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

Lauri Tuppi, Pirkka Ollinaho, Madeleine Ekblo, Vladimir Shemyakin and Heikki Järvinen

We would like to thank the reviewers for their critical and constructive comments. We agree with most of the comments and we have made changes accordingly.

We go through each comment one by one. Structure of the final response for each comment is as follows:
(1) original comment from referee
(2) author's response
(3) changes in manuscript.

Comments by Peter Düben are considered first and comments by Joakim Kjellsson after that.

Comments by Peter Düben and the authors’ answers:

(1) The paper is providing a recipe how to optimise parameters within an ensemble NWP system. Such an optimisation is very difficult to realise and the paper, which is based on many years of experience in parameter optimisation, provides essential guidelines how to do it properly and should therefore be published. However, the presentation of the paper can still be improved and I provide suggestions in the following. The English language could also be improved.
(2) We thank for the feedback, and we have made numerous modifications according your comments. The comments of the second reviewer in particular helped to improve the English language. We also found some clear errors, and now we hope that readability of the text has improved.
(3) Exact locations of modifications to the manuscript can be found in the comments below. The English language has been improved at least on lines 33, 102, 221, 256-258, 263-265, 331-339, 352, 360 and 362.

*[1]
(1) l31: "Semi-realistic" What does this actually mean? I guess all tests in the paper are realistic?
(2) Different degrees of realism can be thought as a scale. At one end there are idealistic experiments, such as tuning experiments with Lorenz-95 model. At the other end there are fully-realistic tuning experiments with full NWP model and with analysis or other real world data as the reference. Our experiments are semi-realistic since they do not use reference data representing the real world but artificial data generated with a control model with fixed parameter values. Convergence tests can be thought being somewhere between idealistic and fully-realistic experiments.
(3) No modifications made to the manuscript.

*[2]
(1) Page 2: I do not find the list of "synopsis guidance" very useful. It is not clear to me what the text in the brackets is meant to be. More detail? The results of this study? Readers should be told somewhere what they should take from this list. I would recommend to replace this list by a list that provides information on the degrees-of-freedoms that are important for optimisation (#parameters, forecast lead time, #ensemble members, minimisation algorithm, #initial conditions, cost function...). You could also think about a table to add information about advantages and disadvantages when increasing or decreasing the degrees-of-freedoms. You could then have another
list of the "boundary conditions" including a couple of points from your list (reproducibility
required? optimisation target known? computational resources available?...).
(2) We are sorry to hear this comment because we thought it would be instructive for reader to first
get first a synopsis, before dwelling into more details and thus get a feel whether the article answers
his/hers concerns. We have made an effort to amend the text to include the Reviewer’s message.
(3) Lines 34-35 and 49-67 have been rewritten.

*[3]
(1) End of introduction: It would help to provide a very brief overview on the sections that will
follow.
(2, 3) We have added a short introduction for the coming sections. It can be found on lines 67-69.

*[4]
(1) Section 2.2.: Maybe I have missed this but do ensemble members use different initial
conditions? Ahh, I found it later on in Table 2. But this information seems to be relevant earlier.
(2) The purpose of section 2.2. is to present the ensemble forecasting tool that was used in the
convergence tests. We agree that readers might want know already at this point that how are the
convergence tests run. Section 3 is dedicated to explain the anatomy of the convergence tests and
ensemble forecasts therein. We think that this separation is after all the clearest way to first present
the tools and then what was done with them. This way the readers will have a fresh image of the
structure of the convergence tests in the memory when they enter to the results section.
(3) We decided not to modify the manuscript.

*[5]
(1) l87-91: I do not understand the discussion. Please re-word and explain jitter and dither...
Personally, I would suggest to have more information on the optimisation methods but I leave this
decision to the authors.
(2) We have rewritten the details of DE in a different way so that readers not familiar to DE would
understand. Jitter and dither act to increase diversity of the parameter population so that they cause
small random variation in the parameter values so that DE does not stick to some specific values.
Jitter corresponds to the approach when the scale factor is randomized 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 randomized for each single
component of the mutant vector in the mutation process. The two optimisation methods themselves
are not in the centre of this study, and very detailed description of the methods can be found from
the references. It was mentioned in the introduction that the general guideline of algorithmic tuning
should apply to any ensemble based optimisation methods. EPPES and DE are merely examples.
We could have used different methods but we had access to these two.
(3) Respective modifications can be found on lines 115-124.

*[6]
(1) l98: Have I missed something? What is the control and what is the perturbed in this case? Do
you compare against the default parameter values? Or re-analysis? At l106 you refer to pseudo-
observations which seems to be the same as the control? – Ahh, later I understood that you take the
default value as truth and optimise towards it. However, it is still unclear how you define pseudo-
observations. Are these point measurements? Or 3D fields? Do you add random noise to represent
measurement uncertainty?
(2) During the entire paper the comparison is done always against the model with default parameter
values. We acknowledge that usage of terms ‘control model’ and ‘pseudo-observations’ might be
confusing. Pseudo-observations mean output of the control model. Pseudo-observations mean
values taken from the control model, and they are 3D fields. For simplicity we do not simulate
measurement uncertainty. We unified the terminology.
Respective changes can be found on lines 131-133.

*[7]
(1) l100: It should be explained why it only probes a "small fraction" of the domain, why it requires interpolation... this may not be clear to all readers.
(2) As ΔZ probes only 850 hPa level is not sensitive to perturbations of geopotential in middle and upper troposphere where perturbations of deep convection has the most influential effects. Interpolation is required because of the grid structure of OpenIFS. OpenIFS has a terrain following hybrid sigma vertical coordinate, which means that the model levels are not aligned with pressure levels in the lower troposphere. In mountainous areas 850 hPa level is ambiguous since it is below the ground.
(3) We have modified the text to explain these points better on lines 136-139.

*[8]
(1) l119-120: I do not understand this.
(2) fCRPS consists of two parts; the first part measures how far the ensemble mean value is from the target value and the second part measures how uncertain the ensemble is i.e. how much there is spread. In the original version of fCRPS, the second part is subtracted from the first part. The philosophy of the original fCRPS is that it penalises ensembles having mean value far from the target and small spread, and mean value close to the target and large spread. However, when the mean is far from the target and the spread is large, the original fCRPS gives a good score even though from the tuning point-of-view the result is in fact bad so in some cases the score is misleading. Therefore, we use the parts separately in order to avoid the cases of misleading results.
(3) No modifications to the manuscript were made.

*[9]
(1) l124: During the first read, I was not sure whether you are using ensembles where each ensemble member is using a different parameter value. This could be clarified.
(2) Each ensemble member uses different parameter value.
(3) We have added a clarification to line 163.

*[10]
(1) l138: Why do you use different parameters for different tests?
(2) L0, L1 and L2 tests use two parameters (θ1, θ2), whereas L3 tests use five parameters (θ1 to θ5). There should have been L0 to L2 instead of just L2.
(3) The error has been fixed on line 179.

*[11]
(1) l188: "discrepancy of model versions" I do not understand this.
(2) The initial conditions have been generated with IFS cycle 43r3 while we used OpenIFS, which was part of IFS cycle 40r1. There are substantial differences between the physical parametrisations between the model versions so this "discrepancy of model versions" may have caused stronger-than-usual spin-up at the beginning of the forecasts. ECMWF collects and publishes the model changes in here https://www.ecmwf.int/en/forecasts/documentation-and-support/changes-ecmwf-model
(3) We added a reference to the web page on line 80.

*[12]
(1) l189: Why should it not converge for a non-linear response if you have enough statistics? At least slowly?
(2) We do not argue that it would not eventually converge but we were surprised that there was visible convergence even with this small statistics. Our cost function was of quadratic form so we
were expecting that the non-linear nature of the model destroys the signal coming from the parameter perturbations totally in longer forecast ranges but obviously the model develops large enough bias, which is detectable even though there are non-linear perturbations.

(3) No modifications were made to the text.

*(13)*

(1) l268: "In the six five-parameter convergence tests the parameter values converge toward the default values during the convergence tests in 20 out of 30 cases" It took me a while to realise that 5*6=30. This needs more explanation.

(2) We have re-worded this and emphasize that all six convergence tests with five parameters are considered at once.

(3) We have re-worded the discussion about the six five-parameter tests on lines 331-339.

*(14)*

(1) l325: CPU hours or node hours? I guess it would help if you could provide some rough information on the computer that was used.

(2) It is talking about core hours meaning the wall clock time of execution multiplied by the number of cores so we have re-worded it clearer. The computer used consisted of Intel Haswell nodes having 24 CPUs each.

(3) We changed CPUs to cores and added very short information about the computer on lines 402-403.

*(15)*

(1) Can you add a brief discussion of local or global minima in the parameter optimisation?

(2) We added a short discussion. Local minimisation of the cost function may happen in algorithmic tuning mainly if the initialisation of the parameter values is bad. If some parameter has too small uncertainty and too large distance to the optimum, other parameters may compensate the error, and the cost function becomes locally minimised. The issue about converging to different values depending on the forecast range is a different point. When the parameters are initialised appropriately and initial condition perturbations active, sticking to some local minimum of the cost function is unlikely.

(3) We have added a short discussion on lines 376-381.

*(16)*

(1) Would the convergence with complexity be an option for optimisation (upgrade from L1→L2→L3→L4 for finer and finer parameter ranges)?

(2) This is one option, which can be used. Especially if the number of parameters is relatively large (say 20), this could be very efficient way to make a shortcut into fine tuning of the parameters. However, the uncertainty of the parameters, which remains at the end, increases as complexity increases meaning that one must be careful to not pose too small parameter ranges because inflation of parameter uncertainty often proves to be more difficult than deflation at least for EPPES. Therefore, the user could get too small uncertainty at the end.

(3) No changes were made regarding this comment.

*(18)*

(1) Is overfitting a problem for lower complexity configurations?

(2) We agree that overfitting is very likely a problem, and this is actually one reason why we do not recommend to use the configuration with the lowest complexity (L0) as such in fully-realistic tuning. ‘Overfitting’ is better wording than ambiguous ‘unwanted results’ so we re-worded some places a bit.

(3) We have changed ‘unwanted results’ to ‘overfitting of the parameters’ on line 196.
(1) If you compare against reanalysis or observations, you will often need to use different initial conditions for the different ensemble members to achieve sufficient spread at the beginning of the predictions. How would this influence the discussion around L1-L4?
(2) Only those convergence tests with the lowest level of complexity (L0) use the same initial conditions. We mention in the text that L0 should be used only to check that the optimisation infrastructure does not have major flaws. In all other convergence tests (L1, L2 and L3) the initial condition perturbations are active so every ensemble member already has unique initial conditions. We also acknowledge that ensemble initial conditions sample the weather states better, and this was the reason to use them in L1, L2 and L3.
(3) No changes were made regarding this comment.

Comments by Joakim Kjellson and authors’ responses:

(1) Overall:
The manuscript describes and evaluates the usability of algorithmic tuning for weather prediction models, in this case the convection scheme in OpenIFS. The manuscript uses two methods although only one is discussed and tested in detail. I’m curious about the method, in particular when applied to low-resolution versions of atmosphere models, since it could be a good way forward to improve climate models. Given that OpenIFS is used in a few climate models and that similar convection schemes are used in other models, this paper can be of interest also outside the weather forecasting community. I recommend this paper for publication after some comments below have been addressed.
(2) We thank for the insightful comments. In particular, we thank for the effort in making concrete suggestions how to re-word certain sections of the manuscript better. We are glad to hear that our paper is potentially interesting also for climate modellers.
(3) Exact locations of modifications to the manuscript can be found in the comments below.

*[1]
(1) The paper is well structured and the results interesting. The writing needs some work, in particular the descriptions of experiments and the latter half of the results section.
(2) We have worked on making the unclear sections more readable. Several small and a couple of larger modifications have been made according to the comments below. We now hope that the manuscript is now easier to understand.
(3) Exact locations are shown in the comments below.

*[2]
(1) I also feel that the paper should clarify right from the start that any closure scheme, in this case convection, is only a parametrisation of the real world, i.e. there is no "true" parameter value.
(2) This is a very good point. The difference between the real world convection and parametrisation might not be clear to all readers so stating this point shortly is useful.
(3) Modifications can be found on lines 14-17.

*[3]
(1) The choice of 850 hPa geopotential as a cost function seems odd and the authors also state that they expected this to be a bad choice. While its good to compare a bad choice to a good choice, I think it would make more sense to have used something else, e.g. precipitation or total-column water etc. I don’t think all the experiments need to be re-done to accommodate this, but I’d like to know exactly why 850 hPa geopotential was chosen and if the authors think some other pressure level or single variable would be more appropriate.
We acknowledge that using a different level or different variables would have likely made the "bad" cost function better. The motivation for using this not-so-suitable cost function is two-fold: 1) We want to illustrate that simply targeting a single skill score of the system (ability to do give skillful predictions targeting a single model output field) is not enough for optimization purposes as it leaves too much room for compensating errors to take place. This is also highlighted in earlier work with EPPES in IFS and ECHAM5 contexts (Ollinaho et al. 2013a, b). We believe that had the choice been precipitation or TCWV instead of 850hPa pressure it would not really have affected the results. We motivate this line of thinking from the procedures of operational weather prediction centres, where model forecast skill changes are verified from multiple perspectives with a holistic view during model code updates (i.e. using different skill scores and multiple model fields). 2) We want to highlight that choosing the target for the optimization carefully really matters, in this sense having a very radical contrast helps to deliver the message. We also do show how fine tuning of the target for optimization also makes a difference (the forecast length at which we target the moist energy norm to be minimized).

We have emphasized that it was an intentional choice to make a radical comparison on lines 348-349.

*[4]
(1) Overall: The authors use "Firstly" and "Secondly" a few times in the paper. This should be replaced by "First" and "Second".
(2) We have changed all of these words accordingly.
(3) Modifications can be found on lines 135, 137, 211, 214, 252, 283 and 308.

*[5]
(1) Line 15: A closure approximates a complex process with an equation and a parameter. As an example, we approximate diffusion/mixing with some high-order derivative and a coefficient. I don’t think it’s correct to say 'the parameter values are not known exactly' because they are only approximations anyway. There is no magic value of e.g. autoconversion rate that will result in a perfect cloud scheme. What you are looking for is the optimal parameter, i.e. the one that results in the smallest error.
(2) We totally agree with the reviewer’s comment: what we search is the “optimal” parameter value rather than some hypothetical “correct” value. The text has been corrected to accommodate this comment.
(3) Modifications can be found on lines 14-17.

*[6]
(1) Line 45: How about time of year, i.e. is there a seasonal dependence of the optimal parameters. Also, how are the optimal parameters dependent on model resolution, if at all?
(2) We have been thinking about the seasonality of the parameters but we ended up in a conclusion that if any seasonality existed, it would mean that there would be some fundamental deficiency in the parametrisation itself. In case of OpenIFS we assume that the model developers have accounted for this possibility so we do not expect any seasonality in the parameter values. The effect of model resolution is a different case. Mauritsen et al. (2012) state that the model parameters must be retuned after every modification of the model. Upgrade of resolution is a major modification. However, it depends case-by-case how model resolution affects each model parameter but changes of a few percentage units are likely. Moreover, in our experiments, we do not expect seasonality since the reference is a control model with fixed known parameter values.
(3) No modifications made regarding this comment.

*[7]
(1) Line 54: I’m a bit confused as to the number of forecasts here. Each test is a 51 member ensemble, done for 52 weeks in a year. And Fig 1 looks like you are using 52 iterations. Does this
mean 51 members x 52 weeks x 52 iterations > 100 000 forecasts? Or is it 51 members and each
weeks is an iteration, i.e. 51 x 52 2600 forecasts? If it is the latter, then your cost function for early
iterations are based on how well OpenIFS simulates winter conditions, while iterations 30 are based
on how well OpenIFS simulates summer. I could imagine that the optimal parameter and the
convergence are quite different in different seasons, so you would probably want to estimate the
optimal parameter set for each season separately.

(2) In Fig 1 there are 50+1 member ensemble run once a week throughout the year meaning 52
ensemble forecasts. Therefore, the total number of forecasts is 51 x 52 = 2600 forecasts. We have
two reasons to not expect the parameter values to depend on the season. The first one is that we
expect that the developers of OpenIFS have taken any possible seasonalities into account in
designing the parametrisations. The second one is that we use control model with fixed parameter
values to generate the reference data used in the convergence tests. We do not use analyses or
reanalyses in this study.

(3) No modifications made regarding this comment.

*[8]
(1) Line 48: Could the authors please add a reference to a paper or documentation for OpenIFS.
(2) Standalone documentation of OpenIFS does not exist. However, the documentation of IFS can
be used instead.
(3) Reference to documantation of IFS cycle 40r1 added on line 74.

*[9]
(1) Line 54 (contd): I would also really like to see an estimate of the computational cost of these
optimisation experiments, e.g. number of core-hours and total number of simulated days.
(2) There already is a rough estimate of core hours and number of simulated days used in two
experiments in lines
(3) No modifications made regarding this comment.

*[10]
(1) Line 98: Is "dD = dx * dy"? Then why not use that? Or "dA" for area. Would make more sense.
(2) We decided to use uniform notation for equations [1] and [2]. Notation “dD” is used in equation
[2.2] of Ehrendorfer et al. (1999), and we decided to use the same notation.
(3) No modifications made regarding this comment.

*[11]
(1) Line 105: I’m confused about this equation. The individual terms seem to have different units. u
2 is m2/s2, while L 2 q 2 /(cpT ) would be in Joule, and the last term I think is in [kg/m3]?
(2) We acknowledge that the moist total energy norm is a collection of different cost functions
scaled so that their weight would be comparable. This also leads to different units with different
terms. As the moist total energy norm is somewhat artificial construction (Marquet et al. 2020),
which does not measure energy as such, units are not important. Our formulation of moist total
energy norm follows equation [2.2] of Ehrendorfer et al. (1999) with the exception that the surface
pressure is treated as in Ollinaho et al. (2014). This comment made us to notice that the temperature
term (c
92p
95
/Tr
106
)[T
108
]2
138
was missing from the equation, and we added it. Moreover, we have divided the
resulting number with the mass of the atmosphere so instead of 1/2 there should be 1/(2M
499
M
502
a
505
) where
M
68
a
71
is the mass of the atmosphere
218
Our calculations have been performed with the temperature term
53
and division by the mass of the atmosphere. Only the formula was wrong.
(3) We added the missing terms and explanations on lines 144 and 146.

*[12]
In eq [1], primes denote an individual ensemble member. In [2], primes denote the difference between ensemble member and pseudo observations. First, it is confusing that primes denote two different things on the same page. Second, please explain what pseudo-obs are.

Yes, the notation is confusing. We have replace prime with asterisk in equation [1]. Pseudo-observations mean the reference data generated with the fixed control model. Pseudo-observations are 3D fields. We have added a more clear explanation for the pseudo-observations.

*[13]*

What is "n"? Is it an index over all tunable parameters?

Yes, “n” is an index over all tunable parameters.

*[14]*

How do the authors measure convergence? Is it done using parameter mean or uncertainty? A "parametric" way could be to smooth curves of mean parameter values and then see where the derivative ∂θ/∂N approaches zero, where N is iteration number. Or do you simply judge this by eye here? I agree that L0 seems to converge the fastest (mean stabilises the quickest and uncertainty decreases quickest). On the other hand L1, L2, L3 converge equally fast in terms of the mean parameter value but L1 has the largest uncertainty in Fig 1b.

Here the convergence is judged by eye from Figure 1. We judge the convergence in two ways: first, how far are the mean values of both parameters from the default values, and second, how large are the uncertainties at the end of the convergence tests. The target is that the parameter mean value becomes the same as the default value, and uncertainty becomes smaller than at the beginning. Therefore, we think that parametric way to measure convergence is not needed. It is true that in L1 test parameter 1 happens to retain large uncertainty during that test. However, we expect that this is caused by random variation during the test as each test has been done only once in Figure 1. Parameter 2 shows less signs of random variation as it is more sensitive to the cost function than parameter 1.

*[15]*

What is meant by "fully-realistic"? L3 should be the most "realistic" in the sense that it better represents the full complexity of the entire OpenIFS model. "Fully realistic" would be optimising all parameters (probably > 100) in OpenIFS with SPPT?

Here “fully-realistic” means using analyses, reanalyses or observations as the reference data. It is true that the usage of word “fully-realistic” might be confusing. We have checked all sentences where the word has been used, and reworded if it was used for something else than referring to “genuine” model tuning with real observations or analyses as the reference data. We changed the word “realism” to “complexity” in order to decrease the likelihood of confusion.

*[16]*

again, "fast convergence" based on what metric?

The convergence has been judged by eye from Figure 2: how far are the final parameter values from the default values, and how much uncertainty is left.

*[17]*

How about changing the sentence to "is explained by two factors. First ... " followed by "Second ...".
We have modified accordingly lines 211-213.

*[18] (1) Line 171: I would remove this sentence and then start the next sentence with "... recommend using a more comprehensive cost function which accounts for more than one... ".
(2, 3) We have modified accordingly line 218.

*[19] (1) Line 174: Since you are tuning convection parameters it would be a more fair comparison to use a $\Delta Z$ taken at 200 or 300 hPa. I'm not saying it needs to be done, but it is worth noting that it would be a better comparison.
(2) The point of choosing $\Delta Z$ with 850 hPa level was to show an example of a simple but very bad cost function. This was an intentional selection for the comparison against the moist total energy norm. We acknowledge that 200 or 300 hPa could lead to a better cost function but it would not change the message we want to deliver. The point is to show that putting effort on careful selection of cost function pays off. Comparison pros and cons of multiple different cost functions is out of scope of this paper.
(3) No additional modifications were made.

*[20] (1) Line 176: I would change the first sentence to "The process of finding a forecast range ... " and then merge with the next sentence so that it ends "of computational resources is done in two steps." The next sentence would be "First ... forecast ranges. " followed by "Second we take the forecast range ... ensemble size. ".
(2, 3) We have modified lines 227-229 as suggested.

*[21] (1) Line 179: Merge "Using L2 realism" with either the previous or following sentence.
(2) We have merged it to the previous sentence.
(3) Modifications can be found on line 229.

*[22] (1) Line 182: I'm struggling to understand what is done here. Do you take the final converged state from Fig 1 c,d, then set parameters to default, and compare how the estimated optimal parameters compare to those you got from Fig 1 c,d? Or do you run a convergence test and compare to the default parameter value as in eq 3? If you are doing the former, please explain this more clearly. If you are doing the latter, then why is low bias good? A large bias could mean that the tuning process worked really well and you got a more optimal parameter value.
(2) Here we take the final parameter values proposed by EPPES and then use equations [3] and [4] to calculate how much the mean value of the parameter deviates from the default value, and how much there is spread among the parameter values so we are doing the latter. Bias is an absolute value of the distance between the mean and default values. Spread is a scaled absolute value of how far each parameter value is from the other values within the ensemble. In our convergence tests bias is always calculated respect to the default parameter value, which is the target of convergence. Therefore, low bias means that the average of the proposed parameter values is close to the target, which means that location of the parameter value distribution is good.
(3) No modifications were made to the text.

*[23] (1) Line 186: Could it also be due to the fact that you are focusing on convection parameters and that convection is short lived? If you run 72 hours then you get an influence of synoptic variability...
which adds extra complexity. If you were tuning orographic wave drag you might want longer forecasts?

(2) Our explanation and your comment actually mean the same. Convection is short lived, which means that the response to perturbations stays approximately linear for a relatively short time. We have not studied the behaviour of other parameters besides the convection parameters but we think that it is likely that different forecast range would be optimal for different parametrisation schemes.

(3) No modifications were made to the text.

*[24]*
(1) Line 194: Change to "We now focus on the forecast range of 24 hours and perform convergence tests with ... ".
(2, 3) We have re-worded line 248 accordingly.

*[25]*
(1) Line 195: Change to "Fig 4 indicates that convergence tests with ensemble size > 20 are stable since convergence tests with smaller ensemble sizes do not show... ".
(2, 3) We have re-worded lines 249-250 but changed “since” to “while”.

*[26]*
(1) Line 197: What is meant by the two sentences starting with "sampling variance". What is "sampling variance" here?
(2) Here “sampling variance” refers to the random variability caused by small sample size: if the ensemble size is small, on some iterations all truly good parameter values might lead to bad scores by chance, which hampers the convergence of the parameters.
(3) No modifications were made to the text.

*[27]*
(1) Line 198: Change to "Fig 4 also enables comparison ... "
(2, 3) We have modified line 252 accordingly.

*[28]*
(1) Line 201: Change to "not seem to be necessary to achieve good convergence."
(2, 3) We have modified lines 256-257 accordingly.

*[29]*
(1) Line 202: Change to "Results here are for θ 2 but the same conclusions can be drawn from θ 1 ..."
(2, 3) We have modified line 257-258 accordingly.

*[30]*
(1) Line 211: I would remove this sentence.
(2, 3) We removed the sentence from line 268.

*[31]*
(1) Line 213: The black box In Fig 3 is clearly visible, but the gray one is not. Please use some other colours, e.g. green and yellow, to make sure the boxes stand out. Also explain in the caption for Fig 3 what the boxes are.
(2) We have replace the colours with green and yellow. We have also added an explanation that the boxes highlight the tests repeated in Figure 5.
(3) Caption of Figure 3 has been modified.

*[32]*
(1) Line 217: Rewrite to "shows that both L1 set-up s yield fairly reproducible convergence."
(2, 3) We have modified lines 274-275 accordingly.

*[33]
(1) Line 224: L1 does not use SPPT, i.e not stochastic in OpenIFS. So why are A1-A4 not identical? Are the initial conditions perturbed differently each time, or is there some other stochastic component activated?
(2) The reason why even L1 or L0 test are not identical is that EPPES takes the random number seed from the wall clock time so the proposed parameter values are different in each convergence test.
(3) No modifications were made to the text.

*[34]
(1) Line 226: The language switches from present to past tense here. Please commit to one throughout the paper.
(2) We changed all those parts to present tense where the change was meaningful. However, in some parts of the text, we use the past tense as a rhetoric means or when we extend discussion of experiments done. In Section 1 we use the past tense in order to advertise what we found. Section 4.5. begins with discussion about tests performed in Sections 4.1. to 4.4. so beginning the section with past tense is natural. Most of Sections 5 and 6 also discuss about the tests in sections 4.1.-4.5. so in those parts using past tense is also natural.
(3) Modifications were made on lines 284, 312, 313 and 366.

*[35]
(1) Line 229: I would rewrite sentence to "The opposite is true for θ 1." Also, I would like to see the plots of this (like Fig 1) in a supplementary materials or appendix.
(2) We have modified the sentence. In our opinion additional figures are not needed since the way how the parameters tend to converge respect to forecast range is clearly seen in Figure 6, which shows the end results of the array of L2 convergence tests. Lines 283-287 serve as an introduction to Figure 6, which is discussed in further details after that.
(3) The sentence has been rewritten on line 286.

*[36]
(1) Line 231: I would replace "(to the left)" by Fig 6a and similarly for the other panel. Then I would add letters a,b to the two subplots.
(2, 3) We have modified the figure, the figure caption and lines 288-289 accordingly.

*[37]
(1) Line 231 232: Words "especially" and "actually" are superfluous and should be removed.
(2,3) We have removed those words from the entire manuscript (lines 289, 290, 301 and 436).

*[38]
(1) Line 232: It is interesting that θ_2 depends so strongly on forecast length, but it is also interesting how it is not so sensitive to ensemble size. Could this be an effect of not using SPPT so that the system lacks spread? Also, why is θ_1 results all over the place? Could the authors comment on this?
(2) We did some convergence tests both with SPPT switched on and off on to see if the parameters then behave similarly as in Figure 6. SPPT does not have a large effect on how the parameter mean values tend to converge so it is unlikely that spread has a large role in the dependence. The amount of spread in ensemble forecasts used for parameter tuning is not as important as in operational forecasting for example. Spread itself does not help to find good parameter values. In fact, Figure 5 shows that SPPT merely hinders the convergence as the uncertainty decreases slower. The cost
function is less sensitive to variation of $\theta_1$ so there is also more variability in the results.
Sensitivity of the parameters is discussed further in Section 5.
(3) Lines 355-359 have been modified.

*[39]
(1) Line 232: I think the rest of this paragraph needs a good rewrite. Here I would rephrase to something like "We now examine the cost functions for two sets of ensemble forecasts (please specify length), one using the default parameters and one using parameters obtained from the optimisation (Fig 6)."
(2) We have rewritten accordingly. The length of the ensemble forecasts is six days, but $\Delta E_m$ is evaluated every six hours in order to see how it behaves respect to forecast length.
(3) We have modified lines 258-259 accordingly.

*[40]
(1) Line 236: Rewrite to "Results show that globally optimal parameter values are different from the their respective default values ..."
(2, 3) We have modified lines 291-293 accordingly.

*[41]
(1) Line 241: The sentence includes a statement "we are unsure whether .. " but ends with a question "does the dependency...". Split the sentences.
(2, 3) We have modified lines 301-302 accordingly.

*[42]
(1) Line 249: If you use "First" in a previous sentence, you should use "Second" here.
(2, 3) We have modified line 308.

*[43]
(1) Line 250: "with respect to"
(2, 3) We have modified line 311 accordingly.

*[44]
(1) Line 250: Merge sentences.
(2, 3) We have modified lines 311-312 accordingly.

*[45]
(1) Lines 253-256: This part is very vague. For instance, do you mean that EPPES and DE are both robust? Or is one robust while the other is not? What is excessively large? How can a user know this in advance?
(2) - We mean that one needs to know and understand the nature and main points of the parametrisation to be tuned. Even the best tools available will not work without careful planning of the tuning experiments. Both of the algorithms are robust but DE is more robust than EPPES since it can converge parameters that are further away from the optimum than EPPES. - Usually in tuning of NWP models, the parameter off-set is less than 10%. However, it can be larger in a case of a completely new parametrisation scheme. With excessively large off-set, we mean of an order of magnitude off-set. - In case of retuning old parametrisation schemes, excessively large initial parameter off-set is very unlikely as the old optimal values are known, and usually the change is less than 10%. At least theoretically in case of new parametrisation schemes, some parameters might be difficult to quantify with other methods than trial-and-error, so then the user can expect large initial off-set. In this case we recommend visual inspection of the model output before tuning; the output must be physically realistic.
(3) No modifications were made to the text.
This makes sense if I'm only looking to optimise 1-2 variables. But if a forecasting centre wants to optimise their entire convection scheme (10 parameters) they would need to go for something larger.

We want to point out that we designed the recipe with two parameters but tested it with five and eight parameters. The results with eight parameters are shown in Figure 7, and the convergence is satisfactory.

No modifications were made to the text.

How are the five-parameter tests different from the L3 tests before?
The only difference to L3 tests before is that SPPT is deactivated.
No modifications were made to the text.

Which additional three parameters are used?
They are explained in Table 1 (RMFDEPS, RHEBC and ENTRDD).
We added reference to Table 1 on line 328.

Remove last sentence and make first sentence on line 263 be "...demanding convergence tests with EPPES."
(2, 3) We have modified lines 325 and 329-330 accordingly.

When you say 20 of 30 or 25 of 30, which ones do you mean? Do some variables converge to off-default variables more often than others? A plot or some more text would help here.
Mean values of both \( \theta_1 \) and \( \theta_3 \) converge to off-default values for four times, and of \( \theta_4 \) for two times. The mean value of each parameter is more than two standard deviations away from the default value once in the six tests. We have added a couple of sentences about those parameters converging the worst.
Modifications have been made on lines 331-339.

"uncertainty" = "model spread"?
Uncertainty means the width of the parameter distribution proposed by EPPES.
No modifications were made to the text.

Do you compute cost function of difference to control run? So we would expect variables to converge to default values?
Yes, control model with default parameter values is used as the reference throughout the paper so we would expect the parameters to converge to their default values.
No modifications were made to the text.

How is the uncertainty chosen? Is this taken from previous optimisation runs, or do you use a guess?
The initial uncertainty is a guess. However, in order to be conservative, we tested that the initial uncertainty is unlikely to allow completely unrealistic model states prior to the convergence tests.
(3) No modifications were made to the text.

*[54]
(1) Line 273: This sentence adds no information. Remove it.
(2) We removed it from here, and also moved the following sentence to the discussion about convergence to off-default values above.
(3) Lines 340-341 removed. The remaining sentence was moved to lines 334-335.

*[55]
(1) Line 279: I disagree that all parameters converge. Parameter 8 does not seem to converge at all. Parameter 1 converges to some extent, but very slowly. How is convergence gauged? By eye? And by what metric, convergence of value or model spread?
(2) Looking at Figure 7, also parameter 8 converges: at the end the off-set is smaller than at the beginning and also the uncertainty decreases about 29%. However, you are correct that parameters 8 and 1 show least convergence. Convergence is judged visually from convergence of mean values (= whether they come closer to the default values or not) and decreases of spread of the parameter distributions.
(3) No modifications were made to the text.

*[56]
(1) Line 282: If it was expected to be bad, why not try another level? 300 hPa would have made more sense maybe, or using surface field like precipitation or total-column precipitable water etc.
(2) The root-mean squared error of 850 hPa geopotential was intentionally chosen to be an example of a cost function, which is very simple but does not work in this case. With the chosen example, we showed that careful selection of the cost function pays off. We think that choosing different level would not have changed the message. One must pay attention to the selection of suitable cost function. More comprehensive cost functions tend to perform better.
(3) Small changes can be found on lines 349-350.

*[57]
(1) Line 284: Could the authors please expand on this? Why are they not equally sensitive?
(2) In OpenIFS θ_1 is active only in the 200 hPa layer beginning from the Earth’s surface. This means that the direct effect of perturbation of θ_1 is confined to the lower troposphere. Moreover, the contribution of θ_1 to the cost function comes almost solely from the moisture term as shallow entrainment controls how the moisture distributes in the lower troposphere. θ_2 controls entrainment of deep convection so the effect is felt in entire troposphere. Perturbations of deep convection affect directly wind and temperature besides humidity, and these fields have been included into moist total energy norm. We added a short text to explain the main points.
(3) Modifications can be found on lines 354-359.

*[58]
(1) Line 295: Could it be that some parameters (e.g. convection which is fast) should be tuned with short forecasts while other parameters (e.g. gravity wave drag which is slower) should use longer forecast times?
(2) This might be possible if the user wishes to tune only one parametrisation scheme at once. However, if the user wishes to tune entire model at once, there is one potential obstacle: the fast perturbations of convection become non-linear and also make the signal of gravity wave drag untraceable when longer forecast time is used. How to make an optimal compromise is left for future studies.
(3) No modifications were made to the text.

*[59]
How do you guess the covariance matrix? If you start with a bad guess of the covariance, does that affect the results?

In the initialisation, only the diagonal is filled with variances, and all covariances are set to zero. In guessing of the initial variances, we made sure that ±2 standard deviation perturbation of the parameter does not lead to visually unrealistic model state. If the initial guess is very bad, then other parameters try to compensate the error leading to large jumps of parameter values at the beginning. The parameters may also converge to grossly off-default values.

No modifications were made to the text.

Table 1: It would also be good with a short description of how each parameter should impact the two different cost functions, which would help the argument on line 284.

In many cases it is not intuitive how the model parameters affect the cost function so figuring out the connections would require a substantial amount of experimentation. We disagree that in-detail studying of the processes how each parameter affects the model fields used by the cost functions falls within the scope of this study. In general, convection is not directly linked to 850 hPa geopotential but it is connected in a way or another either to wind, temperature, humidity or surface pressure.

No modifications were made to the text.

References


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

Lauri Tuppi1, Pirkka Ollinaho2, Madeleine Ekblom1, Vladimir Shemyakin3, and Heikki Järvinen1

1Institute for Atmospheric and Earth System Research / Physics, University of Helsinki, Finland
2Finnish Meteorological Institute, Helsinki, Finland
3School 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 $M(x, \theta)$ denote the time evolution model, where $x$ and $\theta$ are the initial state and the default model parameter vectors, respectively. A necessary condition for an algorithmic tuning process to converge is that $\theta$ is recovered when the tuning process is initialised with perturbed model parameters $\theta'$ 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 due to the fact that the closure parameter values are not known exactly. 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 synopsis guidance can be drawn up: Based on a very large amount of testing, the following general guidance can be drawn up.

- The level of realism (synopsis of our guidance: default model must be recovered in trivial testing when model parameters are off-set slightly; initial state perturbations tend to enhance convergence properties)
- The optimisation target (too simplistic target function may not allow convergence even in trivial testing, or convergence is very slow; convergence benefits of comprehensive target function formulation)
- Efficient use of computations (long convergence testing with a small ensemble and varying initial states is better than a short test with a large ensemble; in OpenIFS, 24 hour forecast range seems to be long enough for good parameter sensitivity and convergence)
- Reproducibility of results (24 hour forecast range seems optimal; at longer ranges than 24 hours, results are less repeatable)
- Can one trust on algorithmic tuning (not blindly, take it as expert-guided)
- Potential pitfalls (initial parameter off-set should not be too large, such as off by factor of two; initial parameter off-set and uncertainty should be proportional; be aware that optimal parameter values can be dependent on the forecast range applied)

**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 (cost 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) (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). Recalculation step has been designed to enhance convergence when the cost function is stochastic. Occasionally truly good parameter values lead to bad cost function values due to other perturbations in the forecast. However, the recalculation step passes the previous population for a new iteration, which is used to confirm that those parameter vectors, which lead to good scores on previous iteration, lead to good cost function values again. 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 ($\Delta Z$):

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

where $Z_{850}$ and $Z^*_{850}$ denote 850 hPa geopotential in the control forecast 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 $\Delta Z$ 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 $\Delta Z$ to perform sub-optimally. Firstly, 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 $\Delta Z$ is not (directly) sensitive to perturbations in the middle and upper troposphere where the most influential effects of perturbed convection are present. Secondly, 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. Finally, the 850 hPa level intersects with the ground level in mountainous areas.

The second cost function is the global moist total energy norm ($\Delta E_m$, 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

$$\Delta E_m = \frac{1}{2M_a} \int_D \int_\eta \left[ u'^2 + v'^2 + \frac{c_p}{T_r} T'^2 + c_q \frac{L^2}{c_p T_r} q'^2 \right] dD \frac{\delta p_r}{\delta \eta} d\eta + \frac{1}{2} \int_D \left[ R \frac{T_r}{p_r} \ln p'_s \right] dD,$$  \hspace{1cm} (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 $\eta$ 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 $\Delta E_m$ for practical use.

We expect that at short forecast ranges, the linkage between variations in the values of $\Delta E_m$ 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 $\theta_n$:

\[
fCRPS_1 = \frac{1}{M} \sum_{j=1}^{M} |\theta'_{j,n} - \theta_{d,n}|
\]

and

\[
fCRPS_2 = \frac{1}{2M(M-1)} \sum_{j=1}^{M} \sum_{k=1}^{M} |\theta'_{j,n} - \theta'_{k,n}|
\]

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 run with 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. The perfect score of each part of fCRPS is zero. 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 realism we use in the convergence tests. The different levels of realism 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. $\Delta Z$ is only tested at L1 level. Most of the effort is put on L2 testing with $\Delta E_m$ 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 realism, 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 $\theta_1$ and $\theta_2$ in L0 to L2 tests and parameters $\theta_1$ to $\theta_5$ 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, 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 realism complexity

We test how much realism complexity should be included in algorithmic tuning. Figure 1 shows four convergence tests with different levels of realism complexity, described in Table 2. As expected the parameters converge slower the higher the level of realism 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 realism complexity could lead to unwanted results 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 realism complexity behave quite reasonably. The uncertainty cannot vanish completely since some uncertainty is always present due to ensemble initial conditions. The only difference between L1, L2 and L3 tests is that more complex tests converge slower.

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 control 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 realism 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 so that more parameters are added 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 $\Delta E_m$ (shown with black solid and dash-dotted lines) and $\Delta Z$ (shown with cyan dotted line and shading in the background). Figure 2 shows that $\Delta E_m$ leads to much faster convergence than $\Delta Z$. The superior performance of $\Delta E_m$ is explained by two factors. Firstly, global integral of several variables catches the signal of parameter perturbations much better than a single level
measure. Secondly, 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.

The results showed that a more comprehensive cost function means faster and more reliable convergence. Therefore, we recommend in general using such a measure that a more comprehensive cost function that accounts for more than one atmospheric level and more than one variable. At least in case of convection parameters 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 possible likely that with some other parameters a cost function other than ΔE_m another cost function than ΔE_m might work better.

4.3 Finding the most efficient set-up

This subsection aims at finding a forecast range and an ensemble size that give satisfactory convergence with a minimal amount of computational resources. The consideration is taken at two steps: first, we compare convergence tests that use different forecast ranges, and then we concentrate on the forecast range with the fastest convergence in order to find optimal ensemble size. 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 realism complexity. We expect that the results obtained with such a high level of realism 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.

Figure 4 concentrates We now focus on the forecast range of 24 hours and shows the evolution of perform convergence tests with different ensemble sizes again measured with fCRPS. Firstly, Figure 4 indicates that convergence tests with large enough ensemble size \( \text{ensemble size} > 20 \) are stable while Three convergence tests having the smallest 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. Secondly, 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 justified by the computational resources point of view. Not seem to be necessary to achieve good convergence. Figure 4 shows the results only for \( \theta_2 \) but the same applies also to \( \theta_1 \). Results here are for \( \theta_2 \) but the same conclusions can be drawn for \( \theta_1 \), although the results with \( \theta_1 \) 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 of 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 whether 24 hours is the optimal forecast range for all parameters.

### 4.4 Reliability of convergence tests

This subsection shows additional evidence to use L1 level of realism and 24 hour forecasts. 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 at L1. However, when the level of realism 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

Firstly, 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 ($\theta_1$ and $\theta_2$) tend to converge to slightly different values depending on the forecast range used. $\theta_2$ tends to converge to a value smaller (larger) value than the default value when forecasts longer (shorter) than 24 hours are used. In this respect $\theta_1$ behaves like a mirror image to $\theta_2$. The opposite is true for $\theta_1$. However, at least $\theta_2$ 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 $\theta_1$ (to the left Figure 6a) and for $\theta_2$ (to the right Figure 6b). Especially convergence of $\theta_2$ seems to depend very strongly on the forecast range used in the convergence tests.

We tested whether it is actually possible to obtain smaller cost function values by running regular ensemble forecasts with the parameter values proposed by the algorithm (see Figure 6) and regular ensemble forecasts with default parameter values. 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 $\Delta E_m$. The tests were repeated with only initial condition perturbations active and with initial condition plus stochastic model perturbations active. Results indicated that the globally optimal parameter values and the parameter default values do not necessarily match. Results show that globally optimal parameter values are different from their respective default values even though the control 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. Firstly, 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. Then, which makes convergence to some local optimum is likely. Both algorithms showed that in such a case, the badly
initialised parameter remained practically unchanged while the other parameters appeared 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 realism complexity (at least initial condition perturbations, possibly also stochastic model perturbations)
– a comprehensive measure is used as a cost function ($\Delta E_m$ 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. Here we use EPPES as the optimisation algorithm.

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 five-parameter convergence tests, the parameter values converge towards the default values during the convergence tests in 20 out of 30 cases. $\theta_1$ converges to an off-default value in four of the cases as does $\theta_3$. $\theta_4$ converges to an off-default value twice. Furthermore, $\theta_2$ tends to converge to a slightly smaller value and $\theta_1$, $\theta_4$ and $\theta_5$ 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 parameters tend to converge in a certain way also in these tests. $\theta_2$ tends to converge to a smaller value and $\theta_1$, $\theta_4$ and $\theta_5$ to a larger value than their respective 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 $\theta_5$, the default value is outside of the uncertainty range. Additional dimensionality 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 tested 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 better 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 $\theta_1$ and $\theta_2$ are not equally sensitive to the components of the moist total energy norm. $\theta_1$ is more sensitive to specific humidity and $\theta_2$ to the other components. $\theta_1$, which is related to the shallow convection, is active only in the lowest 200 hPa layer of the model atmosphere. $\theta_1$ mainly affects how specific humidity is distributed in the layer. Therefore, contribution of $\theta_1$ comes almost only from lower tropospheric specific humidity. $\theta_2$ controls deep convection so it has direct impact on wind, temperature and specific humidity throughout the model troposphere. Therefore, contribution of $\theta_2$ 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 any 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 seemed optimal. When the forecast range is shorter, $\theta_1$ tends to converge to smaller values and $\theta_2$ to larger values than the default parameter value, whereas a longer forecast leads to $\theta_1$ converging to larger value and $\theta_2$ 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 CPU core hours for TL159 model resolution and 4200 CPU 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/pirkka/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: $\mu$, $\Sigma$, $W$, 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 $\theta = \{\theta_1, \ldots, \theta_n\}$ be the closure parameters. In EPPES, the prior guess it that the closure parameter follows a Gaussian distribution $\theta_i \sim N(\mu, \Sigma)$, where $\mu$ is the mean vector of $\theta$ and $\Sigma$ 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. Especially 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


Figure 1. Comparison of convergence tests at different levels of realism complexity. Panels (a) and (b) show the evolution of distribution mean value ($\mu$) and the mean value $\pm 2$ standard deviations uncertainty ($\mu \pm 2\sigma$) for $\theta_1$, and (c) and (d) show the same as (a) and (b) but for $\theta_2$. 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. $\Delta E_m$ is used as the cost function, and the levels of realism 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 $\theta_1$ on the left and $\theta_2$ on the right. The x-axes show running number of iterations. Solid black lines show the evolution of distribution mean values ($\mu$) and black dash-dotted lines the mean values ±2 standard deviations when $\Delta E_m$ is used as cost function. Cyan dotted lines and shading in the background show the same for $\Delta Z$. Default value shows the fixed parameter value used in the control 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 $\theta_2$. 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 $\theta_2$ in convergence tests with L2, $\Delta E_m$, 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 $\theta_2$ in repeated convergence tests with two selected forecast range – ensemble size combinations highlighted in Figure 3. The level of realism 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 $\theta_1$ are on the left in (a) and mean values of $\theta_2$ on the right 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%.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
<th>Short description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTSHALP ($\theta_1$)</td>
<td>2.0</td>
<td>Entrainment rate scaling factor for shallow convection</td>
</tr>
<tr>
<td>ENTRORG ($\theta_2$)</td>
<td>$1.75 \cdot 10^{-3}$ m$^{-1}$</td>
<td>Entrainment per unit length for deep convection</td>
</tr>
<tr>
<td>DETRPEN ($\theta_3$)</td>
<td>$0.75 \cdot 10^{-4}$ m$^{-1}$</td>
<td>Turbulent detrainment per unit length for deep convection</td>
</tr>
<tr>
<td>RPRCON ($\theta_4$)</td>
<td>$1.4 \cdot 10^{-3}$ s$^{-1}$</td>
<td>Conversion factor from cloud water/ice to rain/snow</td>
</tr>
<tr>
<td>RDEPTHS ($\theta_5$)</td>
<td>20000 Pa</td>
<td>Depth of layer for shallow convection</td>
</tr>
<tr>
<td>RMFDEPS ($\theta_6$)</td>
<td>0.3</td>
<td>Fractional massflux for downdrafts at level of free sinking</td>
</tr>
<tr>
<td>RHEBC ($\theta_7$)</td>
<td>0.9</td>
<td>Critical relative humidity below cloud for evaporation</td>
</tr>
<tr>
<td>ENTRDD ($\theta_8$)</td>
<td>$3.0 \cdot 10^{-4}$ m$^{-1}$</td>
<td>Average entrainment per unit length for downdrafts</td>
</tr>
</tbody>
</table>

Table 1. Parameters of the convection scheme of OpenIFS. $\theta_1$ and $\theta_2$ are used the most in this study.
Table 2. Summary of convergence tests with different degrees of realism complexity.

<table>
<thead>
<tr>
<th>Level</th>
<th>Number of parameters</th>
<th>Different initial conditions</th>
<th>Stochastic physics (SPPT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 0 (L0)</td>
<td>2</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Level 1 (L1)</td>
<td>2</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Level 2 (L2)</td>
<td>2</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Level 3 (L3)</td>
<td>5</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Parameter</td>
<td>Mean</td>
<td>Variance</td>
<td>Lower bound</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ENTSHALP ($\theta_1$)</td>
<td>2.0</td>
<td>0.5625</td>
<td>0.5</td>
</tr>
<tr>
<td>ENTRORG ($\theta_2$)</td>
<td>1.75e-3</td>
<td>1.40625e-7</td>
<td>1e-4</td>
</tr>
<tr>
<td>DETRPEN ($\theta_3$)</td>
<td>0.75e-4</td>
<td>2.25e-10</td>
<td>1e-5</td>
</tr>
<tr>
<td>RPRCON ($\theta_4$)</td>
<td>1.4e-3</td>
<td>7.84e-8</td>
<td>1e-4</td>
</tr>
<tr>
<td>RDEPTHS ($\theta_5$)</td>
<td>20000</td>
<td>1.6e7</td>
<td>1000</td>
</tr>
<tr>
<td>RMFDEPS ($\theta_6$)</td>
<td>0.3</td>
<td>0.08</td>
<td>0.1</td>
</tr>
<tr>
<td>RHEBC ($\theta_7$)</td>
<td>0.9</td>
<td>0.01</td>
<td>0.5</td>
</tr>
<tr>
<td>ENTRDD ($\theta_8$)</td>
<td>3.0e-4</td>
<td>0.65e-9</td>
<td>3.0e-5</td>
</tr>
</tbody>
</table>

Table A1. Initial values of the convection parameters for EPPES.
<table>
<thead>
<tr>
<th>Namelist object</th>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxn</td>
<td>5</td>
<td>Length of memory in iterations</td>
</tr>
<tr>
<td>maxstep</td>
<td>0.05</td>
<td>Maximum change of parameter mean value in one iteration</td>
</tr>
<tr>
<td>lognor</td>
<td>0</td>
<td>Use log-normal distribution, 0=no</td>
</tr>
<tr>
<td>useranks</td>
<td>1</td>
<td>Ranking of cost function values instead of using values themselves</td>
</tr>
</tbody>
</table>

*Table A2. Other settings of EPPES.*
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Lower bound</th>
<th>Upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTSHALP ($\theta_1$)</td>
<td>1.0</td>
<td>4.0</td>
</tr>
<tr>
<td>ENTRORG ($\theta_2$)</td>
<td>1.25e-3</td>
<td>2.25e-3</td>
</tr>
<tr>
<td>DETRPEN ($\theta_3$)</td>
<td>5.0e-5</td>
<td>1.0e-4</td>
</tr>
<tr>
<td>RPRCON ($\theta_4$)</td>
<td>1.0e-3</td>
<td>1.8e-3</td>
</tr>
<tr>
<td>RDEPTHS ($\theta_5$)</td>
<td>15000</td>
<td>25000</td>
</tr>
</tbody>
</table>

Table A3. Initial parameter value search area of DE.
<table>
<thead>
<tr>
<th>Namelist object</th>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>0.5</td>
<td>Control for amplification of differential variation</td>
</tr>
<tr>
<td>CR</td>
<td>0.9</td>
<td>Crossover probability</td>
</tr>
<tr>
<td>JP</td>
<td>0.1</td>
<td>Probability of generation jumping</td>
</tr>
<tr>
<td>mutation_type</td>
<td>2</td>
<td>Use the best parameter vector in mutation</td>
</tr>
<tr>
<td>scale_factor_type</td>
<td>5</td>
<td>Scale factor randomisation scheme</td>
</tr>
<tr>
<td>F_l</td>
<td>0.5</td>
<td>Lower boundary for scale factor F</td>
</tr>
<tr>
<td>F_u</td>
<td>1.0</td>
<td>Upper boundary for scale factor F</td>
</tr>
<tr>
<td>pop_function</td>
<td>positive</td>
<td>Limits parameters to be positive</td>
</tr>
<tr>
<td>Jtr</td>
<td>0.01</td>
<td>Scale factor randomisation</td>
</tr>
</tbody>
</table>

Table A4. Other settings of DE.