Necessary conditions for algorithmic tuning of weather prediction models using OpenIFS as an example

Lauri Tuppi[1], Pirkka Ollinaho[2], Madeleine Ekblom[1], Vladimir Shemyakin[3], and Heikki Järvinen[1]

[1]Institute for Atmospheric and Earth System Research / Physics, University of Helsinki, Finland
[2]Finnish Meteorological Institute, Helsinki, Finland
[3]School of Engineering Science, Lappeenranta University of Technology, Lappeenranta, Finland

Correspondence: Lauri Tuppi (lauri.tuppi@helsinki.fi)

Abstract. Algorithmic model tuning is a promising approach to yield the best possible forecast performance of multi-scale multi-phase atmospheric models once the model structure is fixed. The problem is to what degree we can trust algorithmic model tuning. We approach the problem by studying the convergence of this process in a semi-realistic case. Let $M(\boldsymbol{x},\boldsymbol{\theta})$ denote the time evolution model, where $\boldsymbol{x}$ and $\boldsymbol{\theta}$ are the initial state and the default model parameter vectors, respectively. A *necessary condition* for an algorithmic tuning process to converge is that $\boldsymbol{\theta}$ is recovered when the tuning process is initialised with perturbed model parameters $\boldsymbol{\theta'}$ and the default model forecasts are used as pseudo-observations. The aim here is to gauge which conditions are *sufficient* in semi-realistic test setting to obtain reliable results, and thus building confidence on the tuning in fully-realistic cases. A large set of *convergence tests* is carried in semi-realistic cases applying two different ensemble-based parameter estimation methods and the OpenIFS model. The results are interpreted as general guidance for algorithmic model tuning, which we successfully tested in a more demanding case of simultaneous estimation of eight OpenIFS model parameters.

1 Introduction

Numerical weather prediction (NWP) models solve non-linear partial differential equations in discrete and finite representation. Sub-grid scale physical processes, such as cloud micro-physics, are treated in specific closure schemes. Once the model structure is fixed, some parametric uncertainty remains depending on how closure parameter values are specified. Closure schemes are always simplified representations of the real world, so 'universally true' parameter values do not exist. This uncertainty can be conceived as a probability density of the closure parameters: the expected value corresponds to the optimal model skill and the co-variance to their inherent uncertainty. The expected parameter value is the obvious choice for deterministic forecasting while the co-variance can be utilized to represent parametric uncertainty in ensemble forecasting (Ollinaho et al., 2017).

Model tuning is an attempt to unveil some statistics of the probability density of the closure parameters, and algorithmic methods add objectivity and transparency to the process (Hourdin et al., 2017; Mauritsen et al., 2012). It is paramount in algorithmic model tuning that the method applied is able to converge to the correct statistics with limited computing resources.

The aim of this paper is to gauge the circumstances favouring successful model tuning when ensemble-based parameter estimation methods are used. These include, for instance, differential evolution (Storn and Price, 1997) and its variants, genetic

algorithm (Goldberg, 1989), particle swarm optimisation (Kennedy, 2010) and Gaussian importance samplers (e.g. EPPES, Järvinen et al., 2012; Laine et al., 2012). The results may also be useful for more deterministic algorithms, such as multiple very fast simulated annealing (Ingber, 1989), and filter based estimation algorithms such as ensemble Kalman filter and its variants (e.g. Annan et al., 2005; Pulido et al., 2018) as well as particle filters (e.g. Kivman, 2003). Of course, the length of the time window is limited in filter based estimation methods.

The results are interpreted so as to provide guidance into successful tuning exercises and savings in computing time. Convergence testing is always semi-realistic and can provide insight into how to design fully-realistic model tuning exercises. Based on a very large amount of testing, the following general guidance can be drawn up.

Trivial testing: It is important to start model optimisation with some trivial testing before proceeding to more demanding realistic cases. In a recommended trivial test, model parameters are off-set slightly from their default values and optimisation is confined to a known training sample of model initial states and forecasts. In this case, the chosen optimisation method must be able to recover the default parameter values. If the optimisation target function is too simplistic, the process may not converge or convergence is very slow. In the case of ensemble-based sampling, we noted that stochastic initial state enhance the recovery of the parameters.

Efficiency: It is important to consider how to efficiently allocate computing resources. We noted that it is better to perform a long convergence test with a small ensemble and varying initial states rather than a short test with a large ensemble – the variability of atmosphere is thus more robustly sampled. Also, in OpenIFS, it turned out that already a 24-hour forecast range is sufficiently long for parameter convergence, offering potential savings in computing time.

Reproducibility: It is usually necessary that in realistic testing, the optimisation result is verified in an independent sample. With the convection parameters of OpenIFS used in this study, reproducibility is better for a relatively short forecast range (24-hours). At longer ranges, results are less repeatable. Thus, from both the efficiency and reproducibility viewpoints, a 24 hour forecast range seems optimal for the convection parameters used in this study.

Practical concerns: Our testing showed that one should not blindly trust algorithmic tuning – it is an efficient tool that can potentially accelerate model development (Jakob, 2010), but it must be used cautiously. For example, ensemble-based sampling works well in fine-tuning of an already well-performing model but less well if the initial uncertainty of model parameters is large. It is also worth noting the deep-rooted ambiguity in the optimal parameter values which can depend on the forecast range.

The paper contains the following sections: Section 1 introduces the topic of convergence testing, Section 2 presents tools and methods needed for running and evaluating convergence tests, Section 3 presents the convergence test set-ups, Section 4 presents the results, Section 5 contains further discussion and Section 6 concludes.

2 Methods

2.1 OpenIFS and closure parameters of the convection scheme

OpenIFS is the atmospheric forecast model of the Integrated Forecasting System (IFS) of ECMWF. In this study, we use a version based on the model version being operational from November 2013 to May 2015 (cycle 40r1, ECMWF, 2014). Convergence tests are run at two resolutions: TL159 and TL399 corresponding to 125 km and 50 km resolutions, respectively, both with 91 vertical levels. Initial conditions are extracted from the ECMWF operational archive (a control member plus 50 perturbed analyses) for year 2017 to cover different weather regimes and seasons. Each convergence test contains 52 ensemble forecasts, i.e., an ensemble is initialised once per week. The data set of ensemble initial conditions (Ollinaho et al., submitted) has been generated with the IFS cycle 43r3. Thus, some spinup/spindown is possible at early forecast ranges of a few hours. The differences between the two model versions can be found in ECMWF (2019).

We focus on closure parameters of the convection scheme, consisting of a bulk mass flux with an updraught and downdraught pair in each grid box for shallow, deep and mid-level convection (ECMWF, 2014; Bechtold et al., 2008). The parameters, their default values and short descriptions are in Table 1. The tests also involve the use of a stochastic representation of model uncertainty (Stochastically Perturbed Parametrisation Tendencies, (SPPT, Palmer et al., 2009)).

2.2 OpenEPS – ensemble prediction workflow manager

Convergence tests involve running large amounts of ensemble forecasts. Traditionally, ensemble forecasting and research on ensemble methods have been tied to major NWP centres providing operational ensemble forecasts to end-users. Usually these platforms are not suited for academic research. Instead, we use a novel and easily portable ensemble prediction workflow manager (called OpenEPS), developed at the Finnish Meteorological Institute specifically for academic research purposes (https://github.com/pirkkao/OpenEPS).

OpenEPS has been designed for launching, running, and post-processing a large number of ensemble forecast experiments with only a small amount of manual work. OpenEPS is very flexible and can be easily coupled with external applications required in parameter tuning, such as autonomous parameter sampling.

2.3 Optimisation algorithms

Brute force sampling of the parameter space of full-complexity NWP models is computationally far too expensive. Typically, one can afford running perhaps only some tens or a maximum of a few hundred simulations in a tuning experiment. Therefore, the tuning methods need to be sophisticated. In these convergence tests, we use two ensemble based optimisation algorithms: Ensemble prediction and parameter estimation system (EPPES) (Laine et al., 2012; Järvinen et al., 2012) and differential evolution (DE) (Storn and Price, 1997; Shemyakin and Haario, 2018).

EPPES is a hierarchical statistical algorithm, which uses Gaussian proposal distributions, importance sampling, and sequential modelling of parameter uncertainties to estimate model parameters. A parameter sample is drawn from the distribution,

an ensemble forecast is run with these parameter values and the goodness of the parameter values is evaluated by calculating a cost function for each ensemble member. The proposal distribution is sequentially updated such that it shifts towards more favourable parameter values. Here, the shift of parameter mean values between consecutive iterations is limited to a conservative value of 5%.

DE (Storn and Price, 1997) is heuristically based on natural selection. It consists of evolving population of parameter vectors where vectors leading to good cost function values thrive, and produce an offspring, while vectors leading to bad cost function values are eliminated. The population update procedure of DE is achieved by a certain combination of mutation and crossover steps. These steps ensure that the new parameter vectors differ at least slightly from the vectors that already belong to the population. The natural selection is achieved through the selection step, where the elements of the current population compete with the new candidates based on the defined cost function. The fittest are kept alive and proceed to the next generation. Besides the standard DE algorithm, we use: generation jump (Chakraborty, 2008), DE/best/1 mutation strategy (Feoktistov, 2006; Chakraborty, 2008; Qing, 2009) with randomized scale factor by jitter and dither (Feoktistov, 2006; Chakraborty, 2008) and recalculation step (Shemyakin and Haario, 2018). Jitter and dither act to increase the diversity of the parameter population so that they cause small random variation in the parameter values preventing DE from sticking to some specific values. Jitter corresponds to the approach when the scale factor is randomised for every mutant parameter vector in the mutation process by sampling from the given (usually uniform) distribution. Dither corresponds to the approach when the fixed scale factor is slightly randomised for each single component of the mutant vector in the mutation process. The recalculation step enhances the convergence when the cost function is stochastic. This means that occasionally the parameter vector is not updated but passed for a new iteration in order to ensure that only those parameter vectors leading to good scores for two times are used in generation of new vectors.

The main focus here is on the EPPES results. More details about the algorithms and their specific setting are explained in Appendices 1 and 2.

2.4 Optimisation target functions

We will apply two very different optimisation target functions (hereafter cost functions) in the convergence tests. The first one is the global root mean squared error of the 850 hPa geopotential height (ΔZ):

$$\Delta Z = \sqrt{\frac{1}{D}\int_D (Z_{850} - Z^*_{850})^2 dD} \tag{1}$$

where Z_{850} and Z^*_{850} denote 850 hPa geopotential in the pseudo-observations and perturbed forecast at each grid point. D denotes the horizontal domain. Pseudo-observations are default model forecasts with fixed default parameter values. Full fields are three-dimensional but for ΔZ only two-dimensional slices are used. This is a restrictive cost function formulation and used here merely as a useful demonstrator. There are three reasons why we expect ΔZ to perform sub-optimally. First, it exploits information only from a small fraction of the model domain and the upper parts of the model domain remain unconstrained. This means that ΔZ is not (directly) sensitive to perturbations in the middle and upper troposphere where the most influential

effects of perturbed convection are present. Second, it requires substantial interpolation of forecasts and reference data because OpenIFS has a terrain following hybrid sigma vertical coordinate, and hence the model levels are not aligned with pressure levels in lower troposphere. Third, the 850 hPa level intersects with the ground level in mountainous areas.

The second cost function is the global moist total energy norm (ΔE_m, e.g. Ehrendorfer et al., 1999) and it is a very comprehensive integral measure of the distance between two atmospheric states. The moist total energy norm can be written as

$$\Delta E_m = \frac{1}{2M_a} \int_\eta \int_D [u'^2 + v'^2 + \frac{c_p}{T_r} T'^2 + c_q \frac{L^2}{c_p T_r} q'^2] dD \frac{\delta p_r}{\delta \eta} d\eta + \frac{1}{2} \int_D [R \frac{T_r}{p_r} \ln p_s'^2] dD, \tag{2}$$

where u', v', T', q' and $\ln p_s'$ refer to differences between forecast and pseudo-observations in wind components, temperature, specific humidity and the logarithm of surface pressure. M_a is the mass of the atmosphere, c_q is a scaling constant for the moist term, L vaporisation energy of water, c_p specific heat at constant pressure, T_r reference temperature and p_r reference pressure. Here, we set $T_r = 280$ K and $p_r = 1000$ hPa as in (Ollinaho et al., 2014). D and η denote increments of horizontal and vertical integrals. Unlike in Ollinaho et al. (2014) we set c_q to 1 and $d\eta$ to equal the difference of pressure between consecutive model levels. Ollinaho et al. (2014) also show instructions how to discretise ΔE_m for practical use.

We expect that at short forecast ranges, the linkage between variations in the values of ΔE_m and parameter perturbations is detectable, enabling to estimate parameter densities.

2.5 Evaluation of convergence tests

The parameter convergence is measured with fair continuous ranked probability score (fCRPS, Ferro et al., 2008) formulated as the kernel representation (see e.g. Leutbecher, 2018). fCRPS rescales the scores as if the ensembles were infinitely large so that there is no dependence between ensemble size and the score itself. The property of fairness is essential in comparison of convergence tests with different ensemble sizes. fCRPS has not been designed for evaluation of ensembles of parameter values, so a direct application of fCRPS may lead to cancellation of the two terms (see Eq. 6, Leutbecher, 2018) causing difficulties in interpreting the results. Therefore, we use the two terms separately for each parameter θ_n:

$$\text{fCRPS}_1 = \frac{1}{M} \sum_{j=1}^{M} |\theta'_{j,n} - \theta_{d,n}| \tag{3}$$

and

$$\text{fCRPS}_2 = \frac{1}{2M(M-1)} \sum_{j=1}^{M} \sum_{k=1}^{M} |\theta'_{j,n} - \theta'_{k,n}|, \tag{4}$$

where n is the index over tunable parameters, $\theta'_{j,n}$ and $\theta'_{k,n}$ are the parameter values used by ensemble members j and k, $\theta_{d,n}$ is the default parameter value and M is the ensemble size. Each ensemble member is generated applying a unique set of parameter values. The first part (Equation 3) is a measure of how much the ensemble mean parameter value differs from the

default value while the second part (Equation 4) indicates how much spread is associated with the ensemble mean parameter value; in other words how certain or uncertain the algorithm considers the parameter value. Both parts of fCRPS have a perfect score of zero.

3 Set-ups of the convergence tests

Table 2 shows the outline of our experiments. The table explains the different levels of complexity we use in the convergence tests. The different levels of complexity are tested to see which is the optimal way to extract information of the parameter space. On the one hand, keeping convergence tests as simple as possible makes interpretation of the test results easy, but on the other hand, more realistic tests provide information on how the parameter tuning system would perform in fully-realistic tuning tasks.

L0, L1 and L3 tests are performed with one set-up: a forecast range of 48 hours and an ensemble size of 50 members. ΔZ is only tested at L1 level. Most of the effort is put on L2 testing with ΔE_m so they are performed with various combinations of forecast range and ensemble size. The focus is on L2 tests on one hand because we assume that if the convergence is good at this level of complexity, it will be good also at lower levels. On the other hand, convergence in L2 and L3 tests is relatively similar. We use the parameters θ_1 and θ_2 in L0 to L2 tests and parameters θ_1 to θ_5 in L3 tests. In L0 tests all forecasts are initialised from an unperturbed initial state of the control forecast (1 Jan 2017 00UTC). L1, L2 and L3 tests use ensemble initial conditions from 52 dates.

Throughout the paper, the pseudo-observations are generated with a default model with fixed parameter values (see Table 1). Therefore, the target of the convergence is always known (i.e., the default parameter set), and it stays the same during the tests. Analyses, re-analyses or real observations are not used in this study.

4 Results

For brevity, we mainly concentrate on discussing results obtained with EPPES, although most of the convergence tests have been run with both algorithms. Due to the nature of the algorithms, EPPES produces less noise near the end of the convergence tests. Therefore, results generated with EPPES are easier to interpret. However, none of the results produced with DE contradict the results of EPPES.

4.1 Selection of level of complexity

We test how much complexity should be included in algorithmic tuning. Figure 1 shows four convergence tests with different levels of complexity, described in Table 2. As expected the parameters converge slower the higher the level of complexity is as the parameter uncertainties decrease slower. Both parameters converge very fast in the trivial L0 convergence test. Convergence to the default values in the L0 convergence test is trivial as minimisation of the parameter perturbations is the only way to minimise the cost function values. However, we want to emphasize that fully-realistic tuning at L0 level of complexity could

lead to overfitting of the parameters since the parameters would be optimised for that specific weather state only. L0 can still be used to test that the tuning infrastructure works.

L1, L2 and L3 tests resemble more fully realistic model tuning. Figure 1 shows that the convergence tests at different levels of complexity behave quite reasonably. The uncertainty cannot vanish completely since some uncertainty is always present due to ensemble initial conditions.

L1, L2 and L3 tests have a common feature that the parameters tend to converge to some off-default values. This feature is inherent to convergence tests, which use pseudo-observations generated with the default model. The convergence to off-default values will be discussed in further below.

We recommend using L1 (only perturbed initial conditions) in fully-realistic model tuning. L1 is the simplest safe option. Higher levels of complexity do not provide additional information, instead, they only make convergence more difficult. L0 is only recommended for testing purposes. Depending on user needs, L1 can be modified by adding more parameters.

4.2 Selection of optimisation target

Here, we compare convergence tests that use different cost functions. Figure 2 compares L1 convergence tests with ΔE_m (shown with black solid and dash-dotted lines) and ΔZ (shown with cyan dotted line and shading in the background). Figure 2 shows that ΔE_m leads to much faster convergence than ΔZ. The superior performance of ΔE_m is explained by: first, global integral of several variables catches the signal of parameter perturbations much better than a single level measure. Second, perturbations of convection parameters do not affect 850 hPa geopotential directly so 48 hours may not be long enough to develop a traceable signal. Perturbations of convection parameters modify specific humidity, wind components and temperature directly. These fields must change substantially before the signal is seen in geopotential at 850 hPa.

We recommend in general using a more comprehensive cost function that accounts for more than one atmospheric level and more than one variable. Targeted forecast range also plays a crucial role in constructing a suitable cost function, and thus should be carefully chosen based on what type of parameterised processes are being optimised. However, it is likely that with some other parameters another cost function than ΔE_m might work better.

4.3 Finding the most efficient set-up

The process of finding a forecast range and an ensemble size that give satisfactory convergence with a minimal amount of computational resources is done in two steps. First, we compare forecast ranges. Second, we take the forecast range with the fastest convergence in order to find optimal ensemble size using L2 level of complexity. We expect that the results obtained with such a high level of complexity generalise well to lower levels. This step can also be seen as fine tuning of the cost function.

We take the final parameter values proposed by EPPES to calculate the components of fCRPS (Equations (3) and (4)). Low bias and low uncertainty are desired since they indicate that the mean value has converged close to the default value and that the uncertainty is small. However, we do not expect precise convergence to the default values because initial condition and model perturbations are activated in the experiments. Figure 3 shows the components of fCRPS of the final ensemble for the

parameter θ_2 from numerous convergence tests with different forecast ranges and ensemble sizes. From Figure 3 it is obvious that the convergence of θ_2 is the most efficient when the forecast range is 24 hours. It is remarkable that 24 hours and also 48 hours lead to significantly better convergence than 72 hours that was used for example in (Ollinaho et al., 2014). For θ_1, 12, 24 and 48 hour forecasts are the best and roughly equally good (not shown).

The superior convergence with 24 hour forecasts can be explained by relatively linear response of OpenIFS to parameter perturbations, which ΔE_m is able to detect. The sub-optimal performance of 12 hour forecasts compared to 24 hour forecasts may be, at least partly, due to the spin-up related to to discrepancy of model versions. Consequently, we are somewhat uncertain about the true performance at very short forecast ranges. Against our expectations, some convergence occurs also at the longest forecast ranges when the response to parameter perturbations is definitely non-linear. At least some parameter convergence takes place with all forecast ranges but the convergence is by far the fastest at short ranges.

Figure also 3 shows that there is relatively more error in the parameter mean value than there is spread when long or very short forecasts are used. This is discussed further below.

We now focus on the forecast range of 24 hours and perform convergence tests with different ensemble sizes again measured with fCRPS. Figure 4 indicates that convergence tests with ensemble sizes > 20 are stable while convergence tests with smaller ensemble sizes do not show the desired smooth decrease of both parts of the fCRPS. Sampling variance seems to have a strong effect in those cases. Sampling variance seems to play a smaller role when the ensemble size is 20 or larger. Figure 4 also enables comparison of the convergence tests from the resources point-of-view. For example, tests with 50 ensemble members and 20 iterations, and 20 members and 50 iterations both use 1000 forecasts. However, the latter option leads to much better convergence. The same pattern seems to apply to most of the similar pairs. Increasing the ensemble size beyond about 20 members does not seem to be necessary to achieve good convergence. Results here are for θ_2 but the same conclusions can be drawn for θ_1 although the results with θ_1 are less conclusive (not shown). Interestingly, these results are in line with the conclusions of Leutbecher (2018) that in ensemble forecasting related research it is better to have large number of small ensembles than small number of large ensembles.

Based on these results we recommend using relatively short forecasts of 24 hours, at least when convection parameters are concerned. We also recommend using medium-sized ensembles of about 20 members. Very small ensembles of less than 10 members increase sampling variance and destabilise the convergence. Moreover, convergence with DE was practically impossible with small ensembles. We are fairly sure that about 20 members is close to the optimal ensemble size at least for tuning these two parameters. However, we are somewhat uncertain that 24 hours is the optimal forecast range for all parameters.

4.4 Reliability of convergence tests

Two example convergence tests are repeated four times first using L1 and then L2: first one with a sub-optimal set-up of 48 hour forecasts and 20 ensemble members and the second one with a close to optimal set-up of 24 hour forecasts and 26 ensemble members. These set-ups are highlighted in Figure 3. We test whether these convergence tests yield similar results every time i.e. the repeatability.

Figure 5 shows the evolution of the repeated convergence tests measured with fCRPS in a similar fashion as in Figure 4. L1 convergence tests are on the left-hand side and L2 tests on the right-hand side. Labels A1 to A4 refer to the sub-optimal set-up and labels B1 to B4 to the optimal set-up. The left-hand side of Figure 5 shows that both L1 set-ups yield fairly reproducible convergence. However, when the level of complexity is raised to L2, only the more optimal set-up seems to yield repeatable convergence with EPPES. The results obtained with DE are less conclusive as DE tends to fluctuate around the optimum (not shown).

We recommend using such a set-up that is the most likely to yield reliable parameter convergence. At least in our case, the optimal set-up of 24 hour forecasts and about 20 member ensembles is also the most reliable set-up. Also, using only initial condition perturbations (L1) besides the parameter perturbations leads to more reliable convergence than initial condition plus stochastic model perturbations (L2).

4.5 Potential pitfalls

First, during the convergence tests, we noticed that some parameters tend to converge to some off-default values. As an example, the two most used parameters (θ_1 and θ_2) tend to converge to slightly different values depending on the forecast range used. θ_2 tends to converge to a value smaller (larger) value than the default value when forecasts longer (shorter) than 24 hours are used. The opposite is true for θ_1. However, at least θ_2 tends to converge in one-parameter convergence tests in the similar way as in the two-parameter tests. This is illustrated in Figure 6, which shows the mean values of the final parameter for θ_1 (Figure 6a) and for θ_2 (Figure 6b). Convergence of θ_2 seems to depend very strongly on the forecast range used in the convergence tests. We now examine the cost functions for two sets of six hours to six days long ensemble forecasts, one using the default parameters and one using parameters obtained from the optimisation (Fig 6). Both sets of ensemble forecasts were compared to respective control forecasts with ΔE_m. The tests were repeated with only initial condition perturbations active and with initial condition plus stochastic model perturbations active. Results show that globally optimal parameter values are different from their respective default values even though the default model is used as reference. It is indeed possible to obtain lower cost function values with some off-default parameter values than with the default parameter values. This means that the peculiar dependence is not caused by any deficiencies in the cost function or optimisation algorithms. However, convergence tests and fully-realistic tuning are so different that we are unsure whether this dependency even exists at all in fully-realistic tuning. Even if the dependency exists, it is unclear whether it hinders tuning after all.

A potential pitfall might emerge if there is a need to do algorithmic tuning with ensembles of very different sizes. At least the two algorithms, EPPES and DE, are difficult to set up so that they would work satisfactorily regardless of the ensemble size. If the algorithms produce good convergence with small ensembles of $\sim$5 members, the parameter convergence is very slow with medium-sized and large ensembles or the parameters may even diverge. Vice versa, if the algorithms work well with medium-sized and large ensembles, they tend to be unstable with small ensembles.

At least two potential pitfalls are related to bad initialisation of tuning exercises. The first pitfall is that convergence of EPPES suffers from too large initial parameter off-sets, while DE is very robust. For example, in an extreme convergence test, where parameter off-set is an order of magnitude, convergence of EPPES may stop while DE suffers much less. The second

pitfall may be encountered if the initial uncertainty of some parameter is too small with respect to the initial off-set, which makes convergence to some local optimum likely. Both algorithms show that in such a case, the badly initialised parameter remain practically unchanged while the other parameters appear to compensate the error.

We do not recommend completely blind algorithmic tuning. The parameter off-set should not be excessively large, and the initial parameter off-set and uncertainty should be well proportioned. We also recommend to pay attention to selection of the tuning algorithm. In case of tuning very uncertain parameters, we recommend to use robust algorithms, which do not suffer from large parameter off-set.

4.6 A recipe for successful tuning

Our recipe for economic and efficient tuning is summarised below:

- level 1 of complexity (at least initial condition perturbations, possibly also stochastic model perturbations)
- a comprehensive measure is used as a cost function (ΔE_m in our case)
- a relatively short forecast range is used (24 hours in our case)
- a relatively small ensemble size (20 in our case)

Here we put the recipe into test with more demanding convergence tests with EPPES. We run four five-parameter tests with TL159, two five-parameter tests with TL399 and one eight-parameter test with TL159 resolution. The parameters in the five-parameter tests are the same as in the L3 convergence tests. In the eight-parameter test there are three additional parameters from the convection scheme (see Table 1). The parameters are initialised randomly with either 10% too large or too small value, and large uncertainty. The set of initial conditions is the same as before meaning 52 iterations.

We discuss all the four TL159 and the two TL399 convergence tests at once, meaning there are a total of 30 converge cases to be discussed (six experiments all involving five parameters). In these six tests, the parameter values converge towards the default values during the convergence tests in 20 out of 30 cases. θ_1 converges to an off-default value in four of the cases as does θ_3. θ_4 converges to an off-default value twice. Furthermore, θ_2 tends to converge to a slightly smaller value, and θ_1, θ_4 and θ_5 to slightly larger value than their respective default values. In 25 out of 30 cases the final parameter value and the default value are both within two standard deviations uncertainty of each other, and hence the default value is inside the parameter distribution proposed by EPPES. In the remaining five cases the remaining parameter off-set is slightly more than two standard deviations. These five cases distribute so that each parameter ends up outside of two standard deviations distance to the default value for one time. In all 30 cases, the uncertainty of the parameter value decreases during the convergence tests meaning that the parameters do converge even though in some cases they converge to some off-default values.

The results of the eight-parameter convergence test are presented in Figure 7. It shows convergence of the eight parameters in normalised form, and the text boxes in each panel indicate the remaining parameter off-set after 52 iterations. All parameters converge towards their default values. In case of θ_5, the default value is outside of the uncertainty range. Additional dimen-

sionality seems to slow down the convergence only a little, which definitely encourages to use algorithmic tuning methods for large parameter sets.

5 Discussion

The choice of the cost function is an essential part of the tuning problem. In order to illustrate the importance of choosing a suitable cost function, we intentionally chose two radically different cost functions: the root-mean squared error of geopotential at 850 hPa and the moist total energy norm. The former was, as expected, a bad choice, whereas the latter was a clearly more suitable choice. However, the moist total energy norm was not a perfect choice due to the properties of the tunable parameters. The two parameters θ_1 and θ_2 are not equally sensitive to the components of the moist total energy norm. θ_1, which is related to the shallow convection, is active only in the lowest 200 hPa layer of the model atmosphere. θ_1 mainly affects how specific humidity is distributed in the layer. Therefore, contribution of θ_1 comes almost only from lower tropospheric specific humidity. θ_2 controls deep convection so it has direct impact on wind, temperature and specific humidity throughout the model troposphere. Therefore, contribution of θ_2 to the cost function dominates, which may in some cases decrease the overall sensitivity of the moist total energy norm when estimating the two parameters simultaneously. An option would be to use multiple cost functions, having one dedicated for each tunable parameter. However, this could lead to a question of scaling: would each cost function have equal weight or are some of the cost functions considered more important? At the moment we do not have a definitive answer for this.

In our study, we aimed at finding an optimal set-up for convergence tests by studying different combinations of forecast range and ensemble size. Using an ensemble of 20 members and a forecast range of 24 hours gave the best results. When the ensemble size is too small, the sample size will also be small, which could lead to the case of not having a representative sample. A forecast range of 24 hours seems optimal. When the forecast range is shorter, θ_1 tends to converge to smaller values and θ_2 to larger values than the default parameter value, whereas a longer forecast leads to θ_1 converging to larger value and θ_2 to smaller value than the default value. The question whether the parameter values depend on the forecast range is profound. The entire forecast range could also be considered, but may lead to similar scaling issues (e.g. Ollinaho et al., 2013) as when using multiple cost functions.

The two optimisers used in this study, EPPES and DE, have different properties. EPPES converges more slowly and estimates the covariance matrix of the parameters, whereas DE gives faster but less steady convergence. The optimisers could therefore be used at different stages of the optimisation: first, DE could be used as a coarse tuner finding the approximate direction, and then EPPES could be used to fine tune the results. This type of tuning process would be of most use when the parameters are known poorly a priori.

Local minima of the cost function is a potential problem. According to our observations, this may occur if the initialisation of the parameter values is bad. A large initial distance from the optimal value combined with a too small initial uncertainty range might lead to a case where the cost function becomes locally minimised, and the algorithm gets stuck exploring parameter values from around this local minimum. Other parameters may compensate the error, and the cost function becomes locally

minimised. When the parameters are initialised appropriately and initial condition perturbations are active, problems of this sort are less likely.

We compared the perturbed forecasts against the control forecast run with default parameters. In this case, one would expect that forecasts with default parameters would result in minimum cost function, but this turned out not to be the case. This leads us to the question whether changing the values of the model parameters affect properties of the ensemble, such as its spread. In a well-tuned ensemble prediction system not only should the model be as good as possible (i.e., having optimal parameter values) but the relationship between the spread of the ensemble and the ensemble mean skill should be in balance. We leave this question open for future studies.

6 Conclusions

In this paper we have studied the convergence properties of two algorithms used for tuning model physics parameters in a numerical weather prediction model. The tuning process is a computationally demanding task and using an optimal experimental set-up would minimise the amount of computational resources required.

In our experiments we studied two different tuning algorithms and how the convergence properties were affected by (1) the choice of cost function, (2) forecast range, (3) ensemble size, and (4) the complexity of the model set-up (perturbations of initial conditions and stochastic physics turned on or off). In our case, we focused on tuning two parameters of the convection scheme of the OpenIFS model. The model resolution in these tests was T159 (about 125 km).

Our goal was to find an optimal set-up of forecast range and ensemble size with the highest likelihood for fast and reliable convergence; hence, minimising the amount of computations. We ran many convergence tests with different experimental set-ups, calculated the moist total energy norm between the forecasts with perturbed parameters and the control forecast having default parameters values, calculated a fair verification metric (fair-CRPS), and finally compared the experiments against each other. The optimal set-up in our experiments was an ensemble of 20 members and a forecast range of 24 hours.

We tested the optimal set-up for a more complex optimisational task: tuning five and eight parameters at once. In these experiments, the ensemble had 20 members, the forecast range was 24 hours, and the algorithm was run for 52 iterations. Such an experiment would be the same as running a single 1040-day long forecast consuming roughly 400 core hours for TL159 model resolution and 4200 core hours for TL399 model resolution on Intel Haswell computation nodes. These experiments showed that the convergence of most of the parameters was good.

Finally, we conclude our study by answering the question whether algorithmic tuning (of model physics parameters) could be trusted: yes when used with care.

Code and data availability. Basic version of OpenEPS is available under Apache licence version 2.0, January 2004 on Zenodo (https://doi.org/10.5281/zenodo.3759127). Amended version of OpenEPS, which was used in the convergence tests, is also available under Apache licence version 2.0, January 2004 on Zenodo (https://doi.org/10.5281/zenodo.3757601). The amended version contains various

modifications such as set-up scripts for EPPES and DE, scripts for calculating cost function values and scripts for processing and plotting output of convergence tests. Besides the archived versions, we encourage to check out also the maintained versions of basic and amended OpenEPS in Github (https://github.com/pirkkao/OpenEPS and https://github.com/laurituppi/OpenEPS). Licence for using OpenIFS NWP model can be requested from ECMWF user support (openifs-support@ecmwf.int), and the model can be downloaded from ECMWF ftp site (ftp.ecmwf.int). EPPES is available under MIT licence on Zenodo (https://doi.org/10.5281/zenodo.3757580) , and DE is available upon request from vladimir.shemyakin@lut.fi. The initial conditions used in the convergence tests belong to a larger data set. Availability of the data set will be described in (Ollinaho et al., submitted). We want to emphasize that reproducing the results does not require using exactly the same initial conditions than in this paper but any OpenIFS ensemble initial conditions can be used. Output data of the convergence tests is not archived since it can be easily reproduced.

1 Experimental details of EPPES

EPPES needs four hyperparameters: $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, $\boldsymbol{W}$, and n. The two former describes the initial guess for the distribution of the parameters that are to be estimated, whereas the latter two describes how accurate the initial guess is.

Let $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_n\}$ be the closure parameters. In EPPES, the prior guess it that the closure parameter follows a Gaussian distribution $\boldsymbol{\theta_i} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is the mean vector of $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ the covariance matrix.

Details of initialisation of the parameter distribution are listed in Table A1 and other settings of EPPES are summarised in Table A2. The mean values are always multiplied with 0.9 or 1.1 in the initialisation of the convergence tests.

2 Experimental details of DE

DE requires the boundaries for the parameter search domain to be specified. DE does not explicitly limit any searching directions by default, but some constraints can be specified in order to avoid unfeasible parameter values. In our case, we are targeting to only non-negative values.

Initial search domain is specified in Table A3 and other settings written in the namelist file are summarised in Table A4.

Recalculation step is employed every fifth iteration, it substitutes all usual DE steps (mutation, crossover, selection) and just computes/updates the value of the cost function in the current environment for the elements already in the population.

Author contributions. Lauri Tuppi and Pirkka Ollinaho designed the convergence tests, and Lauri Tuppi carried them out. Pirkka Ollinaho provided initial conditions and OpenEPS with comprehensive user support. Vladimir Shemyakin provided hands-on assistance in using DE in the convergence tests. Lauri Tuppi prepared the manuscript with contributions and comments from all co-authors. Heikki Järvinen and Madeleine Ekblom helped with clear formulation of the text. Heikki Järvinen supervised the experimentation and production of the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The authors are grateful to CSC-IT Center for Science, Finland for providing computational resources, and Juha Lento at CSC-IT for user support with the supercomputers. We thank Olle Räty at Finnish Meteorological Institute for assisting in graphical design of Figure 3. We are also thankful to Marko Laine at Finnish Meteorological Institute for the insightful discussions about evaluation of convergence tests and user support for EPPES. The figures have been plotted with help of Python Matplotlib library (Hunter, 2007). CDO (Schulzweida, 2019) was used for post-processing the output of OpenIFS. We would like to acknowledge the funding from the Academy of Finland (grants 333034 and 316939), the Vilho, Yrjö and Kalle Väisälä Foundation of the Finnish Academy of Science and Letters, and funding and support from Doctoral Programme in Atmospheric Sciences of University of Helsinki. Finally, we would like to thank Peter Düben and Joakim Kjellsson for their valuable comments for improving the manuscript further.

References

Annan, J. D., Lunt, D. J., Hargreaves, J. C., and Valdes, P. J.: Parameter estimation in an atmospheric GCM using the Ensemble Kalman Filter, Nonlinear Processes in Geophysics, 12, 363–371, https://doi.org/10.5194/npg-12-363-2005, https://www.nonlin-processes-geophys.net/12/363/2005/, 2005.

Bechtold, P., Köhler, M., Jung, T., Doblas-Reyes, F., Leutbecher, M., Rodwell, M. J., Vitart, F., and Balsamo, G.: Advances in simulating atmospheric variability with the ECMWF model: From synoptic to decadal time-scales, Quarterly Journal of the Royal Meteorological Society, 134, 1337–1351, https://doi.org/10.1002/qj.289, https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.289, 2008.

Chakraborty, U. K.: Advances in Differential Evolution, vol. 143, Springer: Verlag, https://doi.org/10.1007/978-3-540-68830-3, 2008.

ECMWF: IFS documentation. Part IV: Physical processes, CY40R1, https://www.ecmwf.int/sites/default/files/elibrary/2014/9204-part-iv-physical-processes.pdf, 2014.

ECMWF: Changes in ECMWF model, https://www.ecmwf.int/en/forecasts/documentation-and-support/changes-ecmwf-model, 2019.

Ehrendorfer, M., Errico, R. M., and Raeder, K. D.: Singular-Vector Perturbation Growth in a Primitive Equation Model with Moist Physics, Journal of the Atmospheric Sciences, 56, 1627–1648, https://doi.org/10.1175/1520-0469(1999)056<1627:SVPGIA>2.0.CO;2, https://doi.org/10.1175/1520-0469(1999)056<1627:SVPGIA>2.0.CO;2, 1999.

Feoktistov, V.: Differntial Evolution: In Search of Solutions, Springer Science, 2006.

Ferro, C. A. T., Richardson, D. S., and Weigel, A. P.: On the effect of ensemble size on the discrete and continuous ranked probability scores, Meteorological Applications, 15, 19–24, https://doi.org/10.1002/met.45, https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/met.45, 2008.

Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.

Hourdin, F., Mauritsen, T., Gettelman, A., Golaz, J., Balaji, V., Duan, Q., Folini, D., Ji, D., Klocke, D., Qian, Y., Rauser, F., Rio, C., Tomassini, L., Watanabe, M., and Williamson, D.: The Art and Science of Climate Model Tuning, Bulletin of the American Meteorological Society, 98, 589–602, https://doi.org/10.1175/BAMS-D-15-00135.1, https://doi.org/10.1175/BAMS-D-15-00135.1, 2017.

Hunter, J. D.: Matplotlib: A 2D graphics environment, Computing In Science & Engineering, 9, 90–95, https://doi.org/10.1109/MCSE.2007.55, 2007.

Ingber, L.: Very fast simulated re-annealing, Mathematical and Computer Modelling, 12, 967 – 973, https://doi.org/https://doi.org/10.1016/0895-7177(89)90202-1, http://www.sciencedirect.com/science/article/pii/0895717789902021, 1989.

Jakob, C.: Accelerating Progress in Global Atmospheric Model Development through Improved Parameterizations: Challenges, Opportunities, and Strategies, Bulletin of the American Meteorological Society, 91, 869–876, https://doi.org/10.1175/2009BAMS2898.1, https://doi.org/10.1175/2009BAMS2898.1, 2010.

Järvinen, H., Laine, M., Solonen, A., and Haario, H.: Ensemble prediction and parameter estimation system: the concept, Quarterly Journal of the Royal Meteorological Society, 138, 281–288, https://doi.org/10.1002/qj.923, https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.923, 2012.

Kennedy, J.: Particle Swarm Optimization, pp. 760–766, Springer US, Boston, MA, https://doi.org/10.1007/978-0-387-30164-8_630, https://doi.org/10.1007/978-0-387-30164-8_630, 2010.

Kivman, G.: Sequential parameter estimation for stochastic systems, Nonlinear Processes in Geophysics 10, pp. 253–259, 2003.

Laine, M., Solonen, A., Haario, H., and Järvinen, H.: Ensemble prediction and parameter estimation system: the method, Quarterly Journal of the Royal Meteorological Society, 138, 289–297, https://doi.org/10.1002/qj.922, https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.922, 2012.

Leutbecher, M.: Ensemble size: How suboptimal is less than infinity?, Quarterly Journal of the Royal Meteorological Society, 0, https://doi.org/10.1002/qj.3387, https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3387, 2018.

Mauritsen, T., Stevens, B., Roeckner, E., Crueger, T., Esch, M., Giorgetta, M., Haak, H., Jungclaus, J., Klocke, D., Matei, D., Mikolajewicz, U., Notz, D., Pincus, R., Schmidt, H., and Tomassini, L.: Tuning the climate of a global model, Journal of Advances in Modeling Earth Systems, 4, https://doi.org/10.1029/2012MS000154, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2012MS000154, 2012.

Ollinaho, P., Laine, M., Solonen, A., Haario, H., and Järvinen, H.: NWP model forecast skill optimization via closure parameter variations, Quarterly Journal of the Royal Meteorological Society, 139, 1520–1532, https://doi.org/10.1002/qj.2044, https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.2044, 2013.

Ollinaho, P., Järvinen, H., Bauer, P., Laine, M., Bechtold, P., Susiluoto, J., and Haario, H.: Optimization of NWP model closure parameters using total energy norm of forecast error as a target, Geoscientific Model Development, 7, 1889–1900, https://doi.org/10.5194/gmd-7-1889-2014, https://www.geosci-model-dev.net/7/1889/2014/, 2014.

Ollinaho, P., Lock, S.-J., Leutbecher, M., Bechtold, P., Beljaars, A., Bozzo, A., Forbes, R. M., Haiden, T., Hogan, R. J., and Sandu, I.: Towards process-level representation of model uncertainties: stochastically perturbed parametrizations in the ECMWF ensemble, Quarterly Journal of the Royal Meteorological Society, 143, 408–422, https://doi.org/10.1002/qj.2931, https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.2931, 2017.

Ollinaho, P., Carver, G., Lang, S., Tuppi, L., Ekblom, M., and Järvinen, H.: Ensemble prediction using a new dataset of ECMWF initial states, Geoscientific Model Development, submitted.

Palmer, T., Buizza, R., Doblas-Reyes, F., Jung, T., Leutbecher, M., Shutts, G., Steinheimer, M., and Weisheimer, A.: Stochastic parametrization and model uncertainty, ECMWF Technical Memoranda, 598, 1–42, 2009.

Pulido, M., Tandeo, P., Bocquet, M., Carrassi, A., and Lucini, M.: Stochastic parameterization identification using ensemble Kalman filtering combined with maximum likelihood methods, Tellus A: Dynamic Meteorology and Oceanography, 70, 1–17, https://doi.org/10.1080/16000870.2018.1442099, https://doi.org/10.1080/16000870.2018.1442099, 2018.

Qing, A.: Differential Evolution: Fundamentals and Applications in Electrical Engineering, Wiley, https://doi.org/10.1002/9780470823941, 2009.

Schulzweida, U.: CDO User Guide, https://doi.org/10.5281/zenodo.2558193, https://doi.org/10.5281/zenodo.2558193, 2019.

Shemyakin, V. and Haario, H.: Online identification of large-scale chaotic system, Nonlinear Dynamics, 93, 961–975, https://doi.org/10.1007/s11071-018-4239-5, https://doi.org/10.1007/s11071-018-4239-5, 2018.

Storn, R. and Price, K.: Differential Evolution — A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, Journal of Global Optimization, 11, 341–359, https://doi.org/10.1023/A:1008202821328, http://dx.doi.org/10.1023/A:1008202821328, 1997.

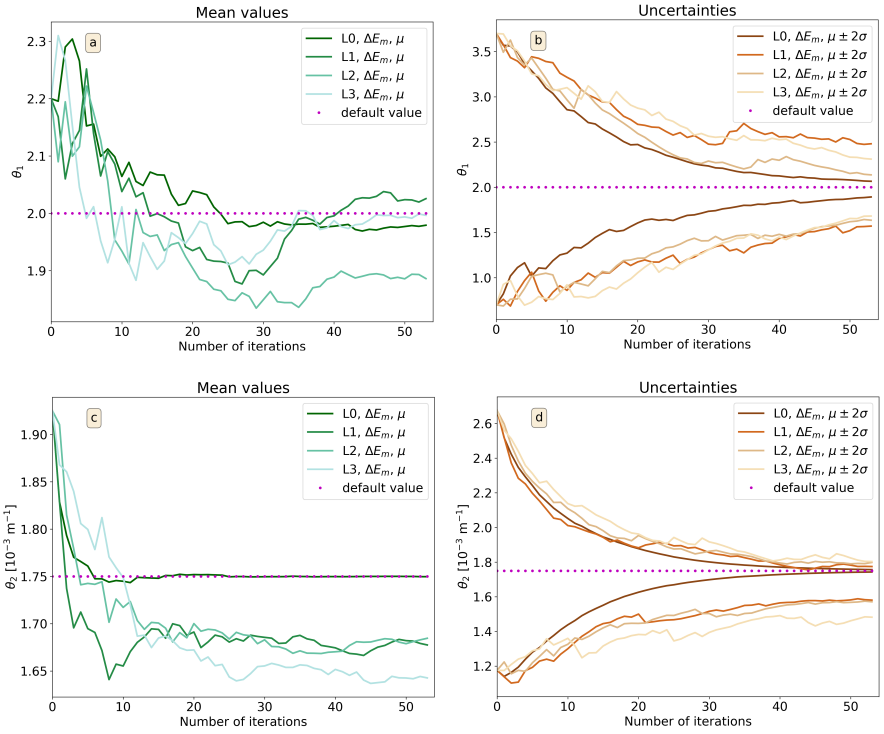

Figure 1. Comparison of convergence tests at different levels of complexity. Panels (a) and (b) show the evolution of distribution mean value (μ) and the mean value ± 2 standard deviations uncertainty ($\mu \pm 2\sigma$) for θ_1, and (c) and (d) show the same as (a) and (b) but for θ_2. The purple dots show the parameter default values. The x-axes show running number of iterations, i.e., how many ensemble forecasts that have been used. ΔE_m is used as the cost function, and the levels of complexity are summarised in Table 2. EPPES is used as the optimiser, the ensemble size is 50 members and the forecast range 48 hours.

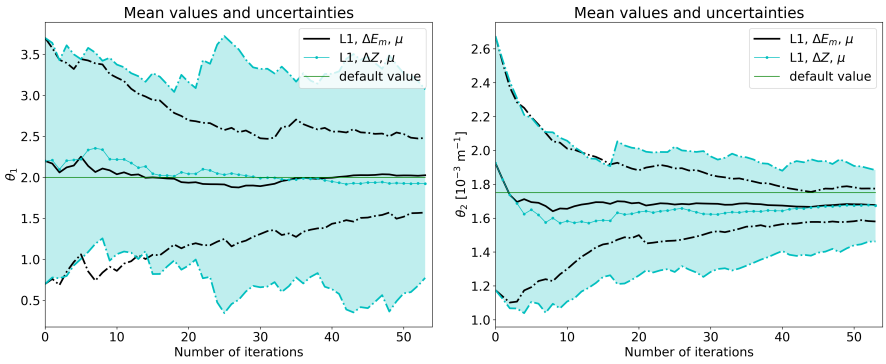

Figure 2. Convergence tests with different cost functions. Convergence of θ_1 on the left and θ_2 on the right. The x-axes show running number of iterations. Solid black lines show the evolution of distribution mean values (μ) and black dash-dotted lines the mean values ± 2 standard deviations when ΔE_m is used as cost function. Cyan dotted lines and shading in the background show the same for ΔZ. Default value shows the fixed parameter value used in the default model. Both convergence tests are L1 tests with 50 ensemble members and 48 hour forecasts. EPPES is used as optimiser.

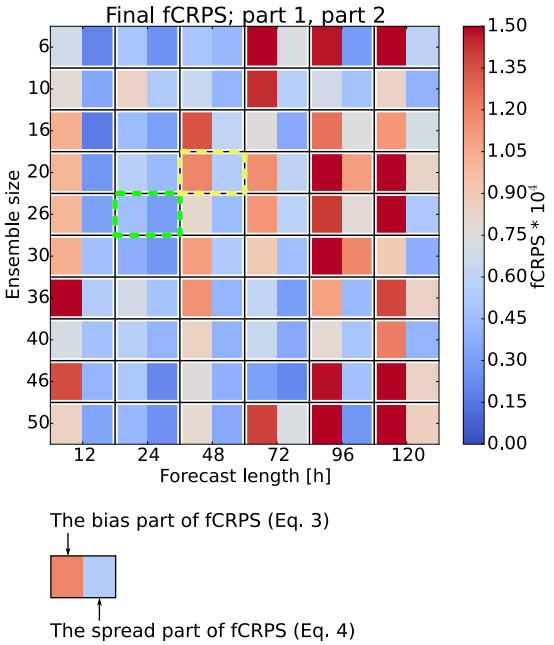

Figure 3. Components of fair CRPS from the final iteration of the convergence tests with various forecast ranges and ensemble sizes. In this example, the optimisation algorithm is EPPES and the parameter is θ_2. The left-hand side of each block represents the average distance of the parameter values from the default value (equation 3), and the right-hand side represents the spread of the parameter value distribution (equation 4). Low values and blue colours of both sides of the blocks indicate good convergence. Green and yellow boxes highlight the tests repeated in Figure 5.

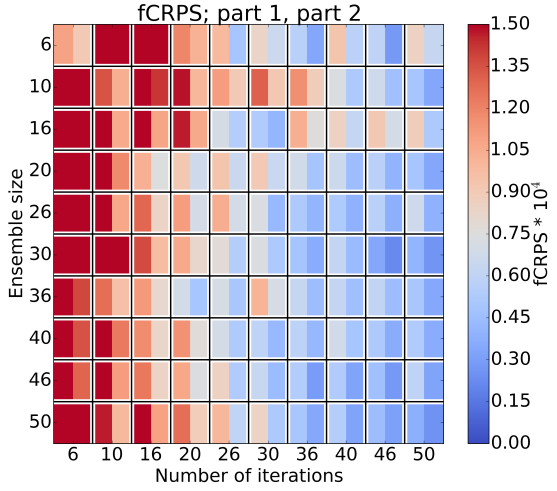

Figure 4. Evolution of fCRPS of θ_2 in convergence tests with L2, ΔE_m, EPPES, 24 hour forecasts and various ensemble sizes. The interpretation of the blocks is the same as in Figure 3. The number of iterations indicates how many iterations of the algorithm have been done, or in other words how many ensemble forecasts have been run. Components of fCRPS have been calculated using equations (3) and (4).

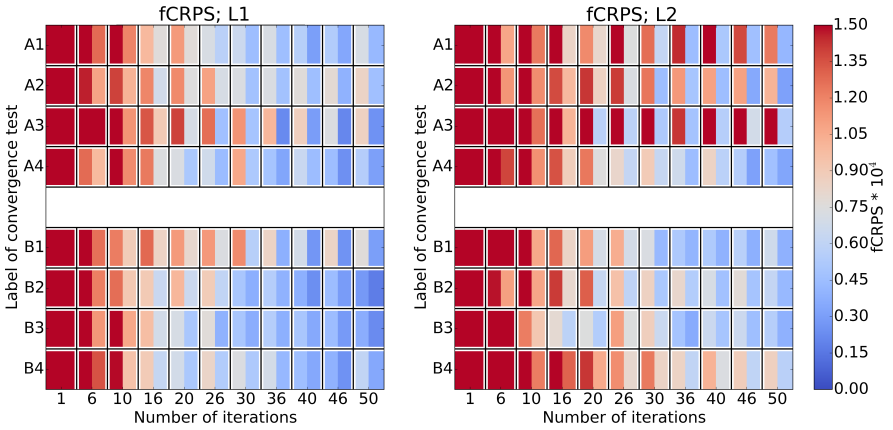

Figure 5. Evolution of θ_2 in repeated convergence tests with two selected forecast range – ensemble size combinations highlighted in Figure 3. The level of complexity is L1 on the left and L2 on the right. Tests A1 to A4 have been run with forecast range of 48 hours and ensemble size of 20 members, and tests B1 to B4 with 24 hours and 26 members. EPPES was used as an optimiser in these examples. Components of fCRPS have been calculated using equations (3) and (4).

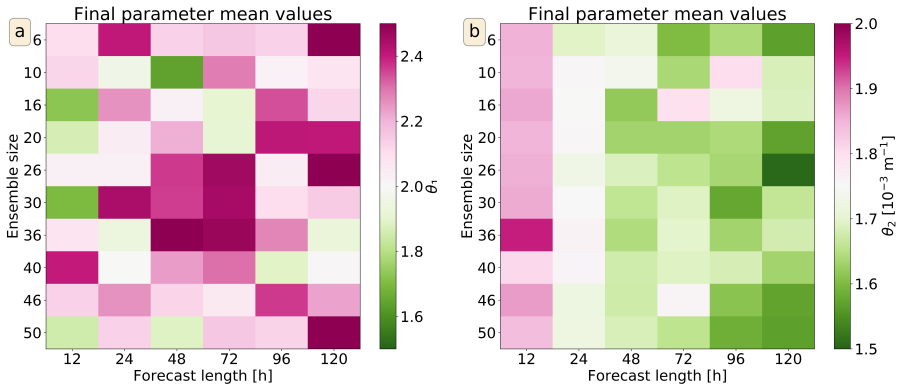

Figure 6. Mean values of the parameter distributions proposed by EPPES at the end of the convergence tests. Mean values of θ_1 are in (a) and mean values of θ_2 in (b). Purple (green) colour means that the final mean values are larger (smaller) than the default value.

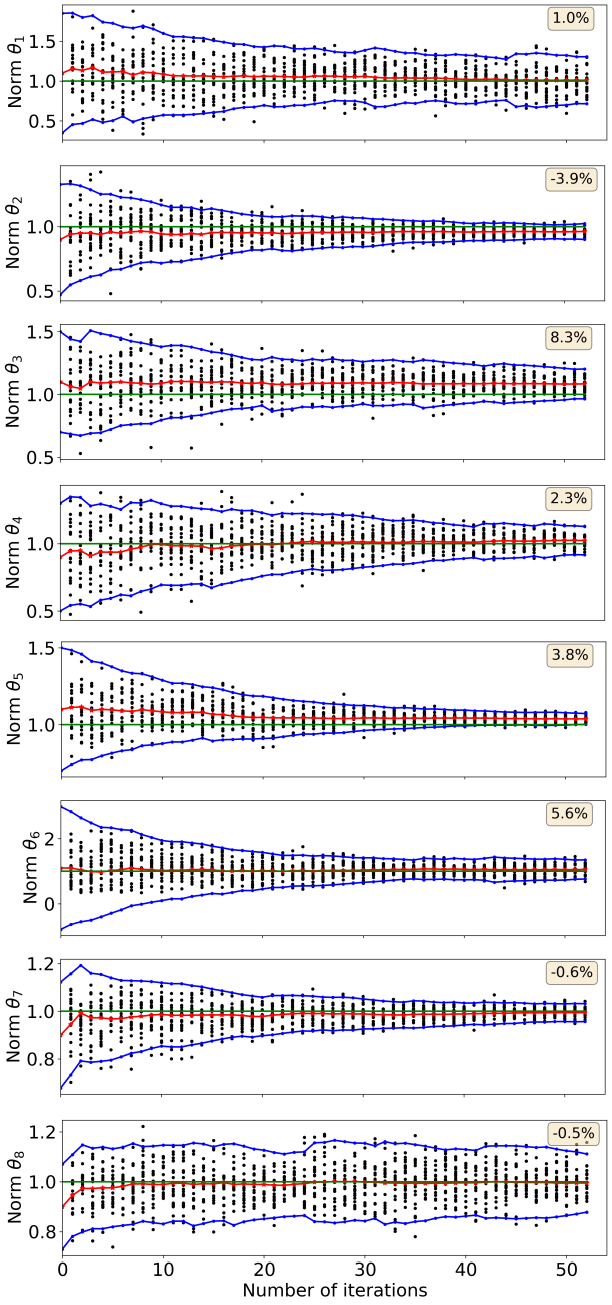

Figure 7. Progress of the convergence in the eight-parameter test. The parameter values and uncertainties have been normalised with their default values. Black dots show sampled parameter values, red line with stars shows parameter mean value, blue lines with dots show mean value ± 2 standard deviations and the green line shows the default parameter value that is 1.0 due to the normalisation. The text boxes indicate the remaining parameter off-set, which is the relative distance between the final parameter mean value and the default value. Initial parameter off-set is randomly plus or minus 10 %.

Parameter	Default value	Short description
ENTSHALP (θ_1)	2.0	Entrainment rate scaling factor for shallow convection
ENTRORG (θ_2)	$1.75 \cdot 10^{-3}$ m^{-1}	Entrainment per unit length for deep convection
DETRPEN (θ_3)	$0.75 \cdot 10^{-4}$ m^{-1}	Turbulent detrainment per unit length for deep convection
RPRCON (θ_4)	$1.4 \cdot 10^{-3}$ s^{-1}	Conversion factor from cloud water/ice to rain/snow
RDEPTHS (θ_5)	20000 Pa	Depth of layer for shallow convection
RMFDEPS (θ_6)	0.3	Fractional massflux for downdrafts at level of free sinking
RHEBC (θ_7)	0.9	Critical relative humidity below cloud for evaporation
ENTRDD (θ_8)	$3.0 \cdot 10^{-4}$ m^{-1}	Average entrainment per unit length for downdrafts

Table 1. Parameters of the convection scheme of OpenIFS. θ_1 and θ_2 are used the most in this study.

	Number of parameters	**Different initial conditions**	**Stochastic physics (SPPT)**
Level 0 (L0)	2	No	No
Level 1 (L1)	2	Yes	No
Level 2 (L2)	2	Yes	Yes
Level 3 (L3)	5	Yes	Yes

Table 2. Summary of convergence tests with different degrees of complexity.

Parameter	Mean	Variance	Lower bound	Upper bound
ENTSHALP (θ_1)	2.0	0.5625	0.5	6.0
ENTRORG (θ_2)	1.75e-3	1.40625e-7	1e-4	1e-2
DETRPEN (θ_3)	0.75e-4	2.25e-10	1e-5	1e-3
RPRCON (θ_4)	1.4e-3	7.84e-8	1e-4	1e-2
RDEPTHS (θ_5)	20000	1.6e7	1000	60000
RMFDEPS (θ_6)	0.3	0.08	0.1	0.6
RHEBC (θ_7)	0.9	0.01	0.5	1.0
ENTRDD (θ_8)	3.0e-4	0.65e-9	3.0e-5	3.0e-3

Table A1. Initial values of the convection parameters for EPPES.

Namelist object	Value	Explanation
maxn	5	Length of memory in iterations
maxstep	0.05	Maximum change of parameter mean value in one iteration
lognor	0	Use log-normal distribution, 0=no
useranks	1	Ranking of cost function values instead of using values themselves

Table A2. Other settings of EPPES.

Parameter	Lower bound	Upper bound
ENTSHALP (θ_1)	1.0	4.0
ENTRORG (θ_2)	1.25e-3	2.25e-3
DETRPEN (θ_3)	5.0e-5	1.0e-4
RPRCON (θ_4)	1.0e-3	1.8e-3
RDEPTHS (θ_5)	15000	25000

Table A3. Initial parameter value search area of DE.

Namelist object	Value	Explanation
F	0.5	Control for amplification of differential variation
CR	0.9	Crossover probability
JP	0.1	Probability of generation jumping
mutation_type	2	Use the best parameter vector in mutation
scale_factor_type	5	Scale factor randomisation scheme
F_l	0.5	Lower boundary for scale factor F
F_u	1.0	Upper boundary for scale factor F
pop_function	positive	Limits parameters to be positive
Jttr	0.01	Scale factor randomisation

Table A4. Other settings of DE.