formula (Blackadar, 1962) for the turbulence length. The higher is tur_len, the higher are the turbulent coefficients (both vertical and horizontal) in the middle-upper atmospheric levels, and consequently the higher are the turbulent fluxes (mixing) for all the variables and tracers	100	150	1000
entr_sc	2,3	entr_sc [m^{-1}] is the mean entrainment rate of boundary layer humidity into the shallow convection clouds. The higher is entr_sc, the more effective is the shallow convection vertical mixing.	0.05e-3	0.3e-3	2e-3
c_soil	2,3	c_soil [no units] is the surface-area index of the evaporating fraction of gridpoints over land: c_soil ∈ [0, c_lnd=2]. The higher is c_soil, the higher is the surface evaporation.	0	1	2
v0snow	2	v0snow [no units] is the factor in the terminal velocity for snow	10	20	30
crsmin	3	crsmin [s/m] is the minimum value of stomatal resistance used by the BATS approach for vegetation transpiration	50	150	200

Table 2: COSMO parameters tuned at different stages of the CALMO project

4. Meta-Model

4.1 Overview

The Meta-Model was widely discussed in COSMO technical report [2], so here we briefly present its basic idea. Following the theory of Meta-Model construction [1,3], for any parameters combination (for example N=3 parameters combination of rlam_heat, tkhmin, tur_len), for a given day "*i*" and a region "*r*", the COSMO field *F* (for example Tmax, Tmin, Pr, etc.) may be approximated by 3-dimensional polynomial of order 2:

$$F_{i,r} \approx F_{d\,i,r} + c_{i,r} + \sum_{n=1}^{N} a_{i,r}^{(n)} x_n + \sum_{\substack{n,m=1 \\ (n \neq m)}}^{N} B_{i,r}^{(n,m)} x_n x_m \qquad (1)$$

Where: $x_1 = \frac{rlam\_heat - rlam\_heat_d}{rlam\_heat_{max} - rlam\_heat_{min}}, x_2 = \frac{tkhmin - tkhmin_d}{tkhmin_{max} - tkhmin_{min}}, x_3 = \frac{tur\_len - tur\_len_d}{tur\_len_{max} - tur\_len_{min}}$

The index d stands for default. For default values of the N=3 parameters, i.e. [x1 = 0, x2 = 0, x3 = 0], the approximated field should be close to $F_{d\,i,r}$. The constants $c_{i,r}$, $a_{i,r}^{(n)}$, $B_{i,r}^{(n,m)}$ are obtained using several COSMO simulations, as described in the following. Each simulation (for given parameters values) yields a set of forecasted values $F_{i,r}$. When sufficient number of simulations is performed, one can interpolate the different known values of $F_{i,r}$ as function of [x1, x2, x3] using the 3D polynomial in eq. (1) above. The sufficient number is 2N + 0.5N(N − 1) + 1, so that for N=3 the sufficient number of simulations to be performed is 10. Next we discuss the ways to increase the quality and the representativeness of such fit. The following factors are important for the interpolation to be realistic (to be able to replace the COSMO simulations):

- The choice of parameters values (combinations) for COSMO simulations should be specific. In this work the design is chosen according to Bellprat (2012) [1] (see also COSMO technical report [2]). Moreover, one

should use as many as possible additional "constrain" simulations (for additional parameters combinations).

- The simulated COSMO field $F_{i,r}$ should not be noisy as function of the parameters. In other words, the sensitivity of $F_{i,r}$ on the parameters should be higher than the noise level. However, various COSMO fields are noisy for various parameters. That issue was discussed and solved in COSMO technical report [2].
- The time periods "*i*" and the regions "*r*" should be chosen to represent a meaningful forecast of the field $F_{i,r}$. In that work we have chosen typical periods of 24 hours (maximum and minimum daily temperature, 24h accumulated precipitation, etc.) and climatically distinguishable regions. The choice of the regions will be discussed in sections 4.2.1 and 4.2.2 below.
- It is important to select many meteorologically important fields F in order to better reflect the weather conditions. Otherwise there is a danger that the calibration procedure will improve specific fields while degrading the overall skill of the forecast. At CALMO-stages 2 and 3 we have included optimization of meteorological profiles characteristics. That will be discussed in section 4.2.3 below.
- The default values of the parameters should be located close to the center of their allowed ranges. Otherwise, in the "empty parameter ranges", the parabolic fit may reach very high (or very low) unrealistic peaks. The problem is, that the default values of rlam_heat, tur_len and entr_sc are significantly shifted from the centers of their allowed ranges: for [0.1 1 10], [100 500 10000], and [0.05e-3 0.3e-3 2e-3], for rlam_heat, tur_len and entr_sc, respectively. That problem will be discussed and solved in section 4.2.4 below.

4.2 Adaptations to the Meta-Model

From CALMO-stage-1 (see COSMO technical report [2]) to CALMO-stages 2 and 3 we have performed several adaptations to the Meta-Model codes:

4.2.1 Option not to average Tmax/Tmin over regions

For observations over Switzerland we use C. Frei [4] gridded data after correction to the elevations of model grid points. Over Italy we use the observations interpolated to the model grid (without correction to the elevations of model grid points), while only the grid points in vicinity of the stations get a value. At CALMO-stage-1 we have divided Switzerland area into 3 regions, and averaged the maximum and minimum 2m temperatures (Tmax and Tmin, respectively) and 24h accumulated precipitation (Pr) over these regions, before comparing with observations. While for precipitation, this averaging reduces the noise, for Tmax and Tmin we lost a lot of information. Just for example, Tmax errors at two different grid points can yield no error on average (see figure 1).

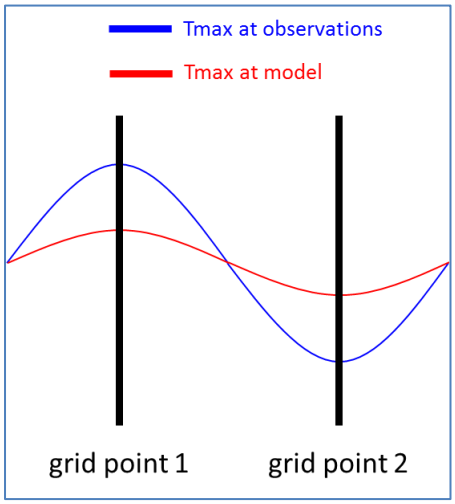

Figure 1: Schematic example for the reason not to average Tmax/Tmin over regions.

Moreover, at CALMO-stages 2 and 3 the Italian data is also analyzed. This data is not gridded (as Swiss one), so that much less grid points are available for comparison of the model to the observations. In that case, region averaging would be based on much less points than over Switzerland.

Therefore at CALMO-stages 2 and 3 we have **added the option** (to the Meta-Model code) not to average Tmax and Tmin over regions, but to calculate the Meta-Model forecast for all the available grid-points in model and observations (about $N_{\Psi regs} \approx 10407$for $\Psi = T\max$ *or* $\Psi = T\min$ for CALMO-stage-2, and similar number for CALMO-stage-3).

4.2.2 Defining new regions for averaging the 24h accumulated precipitation (optional also for Tmax, Tmin)

For observations over Switzerland we use the gridded MeteoSwiss radar composite (corrected by rain gauges) interpolated to the model grid. Over Italy we use the observations interpolated to the model grid, while only the grid points in vicinity of the stations get a value. In order to reduce the noise associated with precipitation fields, the precipitation model and observations values are averaged over $N_{\mathrm{Prregs},mon} = 6$ geographically unique regions, as presented at figure 2:

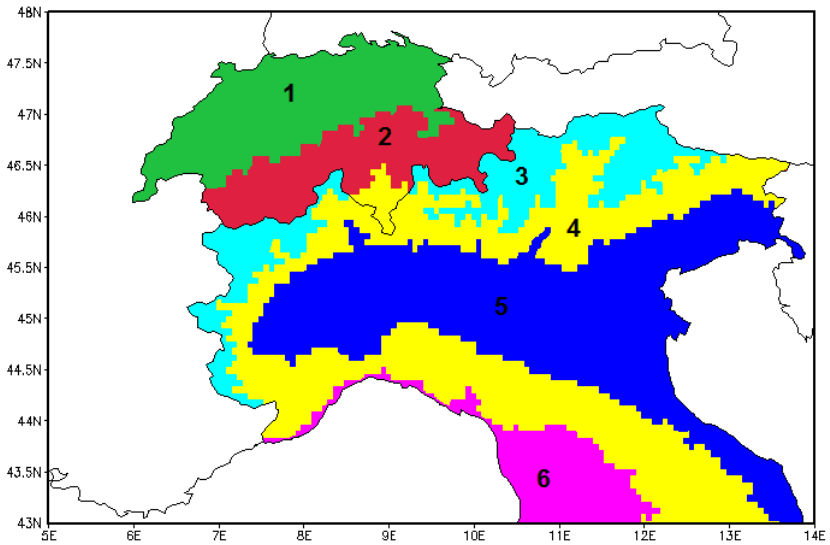

Figure 2: Geographically unique regions for precipitation averaging: 1-green: Swiss plateau (300m<h<1500m); 2-red: Swiss Alps (1500m<h); 3-cyan: Italian Alps (1500m<h); 4-yellow: Italian hills and Ticino (300m<h<1500m); 5-blue: Po Valley (h<300m); 6-magenta: Italian north-west coast (mainly h<300m).

4.2.3 Meta-Model predicts profiles characteristics

At CALMO-stage1 we have used the following fields: Ψ_1 - Daily maximum 2m temperature ($T_{\max}$), Ψ_2 - Daily minimum 2m temperature ($T_{\min}$), Ψ_3 - 24h accumulated precipitation ($\Pr$).

At CALMO-stages 2 and 3 we are using also the soundings data and the associated model profiles (at grid points near the soundings locations). The new verified fields are: Ψ_4 - Convective available potential energy ($CAPE$); Ψ_5 - Convective inhibition (CIN); Ψ_6 - Total column water vapor ($TCWV$); Ψ_7 - Vector wind shear between the levels of 500mb and 700mb ($WS1$); Ψ_8 - Vector wind shear between the levels of 700mb and 850mb ($WS2$); Ψ_9 - Vector wind shear between the levels of 850mb and 1000mb ($WS3$); $\Psi_{10,11,12}$ - Temperatures at 500mb (T500), 700mb (T700) and 850mb (T850), respectively; $\Psi_{13,14,15}$ - Relative humidity at 500mb (T500), 700mb (T700) and 850mb (T850), respectively; $\Psi_{16,17,18}$ - East-west wind component at 500mb (U500), 700mb (U700) and 850mb (U850), respectively; $\Psi_{19,20,21}$ - South-north wind component at 500mb (V500), 700mb (V700) and 850mb (V850), respectively.

There are 11 available soundings at the CALMO-stages 2 and 3 domains, as presented in figure 3:

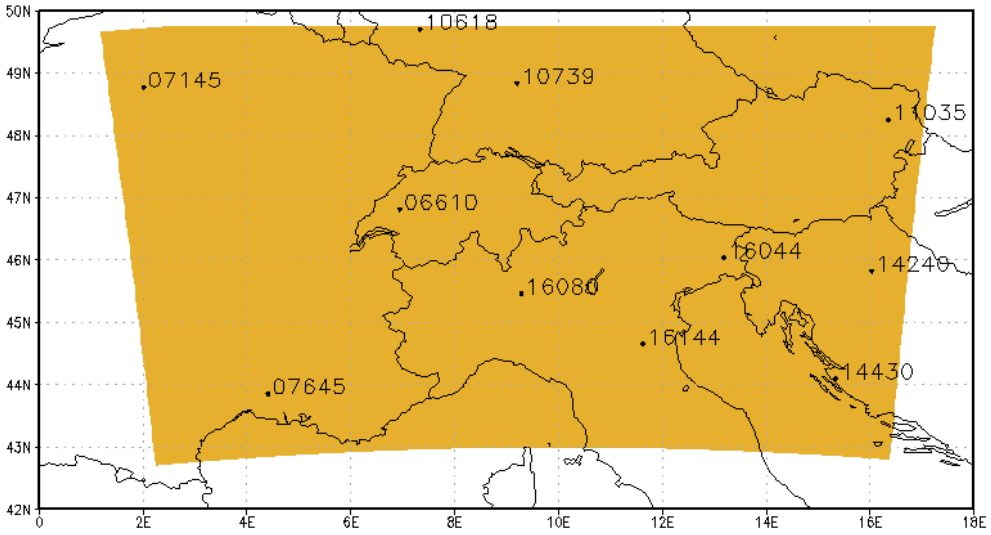

Figure 3: Available soundings inside CALMO-stages 2 and 3 domains.

Most of these soundings available twice per day, yielding about $N_{\Psi regs} \approx 20-25$ ($\Psi = any\ sounding\ field$) profiles over the domain per day (depends on the sounding report quality).

4.2.4 Logarithmic transformation for some of the parameters

As discussed already in CALMO-stage 1 [2], the parabolic fit can accurately represent the dependency of the verified fields in parameters space only if the default values of the parameters are located close to the center of their allowed ranges. Otherwise, in the empty parameter ranges, the parabolic fit may reach very high (or

very low) unrealistic peaks. The solution for that problem is transforming the "problematic" parameter/s to logarithm of the parameter/s. Such transformation "brings" the far away parameter values closer to the others, eliminating the empty parameter ranges, causing the parabolic fit to be more monotonic. In CALMO-stages 2 and 3, the "problematic" parameters are tur_len and entr_sc.

Recently we have developed a method to objectively transform these parameters to logarithmic space:

$x \to \hat{x} \equiv \log\left(\alpha \frac{x - x_{\min}}{x_{\max} - x_{\min}} + \beta \right)$. The demand for the transformed default value to be exactly at the center between the minimum and maximum values, i.e. $\hat{x}_d - \hat{x}_{\min} \overset{!}{=} \hat{x}_d - \hat{x}_{\min}$ defines α and β. Applying the procedure yielded $\alpha = 72, \beta = 0.25$, for tur_len, and $\alpha = 9500, \beta = 210$ for entr_sc.

5. Performance scores

First, we have introduced "user defined weights" $\omega_{\Psi=1,\ldots,21}$ (any positive numbers) for the contributions of various fields. For the results below, we have set:

$\omega_{T\max} = 1,\ \omega_{T\min} = 1,\ \omega_{\Pr} = 1,\ \omega_{CAPE} = 0,\ \omega_{CIN} = 0,\ \omega_{TCWV} = 1,\ \omega_{WS1} = 0.33,\ \omega_{WS2} = 0.33,\ \omega_{WS3} = 0.33,$

$\omega_{T500} = 0.33,\ \omega_{T700} = 0.33,\ \omega_{T850} = 0.33,\ \omega_{RH500} = 0.33,\ \omega_{RH700} = 0.33,\ \omega_{RH850} = 0.33,\ \omega_{U500} = 0.2,$

$\omega_{U700} = 0.2,\ \omega_{U850} = 0.2,\ \omega_{V500} = 0.2,\ \omega_{V700} = 0.2\ ,\ \omega_{V850} = 0.2$.

The fields CAPE and CIN (both observed and simulated) are generally noisy. Moreover, in the soundings data the number of reports (levels) is usually low, making the calculation of CAPE and CIN highly uncertain. Therefore at this work we set zero weights for these fields: $\omega_{CAPE} = 0,\ \omega_{CIN} = 0$.

We have developed 2 optional performance scores for CALMO-stages 2 and 3, which are described below.

5.1 RMSE-type score

In contrast to CALMO-stage1, in stages 2 and 3 the number of regions (or grid-points) for comparing the model with observations very much depends on the forecasted field Ψ (and slightly depends on the month). Therefore the score for parameters combination p takes more complicated form:

$$S_p = \left\{ \frac{1}{12\sum_{\Psi=1}^{21}\omega_\Psi} \sum_{\Psi=1}^{21}\omega_\Psi \sum_{mon=1}^{12} \frac{1}{W_{\Psi,mon} N_{\Psi days,mon} N_{\Psi regs,mon}} \sum_{\Psi regs} \left[\frac{\sum_{\Psi days} \left(F_{\Psi,p,d,r,mon} - O_{\Psi,d,r,mon} \right)^2}{\sigma^2_{\Psi,r,mon}} \right] \right\}^{1/2} \qquad (2)$$

Where the fields Ψ_{1-21} where defined at Section 4.2.3 above.

5.1.1 "Observations variability" is defined per month

The quality of COSMO forecast strongly depends on the region and the season. For example, the forecast with Tmax error of 5K in the Alps at winter may be actually better than with error of 3K in the Swiss Plateau at summer. Therefore one needs to normalize the forecast errors by a value which reflects the forecast complexity for a given day and region. As at CALMO-stage 1, we normalize the forecast field Ψ errors by the

observations standard deviation $\sigma_{\Psi,r,mon}$ at a given region (or grid-point) r over a period of a month $N_{\Psi days,mon} \approx 30$ (the period should not be too short in order to contain large enough sample, but not too long in order to represent the variability of a specific season):

$$\sigma_{\Psi,r,mon} = \sqrt{\frac{1}{N_{\Psi days,mon}} \sum_{\Psi days} \left(O_{\Psi,d,r,mon} - \overline{O}_{\Psi,d,r,mon} \right)^2} \tag{3}$$

5.1.2 Normalization weights

Normalization weights are defined to set equal contributions for the various fields ($N_p = 10000$ - number of parameters combinations):

$$W_{\Psi,mon} = \frac{1}{N_p} \sum_{p=1}^{N_p} \frac{1}{N_{\Psi days,mon} N_{\Psi regs,mon}} \sum_{\Psi regs} \left[\frac{\sum_{\Psi days} \left(F_{\Psi,p,d,r,mon} - O_{\Psi,d,r,mon} \right)^2}{\sigma^2_{\Psi,r,mon}} \right] \tag{4}$$

5.2 COSI-type score

The "COSMO Index" (COSI) was developed by Ulrich Damrath (DWD) and is defined in [5].

We have adapted the score for CALMO use as following:

$$S_p = \frac{1}{12 \sum_{\Psi=1}^{18} \omega_\Psi} \left\{ \sum_{\Psi \neq 3} \omega_\Psi \sum_{mon=1}^{12} \left[1 - \frac{\sum_{\Psi regs} \sum_{\Psi days} \left(F_{\Psi,p,d,r,mon} - O_{\Psi,d,r,mon} \right)^2}{\sum_{\Psi regs} \sum_{\Psi days} \left(O_{\Psi,d-1,r,mon} - O_{\Psi,d,r,mon} \right)^2} \right] + \omega_3 \frac{\sum_{mon=1}^{12} \sum_{\Psi regs} \sum_{\Psi thr} ETS_{p,r,mon,thr}}{N_{\Psi days,mon} N_{\Psi regs,mon}} \right\} \tag{5}$$

where -1/3<ETS<1 (1 is the best) is the threshold dependent (we have chosen region averaged precipitation amounts thresholds of 0.1,1,3,7.5,10mm per 24h) precipitation score:

$$ETS_{p,r,mon,thresh} = \frac{H - \frac{(H+F)(H+M)}{N_{\Psi regs,mon}}}{H + M + F - \frac{(H+F)(H+M)}{N_{\Psi regs,mon}}} \tag{6}$$

where:

H - Number of hits (i.e. both the model and the observations where above the given threshold);

F - Number of "false alarms";

M - Number of misses.

6. Convergence to the optimal parameters combination

6.1 Method

After the Meta-Model is constructed we divide the parameters space into high number of points (parameters combinations), and calculate the score (see section 5) for each of the points in order to find the optimal one. In CALMO-stage 1, we have tuned 3 parameters, dividing the parameters space into 10000 points, i.e. roughly 21 bins for each of the parameters. In CALMO-stage 2, for example, the number of calibrated parameters is N=6,

yielding huge number (about $21^6 \approx 10^8$) of points to be evaluated in order to find the optimal one. However, for computer time reasons it is not possible.

Recently we have developed a method to overcome that problem and converge to the optimal parameters combination. At first iteration we sample 1000 points only and reveal the optimal regions in our N dimensional parameters space (according the uncertainty of the optimal 100 combinations). An example of the convergence after first iteration is presented at figure 4 below.

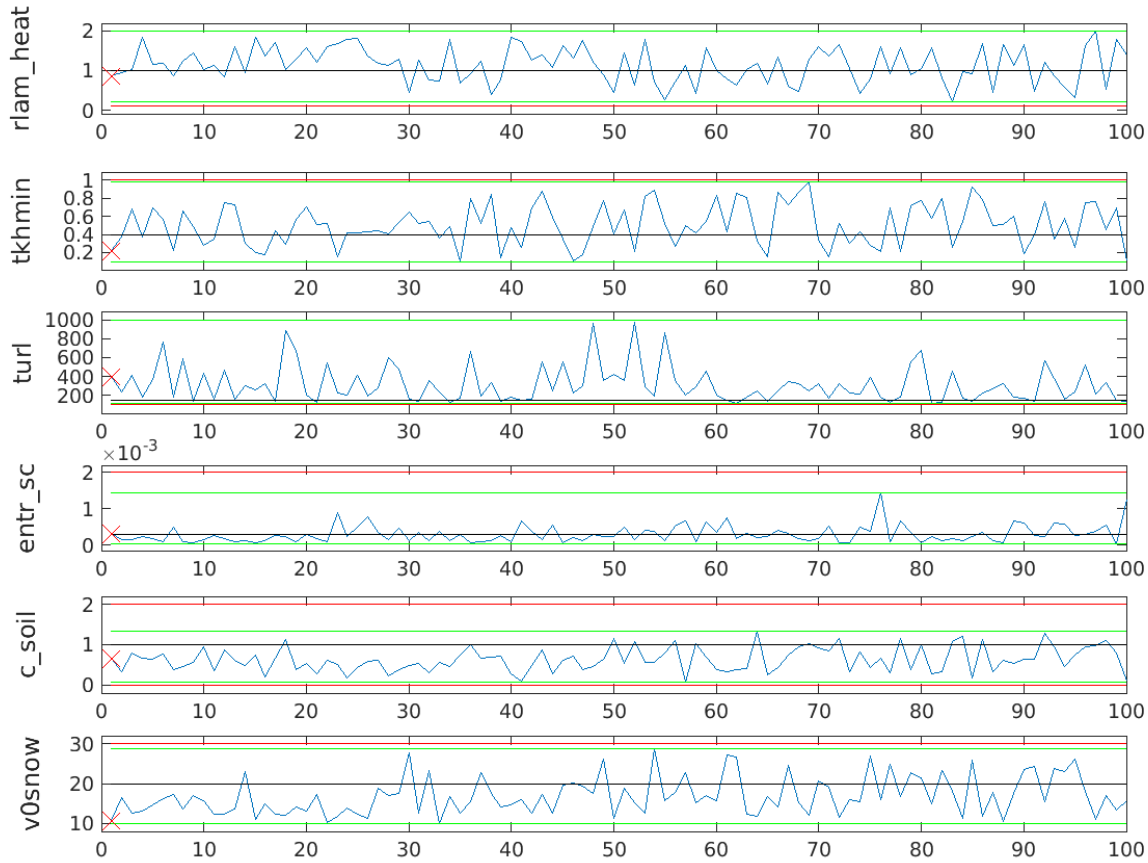

Figure 4: Example of convergence after first iteration. Each panel shows (in blue) the optimal 100 parameters values (in sorted order) among 1000 sampled combinations. The red lines represent the allowed ranges for each parameter, the green lines represent the uncertainty for each parameter after first iteration (following the optimal 100 values). Red crosses represent the best parameters combination after first iteration.

At second iteration we sample those regions (between the green lines at figure 4) by additional 1000 points, and reveal new, smaller, optimal regions (again according the uncertainty of the new optimal 100 combinations). We continue with these iterations until the solution converges to the optimal parameters combination. An example of the converged stage is presented at figure 5 below.

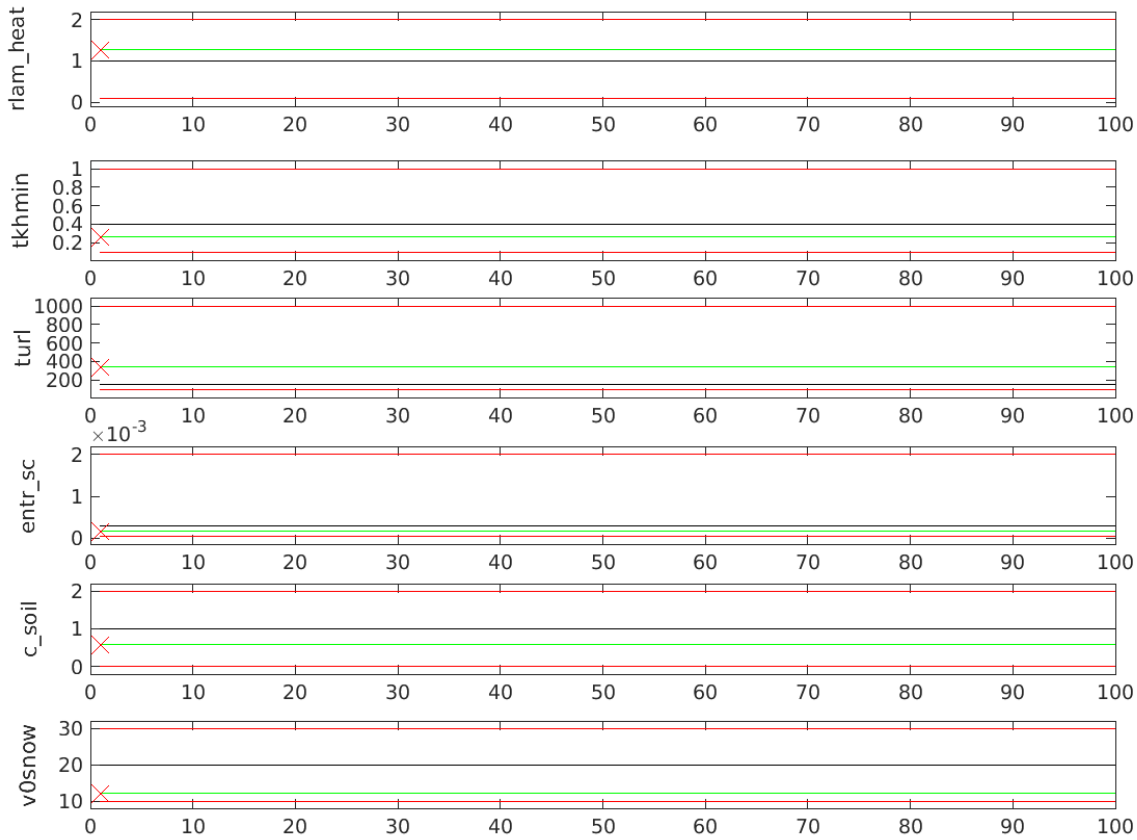

Figure 5: Example of convergence after last (35th) iteration.

6.2 Uncertainty of the optimal parameters combination

A question arises – what is the uncertainty of the optimal parameters combination? In other words, what is the score sensitivity when slightly changing the parameters values with respect to the optimal parameters combination? To answer this question, we have followed the procedure described above, and determined the iteration at which the score reaches 90% of the optimal combination score. We define the parameters uncertainty (between the green lines at figure 4, for example) at that iteration as the uncertainty of the optimal parameters combination.

7. TERRA stand-alone

As part of CALMO-stages 2 and 3, among other parameters it was planned to tune also soil-scheme (TERRA) parameters (for example the hydraulic soil conductivity). In contrast to the regular COSMO parameters, the change in Terra parameters affects the COSMO forecasts with a significant delay (up to several years) via slow adaptation of the soil temperature and moisture profiles. Therefore, in order to tune TERRA parameters for specific year, one has to make the parameter changes several years earlier, and run the COSMO model in a cycle, slowly adapting the soil profiles to the parameter change. Moreover, errors in soil profiles caused by interpolation of soil fields from a coarse model disappear slowly, also on the scale of several years. Therefore, in order to obtain appropriate initial conditions in the soil (without interpolation errors), one again has to make the interpolation of soil fields several years earlier, and run the COSMO model in a cycle, slowly

"forgetting" the interpolation errors. However, performing several years pre-run of the COSMO model (in a cycle mode) is computationally expensive. Instead, it was decided to use the TERRA "stand-alone" (TSA) program driven by COSMO atmospheric analyses (from MeteoSwiss archive). The method was to initialize soil profiles from a coarse model interpolation, change the parameters of TSA (if needed), and run it for several years (prior to the tested year). Then, the obtained soil profiles were installed as initial soil conditions for the COSMO model runs (for the tested year). With we have run TSA for 3 years (2010-2012) with resolutions of 2.2 and 1.1 km, and prepared the soil initial conditions for the COSMO runs at 2013.

8. CALMO stage-2

8.1 Validation of CALMO stage-2 Meta-Model using arbitrary test simulation

In order to validate the Meta-Model quality, additional test simulation was performed for an arbitrary parameters combination [rlam_heat=1.24, tkmmin=0.233, tur_len=363.9, entr_sc=0.000267, c_soil=0.492 and v0snow=12.1], which was not used for building the Meta-Model. That allows comparing the Meta-Model prediction for this specific parameters combination with the real simulation results, over the entire 2013. Figs. 6-9 show scatter plots for maximum daily 2m-temperature (Tmax), minimum daily 2m-temperature (Tmin), 24h accumulated precipitation (Pr), and column integrated water vapor (TCWC), respectively. The y-axes show the Meta-Model estimation with respect to the reference (simulation with default parameters values), while the x-axes show the COSMO simulation results with respect to the reference. For Tmax and Tmin, each point represents grid-point comparison (according method IV as explained in section 8.2 below). For Pr each point represents regions averages. For TCWC each point represents a profile in one of the radiosondes locations.

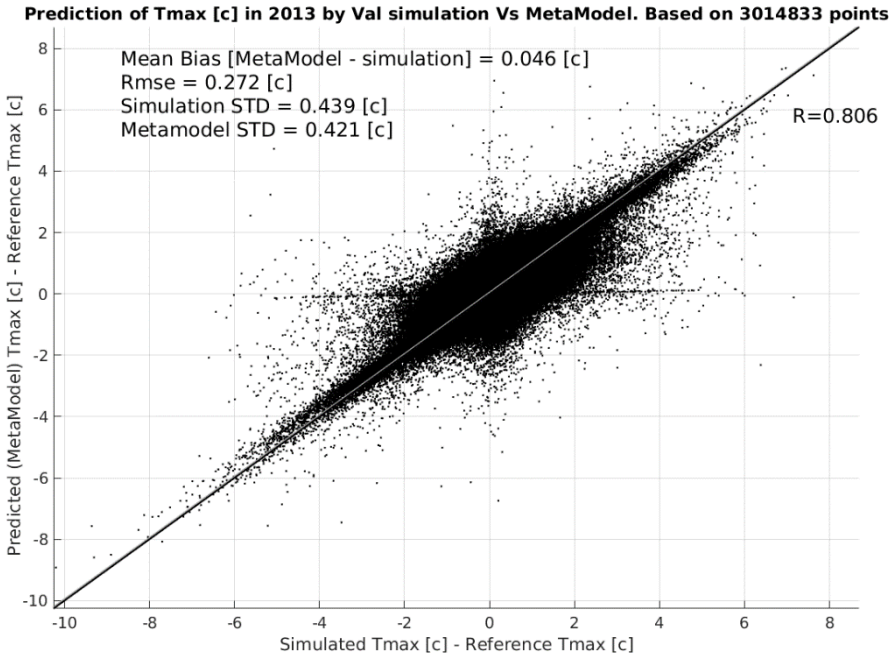

Figure 6: Tmax Meta-Model prediction for the tested parameter combination, vs COSMO simulation results during the year 2013. X axis presents the simulated Tmax minus the reference simulation. Y axis presents the Meta-Model Tmax minus the reference simulation.

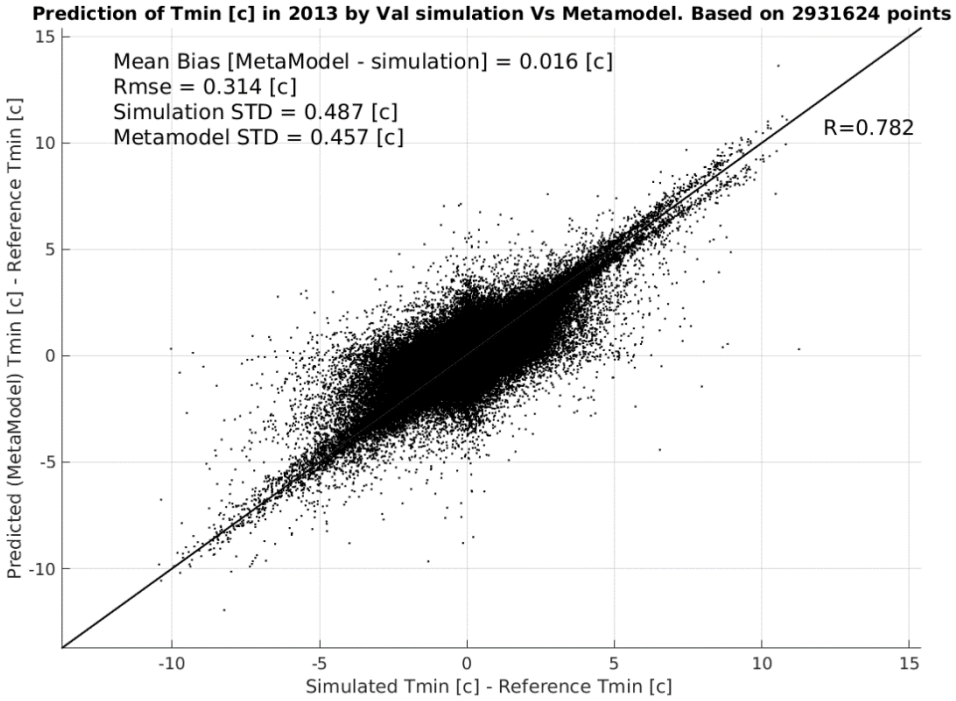

Figure 7: Tmin Meta-Model prediction for the tested parameter combination, vs COSMO simulation results during the year 2013. X axis presents the simulated Tmin minus the reference simulation. Y axis presents the Meta-Model Tmin minus the reference simulation.

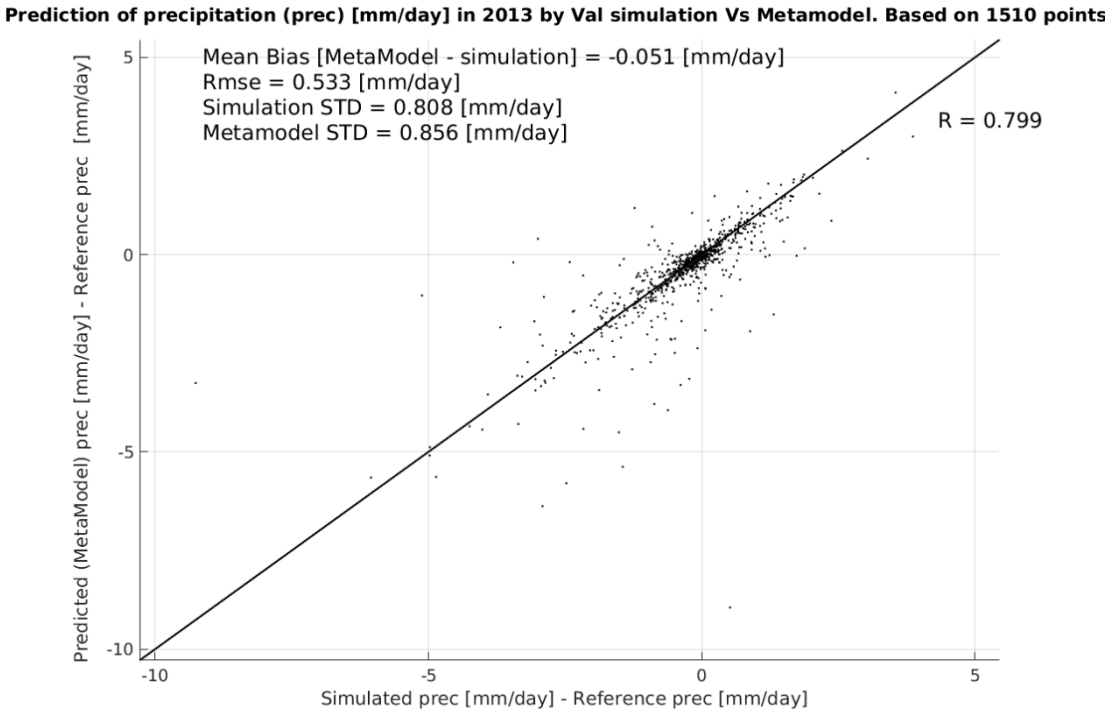

Figure 8: Pr Meta-Model prediction for the tested parameter combination, vs COSMO simulation results during the year 2013. X axis presents the simulated Pr minus the reference simulation. Y axis presents the Meta-Model Pr minus the reference simulation.

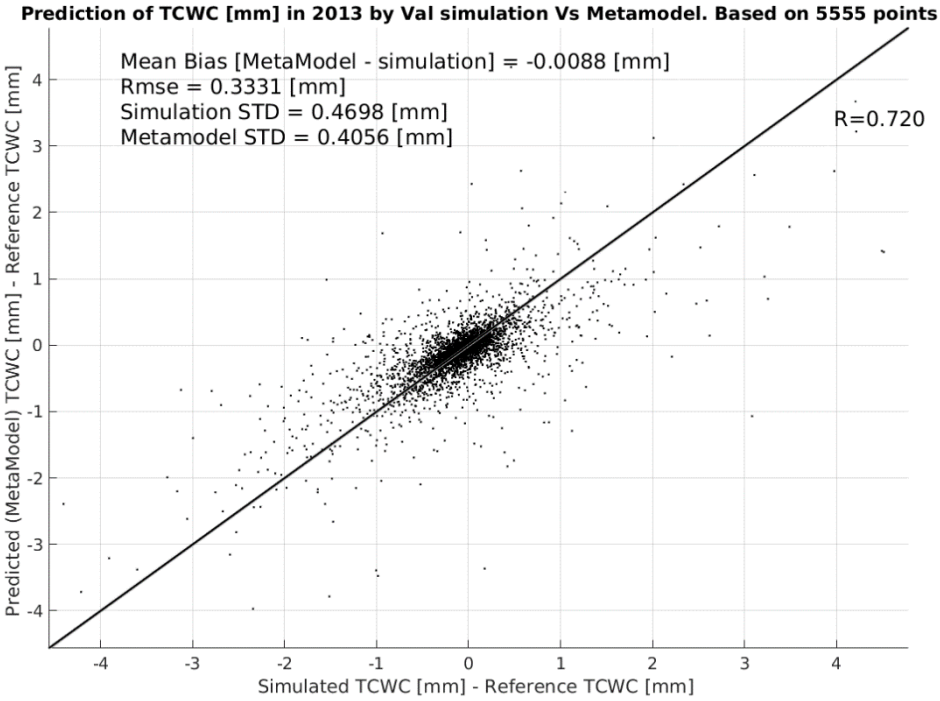

Figure 9: TCWC Meta-Model prediction for the tested parameter combination, vs COSMO simulation results during the year 2013. X axis presents the simulated TCWC minus the reference simulation. Y axis presents the Meta-Model TCWC minus the reference simulation.

Important to mention, that only *one* point (one parameters combination) in 6-dimensional parameters space was analyzed, so that correlations presented in figures 6-9 are not necessarily represent other possible parameters combinations.

However, for the tested parameter combination, the correlations R between the COSMO forecasts and the Meta-Model estimations are generally high. Consequently, the overall method seems to prove itself: one can use the Meta-Model to reproduce COSMO forecasts for various parameters combinations.

8.2 Calibration results for entire year 2013

The calibration was performed using 4 different methods:

I. **Averaging Tmax and Tmin over regions (see Section 4.2.1 above), using RMSE-type score;**
II. **Not averaging Tmax and Tmin over regions, using RMSE-type score;**
III. **Averaging Tmax and Tmin over regions, using or the COSI score;**
IV. **Not averaging Tmax and Tmin over regions, using the COSI score.**

We have used the Meta-Model to calculate the overall score S_p (either RMSE-type (eq. 2) or COSI (eq. 5)) for any given parameters combination. Figures 10-13 present the contours of S_p deviation, i.e. $S_p - \overline{S_p}$, for pairwise parameters combinations only, for the methods I,II,III,IV, respectively. Note that for RMSE-type score lower $S_p - \overline{S_p}$ means "better" parameters combination, while for COSI score, higher $S_p - \overline{S_p}$ is better. One can see that the optimal parameters regions are similar, regardless the method we used.

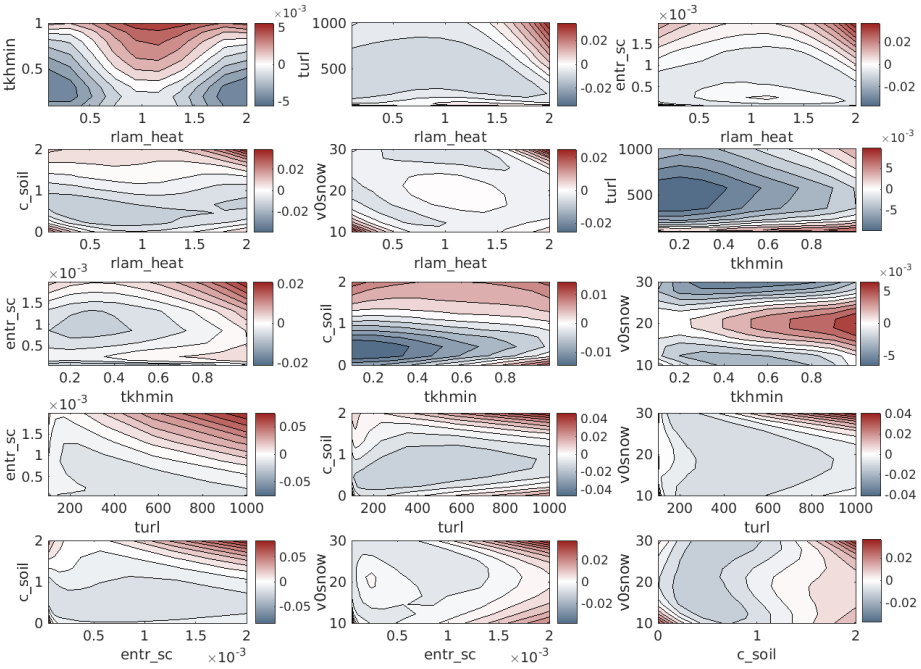

Fig. 10: Method I - Contours of score deviation $S_p - \overline{S_p}$ (eq. 2), for pairwise parameters combinations. Lower $S_p - \overline{S_p}$ areas represent "better" parameters combinations.

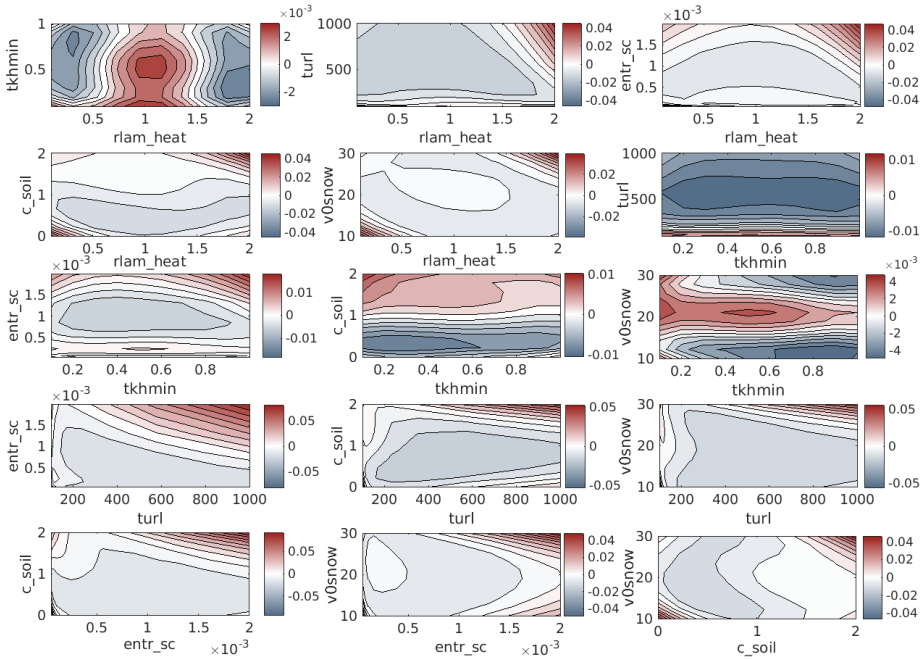

Fig. 11: Method II - Contours of score deviation $S_p - \overline{S_p}$ (eq. 2), for pairwise parameters combinations. Lower $S_p - \overline{S_p}$ areas represent "better" parameters combinations.

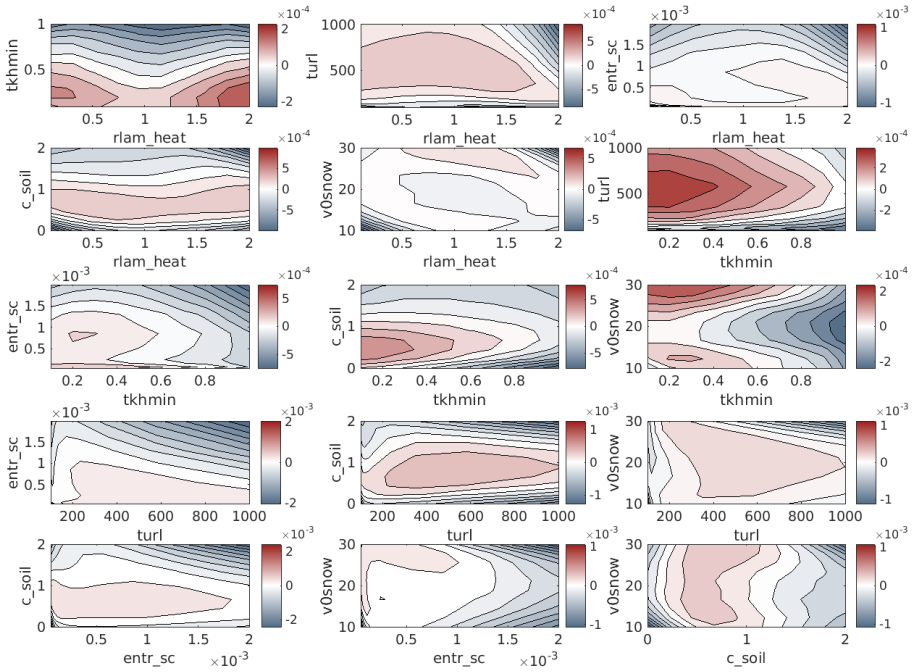

Fig. 12: Method III - Contours of score deviation $S_p - \overline{S_p}$ (eq. 5), for pairwise parameters combinations. Higher $S_p - \overline{S_p}$ areas represent "better" parameters combinations.

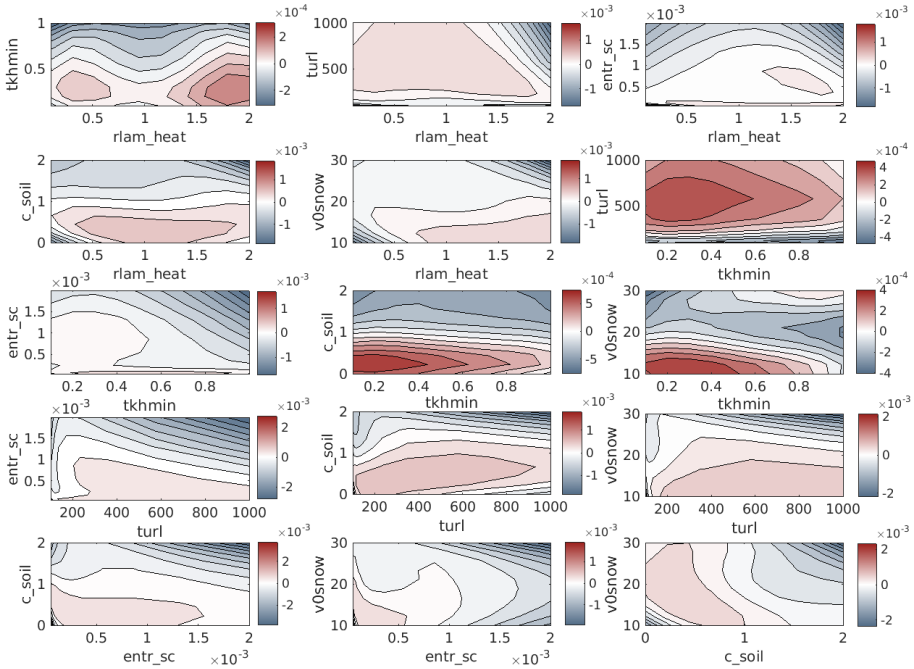

Fig. 13: Method IV - Contours of score deviation $S_p - \overline{S_p}$ (eq. 5), for pairwise parameters combinations. Higher $S_p - \overline{S_p}$ areas represent "better" parameters combinations.

Figures 14 and 15 present S_p scores distributions after first and last iterations, respectively (see Section 6 above), together with the score of the reference (REF) simulation, for methods I-IV.

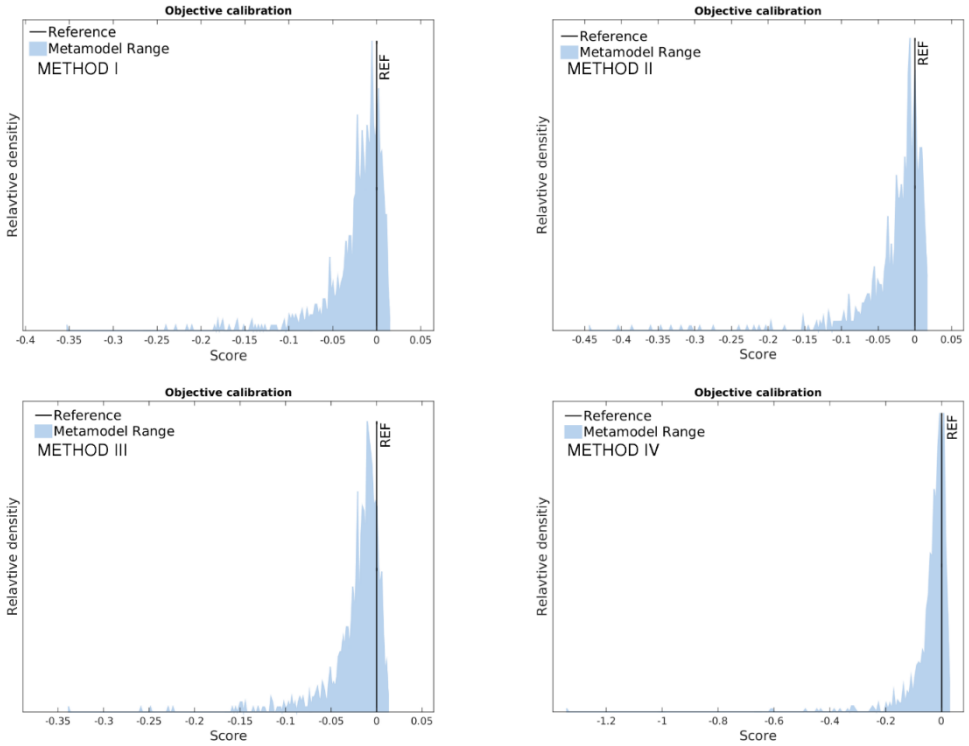

Fig. 14: S_p *scores distributions after* ***first*** *iteration, together with the score of the reference (REF) simulation, for methods I-IV. For convenience, the distributions are presented as function of* $\tilde{S}_p = 1 - S_p / S_{p,REF}$ *for methods I,II and as function of* $\tilde{S}_p = S_p / S_{p,REF} - 1$ *for methods III,IV. Therefore higher* $\tilde{S}_p > 0$ *means better score with respect to the REF simulation.*

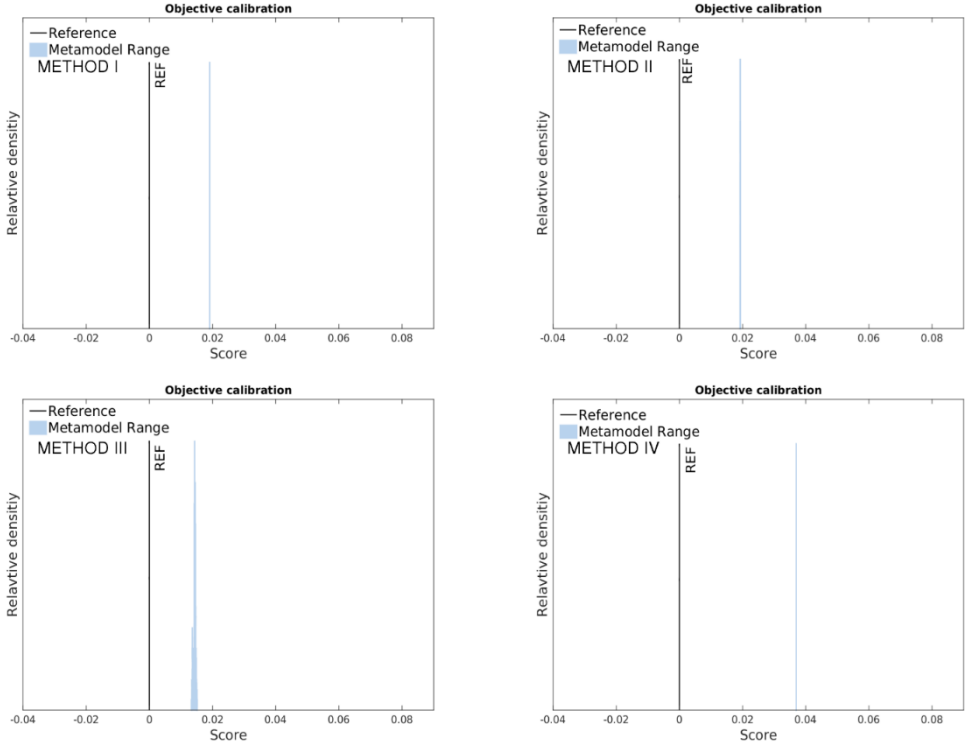

Fig. 15: S_p *scores distributions after* ***last*** *iteration, together with the score of the reference (REF) simulation, for methods I-IV. For convenience, the distributions are presented as function of* $\tilde{S}_p = 1 - S_p / S_{p,REF}$ *for*

methods I,II and as function of $\widetilde{S}_p = S_p / S_{p,REF} - 1$ *for methods III,IV. Therefore higher* $\widetilde{S}_p > 0$ *means better score with respect to the REF simulation.*

Table 3 presents the optimal parameters combinations, as well as their uncertainties (see section 6.2) for the methods I-IV described above:

Method	rlam_heat	Tkhmin	tur_len	entr_sc (10^{-4})	c_soil	v0snow
I	0.724 **0.835** 0.942 [-5.8% +5.6%]	0.179 **0.229** 0.282 [-5.6% +5.9%]	268.0 **309.3** 347.3 [-4.6% +4.2%]	0.643 **0.731** 0.866 [-4.5% +6.9%]	0.623 **0.681** 0.733 [-2.9% +2.6%]	18.7 **19.9** 21.2 [-6.0% +6.5%]
II	0.836 **0.964** 1.077 [-6.7% +5.9%]	0.316 **0.372** 0.442 [-6.3% +7.8%]	390.2 **437.4** 503.9 [-5.2% +7.4%]	0.796 **0.798** 0.938 [-0.01% +0.7%]	0.679 **0.725** 0.760 [-2.3% 1.8%]	17.1 **18.5** 19.3 [-7.0% +4.0%]
III *	**1.009**	**0.155**	**422.3**	**0.832**	**0.735**	**18.8**
IV	1.149 **1.273** 1.390 [-6.5% +6.2%]	0.205 **0.266** 0.351 [-6.8% +9.4%]	294.6 **346.5** 409.9 [-5.8% +7.0%]	1.261 **1.607** 2.104 [-1.8% +2.5%]	0.515 **0.588** 0.664 [-3.7% +3.8%]	11.6 **12.3** 13.3 [-3.5% +5.0%]

Table 3: Optimal parameters combinations and their uncertainties for methods I-IV. *For method III there was no complete convergence to the optimal parameters combination, so the uncertainties are not presented.

Assuming method IV as the most reasonable, the final optimal parameters combination with its uncertainty is:

- rlam_heat=**1.273** instead of the default 1.0. Uncertainty: [1.149 1.390];
- tkhmin=**0.266** instead of the default 0.4; Uncertainty: [0.205 0.351];
- tur_len=**346.5** instead of the default 150; Uncertainty: [294.6 409.9];
- entr_sc=**0.0001607** instead of the default 0.003; Uncertainty: [0.0001261 0.0002104];
- c_soil=**0.588** instead of the default 1.0; Uncertainty: [0.515 0.664];
- v0snow=**12.3** instead of the default 20; Uncertainty: [11.6 13.3].

8.3 Calibration results - seasonal dependence

At this section we analyzed the seasonal dependence of the optimal parameters combination during 2013. For that purpose we have performed parameters calibration for summer 2013 (Jul, Aug and Sep) and winter 2013 (Jan, Feb and Mar), separately. The results for the optimal parameters combinations, as well as their uncertainties are presented in table 4.

Cases	rlam_heat	Tkhmin	tur_len	entr_sc (10^{-4})	c_soil	v0snow
Entire 2013	1.149 **1.273** 1.390 [-6.5% +6.2%]	0.205 **0.266** 0.351 [-6.8% +9.4%]	294.6 **346.5** 409.9 [-5.8% +7.0%]	1.261 **1.607** 2.104 [-1.8% +2.5%]	0.515 **0.588** 0.664 [-3.7% +3.8%]	11.6 **12.3** 13.3 [-3.5% +5.0%]
Summer 2013	0.954 **1.071** 1.164 [-6.2% +4.9%]	0.186 **0.221** 0.270 [-3.9% +5.4%]	352.2 **357**.5 398.9 [-0.6% +4.6%]	4.439 **4.890** 5.495 [-2.3% +3.1%]	1.090 **1.150** 1.205 [-3.0% +2.8%]	20.2 **21.2** 22.3 [-5.5% +5.0%]
Winter 2013	0.982 **1.112** 1.232 [-6.8% +6.3%]	0.791 **0.891** 0.929 [-11.1% +4.2%]	109.8 **117.2** 127.9 [-0.8% +1.2%]	1.387 **1.714** 2.076 [-1.7% +1.9%]	0 **0.041** 0.134 [-2.1% +4.7%]	29.2 **30.0** 30.0 [-4.0% +0.0%]

Table 4: CALMO stage-2 optimal parameters combinations, as well as their uncertainties for method IV for the following cases: all months in 2013, summer 2013 (Jul, Aug and Sep) and winter 2013 (Jan, Feb and Mar).

One can see significant differences at the optimal parameters combinations for summer and winter. This fact reflects different biases of atmospheric fields between the seasons (see for example the first CALMO progress report [2]). However, figuring out the atmospheric fields "responsible" for this behavior is beyond the scope of this report.

9. CALMO stage-3

9.1 Calibration results for January 2013

CALMO-stage-3 calibration was performed using method IV, i.e. **"not averaging Tmax and Tmin over regions, using the COSI score"**. As can be seen in table 1, at that stage we have tuned 5 parameters: tkhmin, tur_len, entr_sc, csoil, crsmin, for the period of ~ 1/1/2013 – 1/2/2013.

As at CALMO-stage 2, we have used the Meta-Model to calculate the overall COSI score S_p (eq. 5) for any given parameters combination. Before presenting the results for the optimal 5-parameters combination, we first investigate the importance of each of the 5 parameters. This is done by performing the calibration several times, each time excluding one of the parameters. Figures 16-20 present the contours of S_p deviation, i.e. $S_p - \overline{S_p}$ (higher $S_p - \overline{S_p}$ is better), for pairwise parameters combinations only, in the following order:

- Case 1: Tuning parameters tkhmin, tur_len, entr_sc, csoil (excluding crsmin) – see figure 16;
- Case 2: Tuning parameters tkhmin, tur_len, entr_sc, crsmin (excluding csoil) – see figure 17;
- Case 3: Tuning parameters tkhmin, tur_len, csoil, crsmin (excluding entr_sc) – see figure 18;
- Case 4: Tuning parameters tkhmin, entr_sc, csoil, crsmin (excluding tur_len) – see figure 19;
- Case 5: Tuning parameters tur_len, entr_sc, csoil, crsmin (excluding tkhmin) – see figure 20.

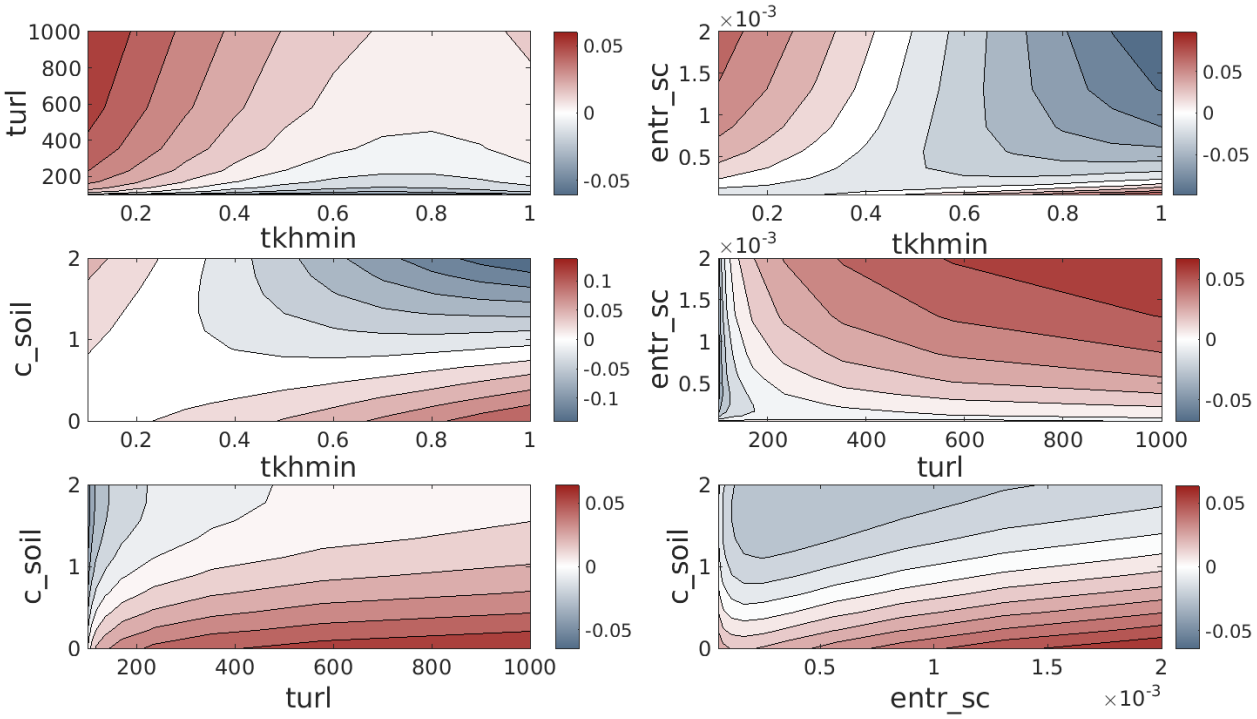

Fig. 16: Contours of score deviation $S_p - \overline{S_p}$ for method IV (eq. 5), for pairwise parameters combinations. Higher $S_p - \overline{S_p}$ areas represent "better" parameters combinations. The tuning is performed for parameters tkhmin, tur_len, entr_sc, csoil (excluding crsmin) – case 1. Period: 1/1/2013-6/2/2013.

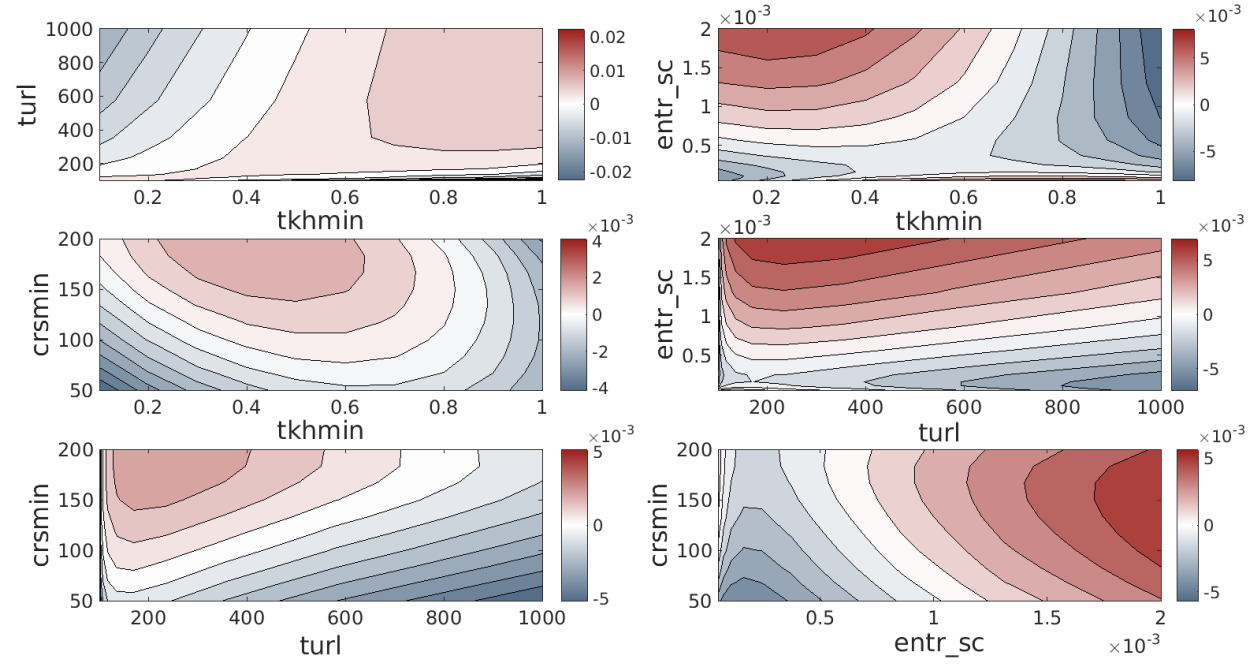

Fig. 17: Contours of score deviation $S_p - \overline{S_p}$ for method IV (eq. 5), for pairwise parameters combinations. Higher $S_p - \overline{S_p}$ areas represent "better" parameters combinations. The tuning is performed for parameters tkhmin, tur_len, entr_sc, crsmin (excluding csoil) – case 2. Period: 1/1/2013-1/2/2013.

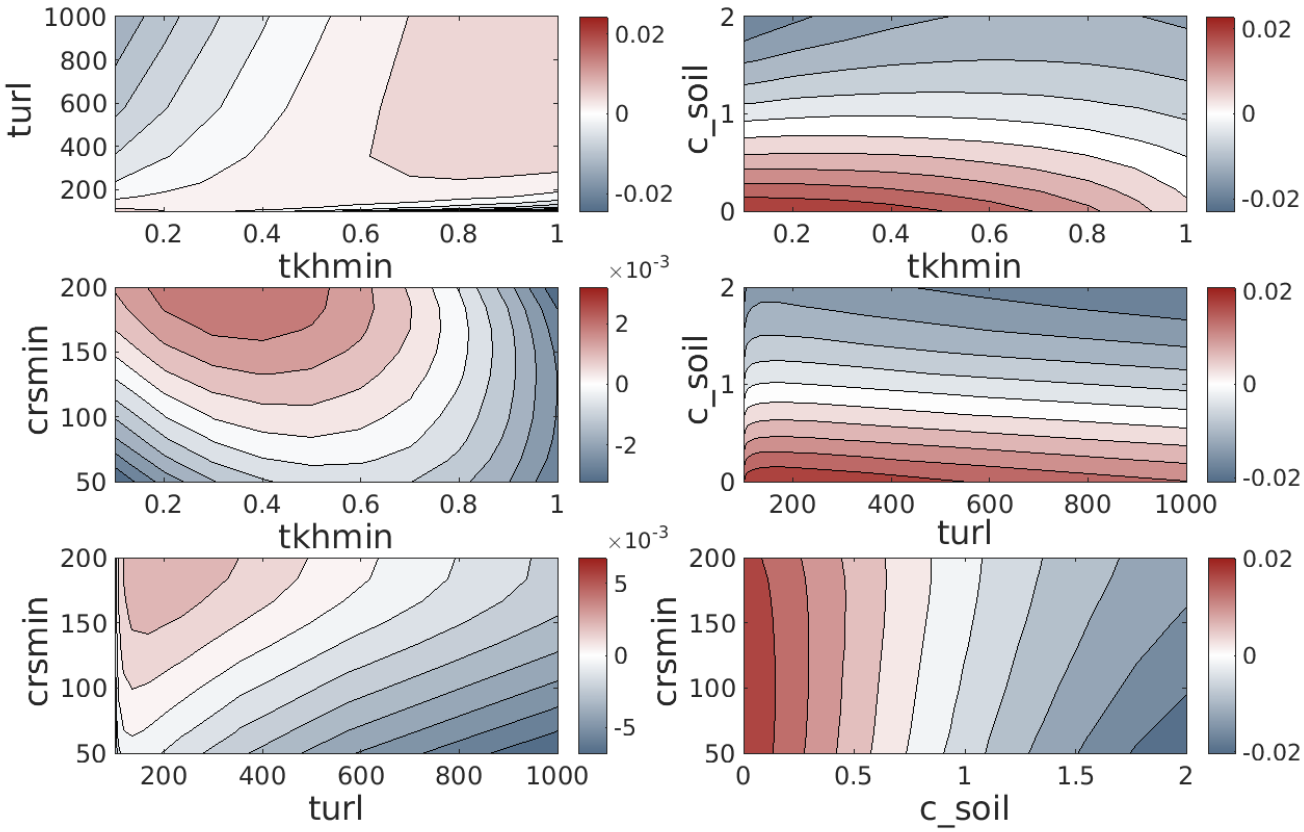

Fig. 18: Contours of score deviation $S_p - \overline{S_p}$ for method IV (eq. 5), for pairwise parameters combinations. Higher $S_p - \overline{S_p}$ areas represent "better" parameters combinations. The tuning is performed for parameters tkhmin, tur_len, csoil , crsmin (excluding entr_sc) – case 3. Period: 1/1/2013-1/2/2013.

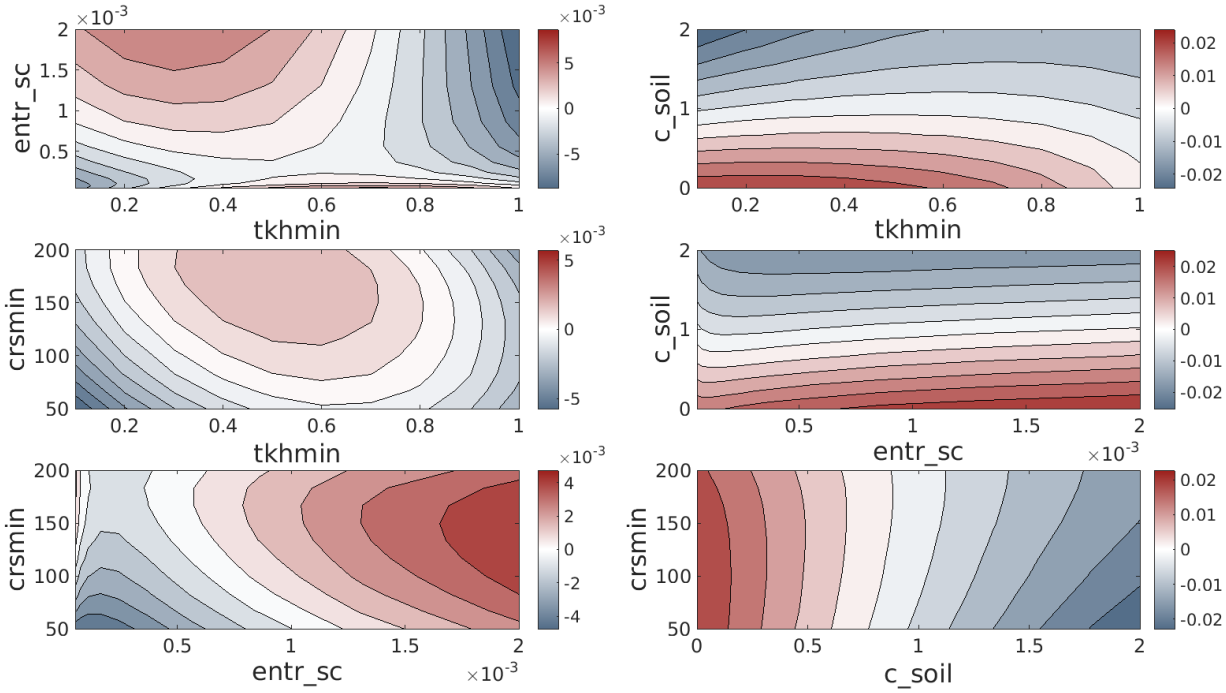

Fig. 19: Contours of score deviation $S_p - \overline{S_p}$ for method IV (eq. 5), for pairwise parameters combinations. Higher $S_p - \overline{S_p}$ areas represent "better" parameters combinations. The tuning is performed for parameters tkhmin, entr_sc, csoil , crsmin (excluding tur_len) – case 4. Period: 1/1/2013-3/2/2013.

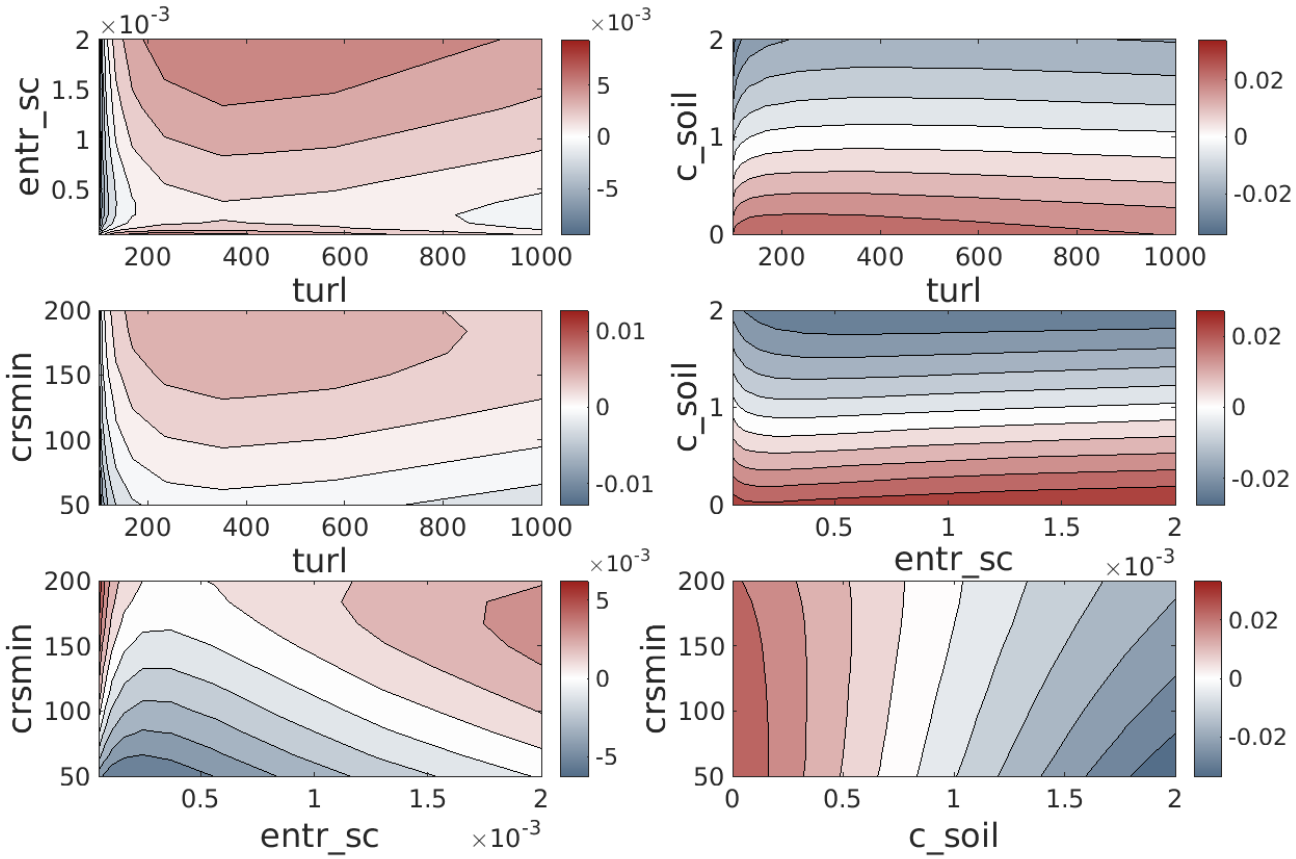

Fig. 20: Contours of score deviation $S_p - \overline{S_p}$ for method IV (eq. 5), for pairwise parameters combinations. Higher $S_p - \overline{S_p}$ areas represent "better" parameters combinations. The tuning is performed for parameters tur_len, entr_sc, csoil , crsmin (excluding tkhmin) – case 5. Period: 1/1/2013-1/2/2013.

As can be seen from figures 16-20, the optimal and worst areas in parameters space differ between the 5 cases. This can be explained by an importance of parameters interactions (fourth term in eq. (1)) with respect the first order parameters variation (third term in eq. (1)). However, the main reason for such behavior can be too small sample (for example, only 48% of the regions were rainy during the 32 days period).

Following the analysis above, we have performed the calibration taking into account all the 5 parameters tkhmin, tur_len, entr_sc, csoil, crsmin. Figure 21 presents the contours of S_p deviation, i.e. $S_p - \overline{S_p}$ (higher $S_p - \overline{S_p}$ is better), for pairwise parameters combinations only.

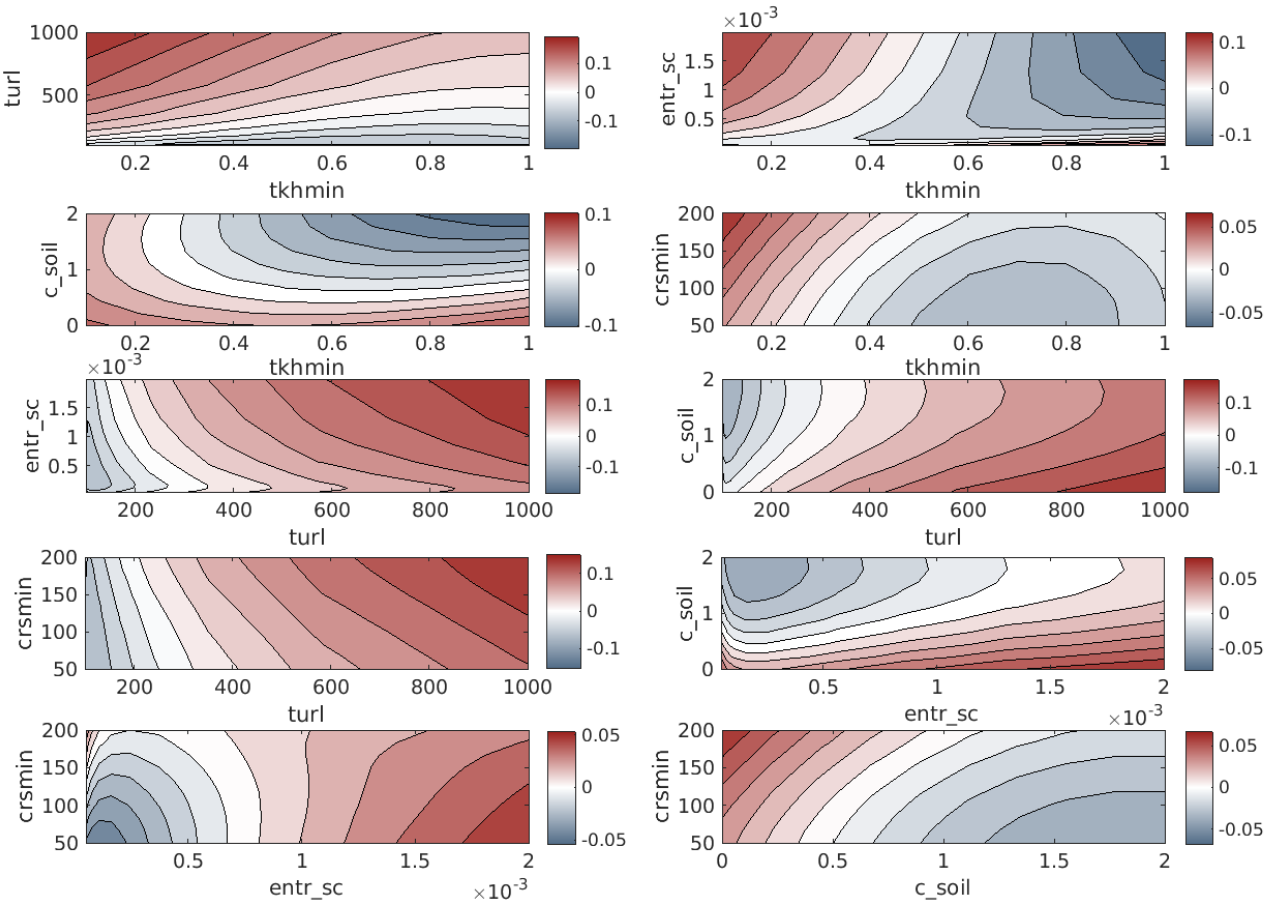

Fig. 21: Contours of score deviation $S_p - \overline{S_p}$ for method IV (eq. 5), for pairwise parameters combinations. Higher $S_p - \overline{S_p}$ areas represent "better" parameters combinations. The tuning is performed for all the 5 parameters tkhmin, tur_len, entr_sc, csoil , crsmin. Period: 1/1/2013-1/2/2013.

Figure 22 presents S_p scores distributions after first (left panel) and last (right panel) iterations, together with the score of the reference (REF) simulation.

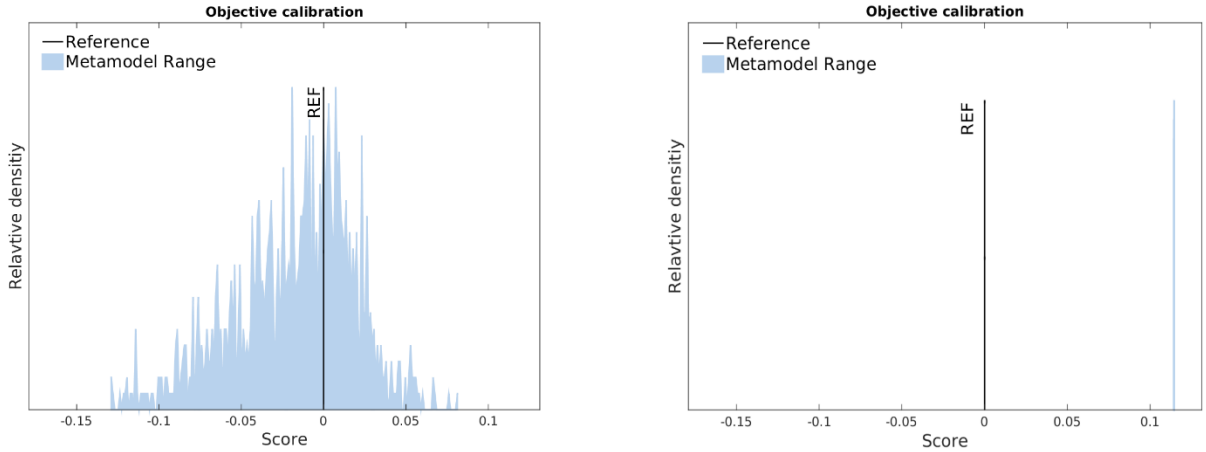

*Fig. 22: S_p scores distributions after **first** iteration (left) and **last** iteration (right), together with the scores of the reference (REF) simulation. For convenience, the distributions are presented as function of $\tilde{S}_p = S_p / S_{p,REF} - 1$. Higher $\tilde{S}_p > 0$ means better score with respect to the REF simulation.*

Table 5 presents the optimal parameters combinations, as well as their uncertainties (see section 6.2), when calibrating 4 parameters (eliminating one parameter each time according cases 1-5 described above) and for the full calibration analysis, tuning all the 5 parameters:

case	tkhmin	tur_len	entr_sc (10^{-4})	c_soil	Crsmin
1	0.986 **1.000** 1.000 [0.0% +1.6%]	100.0 **100.0** 101.0 [-0% +0.1%]	18.0 **20.0** 20.0 [1.0% +0.0%]	1.957 **2.000** 2.000 [2.0% +0.0%]	----------
2	0.988 **1.000** 1.000 [-1.3% +0%]	100.0 **100.0** 100.2 [-0% +1.9%]	7.401 **8.985** 9.012 [-8.1% +0.1%]	----------	150.1 **171.2** 171.6 [-8.4% +0.2%]
3**	**0.101**	**815.2**	----------	**1.793**	**60.0**
4	0.100 **0.100** 0.111 [-0.0% +1.2%]	----------	0.702 **0.704** 0.904 [-0.01% +1.0%]	1.978 **2.000** 2.000 [-1.1% +0%]	50.0 **50.0** 54.2 [-0.0% +1.7%]
5	----------	100.0 **100.0** 100.3 [0.0% +0.03%]	4.284 **20.0** 20.0 [-8.1% +0.0%]	1.971 **2.000** 2.000 [-1.5% +0%]	50.0 **50.0** 54.3 [-0.0% +1.7%]
all 5 param.	0.983 **1.000** 1.000 [-1.9% +0.0%]	104.3 **109.3** 117.2 [-0.6% +0.9%]	18.0 **20.0** 20.0 [-10.3% +0.0%]	1.937 **2.000** 2.000 [-3.2% +0.0%]	186.3 **200.0** 200.0 [-5.5% +0.0%]

Table 5: Optimal parameters combinations and their uncertainties. **For case 3 there was no complete convergence to the optimal parameters combination, so the uncertainties are not presented

Taking into account the uncertainties using also cases 1-5, the final optimal parameters combination (with its uncertainty) is:

- tkhmin=**1** instead of the default 0.4; Uncertainty: [0.983 1];
- tur_len=**109.3** instead of the default 150; Uncertainty: [104.3 117.2];
- entr_sc=**0.002** instead of the default 0.003; Uncertainty: [0.0018 0.002];
- c_soil=**2** instead of the default 1.0; Uncertainty: [1.937 2];
- crsmin=**200** instead of the default 150; Uncertainty: [186.3 200].

One can see, that all the five parameters get their optimal values on the edges of their allowed ranges. As mentioned before, that can be explained by short calibration period i.e. too small sample (for example, only 48% of the regions were rainy during the 32 days period). In addition, more simulations have to be performed to validate the results. In CALMO-stage-3 only one interaction simulation was performed (in addition to the minimum required), while in CALMO-stage-2 13 additional interaction simulations were performed.

9.2 CALMO-2km vs CALMO-1km optimal parameters for January 2013

At this section we addressed the question – does the optimal parameters combination changes with the model resolution, or more specifically, from CALMO-2km to CALMO-1km? As CALMO-1km results are available for January 2013 only, we have calibrated the parameters for CALMO-2km again but this time for January 2013. Table 6 presents the CALMO-2km (Stage-2) and CALMO-1km (Stage-3) optimal parameters combinations, as well as their uncertainties for January 2013.

CALMO Stage	rlam_heat	Tkhmin	tur_len	entr_sc (10^{-4})	c_soil	v0snow
Stage-2 Entire 2013	1.149 **1.273** 1.390 [-6.5% +6.2%]	0.205 **0.266** 0.351 [-6.8% +9.4%]	294.6 **346.5** 409.9 [-5.8% +7.0%]	1.261 **1.607** 2.104 [-1.8% +2.5%]	0.515 **0.588** 0.664 [-3.7% +3.8%]	11.6 **12.3** 13.3 [-3.5% +5.0%]
Stage-2 Jan 2013	0.845 **0.935** 1.002 [-4.7% +3.5%]	0.191 **0.220** 0.262 [-3.2% +4.7%]	559.8 **653.3** 753.0 [-10.4% +11.1%]	2.346 **2.764** 3.242 [-2.1% +2.5%]	0.653 **0.756** 0.841 [-5.2% +4.3%]	11.2 **11.8** 12.3 [-3.0% +2.5%]
Stage-3 Jan 2013	Default value*	0.983 **1.000** 1.000 [-1.9% +0.0%]	104.3 **109.3** 117.2 [-0.6% +0.9%]	18.0 **20.0** 20.0 [-10.3% +0.0%]	1.937 **2.000** 2.000 [-3.2% +0.0%]	Default value*

Table 6: CALMO-2km (Stage-2) and CALMO-1km (Stage-3) optimal parameters combinations, as well as their uncertainties for method IV for January 2013 (in comparison with CALMO-2km for entire 2013 in the first row).
* Note that in CALMO-1km the calibrated parameters are ' tkhmin', 'tur_len', 'entr_sc', 'c_soil' and 'crsmin'

As can be seen from table 6, the optimal parameters for CALMO-2km and CALMO-1km are completely different. We see 4 possible reasons for that:

- The results are not statistically significant due to short calibration period (one month only).
- Different parameters combinations were analyzed in stages-2 and 3. In stage-2 we have tuned the combination of rlam_heat, tkhmin, tur_len, entr_sc, c_soil and v0_snow while in stage-1 we have tuned the combination of tkhmin, tur_len, entr_sc, c_soil and crsmin. Parameters interactions might be significant, so that tuning part of parameters keeping others default is different from tuning them all together. Therefore we cannot state that the comparison performed in table 6 is a “clean experiment”.
- CALMO stage-2 and stage-1 simulations were performed in different ways. Stage-2 runs were initialized every 24 hours, while in stage-1 we have initialized the runs once at 1/1/2013 (using Terra-Standalone pre-runs), and performed single long runs keeping the “soil memory”. This “soil memory” may have a big influence on the model forecasts and the optimal parameters combinations.
- Physical reasons related to the change in resolution from 2.2km (stage-2) to 1.1km (stage-1) are probably significant, but at that stage we cannot state how much.

10. Summary

The CALMO project has several important achievements. In general, we have proved that the CALMO calibration method allows tuning parameters of NWP model. In order to adapt the calibration method of Bellprat O. et al., 2012 [1] to NWP model, we have significantly improved the Meta-Model codes. The main developments are listed below:

- We have added the option not to average 2m-temperature over regions;
- added prediction of profiles characteristics;
- added quality control to the observed and simulated fields;
- added "clever" interpolation of observed 2m-temperature fields to the model grid;
- developed the RMSE-type and COSI scores;
- developed new method for logarithmic transformation for selected parameters;
- developed a method to converge to optimal parameters combination in huge N-dimensional parameters space;
- analyzed the uncertainty of the optimal parameters combination.

These new developments, mainly in the Meta-Model, performance score and the optimization algorithm (sections 4,5,6 above and COSMO technical report [2]) highly increased the reliability of the calibration results. As part of the CALMO project, we have calibrated the COSMO model in resolutions of 7km, 2.2km and 1.1km. For 2.2km and 1.1km resolutions we have used wide verification area, which included Switzerland and north of Italy. We have validated the model performance over many meteorological fields. Moreover, for 2.2km resolution, the calibration period was very long (entire 2013) and the number of the tuned parameters was high (six). These achievements yielded highly qualitative calibration analysis, making the calibration results especially reliable.

Future study may have a lot of interesting and important directions. Using the Meta-Model, one can perform more specific calibrations:

- focusing on specific types of weather conditions (rain, extreme events, stable stratified nights, fogs, etc.);
- focusing on seasons (season-dependent parameters tuning);
- Reducing the noise of the calibration method by matching specific parameters to related fields and weather situations, and performing the tuning for these matches only.
- Analyzing the relative importance of "constrain" and "interaction" simulations for various parameters.

11. Bibliography

[1] Bellprat O. et al., 2012: Objective calibration of regional climate models, Journal of Geophysical Research, 117, D23.

[2] COSMO technical report www.cosmo-model.org/content/model/documentation/techReports/docs/techReport25.pdf

[3] Neelin J. D. et al., 2010: Considerations for parameter optimization and sensitivity in climate models, Proc. of the National Academy of Sciences of the United States of America, 107, 21349-21354

[4] Frei C., 2014: Interpolation of temperature in a mountainous region using non-linear profiles and non-euclidean distances, Int. J. Climatol., **34**, 5, 1585-1605.

[5] http://www.cosmo-model.org/content/consortium/generalMeetings/general2009/wg5-versus2/Ulrich_Damrath_Long_term_time_series_of_COSMO_EU.ppt

12. Meta-Model code

12.1 Algorithm

In order to execute the program, one has to execute main.m. First it reads the user defined definitions from namelist.m. Then it divides the calibration period to several short (~10 day) sub-periods (to save matlab memory) and executes ReadData_and_MetaModel.m for every period, to read observations and simulations data and build the meta-model regressions. After that stage is performed for all the sub-periods, it executes PostProc.m which performs the post-processing and saves the calibration results in .mat format files.

12.1.1 ReadData_and_MetaModel.m

First the observations data is read (via **read_calmo_obs.m**) into datamatrix.obsdata and datamatrix_so.obsdata structures, for near surface fields and sounding derived fields, respectively. The Swiss 2m-temperature (Tmax and Tmin) fields are interpolated to the model grid (via **build_temp_obs.m**) using several optional methods (including the one which takes into account the local observed 2m-temperature profile in the vicinity of the model grid point). Next, the 24-hours precipitation is interpolated to the model grid (via **build_rain_obs.m**). In addition, soundings data is read (via **read_sounding_obs.m**), followed by reading the Italian Tmax, Tmin and precipitation data (from netcdf files). After reading the observations, the simulations data is read (via **read_calmo_sim.m**), while the profiles data is read via **read_profiles_mod.m**. There are several types of simulations data: reference (default parameters combination) simulation is read into datamatrix.refdata structure for near surface fields (and datamatrix_so.refdata for profiles derived fields); min-max simulations (where one of the parameters gets its maximum or minimum value, while the others kept default) are read into datamatrix.moddata structure for near surface fields (and datamatrix_so.moddata for profiles derived fields); interaction simulations (where pair of parameters get their maximum or minimum value, while the others kept default) are read for near surface fields into datamatrix.moddata structure as well (and datamatrix_so.moddata for profiles derived fields); constrain simulations (where one of parameters gets some intermediate value, while the others kept default) are read into datamatrix.constrain structure for near surface fields (and datamatrix_so.constrain for profiles derived fields); validate simulation (where all the parameters get some intermediate value - needed for validating the meta-model at an arbitrary point in parameters space) is read into datamatrix.valdata structure for near surface fields (and datamatrix_so. valdata for profiles derived fields). After the observations and simulations data is read, there is an option (via "avg_T" parameter) to average part of the fields over regions (regions are defined in **regions_bmp.m**, similarly to definition of regions at Frei 2013 for Switzerland) so that the Meta-Model will be built to predict region-averaged fields, rather than the fields at every grid point. The averaging over regions is performed in **Frei_regions.m**. Next, we redefine data arrays to have the structure (fields,days,regions,simulations) (via **gpts_series.m**) and finally delete unrealistic sounding/profiles data (via **del_bad_sounding.m**). Next, the Meta-Models are created via **neelin_e.m** which uses **polyfitn.m** for performing the forecasts fits in N-dimensional parameters space. In case only precipitation is averaged over regions (avg_T=0), the Meta-Model structures metamodel_tmax, metamodel_tmin and metamodel_pr are created, while if temperature data is also averaged (avg_T=1), the Meta-Model structure metamodel_new includes all the data (for temperature and

precipitation). In addition, the profiles Meta-Models are written into the structure metamodel_so (see definitions of the Meta-Model structures in the description of **neelin_e.m** function below). Finally, the Meta-Model structures are saved in .mat format.

12.1.2 PostProc.m

First, the Meta-Model structures are loaded for all the periods. Then we calculate the weights for different fields via **weights_calc.m**, needed to equalize their contributions to the final score (see eq. 4). The weights calculation uses **neelin_p.m**. This function calculates “pseudo-forecast” using the Meta-Models, for an arbitrary parameters combination. Then a big structure "main_data" is created (and saved in .mat format), which includes all the model and observations structures, as well as the Meta-Model structures and the fields weights. Next, the function **planes.m** is called, which plot performance scores for pair-wise parameters cross sections. For that, it uses **neelin_p.m**, as well as the scores calculation **rmse_calc.m** and **cosi_calc.m**, for rmse-type and cosi-type scores, respectively. This is followed by the iterations loop, aimed to converge to the optimal parameters combination (see section 6). At each iteration, the function **lhopt.m** is called which uses **neelin_p.m** to calculate the scores distribution for a specific part of the parameters space (which is getting smaller from iteration to iteration). This distribution is plotted via **histplot.m**. In case the process converged to the optimal parameters combination, or the iterations number reached a predefined maximum ("iterations_num"), the loop is broken. Next, the "good enough" iteration is determined (parameter "iteration_goodenough") as the iteration at which the score reaches 90% of the optimal combination score. We define the parameters uncertainty (between the green lines at figure 4, for example) at that iteration as the uncertainty of the optimal parameters combination. Next **optparam.m** is called to plot the optimal parameters combination (before transforming parameters back from the log representation). In case some of the parameters were transformed to log space (see section 4.2.4), the optimal parameters combination and the uncertainties values are transformed back to the real parameters space via **log_turlen_entrsc.m**. Finally, the uncertainty ranges are saved in UB_reg.mat and LB_reg.mat, and the optimal parameters combination is saved in popt_reg.txt.

12.2 Structure

The Meta-Model code is written in Matlab and uses its Statistical Toolbox. It consists of 33 Matlab (.m) files, which are called in the following order:

1	**main.m** % main program to be called
2	**ReadData_and_MetaModel.m** % Read observations and simulations data, then fit the Meta-Models
3	**namelist.m** % read namelist
4	**sims_def.m** % choose parameters to be tuned
5	**expval_inter_combs.m**
6	**% stage A: Reading observations and simulations data:**
7	**read_calmo_obs.m** % Read observations data
8	**build_temp_obs.m** % Interpolate 2m-temperature observations grid to model grid
9	
10	**build_rain_obs.m** % Interpolate rain observations grid to model grid
11	**var_meta_calmo.m** % correct fields units

12	read_sounding_obs.m % Read sounding observations
13	get_press_levs.m
14	read_calmo_sim.m % Read simulations data
15	var_meta_calmo.m % correct fields units
16	read_profiles_mod.m % Read simulations data
17	get_press_levs.m
18	Frei_regions.m % Average part of the fields over predefined big regions
19	regions_bmp.m % Definition of regions over Switzerland and north Italy
20	gpts_series.m % Redefine data arrays to the following structure: (fields,days,regions,simulations)
21	del_bad_sounding.m % delete unrealistic sounding/profiles data
22	corr_so.m
23	**% stage B: create Meta-Models:**
24	neelin_e.m % create Meta-Models – forecasts fits in N-dimensional parameters space
25	allcomb.m
26	polyfitn.m % N-dimensional 2-nd order polynomial fit
27	**% stage C: Post-processing:**
28	PostProc.m % Post-processing: plot analysis results and calculate the optimal parameters combination
29	namelist.m % read namelist
30	sims_def.m % read parameters to be tuned
31	weights_calc.m % calculate weights for different fields, to equalize their contributions to the final score (assuming user defined weights are uniform)
32	neelin_p.m % calculate “pseudo-forecast” using the Meta-Models
33	ETS.m % calculate rain part of COSI score (in case weights are calculated for COSI score also)
34	planes.m % plot performance scores for pair-wise parameters cross sections
35	allcomb.m
36	divisor.m
37	neelin_p.m % calculate “pseudo-forecast” using the Meta-Models
38	rmse_calc.m % calculate RMSE-type score
39	cosi_calc.m % calculate COSI-type score
40	ETS.m % calculate rain part of COSI score
41	lhopt.m % calculate scores distribution as part of the iterative convergence algorithm
42	neelin_p.m % calculate “pseudo-forecast” using the Meta-Models
43	histplot.m % plot scores distribution as part of the iterative convergence algorithm
44	optparam.m % plot optimal parameters combination (before transforming parameters back from the log representation)
45	allcomb.m
46	log_turlen_entrsc.m % transforming tur_len and entr_sc parameters back from the log representation

12.3 Subroutines

The Meta-Model code includes the following subroutines:

```
function main()

% NAME
%   main
% PURPOSE
%   main program of the CALMO parameters tuning method
```

```
% NOTE
%   Set main definitions at namelist.m file !
% RUN
%   from Bash: "matlab -nodesktop -nosplash -r main"
%   from Matlab: F5 inside main.m
% INPUT
%   -
% OUTPUT
%   calibration results saved in .mat format
% AUTHORS
%   Pavel Khain (pavelkh_il@yahoo.com)
%   Itsik Carmona (carmonai@ims.gov.il)
%   Originally: Omar Bellprat (omar.bellprat@gmail.com)


function ReadData_and_MetaModel(date_min,date_max)

% NAME
%   ReadData_and_MetaModel
% PURPOSE
%   Read observations and simulations data, then fit the Meta-Models
% INPUTS
%   time period: from date_min 'dd-mmm-yyyy' to date_max 'dd-mmm-yyyy'
% OUTPUTS
%   saved (in .mat format) observations and simulations fields as well as
%   the Meta-Models coefficients


function [maindir simuldir obsdir extdir vars vars_2d avg_T vars_sound sims_opt ml
        score w_user lhacc iterations_num best_percent date_min
        date_max]=namelist()

% NAME
%   namelist
% PURPOSE
%   Namelist of the calibration analysis
% INPUTS
%   -
% OUTPUTS
%   maindir - main directory
%   simuldir - "maindir/simuldir": path to simulations files
%   obsdir - "maindir/obsdir": path to observations files
%   extdir - "maindir/extdir": path to "external data" files
%   vars - calibrated fields groups. Can be any combinations of:
%         {'t2m_max','t2m_min','pr','sound'}
%   vars_2d - calibrated 2D fields. Can be any combinations of:
%           {'t2m_max','t2m_min','pr'}
%   avg_T - region average over Precipitation only (avg_T=0), or over
%           Precipitation, Tmax and Tmin (avg_T=1)
%   vars_sound - calibrated profiles fields:
%             {'CAPE','CIN','TCWC','WSHEAR1','WSHEAR2','WSHEAR3','T850mb',
%             'T700mb','T500mb','RH850mb','RH700mb','RH500mb','U850mb',
%             'U700mb','U500mb','V850mb','V700mb','V500mb'}
%   sims_opt - Choose parameters to callibrate and the simulations to use. The
%            possible values for sims_opt and their meaning appear in sims_def.m
%            file
%   ml - Minimum number of days (during given period) for valid soundings data. If
%       less - current sounding fields are not analyzed
%   score - 'rmse' or 'cosi' for RMSE-type and COSI-type scores, respectively
%   w_user - array of user defined weights (for simlicity - from 0 to 1) for
%           calibrated fields:
%           tmax tmin pr cape cin ws1 ws2 ws3 T850mb T700mb T500mb RH850mb
%           RH700mb R500mb U850mb U700mb U500mb V850 V700mb V500mb
```

```
%  lhacc -  Number of experiments to sample parameter space at each iteration
%  iterations_num - Maximum number of iterations
%  best_percent - "winners" percent of lhacc which is used to define the
%              parameters space for the next iteration
%  date_min - beginning of calibration period
%  date_max - end of calibration period



function [paramn,paramnt,range,default,expval,valval,simval,sims_reg,sims_inter,
       sims_con,valcon,param_log,date_min,date_max]=sims_def(sims_opt)


% NAME
%  sims_def
% PURPOSE
%  Choose parameters to tune and the simulations to use
% INPUTS
%  sims_opt - defined by 5-digits number: sims_opt=ABCDE, where:
%           A - number of parameters to calibrate (1,2,3,4,5,6,...)
%           B - serial number of combination for given A
%           C - number of ADDITIONAL (to the minimum required) interaction
%              parameter simulations (interaction terms)
%           D - number of "constrain" 1D simulations (additional simulations
%              where only one parameter is changed from default)
%           E - number of parameters (among A) which are transformed to LOG
%              space
% OUTPUTS
%  paramn - Parameter names
%  paramnt - Parameter names (for TEX interpreter)
%  range - Parameters ranges (min and max)
%  default - Parameters defaults
%  sims_reg - Names of max-min simulations (where only 1 parameter is shifted to
%           its max/min value)
%  sims_inter - Names of interaction simulations (where 2 parameters are shifted
%             to their max/min values)
%  expval - Parameters values for max-min and interaction simulations
%  simval - Name of "val" simulation (where all the parameters are shifted from
%          their default values, in order to validate the Meta-Models)
%  valval - Parameters values for "val" simulation
%  sims_con - Name of "constrain" simulations (where each time one parameter is
%           shifted from its default value, but not to its max/min values)
%  valcon - Parameters values for "constrain" simulations
%  param_log - Array of 0/1 numbers (having length of paramn), where ones stand
%             for parameters which are transformed to log space
%  date_min - The earliest allowed start date ('dd-mmm-yyyy') for chosen sims_opt
%  date_max - The latest allowed end date ('dd-mmm-yyyy') for chosen sims_opt


function [VectorValues]=expval_inter_combs(temp_inter,range,default,paramn)

% NAME
%  expval_inter_combs
% PURPOSE
%  fill expval matrix for sims_def.m
% INPUTS
%  temp_inter - one of the interaction simulations
%  range - Parameters ranges (min and max)
%  default - Parameters defaults
%  paramn - Parameter names
% OUTPUTS
%  expval matrix for sims_def.m
```

```
function [odata odata_s sound_exist]=read_calmo_obs(vars,date_lim,avg_type,
            size_vars_sound)

% NAME
%   read_calmo_obs
% PURPOSE
%   Read observations data from Switzerland and north Italy, as well as soundings
%   data, for specified period
% INPUTS
%   vars - calibrated fields groups. See namelist.m
%   date_lim - Structure which includes the dates range of simulations to be read
%   avg_type - Interpolation method of observations data to model grid. Can be:
%           'near_neighb','simple_mean','weight_mean','clever_mean'
%   size_vars_sound - length(vars_sound) - number of soundings fields
% OUTPUTS
%   odata - Data matrix with dimensions [Field,Day,1,Lon,Lat] (field can be
%         Tmax,Tmin,Pr)
%   odatas - Data matrix for profileswith dimensions [Field,Day,1,Hour,Sounding
%                       location]
%   sound_exist - binary matrix [Day,Hour,Sounding location] with ones where the
%             sounding data exist


function build_temp_obs(avg_type,maxminavg)

% NAME
%   build_temp_obs
% PURPOSE
%   This function interpolates the gridded temperature observations (by C. Frei) to the model grid.
% METHOD
%   When comparing smoothed topography model-grid 2m-temperature whith the
%   observed 2m temperature, one shoud "correct" the observed
%   2m-temperature to correspond the model grid elevation. The correction may be
%   performed using the neighboring grid points 2m-temperature profile,
%   according the recommendation of C. Frei
%   Steps:
%   1. Read any simulation file to obtain the model grid lat/lon
%     (ex:aggregated_LTUR_2013011000.nc)
%   2. Read any simulation file to obtain the model grid altitude
%     (ex:laf2013111600_filtered.nc)
%   3. Read any observations file to obtain the observations grid lat/lon (ex:
%     TmaxD_ch01r.swisscors_201301010000_201302010000.nc)
%   4. Read gridded observations altitude ( ex: topo.swiss1_ch01r.swisscors.nc)
%   5. Interpolate the gridded observations to the model grid using one of the
%     possible methods
% INPUTS
%   avg_type - one of the interpolation methods:
%           'near_neighb','simple_mean','weight_mean','clever_mean'
%   maxminavg - Which field to interpolate: 'Tmax','Tmin','Tavg'
% OUTPUTS
%   saved (in .mat format) interpolated temperature observations


function build_rain_obs()

% NAME
%   build_rain_obs
% PURPOSE
%   This function interpolates the gridded rain observations to the model grid.
% METHOD
%   1. Read any simulation file to obtain the model grid lat/lon
%           (ex:aggregated_LTUR_2013011000.nc)
%   2. Read gridded rain observations you need to interpolate (ex:
```

```
%           CPCH_201301080000_01440_c2.nc)
%   3. Interpolate the gridded observations to the model grid using nearest
%           neighbor
% INPUTS
%   -
% OUTPUTS
%   saved (in .mat format) interpolated precipitation observations
% AUTHOR
%   Pavel Khain (pavelkh_il@yahoo.com)


function output=read_sounding_obs(year,month,day,height_step,
       windshear,windshearopt)

% NAME
%   read_sounding_obs
% PURPOSE
%   Read and interpolate soundings data, calculate sounding characteristics
% INPUTS
%   year
%   month
%   day
%   height_step - The interpolation height step in meters
%   windshear - [v1u,v1d,v2u,v2d,v3u,v3d] - pressure levels for calculating wind shears:
%      v1u - the upper pressure level for wshear1
%      v1d - the bottom pressure level for wshear 1
%      v2u - as mentioned above but for wshear 2
%      v2d - as mentioned abobe but for wshear 2
%      v3u - as mentioned above but for wshear 3
%      v3d - as mentioned above but for wshear 3, whereas 1100 is the surface level or the lowest level ("below surface")
%   windshearopt - windshear calculation method: 'scalar' or 'vector'
% OUTPUTS
%   output - Data matrix with dimensions [Field,Day,1,Hour,Sounding location]


function [mdata mdata_s]=read_calmo_sim(vars,sims,date_lim,sound_exist,
              size_vars_sound)

% NAME
%   read_calmo_sim
% PURPOSE
%   Read simulations data from, for specified period
% INPUTS
%   vars - calibrated fields groups. See namelist.m
%   sims - simulations names to be read. See namelist.m
%   date_lim - Structure which includes the dates range of simulations to be read
%   sound_exist - binary matrix [Day,Hour,Sounding location] with ones where the
%                                 sounding data exist
%   size_vars_sound - length(vars_sound) - number of soundings fields
% OUTPUTS
%   mdata - Data matrix with dimensions [Field,Day,simulation,Lon,Lat] (field can
%                       be Tmax,Tmin,Pr)
%   mdatas - Data matrix with dimensions [Field,Day,simulation,Hour,Sounding
%                       location]


function output = read_profiles_mod(simdir,year,month,day,simtype,height_step,
              sound_exist,windshear,windshearopt)

% NAME
%   read_profiles_mod
% PURPOSE
%   Read and interpolate profiles data, calculate profiles characteristics
```

```
% INPUTS
%   simdir - path to simulations files
%   year
%   month
%   day
%   simtype - simulation name
%   height_step - The interpolation height step in meters
%   sound_exist - binary matrix [Day,Hour,Sounding location] with ones where the
%                                 sounding data exist
%   windshear - [v1u,v1d,v2u,v2d,v3u,v3d] - pressure levels for calculating wind
%           shears:
%      v1u - the upper pressure level for wshear1
%      v1d - the bottom pressure level for wshear 1
%      v2u - as mentioned above but for wshear 2
%      v2d - as mentioned abobe but for wshear 2
%      v3u - as mentioned above but for wshear 3
%      v3d - as mentioned above but for wshear 3, whereas 1100 is the surface
%                       level or the lowest level ("below surface")
%   windshearopt - windshear calculation method: 'scalar' or 'vector'
% OUTPUTS
%   output - Data matrix with dimensions [Field,Day,simulation,Hour,Sounding
%                       location]


function datamatrix_new = Frei_regions(datamatrix,lat,lon,vars,
            avg_T,unify_regions)

% NAME
%   Frei_regions
% PURPOSE
%   Average part of the surface fields over predefined big regions
% METHOD
%   use image file (regions_italy_swiss_for_matlab.bmp) where each region has its
%   color
% INPUTS
%   datamatrix - Structure which includes observations and simulations data
%            for surface fields ('t2m_max','t2m_min','pr'). Dimensions:
%                       [Field,Day,simulation,Lon,Lat]
%   lat - latitudes of model domain
%   lon - longitudes of model domain
%   vars - calibrated fields groups. Can be any combinations of:
%              {'t2m_max','t2m_min','pr','sound'}
%   avg_T - region average over Precipitation only (avg_T=0), or over
%               Precipitation, Tmax and Tmin (avg_T=1)
%   unify_regions - array that defines which regions (out of 1-7) to unify (option
%                       to unify several regions into one bigger)
% OUTPUTS
%   datamatrix_new - Structure which includes observations and simulations
%               data for surface fields ('t2m_max','t2m_min','pr').
%                                   Dimensions: [Field,Day,region,simulation]


function [area] = regions_bmp(lat,lon,img,unify_regions)

% NAME
%   regions_bmp
% PURPOSE
%   Definition of regions over Switzerland and north Italy
% METHOD
%   Analyze image file (regions_italy_swiss_for_matlab.bmp) where each region has
%   its color
% INPUTS
%   lat - latitude
```

```
%   lon - longitude
%   img - image file (regions_italy_swiss_for_matlab.bmp) where each region has
%               its color
%   unify_regions - array that defines which regions (out of 1-7) to unify (option
%                                   to unify several regions into one bigger)
% OUTPUTS
%   area - region number to which lat lon belong


function datamatrix_t = gpts_series(datamatrix,vars,var)

% NAME
%   gpts_series
% PURPOSE
%   Redefine data arrays (which were not averaged over regions) to the following
%   structure: (fields,days,regions,simulations)
% INPUTS
%   datamatrix - Structure which includes observations and simulations data
%               for surface fields ('t2m_max','t2m_min','pr'). Dimensions:
%               [Field,Day,simulation,Lon,Lat]
%   vars - calibrated fields groups. Can be any combinations of:
%         {'t2m_max','t2m_min','pr','sound'}
%   var - specific field group (one of vars)
% OUTPUTS
%   datamatrix_new - Structure which includes observations and simulations
%                   data for surface fields ('t2m_max','t2m_min','pr').
%                                   Dimensions: [Field,Day,region,simulation]


function datatemp=del_bad_sounding(datamatrix_so,ml,sims_reg,sims_inter,
              sims_con,simval,vars_sound)

% NAME
%   del_bad_sounding
% PURPOSE
%   delete unrealistic sounding/profiles data
% INPUTS
%   datamatrix_so - Data matrix for profiles. Dimensions:
%                                   [Field,Day,region,simulation]
%   ml - Minimum number of days (during given period) for valid soundings data. If
%       less - current sounding fields are not analyzed.
%   sims_reg - Names of max-min simulations (where only 1 parameter is shifted to
%             its max/min value)
%   sims_inter - Names of interaction simulations (where 2 parameters are shifted
%               to their max/min values)
%   sims_con - Name of "constrain" simulations (where each time one parameter is
%             shifted from its default value, but not to its max/min values)
%   simval - Name of "val" simulation (where all the parameters are shifted from
%           their default values, in order to validate the Meta-Models)
%   vars_sound - calibrated profiles fields: {'CAPE','CIN','TCWC','WSHEAR1',
%                           'WSHEAR2','WSHEAR3','T850mb','T700mb','T500mb','RH850mb',
%                           'RH700mb','RH500mb','U850mb','U700mb','U500mb',
%                           'V850mb','V700mb','V500mb'}
% OUTPUTS
%   datatemp - Data matrix for profiles. Dimensions: [Field,Day,region,simulation]


function metamodel=neelin_e(parameters, datamatrix_test,vars_2d)

% Quadratic regression metamodel as described in Neelin et al. (2010) PNAS and
% Bellprat (2012)
% NAME
%   neelin_e
```

```
% PURPOSE
%   Fit a mutlivariate quadratic quadratic regressions (metamodels)
% INPUTS
%   From the structure parameters and datamatrix the following fields are
%   processed
%   parameters.experiments:
%         Parameter values for each experiment with the dimension of [N,
%         2*N+N*(N-1)/2]
%         The structure NEEDS to be as follows. Example for 2 parameters
%         (p1,p2):
%         [p1_l dp2 ] ! Low parameter value for p1 default dp2
%         [p1_h dp2 ] ! High parameter value for p1 default dp2
%         [dp1  p2_l] ! Low parameter value for p2 default dp1
%         [dp1  p2_h] ! Hihg parameter value for p2 default dp1
%         [p1_l p2_h] ! Experiments with interaction (no default)
%                   ! Additional experiments used to constrain interaction
%                     terms
%   parameters.range:
%         Range of values for each parameter to normalize the scale
%   parameters.default:
%         Default values of parameters to center the scale
%   datamatrix_test.moddata:
%         Modeldata corresponding to the dimensions of parameter.experiments
%   datamatrix_test.refdata:
%         Modeldata when using default parameter values to center the
%         datamatrix fitted
%   vars_2d - calibrated 2D fields. Can be any combinations of:
%          {'t2m_max','t2m_min','pr'}
% OUTPUT
%   structure metamodel.
%   a: Metamodel parameter for linear terms [N,1]
%   B: Metamodel parameter for quadratic and interaction terms
%     [N,N]. Quadratic terms in the diagonal, interaction terms
%     in the off-diagonal. Matrix symmetric, B(i,j)=B(j,i).
%   c: Metamodel parameter for zero order (constant)


function PostProc(period)
% NAME
%   PostProc
% PURPOSE
%   plot analysis results and calculate the optimal parameters combination
% INPUTS
%   time periods array (more precisely - initial dates of the periods):
%   {'dd-mmm-yyyy','dd-mmm-yyyy',...}
% OUTPUTS
%   saved (in .mat format) analysis results


function [W_fin]=weights_calc(parameters,datamatrix_tmp,metamodel_tmp,
        w_user,score,fields)

% NAME
%   weights_calc
% PURPOSE
%   calculate weights for different fields, to equalize their contributions to the
%   final score (assuming user defined weights are uniform).
% INPUTS
%   parameters - structure parameters (see definitions in
%   ReadData_and_MetaModel.m)
%   datamatrix_tmp - structure datamatrix (see definitions in
%   ReadData_and_MetaModel.m)
%   metamodel_tmp - structure metamodel (see definitions in neelin_e.m)
```

```
%   w_user - array of user defined weights (for simlicity - from 0 to 1) for
%   calibrated fields:
%   score - 'rmse' or 'cosi' for RMSE-type and COSI-type scores, respectively
%   fields - field name (can be 't2m_max','t2m_min','pr',vars_sound)
% OUTPUT
%   W_fin - weights array for different fields


function [dmatrix]=neelin_p(metamodel,parameters,datamatrix,pvector)

% NAME
%   neelin_p
% PURPOSE
%   Forecast using regression metamodel as described in Neelin et al. (2010) PNAS
%   and Bellprat (2012).
% METHOD
%   Predict data using the metamodel for a parameter matrix
% INPUTS
%   From the structure metamodel, parameters and datamatrix the following fields
%   are
%   processed (mind the same naming in the input)
%   metamodel.a:
%         Vector of linear terms of the metamodel [...,N,1] additional
%         data dimensions possible (ex:a~[Regions,Variables,Time,N,1])
%   metamodel.B:
%          Matrix of quadratic and interactions terms [...,N,N] additional
%          data dimensions possible (ex:a~[Regions,Variables,Time,N,N])
%   metamodel.c: Metamodel parameter for zero order (constant)
%   parameters.range:
%          Range of values for each paramter to normalize the scale.
%   parameters.default:
%          Default values of parameters to center the scale
%   datamatrix.reffdata:
%          Modeldata of when using default parameter values to center the
%          datamatrix
%   pvector: Parameter values for one experiment with the
%          dimension of [N,1] N=Number parameters
% OUTPUT
%   dmatrix: Predicted data for parameter experiment


function ets = ETS (Obs,Model,thresrain)

% NAME
%   ETS
% PURPOSE
%   rain forecast. Answers the question: How well did the forecast "yes" events correspond
%   to the observed "yes" events (accounting for hits that would be expected by chance)
%   Range: -1/3 to 1; 0 indicates no skill. Perfect score: 1.
% NOTE
%   be carefull with 0/0 when there is no rain both in nature and model (ex:
%   "summer in Israel") !!!
% INPUTS
%   parameters - structure parameters (see definitions in
%              ReadData_and_MetaModel.m)
%   datamatrix_tmp - structure datamatrix (see definitions in
%                 ReadData_and_MetaModel.m)
%   metamodel_tmp - structure metamodel (see definitions in neelin_e.m)
%   w_user - array of user defined weights (for simlicity - from 0 to 1) for
%          calibrated fields:
%   score - 'rmse' or 'cosi' for RMSE-type and COSI-type scores, respectively
%   fields - field name (can be 't2m_max','t2m_min','pr',vars_sound)
% OUTPUT
```

```
%   ets - rain forecast score


function planes(main_data,parameters,w_user,score,new_calc,predir,param_log)

% NAME
%   planes
% PURPOSE
%   Plot performance scores for pair-wise parameters cross sections
% INPUTS
%   main_data - big structure, which includes the sub-structures:
%   main_data.data - datamatrix structure
%   main_data.metamodel metamodel structure
%   main_data.field - field names
%   main_data.W - weights array for different fields
%   parameters structure:
%   parameters.range:
%         Range of values for each paramter to normalize the scale.
%   parameters.default:
%         Default values of parameters to center the scale
%   datamatrix.reffdata:
%         Modeldata of when using default parameter values to center the
%                       datamatrix
%   w_user - array of user defined weights (for simlicity - from 0 to 1) for
%                calibrated fields:
%         tmax tmin pr cape cin ws1 ws2 ws3 T850mb T700mb T500mb RH850mb
%                       RH700mb R500mb U850mb U700mb U500mb V850 V700mb V500mb
%   score - 'rmse' or 'cosi' for RMSE-type and COSI-type scores, respectively
%   new_calc - 0 or 1: 0 by default, when main_data is devided into cells over
%                         periods. 1 - otherwise
%   predir - path for saving output planes figures
%   param_log - Array of 0/1 numbers (having length of paramn), where ones stand
%                         for parameters which are transformed to log space
% OUTPUT
%   saved planes figures


function score=rmse_score(qfit,obsdata,W,w_user,new_calc)

% NAME
%   rmse_score
% PURPOSE
%   calculate RMSE-type score for Meta-Model predictions (regressions estimations)
% INPUTS
%   qfit - metamodel predictions for given parameter combination
%   obsdata - observations data
%   W - weights for different fields, to equalize their contributions to the final
%       score
%   w_user - array of user defined weights (for simlicity - from 0 to 1) for
%         calibrated fields
%   new_calc - 0 or 1: 0 by default, when main_data is devided into cells over
%           periods. 1 - otherwise
% OUTPUT
%   score - RMSE-type score


function score=cosi_score(qfit,obsdata,W,w_user,new_calc)

% NAME
%   cosi_score
% PURPOSE
%   calculate COSI score for Meta-Model predictions (regressions
%   estimations). Defined on the basis of the COSI score by Ulrich Damrath (DWD)
```

```
% INPUTS
%   qfit - metamodel predictions for given parameter combination
%   obsdata - observations data
%   W - weights for different fields, to equalize their contributions to the final
%       score
%   w_user - array of user defined weights (for simlicity - from 0 to 1) for
%            calibrated fields
%   new_calc - 0 or 1: 0 by default, when main_data is devided into cells over
%              periods. 1 - otherwise
% OUTPUT
%   score - COSI score


function [Sopt, xstar, xopt, sc_stat,UB_next,LB_next]=lhopt(main_data,
              parameters,w_user,score,new_calc,lhacc,tmp_str,iteration,
        best_percent,UB_new,LB_new,param_log)

% NAME
%   lhopt
% PURPOSE
% Optimise model parameters using a latin hypercube sampling. See Bellprat (2012)
% METHOD
%   Create a sample of parameters using a latin hypercube design
%   and predict the model performance of the sample using the metamodel.
% INPUTS
%   main_data - big structure, which includes the sub-structures:
%   main_data.data - datamatrix structure
%   main_data.metamodel metamodel structure
%   main_data.field - field names
%   main_data.W - weights array for different fields
%   parameters structure:
%   parameters.range:
%           Range of values for each paramter to normalize the scale.
%   parameters.default:
%           Default values of parameters to center the scale
%   w_user - array of user defined weights (for simlicity - from 0 to 1) for
%                       calibrated fields:
%           tmax tmin pr cape cin ws1 ws2 ws3 T850mb T700mb T500mb RH850mb
%                       RH700mb R500mb U850mb U700mb U500mb V850 V700mb V500mb
%   score - 'rmse' or 'cosi' for RMSE-type and COSI-type scores, respectively
%   new_calc - 0 or 1: 0 by default, when main_data is devided into cells over
%                       periods. 1 - otherwise
%   lhacc - Number of experiments to sample parameter space at each iteration
%   tmp_str - path to the calibration results
%   iteration - iteration number (of convergence process to the optimal parameters
%                       combination)
%   best_percent - "winners" percent of lhacc which is used to define the
%                                   parameters space for the next iteration
%   UB_new - upper limit of parameters range at given iteration
%   LB_new - lower limit of parameters range at given iteration
%   param_log - Array of 0/1 numbers (having length of paramn), where ones stand
%                       for parameters which are transformed to log space
% OUTPUTS
%   Sopt - Scores for all experiments at given iteration
%   xstar - Latin hypercube parameter experiments at given iteration
%   xopt - Parameter setting with highest score at given iteration
%   sc_stat - score statistics at given iteration
%   UB_next - upper limit of parameters range at NEXT iteration
%   LB_next - lower limit of parameters range at NEXT iteration


function histplot(lhscore,score,best,predir,iteration)
```

```
% NAME
%   histplot
% PURPOSE
%   plot histogram of SCORES for meta-models predictions
% INPUTS
%   lhscore - Scores for all experiments at given iteration
%   score - 'rmse' or 'cosi' for RMSE-type and COSI-type scores, respectively
%   best - 0 or 1: o by default, 1 if a special simulation exists and verified
%   predir - path for saving output planes figures
%   iteration - iteration number (of convergence process to the optimal parameters combination)
% OUTPUTS
%   saved scores histogram for specific iteration


function optparam(parameters,lhscore,lhexp,popt,errm)

% NAME
%   optparam
% PURPOSE
%   Plot optimal parameters combination
%   NOTE: not checked or adapted since early stage of the CALMO project !
% INPUTS
%   parameters - parameters structure
%   lhscore - Scores for all experiments (at last iteration)
%   lhexp - Latin hypercube parameter experiments (at last iteration)
%   popt - Parameter setting with highest score (at last iteration)
%   errm - error of metamodel, set to 0.001 ???
% OUTPUTS
%   Plot optimal parameters combination


function xnolog=log_turlen_entrsc(xlog,paramname)

% NAME
%   log_turlen_entrsc
% PURPOSE
%   convert the optimal parameters (tur_len and entr_sc) values from log-space
%   back to the regular space
% INPUTS
%   xlog - input vector of paramaeters values in log space
%   paramname - parameter names
% OUTPUT
%   xnolog - output vector of paramaeters values (tur_len and entr_sc) in regular %          space
```