Response to reviewer 1

A. Farchi and M. Bocquet

July 24, 2018

We thank Reviewer 1 for insighful comments and suggestions.

1 Specific comments

Comment 1 *Do I understand correctly that your observations in the L96 setup are of every variable at every time step?*

Yes, with the L96 model every variable is observed at every time step.

So your discussion on pre-regularisation in some way is a discussion on whether to use model error or not? Perhaps this could be made more explicit in the main text.

Yes, pre-regularisation is in some way similar to the use of model error. This is already explicit in the text ("using a stochastic extension of the model").

If observations are of every variable at every time step then it will strongly constrain the posterior pdfs to be Gaussian and may well be influencing the results seen. This is alluded to in the main text but I think it would add insight to explicitly state this in the discussion in section 5.1 and that it is also addressed through the use of the barotropic vorticity model (which I understand uses a more sparse observational system in space and time).

Yes, the posterior pdfs are close to Gaussian pdfs, this is already stated in the last paragraph of Sect. 5.1. In Sects. 6.2 and 9.3, we have followed your suggestion and added a remark.

Comment 2 *The paper outlines quite a few different extensions to existing localisation methods. This is stated in the abstract and introduction and when I read the different algorithm descriptions in detail I could find the paragraph were the differences were noted. However, in general I wasn't really left with a strong feeling of where you had introduced new elements and what benefit they had brought i.e. Section 4.4.4 is entirely new work that has the best result for L96 for the state domain localisation but this is only remarked on in the final paragraph of the algorithm description section. If it was possible to include a small summary that highlights the new work and the improvement it brings, either within an already existing concluding section or as something separate, then I think it is a chance to bring your work to the fore. It would also explicitly demonstrate how setting individual schemes in to a general context can bring benefits. This could be split between the state domain localisation and sequential-observation localisation if that was the more natural division.*

The outline of the paper has been slightly modified. We have included, at the end of each theoretical section, a sub-section called "Summary for the LPF* algorithms". This sub-section includes highlights where we clarified what already existed, what is new and what was improved. It includes as well as some discussion about the numerical complexity and the asymptotic limit of the algorithms (which was demanded by the other Reviewer and the associate Editor). Thank you for the suggestion, this is a great addition.

2 Technical corrections

L97 *A capital N has slipped in to representatioN*

Done.

L386 *It should be 'in order to preserve part of the spatial structure held in the prior particles'*

Done.

L404 *So E^r has discontinuities?*

Yes, $\mathbf{E}^r$ has discontinuities, which is why we had to improve the resampling step. Remember that $\mathbf{E}^r$ is the matrix implicitely defined by step 5 of Alg. 1.

L543 *The sentence doesn't really make grammatical sense.*

The sentence has been corrected.

L625 *I assume this is 's' but it would be good to explicitly state it.*

Done. Thank you for the suggestion.

Response to reviewer 2

A. Farchi and M. Bocquet

July 10, 2018

We thank Dr. Stephen G. Penny (Reviewer 2) for his insighful comments and suggestions.

In this article, we describe and compare many different localised PF methods. Contrary to many articles in the PF literature, the algorithmic sections are detailed and many explanations are given about potential numerical choices. The numerical illustrations use several models, not only in one dimension, and an exhaustive exploration of the algorithm parameters is performed. Please keep in mind that this goes well beyond most studies on the subject.

1 General comments

Comment 1) *The relative costs between the methods should be calculated and compared, along with RMSE, in each section. Some discussion of costs is made in passing, but no quantitative analyses are offered until the end where a few methods are compared. I suggest extending this to each of the major direct comparisons at the end of the LPFx and LPFy sections.*

The numerical complexity of each method is now discussed in Sects. 4.5.2 and 7.5.2. For the BV model with high resolution, the computation times are reported in Table 5. For the low-order models however (standard L96, BV with coarse resolution), we did not to add the "companion plots" suggested by the reviewer for several reasons:

1. The computation time highly depends on many other factors than numerical complexity: implementation, programming language, processors architecture... With low-order models, these factors may be very important and therefore the computation time is often irrelevant.
2. In our configurations, the parameters are not reprensentative of realistic applications: the number of grid points is very small, the number of particles is very high, and the number of spatial dimensions is limited (to 2 in our case).
3. This would add approximately a dozen figures to an article which is already long.

In the PF literature, the only article in which we have seen some discussion about complexity and computation time is the one by Penny and Miyoshi (2016). Finally, we want to highlight the fact that conclusions regarding the computational cost of a method cannot be based on test series with low-order models. From our experience, the ranking of an algorithm in computational cost looks very different when using the one-dimensional L96 model or the two-dimensional BV model with high resolution.

Comment 2) *It would be useful to show the results of the EnKF baseline, both in RMSE and computation time.*

The EnKF scores were given in the text for the L96 model and shown in Fig. 17 for the BV model. We have added a new figure (Fig. 4) that shows the score of the EnKF with the L96 model. We have also added horizontal baselines in most LPF figures (Figs. 8, 13, 14, 15, 16, 18, 20 and 21).

Since the local PF variants are not outperforming the EnKF baseline, the authors should consider some special case scenarios in which the PF does outperform the EnKF as a motivation for continued development of the local PFs and to show why local PFs may also have advantages over standard EnKFs.

This is not true, several LPFs do outperfrom the EnKF in the standard configuration for the L96 model (see Fig. 20), which is a première to our knowledge. Besides, the point of this article is not to design LPF algorithms that beat the EnKF in every configuration, but rather to improve the design of LPFs. Contrary to many studies on the PF, we have chosen to use the dynamical models in standard configurations, which allows for a fair comparison with the EnKF.

Following your recommandation, we have added a test series in Appendix C, in a configuration built to make the EnKF fail. In this configuration, we use the same strongly nonlinear observation operator as Poterjoy (2016). However, the interpretation of these results is harder, because some legitimate question can be asked:

1. Is this configuration relevant for realistic models?

2. How good are the score of the LPFs? There is no baseline for comparison (the EnKF does not count since we are outside the scope of its assumptions).

Comment 3) *In general, I find the algorithm names confusing.*

The coding system for the LPFs looks complex. But please keep in mind that this system was designed to distinguish more than 20 different algorithms. Other studies that focus on a limited number of methods could use a much simpler coding system.

In fact, the coding system follows one simple principle: capital letters refer to the main algorithmic steps and subscripts are used to differentiate the methods. This is now explained in the caption of Tables 1 and 3.

The first half of the paper uses a complex coding system, while the second half credits the authors who developed the methods.

We find the criticism unfair, because every time we introduce a new method, we cite the authors that inspired our work if any. Both $\mathrm{LPF^x}$ and $\mathrm{LPF^y}$ algorithms follow the same convention, with different subscripts to refer to different methods. For the $\mathrm{LPF^y}$s, using 'P' and 'RK' as subscripts is a way to distinguish the two different propagation methods and not to credit the author who developed the method. We could have used 'h' and '2' (for 'hybrid' and '2nd order') but obviously this is harder to remember. It serves exactly the same purpose as the subscrits 'e' and 's' (distinction between optimal transport in ensemble space and in state space).

A more consistent and simpler naming convention would be nice from a reader's point of view, and should be used throughout.

Following your suggestion, the naming convention has been modified. The new system is consistent and as simple as it can be given the fact that it is used for more than 20 algorithms. It is explained in details in the caption of Tables 1 and 3.

In addition, a single table describing every algorithm name, what is does, and what section it can be found in, would also help to add clarity.

Description tables for the $\mathrm{LPF^x}$ and $\mathrm{LPF^y}$ algorithms have been added (Tables 2 and 4). Thank you for the suggestion; this is a great addition.

From what I can tell, the S(IR)^ xSR_ a method appears closest to that of Penny and Miyoshi (2016), as it uses smoothing of weights and adaptive "regularisation jitter" based on the ensemble perturbations, and I think this should be given proper credit, as one of the few LPF methods offered in the geophysical literature that has a combination of good performance and low computational cost.

We did cite Penny and Miyoshi (2016) when presenting the smoothing by weights step in Sect. 4.4.1. As shown by our new results with the high-dimensional BV model, the $\mathrm{S(IR)^xS_{PM}R_c}$ (new nomenclature), our generalisation of the LPF of Penny and Miyoshi (2016), does not have a favourable ratio accuracy / computation time.

2 Specific comments

L28 *Also cite Kalnay and Yang regarding the "Running-in-Place" method.*

We have added a reference to the RIP as an important precursor method. Note however that, contrary to the MLEF, the 4D-ETKF and the IEnKS, the RIP is not mathematically consistent.

L39 *comma after hybridisation*

Done.

L43 *[fewer] particles*

Done.

L60 *degree[s] of freedom*

Done.

L63 *geophysical system[s]*

Done.

L85 *I'm assuming y_k:0 is the set of all observations from time t=0 to time t=k, but perhaps you can state that explicitly.*

Yes, this was stated on the same line.

L91 *Of course, there are many different goals in data assimilation. This is a typical goal. My immediate reaction here is that the DA filtering problem consists in estimating pi_k+1/k. This is the goal at least, usually, to make a prediction. Perhaps you can say - "The DA filtering problem consists in estimating pi_k/k and pi_k+1/k with given realizations of y_k:0."*

The sentence has been changed to reflect the fact that estimating $\pi_{k|k}$ may not be the only goal of DA.

L97 *particle representatio[n]*

Done.

L111 *What do you mean by "pure ensemble transformations"?*

"Pure ensemble transformation" means that this is a transformation that act on the ensemble and that should ideally not alter the density. This is explained by the following sentence. However, we agree that the term "pure" can be confusing and we removed it.

L170 *I'm not sure I understand why this is remarkable. Could you elaborate?*

This is remarkable that in w^i_{k+1}, the dependance on $\mathbf{x}^i_{k+1}$ vanishes. We have added a reference to Doucet (2000) for clarification.

L177 *"... to more elaborate algorithms ..."*

Done.

L182 *"... models [have led] to weight degeneracy..."*

Done.

L196 *" ... it [might seem] surprising that, although MC method[s] have..."*

Done.

L214/217 *You switch tenses, first referring to Synder et al. (2008) as a set of authors, and then referring to Snyder et al. (2008) as a paper. Because "et al." means "and others", I prefer the former and recommend changing L217 to: "Snyder et al. (2008) [do] not illustrate..."*

Done.

L223 *"...optimal importance proposal [density]..."*

Done.

L227 *"... does not [primarily] come from ..."*

Done.

L238 *"... [elaborate] models ..."*

Done.

L252 *It seems awkward to begin a sentence with a variable name. Perhaps used instead: "The quantity tao^ 2 would then be defined using"*

The beginning of the sentence has been changed. Thank you for the suggestion.

L266 *While I appreciate the implication of calling this a 'discontinuity', there are some complications in defining a concept of continuity on a discrete model grid. Some discussion should be made regarding this point.*

Indeed, discontinuity here does not refer to the mathematical notion of continuity. Following your suggestion, we have added some explanation.

L281 *Perhaps you could list some of the past examples of this type via citation.*

Done.

L285 *Again, I suggest citing a few example of this type as well.*

Done.

L332 *Not within a circle, but within some general local region. A circle is a common choice.*

We changed the paragraph. Thank you for helping us clarify this point.

L367 *"and decrease[s] exponentially"*

Done.

L371 *The "size" of the blocks using what measure? Number of grid points?*

We clarified this point. Thank you for spotting the imprecision.

L386 *I think "hold" should be "held"*

Done.

L390-415 *I'm not sure if the point was adequately made that neighboring weights can be made arbitrarily smooth by letting the radius of the taper function (r_ s) get large. I.e. as r_ s goes to infinity, the global PF solution is recovered.*

The asymptotic limit of a LPF$^{\mathrm{x}}$ algorithm using smoothing by weights is now discussed in Sect. 4.5.4. When $r \to \infty$, $\mathbf{E}^{\mathrm{r}}$ is not necessarily equivalent to the global PF solution (because the resampling is independant at each grid point). When $r \to \infty$ and $r_{\mathrm{s}} \to \infty$, $\mathbf{E}^{\mathrm{s}}$ is not necessarily equivalent to the global PF solution (again because the resampling is performed independantly at each grid point).

In that sense, I'm not sure why the additional alpha smoothing step is made explicit.

We do not understant your concern about making the "alpha smoothing step" (we guess you mean Eq. (32)) explicit.

L543 *"only [a] big ensemble"*

The sentence has been changed.

L561 *"RMSE offers a" to "RMSE offer a"*

Done.

L562 *I'm not sure it is settled that the RMSE of the mean is an adequate measure of the PF performance, given that the distribution may not have the mean and mode equal.*

We do not entirely understand the point of this remark. One must keep in mind that PFs are suited to compute the mean state and not the mode. Besides, in these weakly non-Gaussian configurations, mean and mode should not be far from each other.

Further, if we are to adopt a PF solution over an EnKF, then we are acknowledging that the primary data assimilation goal is specifically not mean state estimation, but rather estimation of the state distribution.

We do not agree with this remark. Using a PF does not mean that we are not interested in the mean state. And, again, one must keep in mind that PFs are not suited to "estimat[e] the state distribution". Indeed, with a PF, we primarily have an estimation of

$$\int p(\mathbf{x}) f(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \tag{1}$$

for any test function f, but no estimation of $p(\mathbf{x})$.

L572 *"yield[s]"*

The sentence has been removed.

L595 *I don't understand what this first sentence means. What does it mean to have more information than the truth?*

Indeed there was a typo: one should have read "on average more informations than the observations". Sorry about that; it has been corrected.

P25-25 *It would be nice if Figures 4 and 5 were closer to the referencing text. Perhaps you can make that request of the editors.*

We will ask for this in the editing process.

P25 *It appears here that you are using a fixed parameter for the 'regularization jitter'. Have you compared this the LPF of Penny and Miyoshi (2016) that set this value adaptively based on the analysis ensemble spread?*

The discussion on "adaptative" resampling is located in the "coloured noise" section (5.8). In this section, we developped a method that is an extension of the method by Penny and Miyoshi (2016). We compared our method to that of Penny and Miyoshi (2016) (not shown in this article) and always found better accuracy with our extension. We have added a few sentences in section 5.8.2 about this.

L651 *I'm confused how the higher RMSEs of the S(IR_SU)^ xR algorithm indicates an efficient approach. Could the authors elaborate.*

The $\mathrm{S(IR_{su})^xR}$ algorithm is the only one that does not use the "adjustment-minimising" property. If it has a higher RMSE, we believe that it means that the "adjustment-minimising" property is efficient. Following your comment, we reformulated the statement.

What is the RMSE ratio used in Figure 9? Why does the figure caption say "RMSE" while the y-axis says "RMSE ratio"?

The RMSE ratio used in Fig. 10 (new numbering) is detailed in the second sentence of the caption "The scores are displayed in units of the RMSE of ...". The same kind of ratio is used in Figs. 9, 11 and 12.

L675 *I need a reminder at this point - E is the set of ensemble members and X is the set of perturbations around the ensemble mean? Are the x^ i in (46) the columns of E?*

$\mathbf{E}$ is the ensemble matrix (defined in Sect. 2), whose columns are the particles $\mathbf{x}^i$, which is a very common notation in ensemble DA. $\mathbf{X}$ is indeed the set of (normalised) perturbations around the ensemble mean, as defined two lines above. Reminders have been added, thank you for the suggestion.

L697 *change "as following" to "as follows"*

Done.

How is equation (48) different than (46)? Could you instead just say it is defined as in (46) with a new formulation for the Gaussian regularization jitter covariance matrix (49)?

Equation (51) is different from Eq. (49) (new numbering) because it uses the local weights w_n^i instead of the global weights w^i that do not exist in LPFs. We explained in the text why Eqs. (47) to (49) cannot be used (see the second paragraph of Sect. 5.8.2)

L702 *Am I interpreting these figures correctly in that the new proposed approaches are all mostly making the RMSE larger relative to the S(IR)^ xR (in the small-ensemble size cases of interest)?*

Your interpretation is correct. This is discussed in the last paragraph of Sect. 5.8.

L740 *The smoothing appears to have significant benefits. Are there any strategies for how this could be applied if an exhaustive optimization of the parameters is not possible (e.g. for a large system)?*

As shown by Fig. 12, $\alpha_{\mathrm{s}} = 1$ is optimal in this configuration for the L96 model. We have checked that this is the case in most situations where we used the smoothing (in particular with the BV model). However, we could not find an obvious relationship between the optimally tuned values of r and r_{s}. Besides, one should keep in mind that, in the "small-ensemble size cases of interest", the benefits of the smoothing are far less impressive than the benefits of OT (this can be seen in Fig. 16).

L742 *Do you have the baseline RMSE values for the EnKF?*

See the new Fig. 4. We have also added a baseline to Fig. 13.

L743 *"From these results, we conclude that the smoothing by weights step [of Penny and Miyoshi (2016)] is an efficient way of [reducing] the artificial discontinuities [that were] introduced when concatenating the locally updated particles, especially when combined with the coloured noise regularisation jitter method." I should note that the S(IR)xR_ a method appears closest to that presented by Penny and Miyoshi (2016), since their inflation is adaptive and using the terminology here is a regularization jitter scaled by the ensemble anomalies.*

Please keep in mind that the work of Penny and Miyoshi (2016) has been cited in Sect. 4.4.1 (where we introduced the method in the first place) and that the $\mathrm{S(IR)^xS_{PM}R_c}$ algorithm tested in this section is not the LPF of Penny and Miyoshi (2016) but an improvement thereof, which includes: a more general framework that can be applied to different types of resampling, a tapering function, a smoothing radius and a smoothing strength parameters, coloured noise regularisation.

The other corrections ("reducing" and "that were") have been done.

L771 *The results look very nice with the OT approach. Do you have an analysis of the relative costs of each of the methods as a function of system size, observation count, and ensemble size?*

This is detailed in the new section 4.5.2.

L794 *"local LET algorithm" Is that redundant? Perhaps just say "LET algorithm"*

"LET" means linear ensemble transform (introduced in Sect. 2.3 with appropriate citations).

L804 *I think it would be appropriate at this point to provide a companion plot that shows the relative cost for each method as well.*

Please, see the discussion about general comment 1).

L809 *Perhaps you should put the EnKF baseline on the plot as well.*

See the new Fig. 4. We have also added a baseline to Fig. 16.

L816 *"dynamic[s]"*

Done.

L828 *"The ETKF requires at least Ne = 12 ensemble members to avoid divergence." This would imply that the number of positive and neutral Lyapunov exponents of the system is 11.*

You are right.

L835 *It may not hurt to repeat the definition of each algorithm here.*

We now refer to the algorithms' list in the (new) Table 2. Thank you for the suggestion.

L923 *"The SO formalism is elegant." This seems a strange characterization given that the next few sentences describe legitimate problems with the approach.*

We wanted to emphasis that the formalism developed in Sects. 7.2.1 and 7.2.2 looks elegant. The next few sentences raise issues that appear when combining the SO formalism with the PF. These issues are not specific to the SO formalism. Following your commment, we mitigated this remark.

L940 *I suggest either staying consistent with the rest of the paper and defining the section using the algorithm name adopted in the paper - LPF^ y - S(IRP_P)^ yR, or renaming the rest of the algorithms in the paper based on the authors that introduced them.*

The nomenclature for the algorithms has been changed to follow your suggestions (please, see the discussion about general comment 3)). Thank you for spotting the inconsistency in the naming of the subsections. We corrected this point.

L968 *The terms 'ensemble member' and 'particle' are synonymous - they differentiate the same concept developed in two different fields. The term ensemble does not imply a 2 moment method, so the naming convention shouldn't be used for the purpose stated here.*

We now refer to this algorithm as "the second order propagation algorithm".

You are right, the terms "ensemble member" and "particle" designate the same concept but they are used in different contexts. The term "particle" is often used with Bayesian method (or at least with methods based on importance sampling) while the term "ensemble member" is often used with Kalman filters. The naming convention is therefore already commonly being used to distinguish between Bayesian / non-Bayesian methods.

L970 *"one first need[s] to"*

Done.

L971 *This computation is expensive for large systems. Is this computed in ensemble space or model space?*

We stated in Sect. 7.4.3 that only submatrices of $\mathbf{\Sigma}$ need to be computed. Therefore this computation is not that expensive for large systems.

How to implement Eq. (65) is beyond the scope of this article. For our numerical experiments, we computed it in state space. Note that because of the schur product, there is no obvious formula for $\mathbf{\Sigma}$ in ensemble space.

L1009 *"any distance that need[s] to be computed relative[] to the observation site..."*

Done.

Table 2 *The nomenclature table is somewhat helpful, but I'd prefer a full table showing each method for which results are presented, with a description of the method, and the section where it can be found in the text.*

The caption of this table has now more details, such that it is more explicit. Following your recommendations, we have added new tables (Table 2 and 4). Thanks, this is a great addition.

L1013 *If the block computing is required to make the algorithms computationally scalable to large systems, then these are the results that should be reported.*

This is only a discussion about algorithmic possibilities. The block computation is a way to reduce the computation requirements of LPF$^{\mathrm{y}}$s. It should not be required to make the algorithms computationally scalable to large systems.

L1073 *"size Ne grows[,] the RMSE decreases"*

Done.

L1075 *Again, I suggest showing the LETKF baseline RMSEs, as well as the computational costs of each method.*

The RMSE values for the LETKF are in the dedicated figures. We have also added a baseline to Figs. 20 and 21. For the computational cost, please see the discussion about general comment 1).

L1019 *"few but [dis]similar LPF algorithms"*

Done.

Figure 21 *The better of the white noise and colored noise jitter should be used for each method.*

For very small ensemble sizes, the white noise jitter yields lower RMSEs in most test series so far. This is why we used it in this high-dimensional test series.

I have to state again that there should be another case presented in which the LETKF fails and the S()R methods produce superior results.

Please see the discussion about general comment 2).

I very much like the promise of the LPF^ x OT methods. However, I'd like to see the S(IR)^ xSR_ a method of Penny and Miyoshi (2016) presented, which should give a nice balance between parallelizable computational costs and accuracy as measured by RMSE - which was the primary goal of the algorithm.

We originally did not select the $\mathrm{S(IR)^{x}S_{PM}R_{c}}$ for this high-dimensional test series for these reasons:

1. Given the results for the BV model in the coarse resolution configuration, with very small ensemble sizes this algorithm is outperformed by the algorithms using OT resampling.
2. This algorithm is slower than the other algorithm, because in two dimensions, computing the smoothing by weights is numerically expensive.
3. Optimal tuning parameters for this method are harder to find (both because there are more parameters and because the simulations are long).

Following your recommendation, we performed the simulation and reported the results in Fig. 22 and Table 5. The ration accuracy / computation time is not in favor of this method.

Finally, please keep in mind that the $\mathrm{S(IR)^{x}S_{PM}R_{c}}$ algorithm tested here is not the LPF of Penny and Miyoshi (2016) but our improvement thereof, which includes: a more general framework that can be applied to different types of resampling, a tapering function, a smoothing radius and a smoothing strength parameters, coloured noise regularisation.

Review article: Comparison of local particle filters and new implementations

Alban Farchi[1] **and Marc Bocquet**[1]

[1]CEREA, joint laboratory École des Ponts ParisTech and EDF R&D, Université Paris-Est, Champs-sur-Marne, France

Correspondence to: A. Farchi (alban.farchi@enpc.fr)

Abstract. Particle filtering is a generic weighted ensemble data assimilation method based on sequential importance sampling, suited for nonlinear and non-Gaussian filtering problems. Unless the number of ensemble members scales exponentially with the problem size, particle filter (PF) algorithms ~~lead to~~experience weight degeneracy. This phenomenon is a ~~consequence~~manifestation of the curse of dimensionality that prevents ~~one from using~~the use of PF methods for high-dimensional data assimilation. The use of local analyses to counteract the curse of dimensionality was suggested early ~~on~~in the development of PF algorithms. However, implementing localisation in the PF is a challenge because there is no simple and yet consistent way of gluing together locally updated particles ~~together~~ across domains.

In this article, we review the ideas related to localisation and the PF in the geosciences. We introduce a generic and theoretical classification of local particle filter (LPF) algorithms, with an emphasis on the advantages and drawbacks of each category. Alongside with the classification, we suggest practical solutions to the difficulties of local particle filtering, that lead to new implementations and improvements in the design of LPF algorithms.

The LPF algorithms are systematically tested and compared using twin experiments with the one-dimensional Lorenz 40-variables model and with a two-dimensional barotropic vorticity model. The results illustrate the advantages of using the optimal transport theory to design the local analysis. With reasonable ensemble sizes, the best LPF algorithms yield data assimilation scores comparable to those of typical ensemble Kalman filter algorithms even for a mildly nonlinear system.

1 Introduction

The ensemble Kalman filter (EnKF, Evensen, 1994) and its variants are currently among the most popular data assimilation (DA) methods. Because EnKF-like methods are simple to implement, they have been successfully developed and applied to numerous dynamical systems in geophysics such as atmospheric and oceanographic models including in operational conditions (see for example Houtekamer et al., 2005; Sakov et al., 2012a).

The EnKF can be viewed as a subclass of sequential Monte Carlo (MC) methods whose analysis step relies on Gaussian distributions. However, observations ~~to assimilate~~ can have non-Gaussian error distributions, an example being the case of bounded variables — which are frequent in ocean and land surface modeling or in atmospheric chemistry. Most geophysical dynamical models are nonlinear yielding non-Gaussian error distributions (Bocquet et al., 2010). Moreover, recent advances in numerical modeling enable the use of finer resolutions for the models: small scale processes that can increase nonlinearity have then to be resolved.

When the Gaussian assumption is not fulfilled, Kalman filtering is suboptimal. Iterative ~~EnKF~~ensemble Kalman filter and smoother methods have been developed to overcome these limitations, mainly by including variational analysis in the algorithms (Zupanski, 2005; Sakov et al., 2012b; Bocquet and Sakov, 2014), or through heuristic iterations (Kalnay and Yang, 2010). Yet one cannot bypass the Gaussian representation of the conditional density with these latter methods. On the other hand, with particle filter (PF) methods (Gordon et al., 1993; Doucet et al., 2001; Arulampalam et al., 2002; Chen, 2003; van Leeuwen, 2009; Bocquet et al., 2010), all Gaussian and linear hypotheses have been relaxed, allowing a fully Bayesian analysis step. That is why the generic PF is a promising method.

Unfortunately, there is no successful application of it to a significantly high-dimensional DA problem. Unless the number of ensemble members scales exponentially with the problem size, PF methods experience weight degeneracy and ~~lead to~~yield poor estimates of the model state. This phenomenon is a ~~consequence~~symptom of the curse of dimensionality and is the main obstacle to an application of PF algorithms to most DA problems (Silverman, 1986; Kong et al., 1994; Snyder et al., 2008). Nevertheless, the PF has appealing properties – the method is elegant, simple and fast, and it allows for a Bayesian analysis. Part of the research on the PF is dedicated to their application to high-dimensional DA with a focus on four topics: importance sampling, resampling, ~~hybridisation~~hybridisation, and localisation.

Importance sampling is at the heart of PF methods where the goal is to construct ~~a representation~~a sample of the posterior density (the conditional density) given particles from the prior density using importance weights. The use of a proposal transition density is a way to reduce the variance of the importance weights, hence allowing the use of ~~less~~fewer particles. However, importance sampling with a proposal density can lead to more costly algorithms that are not necessarily ~~free~~rid of the curse of dimensionality (Chap. 4 of MacKay, 2003; Snyder et al., 2015). Proposal-density PF methods include the optimal importance particle filter (OIPF, Doucet et al., 2000), whose exact implementation is only available in simple DA problems (linear observation operator and additive Gaussian noise), the implicit particle filter (Chorin and Tu, 2009; Chorin et al., 2010; Morzfeld et al., 2012) which is an extension of the OIPF for ~~non-trivial~~ DA problems using smoothing, the equivalent-weights particle filter (EWPF) and its implicit version (van Leeuwen, 2010; Zhu et al., 2016).

Resampling is the first improvement that was suggested in the bootstrap algorithm (Gordon et al., 1993) to avoid the collapse of a PF based on sequential importance sampling. Common resampling algorithms include the multinomial resampling and the stochastic universal (SU) sampling algorithms. The resampling step allows the algorithm to focus on particles that are more likely, but, as a drawback, it introduces sampling noise. Worse, it may lead to sample impoverishment hence failing to avoid the collapse of the PF if the model noise is insufficient (van Leeuwen, 2009; Bocquet et al., 2010). Therefore it is usual practice to add a regularisation step after the resampling (Musso et al., 2001). ~~Eventually, using~~Using ideas from the optimal transport theory, Reich (2013) designed a resampling algorithm that creates strong bindings between the prior ensemble members and the updated ensemble members.

Hybridising PFs with EnKFs seems a promising approach for the application of PF methods to high-dimensional DA, in which one can hope to take the best of both worlds: robustness of the EnKF and Bayesian analysis of the PF. The balance between the EnKF and the PF analysis must be chosen carefully. Hybridisation especially suits the case where the number of significantly nonlinear ~~degree~~degrees of freedom is small compared to the others. ~~Examples of assimilation using hybrid filters can be found when applied~~ Hybrid filters have been applied for example to geophysical low-order models (Chustagulprom et al., 2016) ~~and in~~and to Lagrangian DA (Apte and Jones, 2013; Slivinski et al., 2015).

In most geophysical ~~system~~systems, distant regions have (almost) independent evolution over short time scales. This idea was used in the EnKF to implement localisation in the analysis (Houtekamer and Mitchell, 2001; Hamill et al., 2001; Evensen, 2003; Ott et al., 2004). In a PF context, localisation could be used to counteract the curse of dimensionality. Yet, if localisation of the EnKF is simple and leads to efficient algorithms (Hunt et al., 2007), implementing localisation in the PF is a challenge because there is no trivial way of gluing together locally updated particles ~~together~~ across domains (van Leeuwen, 2009). The aim of this paper is to review and compare recent propositions of local particle filter (LPF) algorithms (Rebeschini and van Handel, 2015; Lee and Majda, 2016; Penny and Miyoshi, 2016; Poterjoy, 2016; Robert and Künsch, 2017) and to suggest practical solutions to the difficulties of local particle filtering that lead to improvements in the design of LPF algorithms.

Section 2 provides some background on DA and particle filtering. Section 3 is dedicated to the curse of dimensionality with some theoretical elements and illustrations. The challenges of localisation in PF methods are then discussed in Sects. 4 and 7 from two different angles. For both approaches, we propose new implementations of LPF algorithms, which are tested in Sects. 5, 6 and 8 with twin simulations of low-order models. Several of the LPFs are tested in Sect. 9 with twin simulations of a higher dimensional model. ~~Elements of conclusion~~Conclusions are given in Sect. 10.

2 Background

2.1 The data assimilation filtering problem

We follow a state vector $\boldsymbol{x}_k \in \mathbb{R}^{N_x}$ at discrete times $t_k, k \in \mathbb{N}$, through independent ~~observations~~observation vectors $\boldsymbol{y}_k \in \mathbb{R}^{N_y}$. The evolution is assumed to be driven by a hidden Markov model whose initial distribution is $p(\boldsymbol{x}_0)$, whose transition distribution is $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$, and whose observation distribution is $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$.

The model can alternatively be described by

$$\boldsymbol{x}_{k+1} = \mathcal{M}_k(\boldsymbol{x}_k, \boldsymbol{w}_k), \tag{1}$$

$$\boldsymbol{y}_k = \mathcal{H}_k(\boldsymbol{x}_k, \boldsymbol{v}_k), \tag{2}$$

where the random vectors $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ follow the transition and observation distributions.

The components of the state vector $\boldsymbol{x}_k$ are called *state variables* or simply *variables* and the components of the observation vector $\boldsymbol{y}_k$ are called *observations.*

Let $\pi_{k|k}$ be the *analysis* (or *filtering*) density $\pi_{k|k} = p(\boldsymbol{x}_k|\boldsymbol{y}_{k:0})$, where $\boldsymbol{y}_{k:0}$ is the set $\{\boldsymbol{y}_l, l = 0 \ldots k\}$ and let

$\pi_{k+1|k}$ be the *prediction* (or *forecast*) density $\pi_{k+1|k} = p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{k:0})$ with $\pi_{0|-1}$ coinciding with $p(\boldsymbol{x}_0)$ by convention.

The prediction operator P_k is defined by the Chapman–Kolmogorov equation:

$$P_k\left(\pi_{k|k}\right) \triangleq \pi_{k+1|k} = \int p\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k\right)\pi_{k|k}\,\mathrm{d}\boldsymbol{x}_k, \tag{3}$$

and Bayes' theorem is used to define the correction operator C_k:

$$C_{k+1}\left(\pi_{k+1|k}\right) \triangleq \pi_{k+1|k+1} = \frac{p\left(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}\right)\pi_{k+1|k}}{p\left(\boldsymbol{y}_{k+1}|\boldsymbol{y}_{k:0}\right)}. \tag{4}$$

~~The DA filtering problem~~In this article, we consider the DA filtering problem that consists in estimating $\pi_{k|k}$ with given realisations of $\boldsymbol{y}_{k:0}$.

2.2 Particle filtering

The PF is a class of sequential MC methods that produces, from the realisations of $\boldsymbol{y}_{k:0}$, a set of weighted ensemble members (or *particles*) $\left(\boldsymbol{x}_k^i, w_k^i\right), i = 1\ldots N_\mathrm{e}$. The analysis density $\pi_{k|k}$ is estimated through the empirical density:

$$\pi_{k|k}^{N_\mathrm{e}} = \sum_{i=1}^{N_\mathrm{e}} w_k^i\,\delta_{\boldsymbol{x}_k^i}, \tag{5}$$

where the weights are normalised so that their sum is 1 and $\delta_{\boldsymbol{x}}$ is the Dirac distribution centered at $\boldsymbol{x}$.

Inserting the particle ~~representatioN~~representation Eq. (5) in the Chapman–Kolmogorov equation yields

$$P_k\left(\pi_{k|k}^{N_\mathrm{e}}\right) = \sum_{i=1}^{N_\mathrm{e}} w_k^i\,p\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k^i\right). \tag{6}$$

In order to recover a particle representation, the prediction operator P_k must be followed by a sampling step S^{N_e}. In the bootstrap or sampling importance resampling (SIR) algorithm of Gordon et al. (1993), the sampling is performed as follows:

$$\boldsymbol{x}_{k+1}^i \sim p\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k^i\right), \tag{7}$$

$$w_{k+1}^i \leftarrow w_k^i, \tag{8}$$

where $\boldsymbol{x} \sim p$ means that $\boldsymbol{x}$ is a realisation of a random vector distributed according to the probability density function (pdf) p. The empirical density $\pi_{k+1|k}^{N_\mathrm{e}}$ is now an estimator of $\pi_{k+1|k}$.

Applying Bayes' theorem to $\pi_{k+1|k}^{N_\mathrm{e}}$ gives a weight update that follows the principle of importance sampling:

$$w_{k+1}^i \leftarrow w_{k+1}^i\,p\left(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}^i\right). \tag{9}$$

The weights are then renormalised so that they sum to 1.

Finally, an optional resampling step R^{N_e} is added if needed (see Sect. 2.3). In terms of densities, the PF can be summarised by the recursion

$$\pi_{k+1|k+1}^{N_\mathrm{e}} = R^{N_\mathrm{e}} \circ C_{k+1} \circ S^{N_\mathrm{e}} \circ P_k\left(\pi_{k|k}^{N_\mathrm{e}}\right). \tag{10}$$

The additional sampling and resampling operators S^{N_e} and R^{N_e} are ~~pure~~ ensemble transformations that are required to propagate the particle representation of the density. Ideally, they should not alter the densities.

Under reasonable assumptions on the prediction and correction operators and on the sampling and resampling algorithms, it is possible to show that, in the limit $N_\mathrm{e} \to \infty$, $\pi_{k|k}^{N_\mathrm{e}}$ converges to $\pi_{k|k}$ for the weak topology on the set of probability measures over $\mathbb{R}^{N_x}$. This convergence result is one of the main reasons for the interest of the DA community in PF methods. More details about the convergence of PF algorithms can be found in Crisan and Doucet (2002).

Eventually, the focus of this article is on the analysis step, that is the correction and the resampling. Hence, *prior* or *forecast* (respectively *posterior*, *updated* or *analysis*) will refer to quantities before (respectively after) the analysis step.

2.3 Resampling

Without resampling, PF methods are subject to weight degeneracy: after a few assimilation cycles, one particle gets almost all the weight. The goal of resampling is to reduce the variance of the weights by reinitialising the ensemble. After this step, the ensemble is made of N_e equally weighted particles.

In most resampling algorithms, highly probable particles are selected and duplicated while particles with low probability are discarded. It is desirable that the selection of particles has a low impact on the empirical density $\pi_{k|k}^{N_\mathrm{e}}$. The most common resampling algorithms — multinomial resampling, SU sampling, residual resampling and Monte Carlo Metropolis–Hastings algorithm — are reviewed by van Leeuwen (2009).

Resampling introduces sampling noise. On the other hand, not resampling means imparting ~~computation~~computational time to highly improbable particles, that have a very low contribution to the empirical analysis density. Therefore, the choice of the resampling frequency is critical in the design of PF algorithms. Common criteria to decide if a resampling step is needed are based on measures of the degeneracy: for example the maximum of the weights or the effective ensemble size defined by Kong et al. (1994), i.e.

$$N_\mathrm{eff} = \left(\sum_{i=1}^{N_\mathrm{e}}\left(w_k^i\right)^2\right)^{-1} \in \left[1, N_\mathrm{e}\right]. \tag{11}$$

The ~~resampling step and the correction step~~correction and resampling steps of PF methods can be combined and embedded into the so-called *linear ensemble transform* (LET) framework (Bishop et al., 2001; Reich and Cotter, 2015) as follows. Let $\mathbf{E}_k$ be the *ensemble matrix*, that is the $N_x \times N_\mathrm{e}$ matrix whose columns are the ensemble members $\boldsymbol{x}_k^i$. The update of the particles is then given by

$$\mathbf{E}_k \leftarrow \mathbf{E}_k \mathbf{T}, \tag{12}$$

where $\mathbf{T}$ is a $N_\mathrm{e} \times N_\mathrm{e}$ transformation matrix whose coefficients are uniquely determined during the resampling step. In the general LET framework, $\mathbf{T}$ has real coefficients and it is subject to the normalisation constraint

$$\sum_{i=1}^{N_\mathrm{e}} [\mathbf{T}]^{i,j} = 1, \quad j = 1 \ldots N_\mathrm{e}, \tag{13}$$

such that the updated ensemble members can be interpreted as weighted averages of the prior ensemble members. The transformation is said to be first-order accurate if it preserves the ensemble mean (Acevedo et al., 2017), i.e. if

$$\sum_{j=1}^{N_\mathrm{e}} [\mathbf{T}]^{i,j} = N_\mathrm{e} w_k^i, \quad i = 1 \ldots N_\mathrm{e}. \tag{14}$$

In the ~~classical~~"select and duplicate" resampling schemes, the coefficients of $\mathbf{T}$ are in $\{0, 1\}$ meaning that the updated particles are copies of the prior particles. The first-order condition Eq. (14) is then only satisfied on average over realisations of the resampling step. Yet it is sufficient to ensure the weak convergence of $\pi_{k|k}^{N_\mathrm{e}}$ almost surely in the case of the multinomial resampling (Crisan and Doucet, 2002).

If the coefficients of $\mathbf{T}$ are positive reals, the transformation can ~~still~~ be understood as a resampling where the updated particles are composite copies of the prior particles. For example, in the ensemble transform particle filter (ETPF) algorithm of Reich (2013), the transformation is chosen such that it minimises the expected distance between the prior and the updated ensembles (seen as realisations of random vectors) among all possible first-order accurate transformations. This leads to a minimisation problem typical of the discrete optimal transport theory (Villani, 2009):

$$\min_{\mathbf{T} \in \mathcal{T}} \sum_{i,j=1}^{N_\mathrm{e}} [\mathbf{T}]^{i,j} \left\| \boldsymbol{x}_k^i - \boldsymbol{x}_k^j \right\|^2, \tag{15}$$

where $\mathcal{T}$ is the set of $N_\mathrm{e} \times N_\mathrm{e}$ transformation matrices satisfying Eqs. (13) and (14). In this way, the correlation between the prior and the updated ensembles is increased and $\pi_{k|k}^{N_\mathrm{e}}$ still converges toward $\pi_{k|k}$ for the weak topology. In the following, this resampling algorithm will be called *optimal ensemble coupling*.

2.4 Proposal-density particle filters

Let $q(\boldsymbol{x}_{k+1})$ be a density whose support is larger than that of $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$ — i.e. $q(\boldsymbol{x}_{k+1}) > 0$ whenever $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) > 0$. The Chapman–Kolmogorov Eq. (3) can be written:

$$\pi_{k+1|k} = \int \frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)}{q(\boldsymbol{x}_{k+1})} q(\boldsymbol{x}_{k+1}) \pi_{k|k} \, \mathrm{d}\boldsymbol{x}_k. \tag{16}$$

In the importance sampling literature, q is called the *proposal* density and can be used to perform the sampling step S^{N_e} described by Eqs. (7) and (8) in a more general way:

$$\boldsymbol{x}_{k+1}^i \sim q(\boldsymbol{x}_{k+1}), \tag{17}$$

$$w_{k+1}^i \leftarrow w_k^i \frac{p(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i)}{q(\boldsymbol{x}_{k+1}^i)}. \tag{18}$$

Using the proposal density q can lead to an improvement of the PF method if for example q is easier to sample from than p or if q includes information about $\boldsymbol{x}_k$ or $\boldsymbol{y}_{k+1}$ in order to reduce the variance of the importance weights.

The SIR algorithm is recovered with the *standard* proposal $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$, while the *optimal importance* proposal $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k, \boldsymbol{y}_{k+1})$ yields the optimal importance sampling importance resampling (OISIR) algorithm (Doucet et al., 2000). Merging the prediction and correction steps of the OISIR algorithm yields the weight update

$$w_{k+1}^i \leftarrow w_k^i \, p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_k^i). \tag{19}$$

It is remarkable that this formula does not depend on $\boldsymbol{x}_{k+1}$ (Doucet et al., 2000). Hence the optimal importance proposal is optimal in the sense that it minimises the variance of the weights over realisations of $\boldsymbol{x}_{k+1}^i$ — namely 0. Moreover, it can be shown that it also minimises the variance of the weights over realisations of the whole trajectory $\boldsymbol{x}_{k+1:0}^i$ among proposal densities that depend on $\boldsymbol{x}_k$ and $\boldsymbol{y}_{k+1}$ (Snyder et al., 2015).

Although the optimal importance proposal has appealing properties, its computation is non-trivial. For the generic model with Gaussian additive noise described in Appendix A2, when the observation operator $\mathcal{H}$ is linear, the optimal importance proposal can be computed as a Kalman filter analysis as shown by Doucet et al. (2000). However, in the general case there is no analytic form and one must resort to more ~~elaborated~~elaborate algorithms (Chorin and Tu, 2009; Chorin et al., 2010; Morzfeld et al., 2012).

3 The curse of dimensionality

3.1 The weight degeneracy of particle filters

The PF has been successfully applied to low-dimensional DA problems (Doucet et al., 2000). However, attempts to apply the SIR algorithm to medium- to high-dimensional geophysical models ~~lead~~have led to weight degeneracy (e.g., van Leeuwen, 2003; Zhou et al., 2006).

Bocquet et al. (2010) ~~reproduce~~demonstrated weight degeneracy in low-order models, for example in the Lorenz

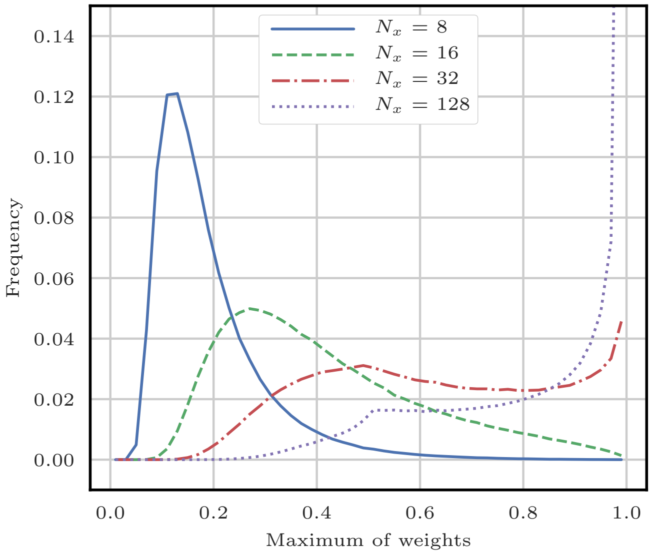


Figure 1. Empirical statistics of the maximum of the weights for one importance sampling step applied to the Gaussian linear model of Appendix A1. The model parameters are $p = 1$, $a = 1$, $h = 1$, $q = 1$, $\sigma = 1$, the ensemble size is $N_\mathrm{e} = 128$ and the system size varies from $N_x = 8$ (well balanced case) to $N_x = 128$ (almost degenerate case).

1996 (L96, Lorenz and Emanuel, 1998) model in the standard configuration described in Appendix A3. They ~~illustrate~~illustrated the empirical statistics of the maximum of the weights for several values of the system size. When the system size is small~~,~~(10 to 20 variables~~,~~) weights are balanced and values close to 1 are infrequent. However, when the system size grows ~~—~~(more than 40 variables~~—~~) weights rapidly degenerate: values close to 1 become more frequent. Ultimately, the frequency of the maximum of the weights peaks to 1.

Similar results ~~occur~~are produced when applying one importance sampling step to the Gaussian linear model described in Appendix A1. For this model, we illustrate the empirical statistics of the maximum of the weights in Fig. 1. Snyder et al. (2008) also computed the required number of particles in order to avoid degeneracy in simulations and found that it scales exponentially with the size of the problem.

This phenomenon, well known in the PF literature, is often referred to as *degeneracy*, *collapse* or *impoverishment* and is a symptom of the curse of dimensionality.

3.2 The equivalent state dimension

At first sight, it ~~seems~~might seem surprising that, although MC ~~method~~methods have a convergence rate independent of the dimension, the curse of dimensionality applies to PF methods. Yet, the correction step C_k is an importance sampling step between the prior and the analysis probability densities. The higher the number of ~~observation components~~observations N_y, the more singular these densities are to each other: random particles from the prior density have an exponentially small likelihood according to the analysis density. This is the main reason for the blow up of the number of particles required for a non-degenerate scenario (Rebeschini and van Handel, 2015).

A quantitative description of the behavior of weights for large values of N_y can be found in Snyder et al. (2008). In this study, the authors first define:

$$\tau^2 = \mathrm{var}\left[\ln\left(p\left(\boldsymbol{y}_k|\boldsymbol{x}_k\right)\right)\right], \tag{20}$$

with the hypothesis that the observation noise is additive and each of its component is independent and identically distributed (iid). Then they derive the asymptotic relationship for only one analysis step:

$$\mathbb{E}\left[\frac{1}{\max\limits_i w_k^i}\right] \underset{N_\mathrm{e}\to\infty}{\sim} 1+\frac{\sqrt{2\ln N_\mathrm{e}}}{\tau}, \tag{21}$$

where $\mathbb{E}$ is the expectation over realisations of the prior ensemble members.

This result means that, in order to avoid the collapse of a PF method, the number of particles N_e must be of order $\exp\left(\tau^2/2\right)$. In simple cases, as the ones considered in the previous sections, τ^2 is proportional to N_y. The dependence of τ on N_x is indirect in the sense that the derivation of Eq. (21) requires N_x to be asymptotically large. In a sense, one can think of τ^2 as an *equivalent state dimension*.

Snyder et al. (2008) then illustrate the validity of the asymptotic relationship Eq. (21) using simulations of the Gaussian linear model of Appendix A1 with a SIR algorithm, for which:

$$\tau^2 = N_y \frac{h^2\left(q^2+a^2p^2\right)}{\sigma^2}\left(1+\frac{3h^2}{2\sigma^2}\left(q^2+a^2p^2\right)\right). \tag{22}$$

Snyder et al. (2008) ~~does~~do not illustrate the validity of Eq. (21) in more general cases, mainly because the computation of τ is non-trivial. The effect of resampling is not investigated either, thought it is clear from simulations that resampling is not enough to avoid filter collapse. Finally, the effect of using proposal densities is the subject of another study by Snyder et al. (2015).

3.3 Mitigating the collapse using proposals

One objective of using proposal densities in PF methods is to reduce the variance of the importance weights as discussed in Sect. 2.4. If one uses the optimal importance proposal density $p\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k,\boldsymbol{y}_{k+1}\right)$ to sample $\boldsymbol{x}_k$ in the prediction and sampling step $S^{N_\mathrm{e}} \circ P_k$, the correction step C_{k+1} consists in matching two identical densities, which leads to a weight update Eq. (19) that does not depend on the realisation of $\boldsymbol{x}_{k+1}$.

Yet, the OISIR algorithm still collapses even for low-order models such as the L96 model with 40 variables (Bocquet et al., 2010). In fact, the curse of dimensionality for any proposal-density PF does not ~~primary~~primarily come from the correction step C_k, but from the recursion in the PF~~ and in~~. In particular it stems from the fact that the algorithm does not correct the particles at earlier times to account for new observations (Snyder et al., 2015). This was a key motivation in the development of the guided SIR algorithm of van Leeuwen (2009), whose ideas were included in the practical implementations of the EWPF algorithm (van Leeuwen, 2010; Ades and van Leeuwen, 2015) as a relaxation step, with moderate success (Browne, 2016).

Snyder et al. (2015) illustrate the validity of Eq. (21) using simulations of the Gaussian linear model of Appendix A1 with an OISIR algorithm, for which:

$$\tau^2 = N_y \frac{a^2p^2h^2}{\sigma^2+h^2q^2}\left(1+\frac{3a^2h^2p^2}{2\left(\sigma^2+h^2q^2\right)}\right), \qquad (23)$$

and they found a good accuracy of Eq. (21) in the limit $N_\mathrm{e} \ll \exp\left(\tau^2/2\right)$. This shows that the use of the optimal importance proposal reduces the number of particles required to avoid the collapse of a PF method. However, ultimately, proposal-density PFs cannot counteract the curse of dimensionality in this simple model and there is no reason to think that they could in more ~~elaborated~~elaborate models (see chapter 29 of MacKay, 2003).

In a generic Gaussian linear model, the equivalent state dimension τ^2 as in Eqs. (22) and (23) is directly proportional to the system size N_x — equal to N_y in this case. For more elaborate models, the relationship between τ^2 and N_x is likely to be more complex and may involve the effective number of degrees of freedom in the model.

3.4 Using localisation to avoid collapse

By considering the definition of τ^2, Eq. (20), one can see that the curse of dimensionality is a consequence of the fact that the importance weights are influenced by all components of the observation vector $\boldsymbol{y}_k$. Yet, a particular state variable and observation can be nearly independent, for example in spatially extended models if they are distant to each other. In this situation, the statistical properties of the ensemble at this state variable (i.e. the marginal density) should not evolve during the analysis step. Yet, this is not the case in PF methods, because of the use of (relatively) low ensemble sizes; even the ensemble mean can be significantly impacted. A good illustration of this phenomenon can be found in Fig. 2 of Poterjoy (2016). In this case, the PF overestimates the information available and equivalently underestimates the uncertainty in the analysis density (Snyder et al., 2008). As a consequence, spurious correlations appear between distant state variables.

This would not be the case in a PF algorithm that would be able to perform local analyses, that is when the influence of each observation is restricted to a spatial neighborhood of its location. The equivalent state dimension τ^2 would then be defined using the maximum number of observations that influence a state variable, which could be kept relatively small even for high-dimensional systems.

In the EnKF literature, this idea is known as *domain localisation* or *local analysis* and was introduced to fix the same kind of issues (Houtekamer and Mitchell, 2001; Hamill et al., 2001; Evensen, 2003; Ott et al., 2004). Technical implementations of domain localisation in EnKF methods is as easy as implementing a global analysis and the local analyses can be carried out in parallel (Hunt et al., 2007). ~~The~~By contrast, the application of localisation techniques in PF methods is discussed in Snyder et al. (2008); van Leeuwen (2009); Bocquet et al. (2010) with an emphasis on two major difficulties.

The first issue is that the variation of the weights across local domains irredeemably breaks the structure of the global particles. There is no trivial way of recovering this global structure, i.e. gluing together the locally updated particles. Global particles are required for the prediction and sampling step $S^{N_\mathrm{e}} \circ P_k$ in all PF algorithms, where the model $\mathcal{M}_k$ is applied to each individual ensemble member.

Secondly, if not carefully constructed, this gluing together could lead to balance problems and sharp gradients in the fields (van Leeuwen, 2009). In EnKF methods, these issues are mitigated by using smooth functions to taper the influence of the ~~observation components~~observations. The smooth dependency of the analysis ensemble on the observation precision reduces imbalance (Greybush et al., 2011). Yet, in most PF algorithms, there is no such smooth dependency. From now on, this issue will be called "imbalance" or "discontinuity" issue. The word "discontinuity" does not point to the discrete nature of the model field on the grid, but, inspired by the mathematical notion of continuity, to large unphysical gaps appearing in the discrete model field.

3.5 Two types of localisation

From now on, we will assume that our DA problem has a well-defined spatial structure:

- each ~~component of $\boldsymbol{x}_k$, hereafter called *state variable* or *grid point*,~~state variable is attached to a location, the *grid point*;
- each ~~component of $\boldsymbol{y}_k$, hereafter called *observation site*~~observation is attached to a location~~as well~~, the *observation site* or simply the *site* (~~local~~ observations are assumed local);
- there is a distance function between locations.

The goal is to be able to define notions such as "the distance between an observation site and a grid point", "the distance between two grid points" or "the center of a group of grid points". In realistic models, these concepts need to be related to the underlying physical space.

In the ~~next~~following sections, we discuss algorithms that address the two issues of local particle filtering (gluing and imbalance) and lead to implementations of domain localisation in PF methods. We divide the solutions into two categories.

In the first approach, independent analyses are performed ~~for~~at each grid point by using only the observation sites that influence this grid point. This leads to algorithms that are easy to define, to implement and to parallelise. However, there is no obvious relationship between ~~grid points~~state variables, which could be problematic with respect to the imbalance issue. This approach is used for example by Rebeschini and van Handel (2015); Penny and Miyoshi (2016); Lee and Majda (2016); Chustagulprom et al. (2016). ~~We will call this approach~~In this article, we call it *state–domain* (and later *state–block–domain*) localisation.

In the second approach, an analysis is performed ~~for~~at each observation site. When assimilating ~~an~~the observation ~~of~~at a site, we partition the state space: nearby grid points are updated while distant grid point remain unchanged. In this formalism, ~~observation sites~~observations need to be assimilated sequentially, which makes the algorithms harder to define and to parallelise but may mitigate the imbalance issue. This approach is used for example by Poterjoy (2016). ~~We will call this approach~~In this article, we call it *sequential–observation* localisation.

4 State–domain localisation for particle filters

From now on, the time subscript k is systematically dropped for clarity and the conditioning with respect to prior quantities is implicit. The superscript $i \in \{1 \dots N_\mathrm{e}\}$ is the member index, the subscript $n \in \{1 \dots N_x\}$ is the state variable or grid point index, the subscript $q \in \{1 \dots N_y\}$ is the observation or observation site index, the subscript $b \in \{1 \dots N_\mathrm{b}\}$ is the block index (the concept of block is defined in Sect. 4.2).

4.1 Introducing localisation in particle filters

Localisation is generally introduced in PF methods by allowing the analysis weights to depend on the spatial position. In the (global) PF, the marginal of the analysis density for each ~~grid point $n = 1 \dots N_x$~~state variable is

$$p(x_n) = \sum_{i=1}^{N_\mathrm{e}} w^i \, \delta_{x_n^i}, \tag{24}$$

whose localised version is

$$p(x_n) = \sum_{i=1}^{N_\mathrm{e}} w_n^i \, \delta_{x_n^i}. \tag{25}$$

The local weights w_n^i depend on the spatial position through the grid point index n.

With local analysis weights, the marginals of the analysis density are uncoupled. This is the reason why localisation was introduced in the first place, but, as a drawback, the full analysis density is not known. The ~~most simple~~simplest fix is to approximate the full density as the product of its marginals:

$$p(\boldsymbol{x}) = \prod_{n=1}^{N_x} \sum_{i=1}^{N_\mathrm{e}} w_n^i \, \delta_{x_n^i}, \tag{26}$$

which is a weighted sum of the $N_\mathrm{e}^{N_x}$ possible combinations between all particles.

In summary, in LPF methods, we keep the generic MC structure described in Sect. 2.2. The prediction and sampling step is not modified. The correction step is adjusted to allow the analysis density to have the form given by Eq. (26). In particular, one has to define the local analysis weights w_n^i; this point will be discussed in Sect. 4.2.2. ~~Finally, the resampling step, that was optional in (global) PF methods, becomes mandatory to reconstitute global particles, which are required for the next assimilation cycle. Practical algorithms to construct these N_e updated particles will be introduced in Sect. 4.4.~~ Finally, global particles are required for the next assimilation cycle and they are obtained as follows. A local resampling is first performed *independently for each grid point*. The locally resampled particles are then assembled into global particles. The local resampling step is discussed in detail in Sect. 4.4.

4.2 Extension to state–block–domain localisation

The principle of localisation in the PF, and in particular Eq. (26), can be included into a more general state–block–domain (SBD) localisation formalism. The state space is divided into ~~*local*~~(local state) *blocks* with the additional constraint that the weights should be constant over the blocks. The resampling ~~is then~~then has to be performed ~~independently for each block~~*independently for each block*.

In the block particle filter algorithm of Rebeschini and van Handel (2015), the local weight of a block is computed using the observation sites that are located inside this block. However, in general nothing prevents one from using the observation sites inside a *local domain* potentially different from the block. This is the case in the LPF of Penny and Miyoshi (2016), in which the ~~local~~ blocks have size 1 grid point while the size of the local domains is controlled by a localisation radius.

To summarise, LPF algorithms using the SBD localisation formalism, hereafter called LPF^x algorithms[1], are characterised by

- the geometry of the ~~local (state)~~ blocks over which the weights are constant;

---

[1] The x exponent emphasises the fact that we perform one analysis per (local state) block.

- the local domain of each block, which gathers all observation sites used to compute the local weight;
- the local resampling algorithm.

Most LPFs (e.g. those described in Rebeschini and van Handel, 2015; Penny and Miyoshi, 2016; Lee and Majda, 2016) in the literature can be seen to adopt this SBD formalism.

4.2.1 The local state blocks

Using parallelepipedic ~~local~~ blocks is a standard geometric choice (Rebeschini and van Handel, 2015; Penny and Miyoshi, 2016). It is easy to conceive and to implement and it offers a potentially interesting degree of freedom: the *block shape*. Using ~~bigger~~larger blocks decreases the proportion of block boundaries and hence the bias in the local analyses. On the other hand, it also means less freedom to counteract the curse of dimensionality.

In the clustered particle filter algorithms of Lee and Majda (2016), the blocks are centered around the observation sites. The potential gains of this method are unclear. Moreover, when the ~~observation~~ sites are regularly distributed over the space — which is the case in the numerical examples of Sects. 5 and 6 — there is no difference with the standard method.

4.2.2 The local domains

~~In the analyses of local EnKF methods, the general idea of localisation is that the analysis at one grid point is computed by using only the observation sites that lie inside a circle centered in this grid point. The radius of this circle is a free parameter often called the *localisation radius*.~~ The general idea of domain localisation in the EnKF is that the analysis at one grid point is computed using only the observation sites that lie within a local region around this grid point, hereafter called the *local domain*. For instance in two dimensions a common choice is to define the local domain of a grid point as a disk, centered at this grid point, and whose radius is a free parameter called the *localisation radius*. The same principle can be applied to the SBD localisation formalism: the local domain of a block will be a ~~circle~~disk whose center coincides with that of the block and whose radius will be a free parameter.

The terminology adopted here (~~circle~~disk, radius...) fits two-dimensional spatial spaces. Yet, most geophysical models have a three-dimensional spatial structure, with typical uneven vertical scales usually much shorter than horizontal scales. For these models, the geometry of the local domains should be adapted accordingly.

Increasing the localisation radius allows one to take more observation sites into account hence reducing the bias in the local analysis. It is also a means to reduce the spatial inhomogeneity by making the weights smoother in space.

The smoothness of the local weights is an important property. Indeed, spatial discontinuities in the weights can lead to spatial discontinuities in the updated particles. ~~Still picking~~Again lifting ideas from the local EnKF methods, the smoothness of the weights can be improved by tapering the influence of an observation site with respect to its distance to the block center as follows. For the (global) PF, assuming that the observation sites are independent, the unnormalised weights are computed according to

$$w^i = \prod_{q=1}^{N_y} p\left(y_q|\boldsymbol{x}^i\right). \tag{27}$$

Following Poterjoy (2016), it becomes for ~~a~~an LPF:

$$w_b^i = \prod_{q=1}^{N_y} \left\{ \alpha + G\left(\frac{d_{q,b}}{r}\right)\left(p\left(y_q|\boldsymbol{x}^i\right) - \alpha\right)\right\}, \tag{28}$$

where α is a constant that should be of the same order as ~~$\max p\left(\boldsymbol{y}|\boldsymbol{x}\right)$~~the maximum value of $p\left(\boldsymbol{y}|\boldsymbol{x}\right)$, $d_{q,b}$ is the distance between the q-th observation site ~~q~~ and the center of the b-th block~~ b~~, r is the localisation radius and G is the taper function: $G(0)=1$ and $G(x)=0$ if x is larger than 1, with a smooth transition. A popular choice for G is the *Gaspari–Cohn* function (Gaspari and Cohn, 1999). If the observation error is an iid Gaussian additive noise with variance σ^2, one can use an alternative "Gaussian" formula for w_b^i directly inspired from local EnKF methods:

$$\ln w_b^i = -\frac{1}{2\sigma^2}\sum_{q=1}^{N_y} G\left(\frac{d_{q,b}}{r}\right)\left(y_q - \mathcal{H}_q\left(\boldsymbol{x}^i\right)\right)^2. \tag{29}$$

Equations (28) and (29) differ. Still they are equivalent in the asymptotic limit $r \to 0$ and $\sigma \to \infty$.

4.2.3 Algorithm summary

Algorithm 1 describes the analysis step for a generic LPF^x. The algorithm parameters are: the ensemble size N_e, the geometry of the blocks and the localisation radius r used to compute the local weights with Eq. (28) or (29). N_b is the number of blocks and $\mathbf{E}_{|b}$ is the restriction of the ensemble matrix $\mathbf{E}$ to ~~state~~the b-th block ~~b~~ (i.e. the rows of $\mathbf{E}$ corresponding to ~~state variables~~grid points that are located within the b-th block~~ b~~). $\mathbf{E}_{|b}$ is a $N_x/N_\mathrm{b} \times N_\mathrm{e}$ matrix.

In this algorithm, and in the rest of this article, the ensemble matrix $\mathbf{E}$ and the particles $\boldsymbol{x}^i$ (its columns) are used interchangeably. Note that in most cases, steps 3, 5 and 6 can be merged into one step.

An illustration of the definition of ~~local~~ blocks and local domains is displayed in Fig. 2.

4.3 Beating the curse of dimensionality

The feasibility of PF methods using SBD localisation is discussed by Rebeschini and van Handel (2015) through the ex-

Algorithm 1 Analysis step for a generic LPF^x

```
Require: Prior (forecast) ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_\mathrm{e}$
1: for $b = 1$ to $N_\mathrm{b}$ do
2:     Compute the local weights $w_b^i$ using Eq. (28) or (29)
3:     Resample the local ensemble $\mathbf{E}_{|b}$ with weights $w_b^i$ as $\mathbf{E}^\mathrm{r}_{|b}$
4: end for
5: ~~Concatenate~~Assemble the locally resampled ensembles $\mathbf{E}^\mathrm{r}_{|b}$ as $\mathbf{E}^\mathrm{r}$
6: Update the ensemble: $\mathbf{E} \leftarrow \mathbf{E}^\mathrm{r}$
7: return Updated (analysis) ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_\mathrm{e}$.
```

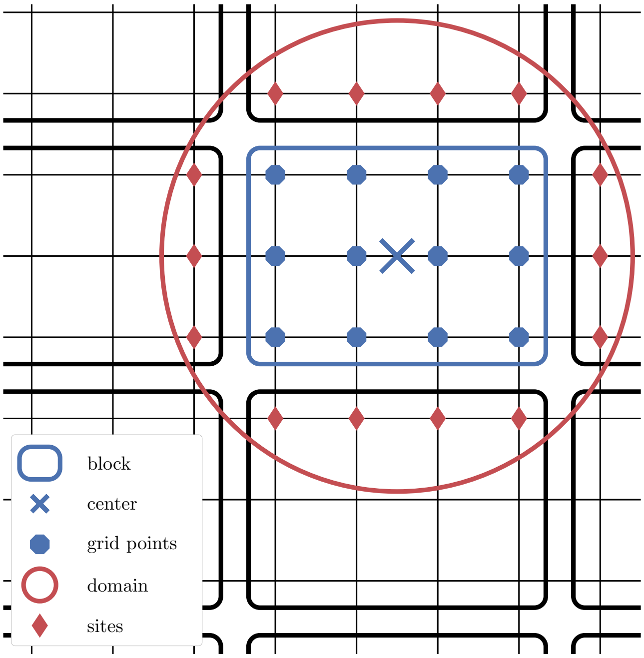


Figure 2. Example of geometry in the SBD localisation formalism for a two-dimensional space. The focus is on the block in the middle which gathers 12 grid points. The local domain is circumscribed by a circle around the block center with potential observation sites outside the ~~local~~ block.

ample of their block particle filter algorithm. In this algorithm, the distinction between blocks and local domains ~~and local blocks~~ does not exist. The influence of each observation ~~site~~ is not tapered and the resampling is performed independently for each block, regardless of the boundaries between blocks.

The main mathematical result is that, under reasonable hypotheses, the error on the analysis density for this ~~LPF~~algorithm can be bounded by the sum of a bias and a variance term. The bias term is related to the block boundaries and ~~decrease~~decreases exponentially with the diameter of the ~~blocks~~blocks, in number of grid points. It is due to the fact that the correction is not Bayesian any more since only a subset of ~~observation sites~~observations is used to update each block. The exponential decrease is a demonstration of the *decay of correlations* property. The variance term is common to all MC methods and scales with $\exp(K)/\sqrt{N_\mathrm{e}}$~~, where K is the size of the blocks — in fact the size of the domain but here they are the same — and not the space dimension any more~~. For global MC methods, K is the state dimension, whereas here K is the number of grid points inside each block. This implies that ~~LPF~~LPF^x algorithms can indeed beat the curse of dimensionality with reasonably large ensembles.

4.4 The local resampling

Resampling from the analysis density given by Eq. (26) does not cause any theoretical or technical issue. One just needs to apply any resampling algorithm (e.g. those described in Sect. 2.3) locally to each block using the local weights. Global particles are then obtained by ~~concatenating~~assembling the locally resampled particles. By doing so, adjacent blocks are fully uncoupled — this is the same remark as when we constructed the analysis density Eq. (26) from its marginals Eq. (25). Once again, this is beneficial, since uncoupling is what counteracts the curse of dimensionality.

On the other hand, blind ~~concatenating~~assembling is likely to lead to ~~artificial~~unphysical discontinuities in the updated particles, regardless of the spatial smoothness of the analysis weights. More precisely, ~~we want to build~~one builds *composite particles*, that is when the i-th updated particle is ~~the concatenation~~composed of the j-th particle on one block and of the k-th particle on an adjacent block with $j \neq k$ — as shown by Fig. 3 in one dimension. There is no guarantee that the j-th and the k-th local particles are *close* and that ~~their concatenation~~assembling them will represent a physical state. ~~Pathological situations can be easily conceived in the case of a multimodal underlying density: artificial composite particles mixing more than one mode are likely to be formed.~~

In order to ~~minimise artificial~~mitigate the unphysical discontinuities, the analysis weights must be spatially smooth, as mentioned in Sect. 4.2.2. Moreover, the resampling scheme must have some "regularity", in order to preserve part of the spatial structure ~~hold~~held in the prior particles. This is a challenge due to the stochastic nature of the resampling algorithms; potential solutions are presented hereafter.

4.4.1 Applying a smoothing by weights

A first solution is to smooth out ~~the~~ potential unphysical discontinuities by averaging in space the locally resampled ensemble as follows. This method was introduced by Penny and Miyoshi (2016) in their LPF and called *smoothing by weights*.

~~For each state block b, let~~Let $\mathbf{E}^\mathrm{r}_b$ be the matrix of the ensemble computed by applying the resampling method to the global ~~ensemble~~ensemble, weighted by the local weights w_b^i of the b-th block. $\mathbf{E}^\mathrm{r}_b$ is ~~a~~an $N_x \times N_\mathrm{e}$ matrix different from the $N_x/N_\mathrm{b} \times N_\mathrm{e}$ matrix $\mathbf{E}^\mathrm{r}_{|b}$ ~~implicitly defined by step 3~~

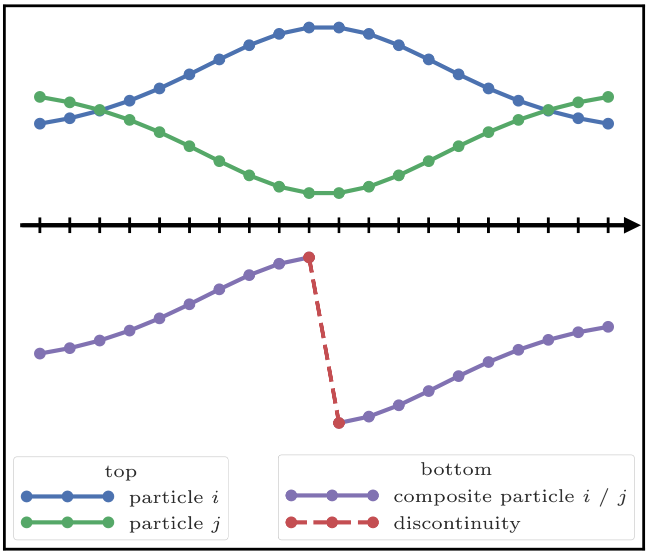


Figure 3. Example of one-dimensional concatenation of particle i on the left and particle j on the right. Top: the prior particles i and j. Bottom: the composite particle, concatenation of i and j. In this situation, a large ~~and artificial~~unphysical discontinuity appears at the boundary.

~~of Algorithm 1~~defined in Sect. 4.2.3. We then define the smoothed ensemble matrix $\mathbf{E}^{\mathrm{s}}$ by

$$[\mathbf{E}^{\mathrm{s}}]^i_n = \frac{\sum\limits_{b=1}^{N_{\mathrm{b}}} G\left(\frac{d_{n,b}}{r_{\mathrm{s}}}\right)[\mathbf{E}^{\mathrm{r}}_b]^i_n}{\sum\limits_{b=1}^{N_{\mathrm{b}}} G\left(\frac{d_{n,b}}{r_{\mathrm{s}}}\right)}, \tag{30}$$

where $d_{n,b}$ is the distance between the n-th grid point ~~n~~ and the center of the b-th block~~ b~~, r_{s} is the *smoothing radius*, a free parameter potentially different from r and G is a taper function, potentially different from the one used to compute the local weights.

~~For example, consider the SU sampling algorithm as the resampling method, and let ϕ_b be~~If the resampling is performed using a "select and duplicate" algorithm (see Sect. 2.3), for example the SU sampling algorithm, then define ϕ_b as the resampling map ~~at~~for the b-th block~~ b~~, i.e. the map computed with the local weights w^i_b such that $\phi_b(i)$ is the index of the i-th ~~resampled~~selected particle. $\mathbf{E}$ being the prior ensemble matrix, Eq. (30) becomes

$$[\mathbf{E}^{\mathrm{s}}]^i_n = \frac{\sum\limits_{b=1}^{N_{\mathrm{b}}} G\left(\frac{d_{n,b}}{r_{\mathrm{s}}}\right)[\mathbf{E}]^{\phi_b(i)}_n}{\sum\limits_{b=1}^{N_{\mathrm{b}}} G\left(\frac{d_{n,b}}{r_{\mathrm{s}}}\right)}. \tag{31}$$

Finally, the ensemble is updated as

$$\mathbf{E} \leftarrow \alpha_{\mathrm{s}}\mathbf{E}^{\mathrm{s}} + (1-\alpha_{\mathrm{s}})\,\mathbf{E}^{\mathrm{r}}, \tag{32}$$

where $\mathbf{E}^{\mathrm{r}}$ ~~be~~is the resampled ensemble matrix implicitly defined by step 5 of Algorithm 1 ~~,~~and α_{s} is the *smoothing strength*, a free parameter in $[0,1]$ that controls the intensity of the smoothing. $\alpha_{\mathrm{s}} = 0$ means that no smoothing is performed and $\alpha_{\mathrm{s}} = 1$ means that the analysis ensemble is totally replaced by the smoothed ensemble.

Algorithm 2 describes the analysis step for a generic LPF^x with smoothing by weights. The original LPF of Penny and Miyoshi (2016) can be recovered if:

- blocks have size 1 grid point (hence there is no distinction between grid points and blocks);
- the local weights are computed using Eq. (29);
- G is a top hat function;
- the resampling method is the SU sampling algorithm;
- r_{s} is set to be equal to r;
- α_{s} is set to 0.5.

The method described here is ~~an extension~~a generalisation of ~~this~~their algorithm.

Note that when the resampling method is the SU sampling algorithm, the matrices $\mathbf{E}^{\mathrm{r}}_b$ do not need to be explicitly computed. One just has to store in memory the resampling maps ϕ_b, $b = 1\dots N_{\mathrm{b}}$ and then use Eq. (31) to obtain the smoothed ensemble matrix $\mathbf{E}^{\mathrm{s}}$.

The smoothing by weights step is an ad-hoc fix to reduce ~~afterwards the artificial~~potential unphysical discontinuities after they have been introduced in the local resampling step. Its necessity hints that there is room for improvement in the design of the local resampling algorithms.

Algorithm 2 Analysis step for a generic LPF^x with smoothing by weights

```
Require: Prior ensemble $\boldsymbol{x}^i$, $i = 1\dots N_e$
for $b = 1$ to $N_b$ do
    Compute the local weights $w^i_b$ using Eq. (28) or (29)
    Resample the local ensemble $\mathbf{E}_{|b}$ with weights $w^i_b$ as $\mathbf{E}^r_{|b}$
    Resample the global ensemble $\mathbf{E}$ with weights $w^i_b$ as $\mathbf{E}^r_b$
end for
~~Concatenate~~Assemble the locally resampled ensembles $\mathbf{E}^r_{|b}$ as $\mathbf{E}^r$
Compute the smoothed ensemble matrix $\mathbf{E}^s$ using Eq. (30)
Update the ensemble matrix $\mathbf{E}$ using Eq. (32)
return Updated ensemble $\boldsymbol{x}^i$, $i = 1\dots N_e$.
```

4.4.2 Refining the sampling algorithms

In this section, we study several properties of the local resampling algorithm that might help dealing with the discontinuity issue: balance, adjustment and random numbers.

A "select and duplicate" sampling algorithm is said to be *balanced* if, for $i=1\dots N_\mathrm{e}$ the number of copies of the i-th particle selected by the algorithm does not differ by more than one unity from $w^i N_\mathrm{e}$. For example, this is the case of the SU sampling but not the multinomial resampling algorithm.

~~In general, the order of the resampled particles does not matter. Hence, once the updated particles have been selected, one can reorder the particle indices to maximise the number of indexes $i \in \{1\dots N_\mathrm{e}\}$ such that the i-th updated particle is a copy of the i-th original particle. If this property is satisfied, we say that the resampling algorithm is *adjustment-minimising*.~~ A "select and duplicate" sampling algorithm is said to be *adjustment-minimising* if the indices of the particles selected by the algorithm are reordered to maximise the number of indices $i \in \{1\dots N_\mathrm{e}\}$ such that the i-th updated particle is a copy of the i-th original particle. The SU sampling and the multinomial resampling algorithms can be simply modified to yield adjustment-minimising resampling algorithms.

While performing the resampling independently for each block, one can use the same random number(s) in the local resampling of each block.

~~In the naive local ensemble Kalman particle filter algorithm of Robert and Künsch (2017), the problem of artificial discontinuities is taken care in the following way:~~

- ~~the resampling algorithm is balanced;~~
- ~~the same random number(s) is (are) used over all blocks;~~
- ~~the resampling algorithm is adjustment-minimising.~~

Using the same random number(s) for the resampling of all blocks ~~cancels~~avoids a stochastic source of ~~artificial spatial discontinuities~~unphysical discontinuity. Choosing balanced and adjustment-minimising resampling algorithms is an attempt to include some kind of continuity in the map $\{\text{local weights}\} \mapsto \{\text{locally updated particles}\}$ by minimising the occurrences of composite particles. However, these properties cannot eliminate all sources of ~~artificial~~unphysical discontinuity. Indeed, ultimately composite particles will be built — if not then localisation would not be necessary — and there is no mechanism to reduce ~~the potential~~unphysical discontinuities in them. These properties have been first introduced in the "naive" local ensemble Kalman particle filter of Robert and Künsch (2017).

4.4.3 Using optimal transport in ensemble space

As mentioned in Sect. 2.3, using the optimal transport (OT) theory to design a resampling algorithm was first investigated in the ETPF algorithm of Reich (2013).

Applying ~~the~~ optimal ensemble coupling to the SBD localisation frameworks results in a local LET resampling method, whose local transformation at each block $\mathbf{T}_b$ solves the discrete OT problem

$$\min_{\mathbf{T}_b \in \mathcal{T}_b} \sum_{i,j=1}^{N_\mathrm{e}} \left[\mathbf{T}_b\right]^{i,j} c_b^{i,j}, \tag{33}$$

where $\mathcal{T}_b$ is the set of $N_\mathrm{e} \times N_\mathrm{e}$ transformations satisfying the normalisation constraint Eq. (13) and the local first-order accuracy constraint

$$\sum_{j=1}^{N_\mathrm{e}} \left[\mathbf{T}_b\right]^{i,j} = N_\mathrm{e} w_b^i, \quad i=1\dots N_\mathrm{e}. \tag{34}$$

In the ETPF, the coefficients $c^{i,j}$ were chosen as the squared L^2 distance between the whole i-th and j-th particles ~~i and j~~ as in Eq. (15). Since we perform a local resampling step, it seems more appropriate to use a local criterion, such as

$$c_b^{i,j} = \sum_{n=1}^{N_x} \left(x_n^i - x_n^j\right)^2 G\left(\frac{d_{n,b}}{r_\mathrm{d}}\right), \tag{35}$$

where $d_{n,b}$ is the distance between the grid point n and the center of block b, r_d is ~~a~~ the *distance radius*, another free parameter and G is a taper function, potentially different from the one used to compute the local weights.

To summarise, Algorithm 3 describes the analysis step for a generic LPF^x that uses optimal ensemble coupling as local resampling algorithm.

Algorithm 3 Analysis step for a generic LPF^x using optimal ensemble coupling for the local resampling

```
Require: Prior ensemble $\boldsymbol{x}^i$, $i = 1\dots N_\mathrm{e}$
for $b = 1$ to $N_\mathrm{b}$ do
    Compute the local weights $w_b^i$ using Eq. (28) or (29)
    Compute the local coefficients $c_b^{i,j}$ with Eq. (35)
    Solve the minimisation problem Eq. (33) for $\mathbf{T}_b$
    Transform local ensemble: $\mathbf{E}^\mathrm{t}_{|b} \leftarrow \mathbf{E}_{|b}\mathbf{T}_b$
end for
~~Concatenate~~Assemble the locally transformed ensemble $\mathbf{E}^\mathrm{t}_{|b}$ as $\mathbf{E}^\mathrm{t}$
Update the ensemble as $\mathbf{E} \leftarrow \mathbf{E}^\mathrm{t}$
return Updated ensemble $\boldsymbol{x}^i$, $i = 1\dots N_\mathrm{e}$
```

Note that localisation was first included in the ETPF algorithm by Chustagulprom et al. (2016), in a similar way as the SBD localisation formalism. ~~However,~~In this article, we are interested in the optimal ensemble coupling for its continuity properties, while the focus of Chustagulprom et al. (2016) was to include the ETPF algorithm in a PF–EnKF hybridisation context. They ~~use~~used the local weights defined by Eq. (29), but they only ~~consider~~considered blocks of size 1 grid point and local coefficients given by

$$c_n^{i,j} = \left(x_n^i - x_n^j\right)^2, \tag{36}$$

in contrast to Eq. (35). ~~On the other hand, in this study we are interested in the optimal ensemble coupling for its continuity properties.~~

On each ~~local~~ block, the linear transformation establishes a strong connection between the prior and the updated ensembles. Moreover, there is no stochastic variation of the coupling at each block. This means that the spatial coherence can be (at least partially) transferred from the prior to the updated ensemble.

~~Using optimal ensemble coupling for the resampling step in a LPF^x algorithm is computationally more expensive than the traditional SU sampling algorithm since it requires to solve one optimisation problem for each local block. The discrete OT problem Eq. (33) to solve is a particular case of the minimum-cost flow problem and can be solved quite efficiently using the network simplex algorithm with complexity of order N_{e}^3 (Pele and Werman, 2009). Moreover, the computation of the local block transformations can be carried out in parallel. For realistic DA applications, the number of particles should be small — no more than 100 or 200 — hence the additional computational cost of solving the discrete OT problem should not be prohibitive.~~

4.4.4 Using optimal transport in state space

In Sect. 4.4.3, the discrete OT theory was used to compute a linear transformation between the prior and the updated ensembles. Following these ideas, we would like to use OT directly in state space. In more than one spatial dimension, the continuous OT problem is highly non-trivial and numerically ~~very costly~~challenging (Villani, 2009). Therefore, we will restrict ourselves to the case where blocks have size 1 grid point. Hence there is no distinction between blocks and grid points.

~~At each grid point~~For each state variable n, we define the prior (marginal) pdf p_n^{f} as the empirical density of the unweighted prior ensemble $\left\{x_n^i, i=1\ldots N_{\mathrm{e}}\right\}$ and the analysis pdf p_n^{a} as the empirical density of the prior ~~ensemble~~ensemble, weighted by the analysis weights $\left\{\left(x_n^i, w_n^i\right), i=1\ldots N_{\mathrm{e}}\right\}$. We seek the map T_n that solves the following OT problem:

$$\min_{T\in\mathcal{T}_n^{\mathrm{f}\to\mathrm{a}}} \int \left|x_n - T\left(x_n\right)\right|^2 \mathrm{d}x_n, \tag{37}$$

where $\mathcal{T}_n^{\mathrm{f}\to\mathrm{a}}$ is the set of maps T that transport p_n^{f} into p_n^{a}:

$$p_n^{\mathrm{f}} = p_n^{\mathrm{a}} \circ T \cdot \mathrm{Jac}\left(T\right), \tag{38}$$

with $\mathrm{Jac}\left(T\right)$ being the absolute value of the determinant of the Jacobian matrix of T.

In one dimension, this transport map is also known ~~as~~to be the *anamorphosis* from p_n^{f} to p_n^{a} and its computation is immediate:

$$T_n = \left(c_n^{\mathrm{a}}\right)^{-1} \circ c_n^{\mathrm{f}}, \tag{39}$$

where c_n^{f} and c_n^{a} are the cumulative density function (cdf) of p_n^{f} and p_n^{a} respectively. ~~In practice,~~Since T_n maps the prior ensemble to an ensemble whose empirical density is p_n^{a}~~. Therefore~~, the images of the prior ensemble members by T_n are suitable candidates for updated ensemble members.

The computation of T_n using Eq. (39) requires a continuous representation for the empirical densities p_n^{f} and p_n^{a}. An appealing approach to obtain ~~the continuous representation~~it is to use the kernel density estimation (KDE) theory (Silverman, 1986; Musso et al., 2001). In this context, the prior density can be written

$$p_n^{\mathrm{f}}\left(x_n\right) = \alpha_n^{\mathrm{f}} \sum_{i=1}^{N_{\mathrm{e}}} K\left(\frac{x_n - x_n^i}{h\sigma_n^{\mathrm{f}}}\right), \tag{40}$$

while the updated density is

$$p_n^{\mathrm{a}}\left(x_n\right) = \alpha_n^{\mathrm{a}} \sum_{i=1}^{N_{\mathrm{e}}} w_n^i K\left(\frac{x_n - x_n^i}{h\sigma_n^{\mathrm{a}}}\right). \tag{41}$$

K is the regularisation kernel, h is the bandwidth, a free parameter, σ_n^{f} and σ_n^{a} are the empirical standard deviation of respectively the unweighted ensemble $\left\{x_n^i, i=1\ldots N_{\mathrm{e}}\right\}$ and the weighted ensemble $\left\{\left(x_n^i, w_n^i\right), i=1\ldots N_{\mathrm{e}}\right\}$ and α_n^{f} and α_n^{a} are normalisation constants.

According to the KDE theory, when the underlying distribution is Gaussian, the optimal ~~pattern~~shape for K is the Epanechnikov kernel (quadratic functions). Yet, there is no reason to think that this will also be the case for the prior density. Besides, the Epanechnikov kernel, having a finite support, generally leads to a poor representation of the distribution tails and it is a potential source of indetermination in the definition of the cdfs. That is why it is more common to use a Gaussian kernel for K. However, in this case, the computational cost associated to the cdf of the kernel (the error function) becomes significant. Hence, as an alternative, we choose to use the Student's t-distribution with two degrees of freedom. It ~~looks close~~is similar to a Gaussian~~,~~but it has heavy tails and its cdf is fast to compute. It was also shown to be a better representation of the prior density than a Gaussian in an EnKF context (Bocquet et al., 2015).

To summarise, Algorithm 4 describes the analysis step for a generic LPF^x that uses anamorphosis as local resampling algorithm.

The local resampling algorithm using anamorphosis is, ~~together with~~as well as the algorithm using optimal ensemble coupling, a deterministic transformation. This means that ~~potentially large artificial~~unphysical discontinuities due to different random realisations over the grid points are avoided. As explained by Poterjoy (2016), in such an algorithm the updated ensemble members have the same *quantiles* as the prior ensemble members. The quantile property should be to some extent regular in space — for example if the ~~space~~spatial discretisation is fine enough — and this kind of regularity is transferred in the updated ensemble.

Algorithm 4 Analysis step for a generic LPF^x using anamorphosis for the local resampling

```
Require: Prior ensemble $\boldsymbol{x}^i, i = 1 \dots N_\mathrm{e}$
for $n = 1$ to $N_x$ do
    Compute the local weights $w_n^i$ using Eq. (28) or (29)
    Compute the empirical standard deviations $\sigma_n^\mathrm{f}$ and $\sigma_n^\mathrm{a}$
    Compute $c_n^\mathrm{f}$ and $c_n^\mathrm{a}$ by integrating Eqs. (40) and (41)
    for $i = 1$ to $N_\mathrm{e}$ do
        Compute $p^i = c_n^\mathrm{f}\left(x_n^i\right)$
        Solve $c_n^\mathrm{a}\left(x_n^i\right) = p^i$ for the updated local particle $x_n^i$
    end for
end for
return Updated ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_\mathrm{e}$.
```

When defining the prior and the corrected densities with Eqs. (40) and (41), we introduce some regularisation whose magnitude is controlled through the bandwidth parameter h. Regularisation is necessary to obtain continuous pdfs. Yet, it introduces an additional bias in the analysis step. Typical values of h should be around 1, with larger ensemble sizes N_e requiring smaller values for h. More generally, regularisation is widely used in PF algorithms as a fix to avoid (or at least limit the impact of) weight degeneracy, though its implementation (see Sect. 5.2) is usually different from the method used in this section.

The refinements of the resampling algorithms suggested in Sect. 4.4.2 were designed to minimise the number of ~~artificial~~unphysical discontinuities in the local resampling step. The goal of the smoothing by weights step is to ~~reduce the impact of the~~mitigate potential unphysical discontinuities after they have been introduced. On the other hand, the local resampling algorithms based on OT are designed to ~~reduce the~~mitigate the unphysical discontinuities themselves. The main difference between the algorithm based on optimal ensemble coupling and the one based on anamorphosis is that the first one is formulated in the ensemble space whereas the second one is formulated in the state space. That is to say in the first case we build an ensemble transformation $\mathbf{T}_b$ whereas in the second case we build a state transformation T_n.

Due to computational considerations, the optimisation problem Eq. (37) was only considered in one dimension. Hence, contrary to the local resampling algorithm based on optimal ensemble coupling, the one based on anamorphosis is purely one-dimensional and can only be used with blocks of size 1 grid point.

The design of the resampling algorithm based on anamorphosis has been inspired from the kernel density distribution mapping (KDDM) step of the LPF algorithm of Poterjoy (2016) which will be introduced in Sect. 7.3. However, the use of OT has different purposes. In our algorithm, we use the anamorphosis transformation to sample particles from the analysis density, whereas the KDDM step of Poterjoy (2016) is designed to correct the posterior particles — they have already been transformed — with consistent high-order statistical moments.

4.5 Summary for the LPF^x algorithms

4.5.1 Highlights

In this section, we have constructed a generic SBD localisation framework, which we have used to define the LPF^{x}s, our first category of LPF methods. The LPF^x algorithms are characterised by: the geometry of the blocks and domains (i.e. the definition of the local weights) and the resampling algorithm. As shown by Rebeschini and van Handel (2015), the LPF^x algorithms have potential to beat the curse of dimensionality. However, unphysical discontinuities are likely to arise after the assembling of locally resampled particles (van Leeuwen, 2009). In this section, we have proposed to mitigate these discontinuities by improving the design of the local resampling step. We distinguished four approaches:

1. A smoothing by weights step can be applied after the local resampling step in order to reduce potential unphysical discontinuities. Our method is a generalisation of the original smoothing designed by Penny and Miyoshi (2016) that includes spatial tapering, a smoothing strength and that is suited to the use of state blocks.

2. Simple properties of the local resampling algorithms can be used in order to minimise the occurences of unphysical discontinuity as shown by Robert and Künsch (2017).

3. Using the principles of discrete OT, we have proposed a resampling algorithm based on a local version of the ETPF of Reich (2013). This algorithm is similar to the PF part of the PF–EnKF hybrid derived by Chustagulprom et al. (2016) but it includes a more general transport cost and it is suited to the use of blocks and any resampling algorithm. By construction, the distance between the prior and the analysis local ensembles is minimised.

4. Combining the continuous OT problem with the KDE theory, we have derived a new local resampling algorithm based on anamorphosis. We have shown how it helps mitigate the unphysical discontinuities.

In Sect. 4.5.2, we discuss the numerical complexity and in Sect. 4.5.4 the asymptotic limits of the proposed LPF^x algorithms. In Sect. 4.5.3, we propose guidelines that should inform our choice of the key parameters when implementing these algorithms.

4.5.2 Numerical complexity

We define the auxiliary quantities $N_\mathrm{b}^\ell(R)$, $N_x^\ell(R)$ and $N_y^\ell(R)$ by

$$N_x^\ell(R) = \max_{b\in\{1\dots N_\mathrm{b}\}} \mathrm{Card}\left\{n\in\{1\dots N_x\}\backslash d_{n,b}\leq R\right\}, \quad (42)$$

$$N_\mathrm{b}^\ell(R) = \max_{n\in\{1\dots N_x\}} \mathrm{Card}\left\{b\in\{1\dots N_\mathrm{b}\}\backslash d_{n,b}\leq R\right\}, \quad (43)$$

$$N_y^\ell(R) = \max_{q\in\{1\dots N_\mathrm{b}\}} \mathrm{Card}\left\{q\in\{1\dots N_y\}\backslash d_{q,b}\leq R\right\}. \quad (44)$$

$N_y^\ell(R)$ is the maximum number of observation sites in a local domain of radius R. $N_\mathrm{b}^\ell(R)$ and $N_x^\ell(R)$ are the corresponding quantities for the neighborhood grid points and blocks. In a d-dimensional spatial space, these quantities are at most proportional to R^d.

The complexity of the LPF^x analysis is the sum of the complexity of computing all local weights and the complexity of the resampling. Using Eq. (28) or (29), we conclude that the complexity of computing the local weights is $\mathcal{O}\left(N_\mathrm{e}T_\mathcal{H} + N_\mathrm{b}N_\mathrm{e}N_y^\ell(r)\right)$, which depends on the localisation radius r and on the complexity $T_\mathcal{H}$ of applying the observation operator $\mathcal{H}$ to a vector. In the following paragraphs we detail the complity of each resampling algorithm.

When using the multinomial resampling of the SU sampling algorithm for the local resampling, the total complexity of the resampling step is $\mathcal{O}\left(N_xN_\mathrm{e}\right)$.

When using optimal ensemble coupling, the resampling step is computationally more expensive because it requires to solve one optimisation problem for each block. The minimisation coefficients Eq. (35) are computed with complexity $\mathcal{O}\left(N_\mathrm{e}^2N_x^\ell(r_\mathrm{d})\right)$, which depends on the distance radius r_d. The discrete OT problem Eq. (33) is a particular case of the minimum-cost flow problem and can be solved quite efficiently using the network simplex algorithm with complexity $\mathcal{O}\left(N_\mathrm{e}^3\right)$ (Pele and Werman, 2009). Applying the transformation to the block has complexity $\mathcal{O}\left(N_xN_\mathrm{b}^{-1}N_\mathrm{e}^2\right)$. Finally, the total complexity of the resampling step is $\mathcal{O}\left(N_\mathrm{b}N_\mathrm{e}^2N_x^\ell(r_\mathrm{d}) + N_\mathrm{b}N_\mathrm{e}^3 + N_xN_\mathrm{e}^2\right)$.

When using optimal transport in state space, every one-dimensional anamorphosis is computed with complexity $\mathcal{O}\left(N_p\right)$ where N_p is the one-dimensional resolution for each state variable. Therefore the total complexity of the resampling step is $\mathcal{O}\left(N_xN_\mathrm{e}N_p\right)$.

When using the smoothing by weights step with the multinomial resampling or the SU sampling algorithm, the smoothed ensemble Eq. (31) is computed with complexity $\mathcal{O}\left(N_xN_\mathrm{e}N_\mathrm{b}^\ell(r_\mathrm{s})\right)$, which depends on the smoothing radius r_s and the updated ensemble Eq. (32) is computed with complexity $\mathcal{O}\left(N_xN_\mathrm{e}\right)$. Therefore, the total complexity of the resampling and the smoothing steps is $\mathcal{O}\left(N_xN_\mathrm{e}N_\mathrm{b}^\ell(r_\mathrm{s})\right)$.

For comparison, the more costly operation in the local analysis of a local EnKF algorithm is to compute the singular value decomposition of a $N_y^\ell(r)\times N_\mathrm{e}$ matrix, which has complexity $\mathcal{O}\left(N_y^\ell(r)N_\mathrm{e}^2\right)$ assuming that $N_\mathrm{e}\leq N_y^\ell(r)$. The total complexity for a local EnKF algorithm depends on the specific implementation but should be at least $\mathcal{O}\left(N_\mathrm{b}N_y^\ell(r)N_\mathrm{e}^2\right)$.

In this complexity analysis, the influence of the parameters r, r_d and r_s is explicitly shown because a practitioner must be aware of the numerical cost of increasing these parameters. Since the resampling is performed independently for each block, this algorithmic step (which is the most costly step in practice) can be carried out in parallel, allowing a theoretical gain up to a factor N_b.

4.5.3 ~~Parameter choice and asymptotic limit~~Choice of key parameters

~~The localisation radius r controls the impact of the curse of dimensionality. Therefore, *relevant* values for r should be relatively small — smaller than what would be required for an EnKF using domain localisation for example — to avoid experiencing immediate weight degeneracy. For realistic models with two or more spatial dimensions, the number of potential observation sites in the local domain grows as r^2 or more. Therefore, optimal values for the localisation radius should be really small, maybe too small for the method to follow the truth trajectory. If this is the case, then localisation alone would not be enough to make the PF operational.~~The localisation radius r controls the number of observation sites in the local domains $N_y^\ell(r)$ and the impact of the curse of dimensionality. To avoid immediate weight degeneracy, r should therefore be relatively small — smaller than what would be required for an EnKF using domain localisation for example. This is especially true for realistic models with two or more spatial dimensions in which $N_y^\ell(r)$ grows as r^2 or more. In this case, it can happen that the localisation radius r have to be too small for the method to follow the truth trajectory (either because too much information is ignored, or because there is too much spatial variation in the local weights) which would mean that localisation alone would not be enough to make PF methods operational.

For a local EnKF algorithm, gathering grid points into ~~local state~~ blocks is an approximation that reduces the numerical cost of the analysis steps by reducing the number of local analyses to perform. For ~~a~~an LPF^x algorithm, the local analyses should in general be ~~much faster because there is no matrix inversion to perform~~faster (see the complexity analysis in Sect. 4.5.2). In this case, using ~~bigger state~~larger blocks is a way to decrease the proportion of block borders, which are potential spots for ~~artificial~~unphysical discontinuities. However, increasing the size of the ~~local state~~ blocks reduces the number of degrees of freedom to counteract the curse of dimensionality. It also introduces an additional bias in the local weight update, Eq. (28) or (29), since the local weights are computed relatively to the block centers. This issue was identified by Rebeschini and van Handel (2015) as a source of spatial inhomogeneity of the error. ~~In any case, the local state~~For these reasons, the blocks should be small (no more than a few grid points)~~only big ensemble could benefit from the use of larger state blocks~~. Only large ensembles could potentially benefit from larger blocks.

More discussion regarding the choice of the localisation radius r and the number of blocks N_{b}, but also regarding the choice of other parameters (the smoothing radius r_{s}, the smoothing strength α_{s}, the distance radius r_{d} and the regularisation bandwidth h) can be found in Sect. 5.

4.5.4 Asymptotic limit

~~The analysis step of LPF^{x}s using the multinomial resampling or the SU sampling as resampling algorithm is equivalent to a PF analysis step in the asymptotic limit $r \to \infty$ if the same random number(s) is (are) used for the resampling of all state blocks or if there is only one state block. The analysis step of LPF^{x}s using optimal ensemble coupling is equivalent to the ETPF analysis step in the asymptotic limit $r \to \infty$ and $r_{\mathrm{d}} \to \infty$ for the localisation and the distance radius, respectively. Finally, even in the asymptotic limit $h \to 0$ and $r \to \infty$, we could not find a proof that the analysis step of LPF^{x}s using anamorphosis is asymptotically unbiased.~~

An essential property of PF algorithms is that they are asymptotically Bayesian: as stated in Sect. 2.2, under reasonable assumptions the empirical analysis density converges to the true analysis density for the weak topology on the set of probability measures over $\mathbb{R}^{N_x}$ in the limit $N_{\mathrm{e}} \to \infty$. In this section, we study under which conditions the LPF^x analysis can be equivalent to a (global) PF analysis and therefore be asymptotically Bayesian.

In the limit of very large localisation radius, $r \to \infty$, the local weights Eq. (28) and (29) are equal to the (global) weights of the (global) PF. However, this does not imply that the LPF^x analysis is equivalent to a PF analysis because the resampling is performed independently for each block. Yet we can distinguish the following cases in the limit $r \to \infty$:

- When using independent multinomial resampling or SU sampling for the local resampling, if one uses the same random number for all blocks (this property is always true if $N_{\mathrm{b}} = 1$), then the LPF^x analysis is equivalent to the analysis of the PF.
- When using the smoothing by weigths step with the multinomial resampling or the SU sampling, if one uses the same random number for all blocks then the smoothed ensemble Eq. (31) is equal to the (locally) resampled ensemble and the smoothing has no effect: we are back to the first case.
- When using optimal ensemble coupling for the local resampling, in the limit $r_{\mathrm{d}} \to \infty$, the LPF^x analysis is equivalent to the analysis of the (global) ETPF.

For other cases, we cannot give a firm conclusion:

- When using independent multinomial resampling or SU sampling for the local resampling with different random number for all blocks, then the updated particles are distributed according to the product of the marginal analysis density Eq. (26), which is in general different from the analysis density even in the limit $r \to \infty$.
- For the same reason, when using anamorphosis for the local resampling, we could not find a proof that the LPF^x analysis is asymptotically Bayesian, even in the limit $h \to 0$ and $r \to \infty$.
- When using the smoothing by weigths step with the multinomial resampling or the SU sampling, in the limit $r \to \infty$ and $r_{\mathrm{s}} \to \infty$ the smoothed ensemble Eq. (31) can be different from the updated ensemble of the global PF because the resampling is performed independently for each block.

5 Numerical illustration of LPF^x algorithms with the Lorenz-96 model

5.1 Model specifications

In this section, we illustrate the performance of LPF^{x}s with twin simulations of the L96 model in the standard (mildly nonlinear) configuration described in Appendix A3. For this series of experiments, as for all experiments in this paper, the synthetic truth is computed without model error. This is usually a stringent constraint for the PF methods for which accounting for model error is a means for regularisation. But on the other hand it allows for a fair comparison with the EnKF and it ~~overcomes~~avoids the issue of defining a realistic model noise.

The distance between the truth and the analysis is measured with the average analysis root mean square error, hereafter simply called the RMSE. To ensure the convergence of the statistical indicators, the runs are at least $5 \times 10^4 \Delta t$ long with an additional $10^3 \Delta t$ spin-up period. An advantage of using PF methods is that it should asymptotically yield sharp though reliable ensembles~~, properties which cannot a priori~~. This may not be entirely reflected in the RMSE. However, not only does the RMSE ~~offers~~offer a clear ranking of the algorithms but it is an indicator that measures the adequacy to the primary goal of data assimilation, i.e. mean state estimation. Moreover, for a sufficiently cycled DA problem, it seems likely that good RMSE scores can only be achieved with ensembles of good quality in the light of most other indicators. Nonetheless, in addition to the RMSE, rank histograms meant to assess the quality of the ensembles are computed and reported in Appendix D for a selection of experiments.

For the localisation, we assume that the grid points are positioned on an axis with a regular spacing of 1 unit of length and periodic boundary conditions consistent with the system

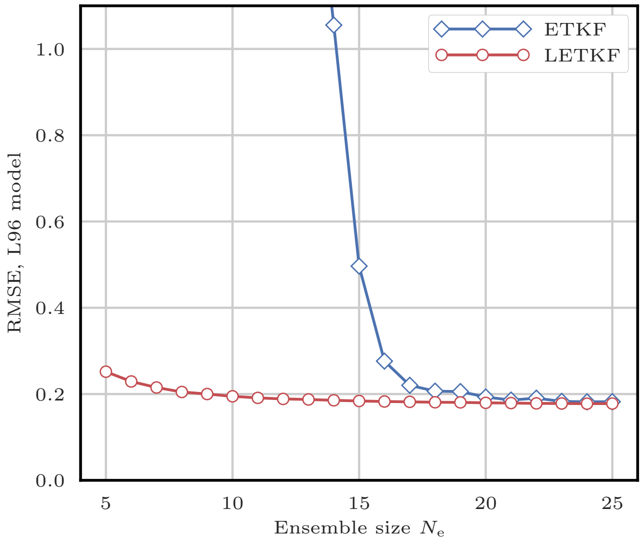


Figure 4. RMSE as a function of the ensemble size N_e for the ETKF and the LETKF.

size. Therefore, the local domain centered on the n-th grid point ~~n~~ is composed of the points $\{n-\lfloor r\rfloor,\dots,n+\lfloor r\rfloor\}$, where $\lfloor r\rfloor$ is the integer part of the localisation radius and the N_b ~~local~~ blocks consist of N_x/N_b consecutive grid points.

This filtering problem has been widely used to asses the performance of DA algorithms. In this configuration, nonlinearities in the model are rather ~~weak~~mild and representative of synoptic scale meteorology and the error distributions are close to Gaussian. ~~A typical EnKF implementation (e.g. the ensemble transform Kalman filter, ETKF in short) with an ensemble of 20 members does not require localisation and yield an RMSE around $0.18-0.19$. With optimally tuned localisation and a ensemble of only 10 members the mean RMSE is around 0.20.~~ As a reference, the evolution of the RMSE as a function of the ensemble size N_e is shown in Fig. 4 for the ensemble transform Kalman filter (ETKF) and its local version (LETKF). For each value of N_e, the multiplicative inflation parameter and the localisation radius (for the LETKF) are optimally tuned to yield the lowest RMSE. In most of the following figures related to the L96 test series, we draw a baseline at 0.2, roughly the RMSE of the LETKF with $N_\mathrm{e}=10$ particles. Note that slightly lower RMSE scores can be achieved with larger ensembles.

5.2 Perfect model and regularisation

The application of PF algorithms to this chaotic model without error leads to a fast collapse. Even with stochastic models that account for some model error, PF algorithms experience weight degeneracy when the model noise is too low. Therefore, PF practitioners commonly include some additional jitter to mitigate the collapse (e.g. Pham, 2001). As described by Musso et al. (2001), jitter can be added in two different ways.

5.2.1 Pre-regularisation

Firstly, the prediction and sampling step, Eq. (7), can be performed using a stochastic extension of the model:

$$\boldsymbol{x}_{k+1}^{i}-\mathcal{M}\left(\boldsymbol{x}_{k}^{i}\right)=\boldsymbol{w}_{k}\sim\mathcal{N}\left(\mathbf{0},q^{2}\mathbf{I}\right), \tag{45}$$

where $\mathcal{M}$ is the model associated to the integration scheme of the ordinary differential equations (ODEs) and q is a tunable parameter. This jitter is meant to compensate for the deterministic nature of the given model. In this case, the truth could be seen as a trajectory of the perturbed model Eq. (45) with a realisation of the noise that is identically zero. In the literature, this method is called pre–regularisation (Le Gland et al., 1998) because the jitter is added before the correction step.

5.2.2 Post-regularisation

Secondly, a regularisation step can be added after a full analysis cycle:

$$\boldsymbol{x}_{k+1}^{i}\leftarrow\boldsymbol{x}_{k+1}^{i}+\boldsymbol{u},\quad\boldsymbol{u}\sim\mathcal{N}\left(\mathbf{0},s^{2}\mathbf{I}\right), \tag{46}$$

where s is a tunable parameter. As opposed to the first method, it can be seen as a jitter before integration: the noise is integrated by the model before the next analysis step, while smoothing potential ~~unrealistic~~unphysical discontinuities. ~~Somehow~~In some ways this method is similar to ~~ensemble~~additive inflation in EnKF algorithms ~~that is meant to compensate for the finite size N_e of the ensemble~~. It is called post–regularisation (Musso and Oudjane, 1998; Oudjane and Musso, 1999) because the jitter is added after the correction step.

5.2.3 Numerical complexity and asymptotic limit

Both regularisation steps have numerical complexity $\mathcal{O}\left(N_xN_\mathrm{e}T_\mathrm{r}\right)$, with T_r being the complexity of drawing one random number according to the univariate standard normal law $\mathcal{N}\left(0,1\right)$.

The exact LPF is recovered in the limit $q\to0$ and $s\to0$.

5.2.4 Standard S(IR)^{x}R algorithm

With optimally tuned jitter for the standard L96 model, the bootstrap PF algorithm requires about 200 particles to give on average more ~~informations~~information than the ~~truth~~observations.[2] With 10^3 particles, ~~the~~its RMSE is around 0.6 and with 10^4 it is around 0.4.

---

[2] We have proven in this case that the RMSE, when computed between the observations $\boldsymbol{y}_k$ and truth $\boldsymbol{x}_k$, has an expected value of 0.98.

We define the standard S(IR)^{x}R algorithm — sampling, importance, resampling, regularisation, the x exponent meaning that steps in parentheses are performed locally ~~on~~for each ~~state~~ block — as the LPF^x ~~algorithm~~ ~~(see~~ Algorithm 1) with the following characteristics:

- grid points are gathered into N_b blocks of N_x/N_b connected grid points;
- jitter is added after the integration using Eq. (45) with a standard deviation controlled by q;
- the local weights are computed using the Gaussian tapering of observation ~~site~~ influence given by Eq. (29), where G is the piecewise rational function of Gaspari and Cohn (1999);
- the local resampling is performed independently for each block with the adjustment-minimising SU sampling algorithm;
- jitter is added at the end of each assimilation cycle using Eq. (46) with a standard deviation controlled by s.

The standard deviation of the jitter after integration (q) and before integration (s) shall be called "integration jitter" and "regularisation jitter", respectively. The S(IR)^{x}R algorithm has five parameters: $(N_\mathrm{e}, N_\mathrm{b}, r, q, s)$. All algorithms tested in this section are variants of this standard algorithm and are named ~~$\mathrm{S}(\mathrm{I}_*\ast)^x\ast\mathrm{R}_*$~~$\mathrm{S}(\alpha\beta)^x\gamma\delta$ with the conventions detailed in Table 1. Table 2 lists all LPF^x algorithms tested in this section and reports their characteristics according to the convention of Table 1.

5.3 Tuning the localisation radius

We first check that, in this standard configuration, localisation is working by testing the S(IR)^{x}R algorithm with $N_\mathrm{b} = 40$ blocks of size 1 grid point. We take $N_\mathrm{e} = 10$ particles, $q = 0$ (perfect model) and several values for the regularisation jitter s. The evolution of the RMSE as a function of the localisation radius r is shown in Fig. 5. With SBD localisation, the LPF yields an RMSE around 0.45 in a regime where the bootstrap PF algorithm is degenerate. The compromise between bias (small values of r, too much information is ~~dropped~~ignored or there is too much spatial variation in the local weights) and variance (large values of r, the weights are degenerate) reaches an optimum around $r = 3$ grid points. As expected, the local domains are ~~very~~quite small (5 observation sites) in order to efficiently counteract the curse of dimensionality.

5.4 Tuning the jitter

To evaluate the efficiency of the jitter, we experiment with the S(IR)^{x}R algorithm with $N_\mathrm{e} = 10$ particles, $N_\mathrm{b} = 40$ blocks of size 1 grid point and a localisation radius $r = 3$ grid points. The evolution of the RMSE as a function of the integration jitter q is shown in Fig. 6 and as a function of the regularisation jitter s in Fig. 7.

Table 1. Nomenclature conventions for the $\mathrm{S}(\alpha\beta)^x\gamma\delta$ algorithms. Capital letters refer to the main algorithmic ingredients: "I" for importance, "R" for reampling or regularisation, "T" for transport, "S" for smoothing. Subscripts are used to distinguish the methods in two different ways. Lower case subscripts refer to explicit concepts used in the method: "ng" for non-Gaussian, "su" for stochastic universal, "s" for state space and "c" for colour; while upper case subscripts refer to the work that inspired the method: "PM" for Penny and Miyoshi (2016) and "R" for Reich (2013). For simplicity, some subscripts are omitted: "g" for Gaussian, "amsu" for adjustment-minimising stochastic universal and "w" for white. Finally note that we used the subscript "d" (for deterministic) to indicate that the same random numbers are used for the resampling over all blocks.

α	Local importance weights (Sect. 4.2.2)
I_ng	Eq. (28) [non-Gaussian]
I	Eq. (29) [Gaussian]
β	Local resampling algorithm (Sect. 4.4)
R_su	SU sampling algorithm
R_d	adjustment-minimising SU sampling algorithm with the same random numbers over all blocks
R	adjustment-minimising SU sampling algorithm
T_R	optimal transport in ensemble space
T_s	optimal transport in state space
γ	Smoothing-by-weights (Sect. 4.4.1)
S_PM	enabled
–	disabled
δ	Regularisation method (Sects. 5.2 and 5.8)
R	white noise method
R_c	coloured noise method

From these results, we can identify two regimes:

- with low regularisation jitter ($s < 0.15$), the filter stability is ensured by the integration jitter, with optimal values around $q = 1.25$;
- with low integration jitter ($q < 0.5$), the stability is ensured by the regularisation jitter, with optimal values around $s = 0.26$.

As expected, adding jitter before integration (i.e. with s) yields significantly better results. This indicates that the model integration indeed smoothes the jitter out and removes ~~artificial~~unphysical discontinuities for the correction step. We observed the same tendency for most LPFs tested in this ~~study~~article.

In the rest of this section, we take zero integration jitter ($q = 0$) and the localisation radius r and the regularisation

Table 2. List of all LPF^x algorithms tested in this article. For each algorithm, the main characteristics are reported with appropriate references. The last column indicate the section in which benchmarks based on the L96 model can be found.

Algorithm	Local importance weights (Sect. 4.2.2)		Local resampling algorithm (Sect. 4.4)	Subsection	Smoothing by weights (Sect. 4.4.1)			Regularisation method (Sects. 5.2 and 5.8)		L96 benchmark sections
	Eq. (28) [non-Gaussian]	Eq. (29) [Gaussian]			– [disabled]	Eq. (30) [enabled]	Eq. (31) [enabled]	Eq. (46) [white]	Eq. (50) [colour]	
$S(IR)^xR$		✓	adjustment-minimising SU sampling	4.4.2	✓			✓		5.3 to 5.12
$S(I_{ng}R)^xR$	✓		adjustment-minimising SU sampling	4.4.2	✓			✓		5.6
$S(IR_{su})^xR$		✓	SU sampling	–	✓			✓		5.7
$S(IR_d)^xR$		✓	adjustment-minimising SU sampling with the same random numbers	4.4.2	✓			✓		5.7
$S(IR)^xR_c$		✓	adjustment-minimising SU sampling	4.4.2	✓				✓	5.8 to 5.11
$S(IR)^xS_{PM}R$		✓	adjustment-minimising SU sampling	4.4.2			✓	✓		5.9
$S(IR)^xS_{PM}R_c$		✓	adjustment-minimising SU sampling	4.4.2			✓		✓	5.9, 5.12
$S(IT_R)^xR$		✓	optimal ensemble coupling	4.4.3	✓			✓		5.10, 5.12
$S(IT_R)^xR_c$		✓	optimal ensemble coupling	4.4.3	✓				✓	5.10
$S(IT_R)^xS_{PM}R$		✓	optimal ensemble coupling	4.4.3		✓		✓		5.10
$S(IT_R)^xS_{PM}R_c$		✓	optimal ensemble coupling	4.4.3		✓			✓	5.10
$S(IT_s)^xR$		✓	anamorphosis	4.4.4	✓			✓		5.11, 5.12
$S(IT_s)^xR_c$		✓	anamorphosis	4.4.4	✓				✓	5.11
$S(IT_s)^xS_{PM}R$		✓	anamorphosis	4.4.4		✓		✓		5.11

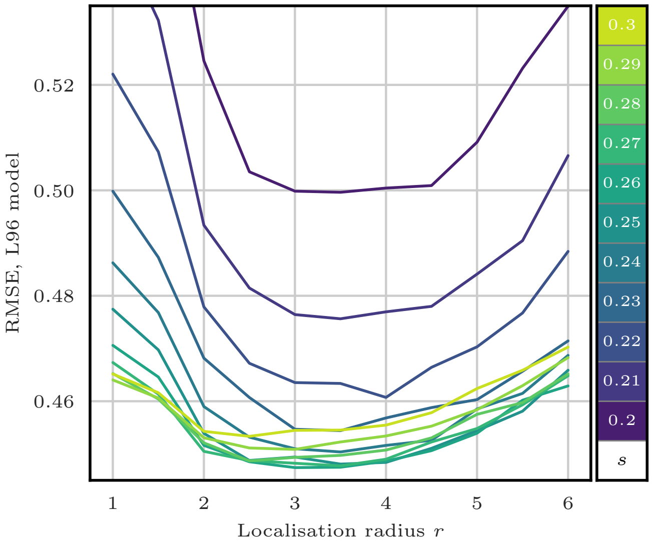


Figure 5. RMSE as a function of the localisation radius r for the S(IR)^{x}R algorithm with $N_e = 10$ particles, $N_b = 40$ blocks of size 1 grid point and no integration jitter ($q = 0$). For each r, several values for the regularisation jitter s are tested as shown by the ~~color~~colour scale.

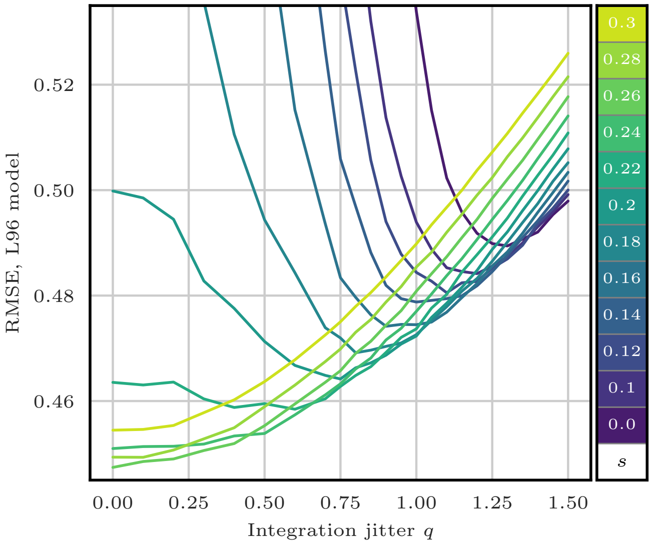


Figure 6. RMSE as a function of the integration jitter q for the S(IR)^{x}R algorithm with $N_e = 10$ particles, $N_b = 40$ blocks of size 1 grid point and a localisation radius $r = 3$ grid points. For each q, several values for the regularisation jitter s are tested as shown by the ~~color~~colour scale.

jitter s are systematically tuned to yield the lowest RMSE score.

5.5 Increasing the size of the ~~local~~ blocks

1420 To illustrate the influence of the size of the ~~local~~ blocks, we compare the RMSEs obtained by the S(IR)^{x}R algorithm

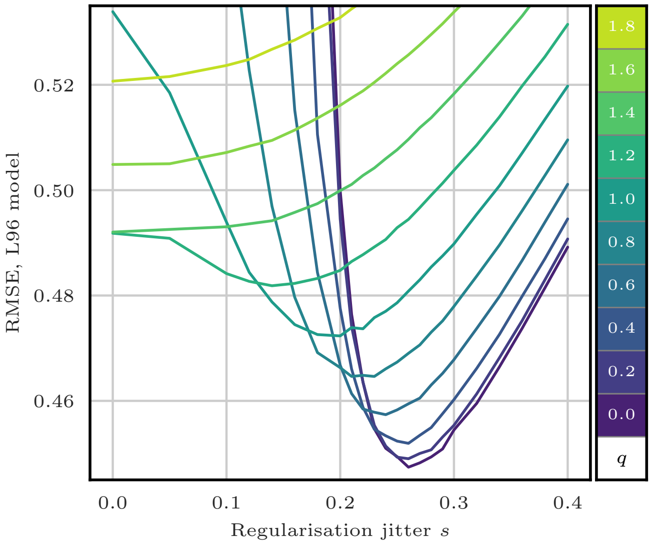


Figure 7. RMSE as a function of the regularisation jitter s for the S(IR)^{x}R algorithm with $N_\mathrm{e} = 10$ particles, $N_\mathrm{b} = 40$ blocks of size 1 grid point and a localisation radius $r = 3$ grid points. For each s, several values for the integration jitter q are tested as shown by the ~~color~~colour scale.

with various fixed number of blocks N_b. The evolution of the RMSE as a function of the ensemble size N_e is shown in Fig. 8. For small ensemble sizes, using larger blocks is inefficient, because of the need for degrees of freedom to counteract the curse of dimensionality. Only very ~~big~~large ensembles benefit from ~~the use of~~using large blocks as a consequence of the reduction of proportion of block boundaries, potential spots for ~~artificial~~unphysical discontinuities.

From now on, unless specified otherwise, we systematically test our algorithms with $N_\mathrm{b} = 40$, 20, 10 blocks of respectively 1, 2, 4 grid points and we keep the best RMSE score.

5.6 Choice of the local weights

To illustrate the influence of the definition of the local weights, we compare the RMSEs of the S(IR)^{x}R and the S(I$_\mathrm{ng}$R)^{x}R algorithms. These two variants only differ by their definition of the local importance weights: the S(IR)^{x}R algorithm uses the Gaussian tapering of observation influence defined by Eq. (29) while the S(I$_\mathrm{ng}$R)^{x}R algorithm uses the non-Gaussian tapering given by Eq. (28).

Figure 9 shows the evolution of the RMSE as a function of the ensemble size N_e. The Gaussian version of the definition of the weights always yields better results. This is probably a consequence of the ~~approximate Gaussianity of this configuration~~fact that, in this configuration nonlinearities are ~~weak~~mild and the error distributions are close to Gaussian. In the following, we always use Eq. (29) to define the local weights.

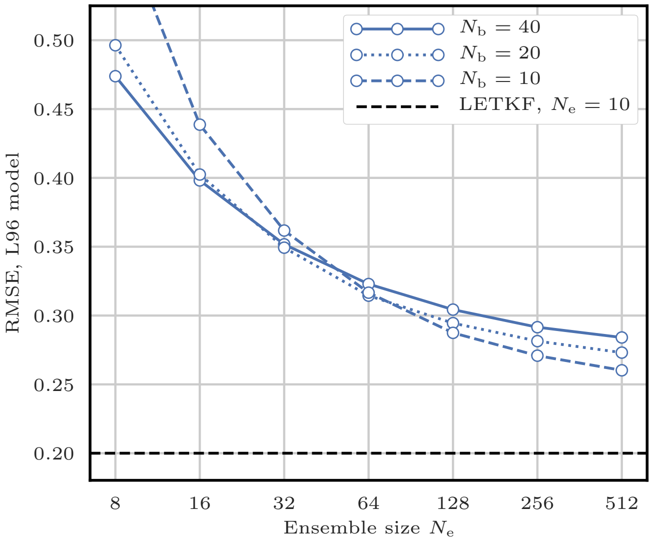


Figure 8. RMSE as a function of the ensemble size N_e for the S(IR)^{x}R algorithm with respectively $N_\mathrm{b} = 40, 20, 10$ blocks of size $1, 2, 4$ grid points.

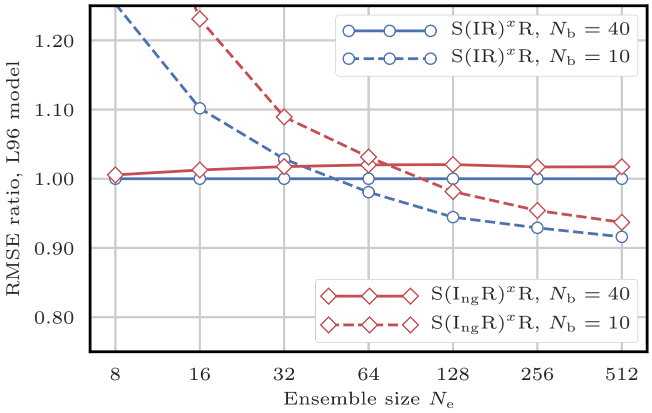


Figure 9. RMSE as a function of the ensemble size N_e for the S(IR)^{x}R and the S(I$_\mathrm{ng}$R)^{x}R algorithms with respectively $N_\mathrm{b} = 40, 10$ blocks of size $1, 4$ grid points. The scores are displayed in units of the RMSE of the S(IR)^{x}R algorithm with $N_\mathrm{b} = 40$ blocks of size 1 grid point.

5.7 Refining the stochastic universal sampling

~~Here, the sampling algorithms proposed in Sect. 4.4.2 are tested.~~In this section, we test the refinements of the sampling algorithms proposed in Sect. 4.4.2. To do this we compare, the RMSEs of the S(IR)^{x}R algorithms with those of:

- the S(IR$_\mathrm{d}$)^{x}R algorithm, for which the same random numbers are used for the resampling of each block;
- the S(IR$_\mathrm{su}$)^{x}R algorithm, which uses the SU sampling algorithm without the adjustment-minimising property.

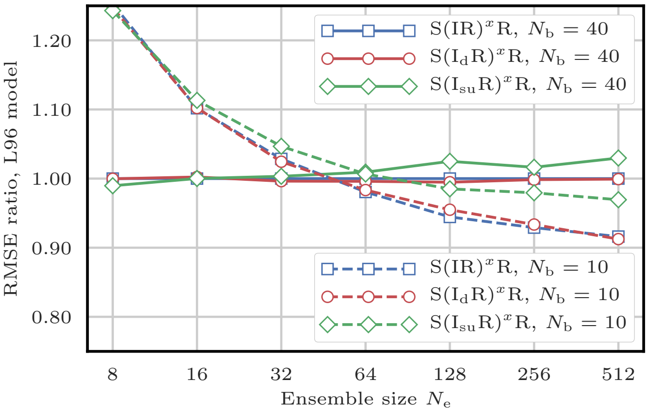


Figure 10. RMSE as a function of the ensemble size N_e for the $\mathrm{S(IR)}^x\mathrm{R}$, the $\mathrm{S(IR_d)}^x\mathrm{R}$ and the the $\mathrm{S(IR_{su})}^x\mathrm{R}$ algorithms with respectively $N_\mathrm{b} = 40, 10$ blocks of size $1, 4$ grid points. The scores are displayed in units of the RMSE of the $\mathrm{S(IR)}^x\mathrm{R}$ algorithm with $N_\mathrm{b} = 40$ blocks of size 1 grid point.

Figure 10 shows the evolution of the RMSE as a function of the ensemble size N_e. ~~The higher RMSEs of the $\mathrm{S(IR_{su})}^x\mathrm{R}$ algorithm, especially for large ensembles, show~~The $\mathrm{S(IR_{su})}^x\mathrm{R}$, the only algorithm that does not satisfy the adjustment-minimising property, yields higher RMSEs. This shows that the adjustment-minimising property is indeed an efficient way of reducing the number of ~~artificial~~unphysical discontinuities introduced during the resampling step.

However, using the same random number for the resampling of each block does not produce significantly lower RMSEs. This method is insufficient to reduce the number of ~~artificial~~unphysical discontinuities introduced when ~~concatenating~~assembling the locally updated particles. This is probably a consequence of the fact that the SU sampling algorithm only uses one random number to compute the resampling map. ~~It is also a hint at the probably low influence of the specific realisation of this random number on long-term statistical properties.~~It also suggests that the specific realisation of this random number has a weak influence on long-term statistical properties.

In the following, when using the SU sampling algorithm, we always choose its adjustment-minimising form but we do not enforce the same random numbers over different blocks.

5.8 Colourising the regularisation

5.8.1 Colourisation for global PFs

According to Eqs. (45) and (46), the regularisation jitters are white noises. In realistic models, different state variables may take their values in disjoint intervals (e.g., the temperature takes values around 300 K and the wind speed can take its values between -10 and $10\ \mathrm{m\,s^{-1}}$) which makes these jittering methods inadequate.

It is hence a common procedure in ensemble DA to scale the regularisation jitter with statistical properties of the ensemble. In a (global) PF context, practitioners often "colourise" the Gaussian regularisation jitter with the empirical covariances of the ensemble as described by Musso et al. (2001). ~~The regularisation jitter being added after the resampling step, in order to mitigate the effect of resampling noise, the jitter is scaled with the weighted ensemble before resampling.~~Since the regularisation jitter is added after the resampling step, it is scaled with the weighted ensemble before resampling in order to mitigate the effect of resampling noise.

More precisely, the regularisation jitter has zero mean and $N_x \times N_x$ covariance matrix given by

$$[\mathbf{\Sigma}]_{n,m} = \frac{\hat{h}}{1 - \sum\limits_{i=1}^{N_\mathrm{e}} \left(w^i\right)^2} \sum_{i=1}^{N_\mathrm{e}} w^i \left(x_n^i - \overline{x}_n\right)\left(x_m^i - \overline{x}_m\right), \quad (47)$$

where $\hat{h}$ is the bandwidth, a free parameter, and $\overline{x}_n$ is the ensemble mean ~~at~~for the n-th state variable~~n~~:

$$\overline{x}_n = \frac{1}{N_e} \sum_{i=1}^{N_\mathrm{e}} x_n^i. \quad (48)$$

In practice, the $N_x \times N_\mathrm{e}$ anomaly matrix $\mathbf{X}$ is defined by

$$[\mathbf{X}]_{n,i} = \sqrt{\frac{\hat{h} w^i}{1 - \sum\limits_{i=1}^{N_\mathrm{e}} \left(w^i\right)^2}} \left(x_n^i - \overline{x}_n\right), \quad (49)$$

and the regularisation is added as

$$\mathbf{E} \leftarrow \mathbf{E} + \mathbf{X}\mathbf{Z}, \quad (50)$$

~~with~~where $\mathbf{E}$ is the ensemble matrix and $\mathbf{Z}$ ~~being~~is a $N_\mathrm{e} \times N_\mathrm{e}$ random matrix whose coefficients are distributed according to a normal law such that $\mathbf{XZ}$ is a sample from the Gaussian distribution with zero mean and covariance matrix $\mathbf{\Sigma}$. In this case, the regularisation fits in the LET framework with a random transformation matrix.

Colourisation could ~~as well~~ be added as well to the integration jitter. However in this case, scaling the noise with the ensemble is less justified than for the regularisation jitter. Indeed, the integration noise is inherent to the perturbed model that is used to evolve each ensemble member independently. Hence PF practitioners often take a time–independent Gaussian integration noise whose covariance matrix does not depend on the ensemble but includes some off-diagonal terms based on the distance between ~~state variables~~grid points (e.g., Ades and van Leeuwen, 2015). However, as we mentioned in Sect. 5.4, we do not use integration jitter for the rest of this ~~section~~article.

5.8.2 Colourisation for LPFs

The 40 variables of the L96 model in its standard configuration are statistically homogeneous with short-range correlations. This is the main reason of the efficiency of the white noise jitter in the S(IR)^{x}R algorithm and its variants tested so far. We still want to investigate the potential gains of using ~~colored~~coloured jitters in LPF^{x}s.

In the analysis step of LPF^x~~s, there is a unique weight for each state variable n~~algorithms, at each grid point there is a different set a local weights w_n^i, therefore it is not possible to compute the covariance of the regularisation jitter with Eq. (47). We propose two different ways of circumventing this obstacle.

A first approach could be to scale the regularisation with the locally resampled ensemble, since in this case all weights are equal. This is the approach followed by Reich (2013) and Chustagulprom et al. (2016) under the name "particle rejuvenation". However, this approach systematically leads to higher RMSEs for the S(IR)^{x}R algorithm (not shown here). This can be potentially explained by two factors. ~~Firstly~~Firstly, the resampling could introduce noise in the computation of the anomaly matrix $\mathbf{X}$. Secondly, the fact that the resampling is performed independently for each block perturbs the propagation of multivariate properties (such as sample covariance) over different blocks.

In a second approach, the anomaly matrix $\mathbf{X}$ is defined by the weighted ensemble before resampling, i.e. using the local weights w_n^i, as ~~following~~follows:

$$[\mathbf{X}]_{n,i} = \sqrt{\frac{\hat{h} w_n^i}{1 - \sum\limits_{i=1}^{N_\mathrm{e}} \left(w_n^i\right)^2}} \left(x_n^i - \overline{x}_n\right). \tag{51}$$

In this case, the Gaussian regularisation jitter has covariance matrix:

$$[\mathbf{\Sigma}]_{n,m} = \sum_{i=1}^{N_\mathrm{e}} \frac{\hat{h}\sqrt{w_n^i w_m^i}\left(x_n^i - \overline{x}_n\right)\left(x_m^i - \overline{x}_m\right)}{\sqrt{\left(1 - \sum\limits_{i=1}^{N_\mathrm{e}} \left(w_n^i\right)^2\right)\left(1 - \sum\limits_{i=1}^{N_\mathrm{e}} \left(w_m^i\right)^2\right)}}, \tag{52}$$

which is ~~an extension~~a generalisation of Eq. (47). This method can as well be seen as a generalisation of the adaptative inflation used by Penny and Miyoshi (2016). For their adaptative inflation, Penny and Miyoshi (2016) only computed the diagonal of the matrix $\mathbf{X}$ and fixed the bandwidth parameter $\hat{h}$ to 1. Our approach yields lowest RMSEs in all tested cases, which is most probably due to the tuning of the bandwidth parameter $\hat{h}$.

5.8.3 Numerical complexity and asymptotic limit

The coloured regularisation step has complexity $\mathcal{O}\left(N_x N_\mathrm{e}^2\right)$. It is slightly more costly than using the white noise regularisation step due to the matrix product Eq. (50).

The exact LPF is recovered in the limit $\hat{h} \to 0$.

5.8.4 Illustrations

We ~~then~~ experiment with the S(IR)^{x}R$_\mathrm{c}$ algorithm, in which the regularisation jitter is colourised as described by Eqs. (50) and (51). In this algorithm, the parameter s (regularisation jitter standard deviation) is replaced by the bandwidth parameter $\hat{h}$, hereafter simply called regularisation jitter. The evolution of the RMSE as a function of $\hat{h}$ for the S(IR)^{x}R$_\mathrm{c}$ algorithm (not shown here) is very similar to the evolution of the RMSE as a function of s for the S(IR)^{x}R algorithm. In the following, when using the coloured regularisation jitter method, $\hat{h}$ is systematically tuned to yield the lowest RMSE score.

Figure 11 shows the evolution of the RMSE as a function of the ensemble size N_e for the S(IR)^{x}R and the S(IR)^{x}R$_\mathrm{c}$ algorithms. These two variants only differ by the regularisation method. The S(IR)^{x}R algorithm uses white regularisation jitter while the S(IR)^{x}R$_\mathrm{c}$ algorithm uses coloured regularisation jitter. For small ensembles, the S(IR)^{x}R$_\mathrm{c}$ algorithm yields higher RMSEs, ~~but for big ensembles, the RMSEs are slightly better when the regularisation jitter is colourised.~~whereas it shows slightly better RMSEs for larger ensembles. Depending on the block size, the transition between both regimes happens around $N_\mathrm{e} = 32$ to 64 particles. The higher RMSEs of the S(IR)^{x}R$_\mathrm{c}$ algorithm for small ensembles may have two potential explanations. Firstly, even if the L96 model in its standard configuration is characterised by short-range correlations, the covariance matrix $\mathbf{\Sigma}$ is a high-dimensional object that is poorly represented with a weighted ensemble. Secondly,~~ despite the relative Gaussianity of this configuration,~~ the analysis distribution for small ensemble may be too different from a Gaussian for the coloured regularisation jitter method to yield better results even though in this mildly nonlinear configuration the densities are close to Gaussian.

5.9 Applying a smoothing by weights

In this section, we look for the potential benefits of adding a smoothing by weights step as presented in Sect. 4.4.1, by testing the S(IR)^{x}S$_\mathrm{PM}$R and the S(IR)^{x}S$_\mathrm{PM}$R$_\mathrm{c}$ algorithms. These algorithms only differ from the S(IR)^{x}R and the S(IR)^{x}R$_\mathrm{c}$ algorithms by the fact that they add a smoothing by weights step as specified in Algorithm 2.

Alongside with the smoothing by weights step come two additional tuning parameters: the smoothing strength α_s and the smoothing radius r_s. We first investigate the influence of theses parameters. Figure 12 shows the evolution of the RMSE as a function of the smoothing radius r_s for the S(IR)^{x}S$_\mathrm{PM}$R with $N_\mathrm{e} = 10$ particles, $N_\mathrm{b} = 40$ blocks of size 1 grid point for several values of the smoothing strength α_s. As before, the localisation radius r and the regularisation jitter s are optimally tuned.

At a fixed smoothing strength $\alpha_\mathrm{s} > 0$, starting from $r_\mathrm{s} = 1$ grid point (no smoothing), the RMSE decreases when r_s in-

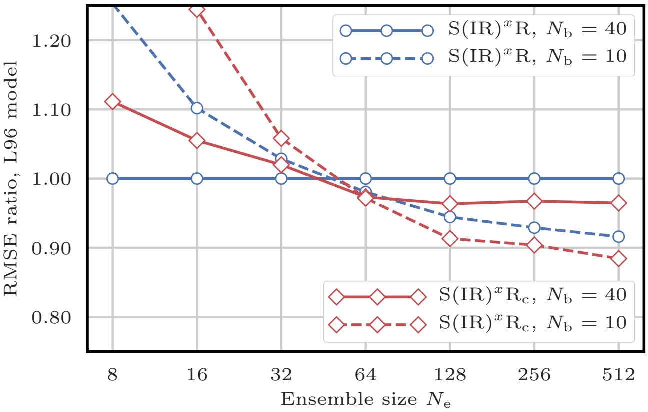


Figure 11. RMSE as a function of the ensemble size N_e for the S(IR)^{x}R and the S(IR)^{x}R$_\mathrm{c}$ algorithms with respectively $N_\mathrm{b} = 40, 10$ blocks of size $1, 4$ grid points. The scores are displayed in units of the RMSE of the S(IR)^{x}R algorithm with $N_\mathrm{b} = 40$ blocks of size 1 grid point.

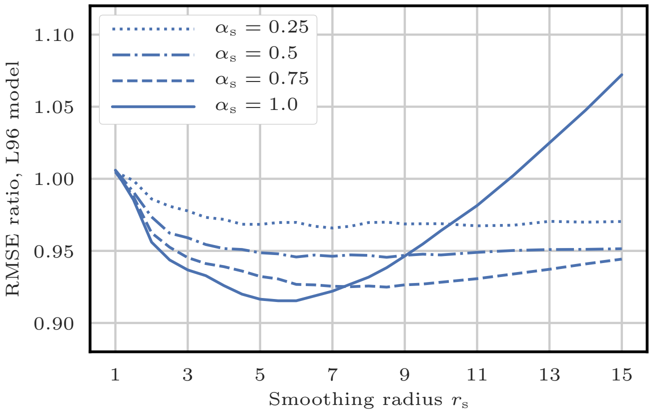


Figure 12. RMSE as a function of the smoothing radius r_s for the S(IR)^{x}S$_\mathrm{PM}$R algorithms with $N_\mathrm{e} = 16$ particles and $N_\mathrm{b} = 40$ blocks of size 1 grid point for several values of the smoothing strength α_s. The scores are displayed in units of the RMSE of the S(IR)^{x}R algorithm with $N_\mathrm{e} = 16$ particles and $N_\mathrm{b} = 40$ blocks of size 1 grid point.

creases. It reaches a minimum and then increases again. In this example, the optimal smoothing radius r_s ~~was found~~lies between 5 and 6 grid points for a smoothing strength $\alpha_\mathrm{s} = 1$, with corresponding optimal localisation radius r between 2 and 3 grid points and optimal regularisation jitter s around 0.45 (not shown here). For comparison, the optimal tuning parameters for the S(IR)^{x}R algorithm in the same configuration were r between 4 and 5 grid points and s around 0.2.

Based on extensive tests of the S(IR)^{x}S$_\mathrm{PM}$R and the S(IR)^{x}S$_\mathrm{PM}$R$_\mathrm{c}$ algorithms with N_e ranging from 8 to 128 particles (not shown here), we ~~found~~conclude that:

- in general $\alpha_\mathrm{s} = 1$ is optimal, or at least only slightly suboptimal;
- optimal values for r and s are larger with the smoothing by weights step than without it;
- optimal values for r and r_s are not related and must be tuned separately.

In the following, when using the smoothing by weights, we take $\alpha_\mathrm{s} = 1$ and r_s is tuned to yield the lowest RMSE score — alongside with the tuning of the localisation radius r and the regularisation jitter s or $\hat{h}$. Figure 13 shows the evolution of the RMSE as a function of the ensemble size N_e for the S(IR)^{x}S$_\mathrm{PM}$R and the S(IR)^{x}S$_\mathrm{PM}$R$_\mathrm{c}$ algorithms. The S(IR)^{x}S$_\mathrm{PM}$R algorithm yields systematically lower RMSEs than the standard S(IR)^{x}R. However, as the ensemble size N_e grows, the gain in RMSE score becomes very small. With $N_\mathrm{e} = 512$ particles, there is almost no difference between both scores. In this case, the optimal smoothing radius r_s is around 5 grid points, much smaller than the optimal localisation radius r around 15 grid points, such that the smoothing by weights step does not modify much the analysis ensemble. The S(IR)^{x}S$_\mathrm{PM}$R$_\mathrm{c}$ algorithm also yields lower RMSEs than the S(IR)^{x}R$_\mathrm{c}$ algorithm. Yet, in this case, the gain in RMSE is still significant for large ensembles and with $N_\mathrm{e} = 512$ particles, the RMSEs are even comparable to those of the EnKF.

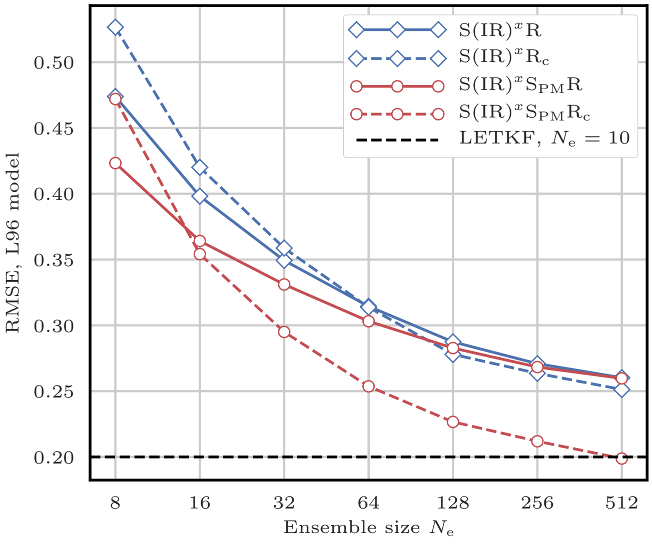


Figure 13. RMSE as a function of the ensemble size N_e for S(IR)^{x}R, the S(IR)^{x}R$_\mathrm{c}$, the S(IR)^{x}S$_\mathrm{PM}$R and the S(IR)^{x}S$_\mathrm{PM}$R algorithms.

From these results, we conclude that the smoothing by weights step is an efficient way of ~~reduce afterwards~~mitigating the ~~artificial~~unphysical discontinuities ~~introduced~~that were introduced when ~~concatenating~~assembling the locally updated particles, especially when combined with the coloured noise regularisation jitter method.

5.10 Using optimal transport in ensemble space

In this section, we evaluate the efficiency of using the optimal transport in ensemble space as a way to ~~reduce~~mitigate the ~~artificial~~unphysical discontinuities of the local resampling step by experimenting the $\mathrm{S(IT_R)}^x\mathrm{R}$ and the $\mathrm{S(IT_R)}^x\mathrm{R_c}$ algorithms. These algorithms only differ from the $\mathrm{S(IR)}^x\mathrm{R}$ and the $\mathrm{S(IR)}^x\mathrm{R_c}$ algorithms by the fact that they use optimal ensemble coupling for the local resampling as described by Algorithm 3.

For each block, the local linear transformation is computed by solving the minimisation problem Eq. (33), which can be seen as a particular case of the minimum–cost flow problem. Therefore, we ~~chose~~choose to compute its numerical solution with the graph library LEMON (Dezső et al., 2011). As described in Sect.4.4.3, this method is characterised by an additional tuning parameter: the distance radius r_d. We have investigated the influence of the parameters N_b and r_d by performing extensive tests of the $\mathrm{S(IT_R)}^x\mathrm{R}$ and the $\mathrm{S(IT_R)}^x\mathrm{R_c}$ algorithms with N_e ranging from 8 to 128 particles (not shown here) and draw the following conclusions.

Optimal values for the distance radius r_d ~~were found to be~~are much smaller than the localisation radius, most of the time even smaller than 2 grid points. Using $r_\mathrm{d} = 1$ grid point yields RMSEs that are only very slightly suboptimal. In this case, the local coefficients $c_b^{i,j}$ computed by Eq. (35) are equal to those defined by Eq. (36). Moreover, in all situations ~~we found that~~ using $N_\mathrm{b} = 20$ blocks of size 2 grid points systematically yields higher RMSEs than using $N_\mathrm{b} = 40$ blocks of size 1 grid point.

In the following, when using the optimal ensemble coupling algorithm, we take $r_\mathrm{d} = 1$ grid point and $N_\mathrm{b} = 40$ blocks of size 1 grid point. Figure 14 shows the evolution of the RMSE as a function of the ensemble size N_e for the $\mathrm{S(IT_R)}^x\mathrm{R}$ and the $\mathrm{S(IT_R)}^x\mathrm{R_c}$ algorithms. Using optimal ensemble coupling for the local resampling step always yields significantly lower RMSEs than using the SU sampling algorithm. Yet in this case, using the coloured noise regularisation jitter method does not improve the RMSEs for very large ensembles.

We have also performed extensive tests with N_e ranging from 8 to 128 particles on the $\mathrm{S(IT_R)}^x\mathrm{S_{PM}R}$ and the $\mathrm{S(IT_R)}^x\mathrm{S_{PM}R_c}$ algorithms in which the optimal ensemble coupling resampling method is combined with the smoothing by weights method (not shown here). Our implementations of these algorithms are numerically more costly. For small ensembles ($N_\mathrm{e} \leq 32$ particles), ~~we have obtained RMSEs barely smaller than those obtained with the $\mathrm{S(IT_R)}^x\mathrm{R}$ and the $\mathrm{S(IT_R)}^x\mathrm{R_c}$ algorithms.~~ the RMSEs of the $\mathrm{S(IT_R)}^x\mathrm{S_{PM}R}$ and the $\mathrm{S(IT_R)}^x\mathrm{S_{PM}R_c}$ algorithms are barely smaller than those of the $\mathrm{S(IT_R)}^x\mathrm{R}$ and the $\mathrm{S(IT_R)}^x\mathrm{R_c}$ algorithms. With larger ensembles, we could not find a configuration where using the smoothing by weights yields better RMSEs.

The facts that neither the use of larger blocks, nor the smoothing by weights does significantly improve the RMSE score when using optimal ensemble coupling indicate that this local resampling method is indeed an efficient way of ~~reducing~~mitigating the ~~artificial~~unphysical discontinuities inherent to ~~the concatenation of~~assembling the locally updated particles.

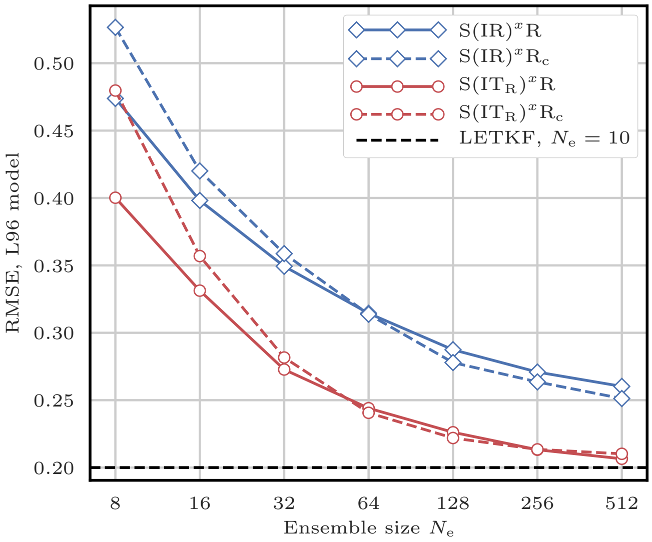


Figure 14. RMSE as a function of the ensemble size N_e for the $\mathrm{S(IR)}^x\mathrm{R}$, the $\mathrm{S(IR)}^x\mathrm{R_c}$, the $\mathrm{S(IT_R)}^x\mathrm{R}$ and the $\mathrm{S(IT_R)}^x\mathrm{R_c}$ algorithms.

5.11 Using continuous optimal transport

In this section, we test the efficiency of using the optimal transport in state space as a way to ~~reduce~~mitigate the ~~artificial~~unphysical discontinuities of the local resampling step by experimenting the $\mathrm{S(IT_s)}^x\mathrm{R}$ and the $\mathrm{S(IT_s)}^x\mathrm{R_c}$ algorithms. These algorithms only differ from the $\mathrm{S(IR)}^x\mathrm{R}$ and the $\mathrm{S(IR)}^x\mathrm{R_c}$ algorithms by the fact that they use anamorphosis for the local resampling as described by Algorithm 4.

As mentioned in Sect. 4.4.4, the local resampling algorithm based on anamorphosis uses blocks of size 1 grid point. Hence, when using the $\mathrm{S(IT_s)}^x\mathrm{R}$ and the $\mathrm{S(IT_s)}^x\mathrm{R_c}$ algorithms, we take $N_\mathrm{b} = 40$ blocks of size 1 grid point. The definition of the state transformation map T is based on the prior and corrected densities given by Eqs. (40) and (41) using the Student's t-distribution with two degrees of freedom for the regularisation kernel K. It is characterised by an additional tuning parameter: h, hereafter called *regularisation bandwidth* — different from the regularisation jitter $\hat{h}$. We have investigated the influence of the regularisation bandwidth h by performing extensive tests of the $\mathrm{S(IT_s)}^x\mathrm{R}$ and the $\mathrm{S(IT_s)}^x\mathrm{R_c}$ algorithms with N_e ranging from 8 to 128 particles (not shown here). For small ensembles ($N_\mathrm{e} \leq 16$ particles), optimal values for h ~~were found~~lie between 2 and 3, the RMSE score obtained with $h = 1$ being very slightly suboptimal. For larger ensembles, we did not find any significant difference between $h = 1$ and larger values.

In the following, when using the anamorphosis resampling algorithm, we take the standard value $h = 1$. Figure 15 shows the evolution of the RMSE as a function of the ensemble size N_e for the $\mathrm{S}(\mathrm{IT}_\mathrm{s})^x\mathrm{R}$ and the $\mathrm{S}(\mathrm{IT}_\mathrm{s})^x\mathrm{R}_\mathrm{c}$ algorithms. These algorithms yield RMSEs even lower than the algorithms using optimal ensemble coupling. However in this case, using the coloured noise regularisation jitter method always yields significantly higher RMSEs than using the white noise regularisation method. It is probably a consequence of the fact that some coloured regularisation is already introduced in the nonlinear transformation process through the kernel representation of the densities with Eqs. (40) and (41). It may also be a consequence of the fact that the algorithms using anamorphosis for the local resampling step cannot be written as a local LET algorithm, contrary to the algorithms using the SU sampling or the optimal ensemble coupling algorithms.

We have also performed extensive tests with N_e ranging from 8 to 128 particles on the $\mathrm{S}(\mathrm{IT}_\mathrm{s})^x\mathrm{S}_\mathrm{PM}\mathrm{R}$ algorithm, in which the anamorphosis resampling method is combined with the smoothing by weights method (not shown here). As for the $\mathrm{S}(\mathrm{IT}_\mathrm{R})^x\mathrm{S}_\mathrm{PM}\mathrm{R}$ and the $\mathrm{S}(\mathrm{IT}_\mathrm{R})^x\mathrm{S}_\mathrm{PM}\mathrm{R}_\mathrm{c}$ algorithms, our implementation is significantly numerically more costly and ~~we found that~~ adding the smoothing by weights step only yields minor RMSE improvements.

These latter remarks, alongside with significantly lower RMSE for the $\mathrm{S}(\mathrm{IT}_\mathrm{s})^x\mathrm{R}$ algorithm than for the $\mathrm{S}(\mathrm{IR})^x\mathrm{R}$ indicate that the local resampling method based on anamorphosis is, as well as the method based on optimal ensemble coupling, an efficient way of ~~reducing~~mitigating the ~~artificial~~unphysical discontinuities inherent to ~~the concatenation of~~assembling the locally updated particles.

5.12 Summary

~~As a summary~~To summarise, Fig. 16 shows the evolution of the RMSE as a function of the ensemble size N_e for the main LPF^xs tested in this section. For small ensembles ($N_\mathrm{e} \leq 32$ particles), the algorithms using OT-based resampling methods clearly yield lower RMSEs than the other algorithms. For large ensemble ($N_\mathrm{e} \geq 128$ particles), combining the smoothing by weights with the coloured noise regularisation jitter methods yields equally good scores as the algorithms using OT. For $N_\mathrm{e} = 512$ particles (the largest ensemble size tested ~~in this study~~with the L96 model), the best RMSE scores obtained with LPF^xs become comparable to those of the EnKF.

In this standard, mildly nonlinear configuration where error distributions are close to Gaussian, the EnKF performs very well and the LPF^x algorithms tested in this section do not clearly yield lower RMSE scores than the ETKF and the LETKF. Our objective is not to design LPF algorithms that beat the EnKF in all situations, but rather to incrementally improve the PF. However, specific configurations in which the EnKF fails and the PF succeeds can easily be conceived by increasing nonlinearities. Such a configuration is studied in Appendix C.

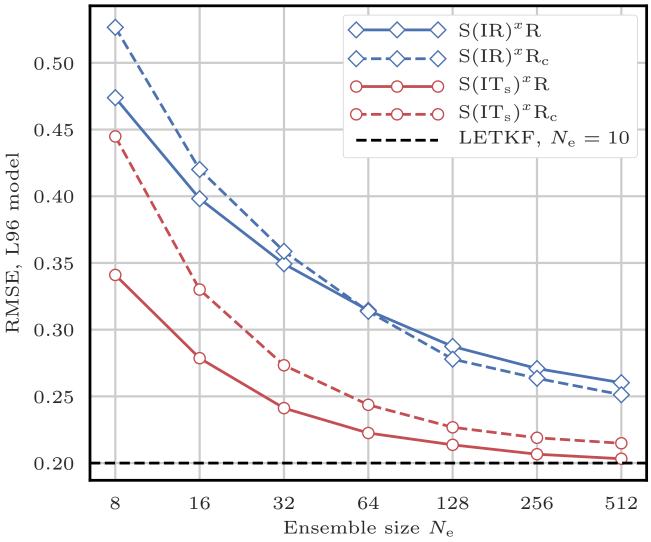


Figure 15. RMSE as a function of the ensemble size N_e for the $\mathrm{S(IR)}^x\mathrm{R}$, the $\mathrm{S(IR)}^x\mathrm{R}_\mathrm{c}$, the $\mathrm{S(IT_s)}^x\mathrm{R}$ and the $\mathrm{S(IT_s)}^x\mathrm{R}_\mathrm{c}$ algorithms.

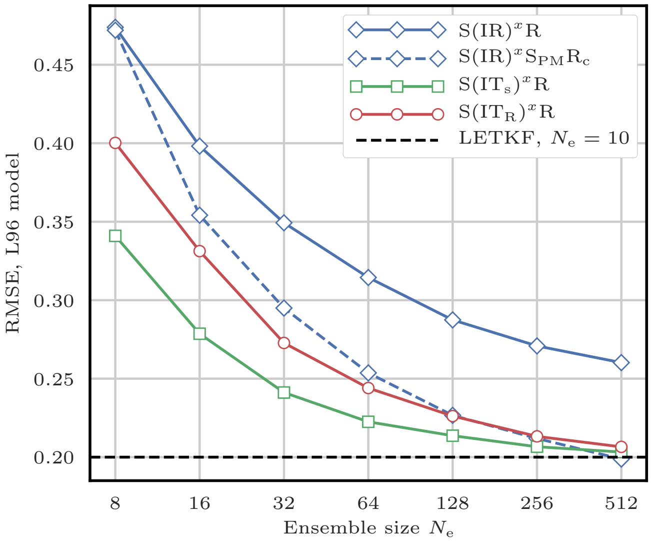


Figure 16. RMSE as a function of the ensemble size N_e for the main LPF^xs tested in this section.

As a complement to this RMSE test series, rank histograms for several LPFs are computed, reported and discussed in Appendix D.

6 Numerical illustration of the LPF^x algorithms with a barotropic vorticity model

6.1 Model specifications

In this section, we illustrate the performance of LPF^{x}s with twin simulations of the barotropic vorticity (BV) model in the coarse resolution (CR) configuration described in Appendix A4.1. Using this configuration yields a DA problem of size $N_x = 1024$ and $N_y = 256$. As mentioned in Appendix A4.1, the spatial resolution is enough to capture the ~~dynamic~~dynamics of a few vortices and the model integration is not too expensive, such that we can perform extensive tests with small to moderate ensemble sizes.

As with the L96 model, the distance between the truth and the analysis is measured with the average analysis RMSE. The runs are $9 \times 10^3 \Delta t$ long with an additional $10^3 \Delta t$ spin-up period, more than enough to ensure the convergence of the statistical indicators.

For the localisation, we use the underlying physical space with the Euclidean distance. The geometry of the ~~local~~ blocks and domain are constructed as described by Fig. 2. Specifically, ~~local~~ blocks are rectangles and local domains are disks, with the difference that the doubly periodic boundary conditions are taken into account.

6.2 Scores for the EnKF and the PF

As a reference, we first compute the RMSEs of the EnKF with this model. Figure 17 shows the evolution of the RMSE as a function of the ensemble size N_e for the ~~(global) ETKF and the local ETKF (LETKF).~~ETKF and the LETKF. For each value of N_e, the inflation parameter and the localisation radius (only for the LETKF) are optimally tuned to yield the lowest RMSE.

The ETKF requires at least $N_e = 12$ ensemble members to avoid divergence. The best RMSEs are approximately 20 times smaller than the observation standard deviation (~~0.3~~$\sigma = 0.3$). Even with only $N_e = 8$ ensemble members, the LETKF yields RMSEs at least 10 times smaller than the observation standard deviation, showing that in this case localisation is working as expected. In this configuration, the observation sites are uniformly distributed over the spatial domain. This constrains the posterior pdfs to be close to Gaussian, which explains the success of the EnKF in this DA problem.

With $N_e \leq 1024$ particles, we could not find a ~~set~~combination of tuning parameters with which the bootstrap filter or the ETPF yield RMSEs significantly lower than 1. In the following figures related to this BV test series, we draw a baseline at $\sigma/20$, which is roughly the RMSE of the ETKF and the LETKF with $N_e = 12$ particles. Note that slightly lower RMSE scores can be achieved with larger ensembles.

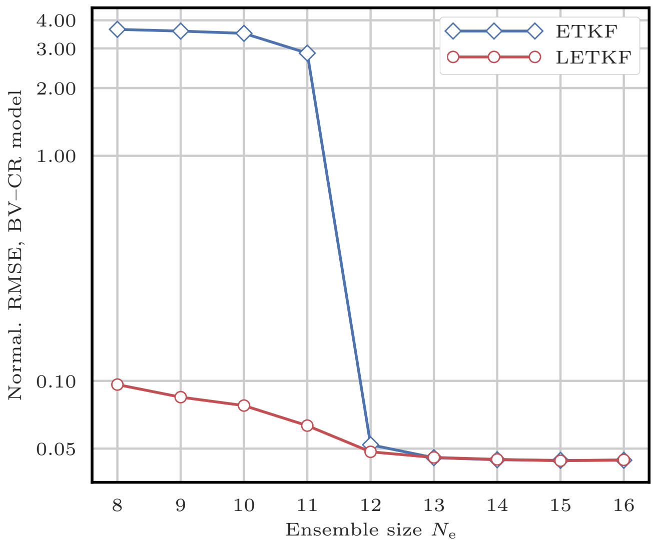


Figure 17. RMSE as a function of the ensemble size N_e for the ETKF and the LETKF. The scores are displayed in units of the observation standard deviation σ.

6.3 Scores for the LPF^x algorithms

In this section, we test the ~~LPF^{x}s~~LPF^x algorihtms with N_e ranging from 8 to 128 particles. ~~For clarity, the naming conventions of the algorithms are the same as in Sect. 5.~~The nomenclature for the algorithms is the same as in Sect. 5. In particular, all algorithms tested in this Section are in the list reported in Table 2.

For each ensemble size N_e we use similar parameter tuning methods as for the L96 model as follows:

- we take zero integration jitter: $q = 0$;
- the localisation radius r is systematically tuned to yield the lowest RMSE score;
- the regularisation jitter s (or $\hat{h}$ when using the coloured noise regularisation jitter method) is systematically tuned as well;
- for the algorithms using the SU sampling algorithm (i.e. the S(IR)$^x**$ variants) we test four values for the number of ~~local~~ blocks N_b, and we keep the best RMSE score:
  - 1024 blocks of shape 1×1 grid point;
  - 256 blocks of shape 2×2 grid points;
  - 64 blocks of shape 4×4 grid points;
  - 16 blocks of shape 8×8 grid points;
- for the algorithms using optimal ensemble coupling or anamorphosis (i.e. the S(IT$_*$)$^x*$ variants) we only test blocks of shape 1×1 grid point;

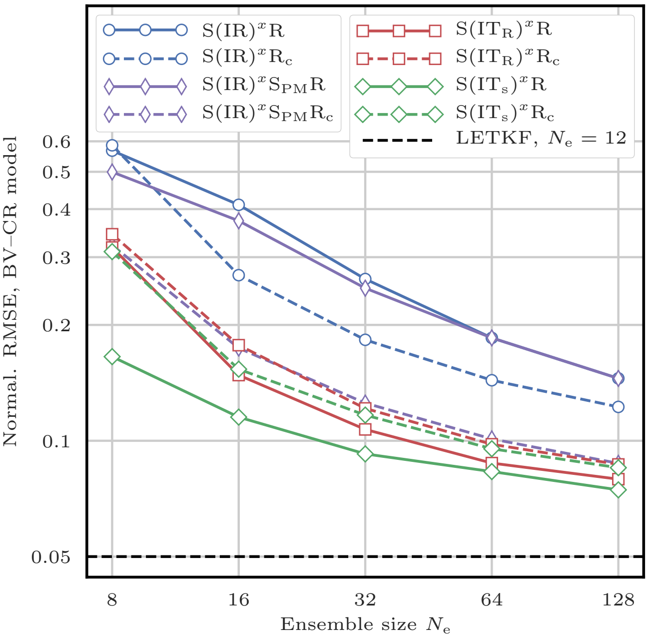


Figure 18. RMSE as a function of the ensemble size N_e for the ~~main~~ LPF^{x}s. The scores are displayed in units of the observation standard deviation σ.

- when using the smoothing by weights method, we take the smoothing strength $\alpha_\mathrm{s}=1$ and the smoothing radius r_s is optimally tuned to yield the lowest RMSE score;
- when using the optimal ensemble coupling for the local resampling step, the distance radius r_d is optimally tuned to yield the lowest RMSE score;
- when using the anamorphosis for the local resampling step, we take the regularisation bandwidth $h=1$.

Figure 18 shows the evolution of the RMSE as a function of the ensemble size N_e for the ~~main~~ LPF^{x}s. Most of the conclusions drawn with the L96 model remain true with the BV model. The best RMSE scores are obtained with algorithms using OT-based resampling methods. Combining the smoothing by weights with the coloured noise regularisation jitter methods yields almost equally good scores as the algorithms using OT. Yet, some differences can be pointed out.

With such a large model, we expected the coloured noise regularisation jitter method to be much more effective than the white noise method because the colourisation reduces ~~the~~ potential spatial discontinuities ~~of~~in the jitter. We observe indeed that the S(IR)^{x}R$_\mathrm{c}$ and the S(IR)^{x}S$_\mathrm{PM}$R$_\mathrm{c}$ algorithms yield significantly lower RMSEs than the S(IR)^{x}R and the S(IR)^{x}S$_\mathrm{PM}$R algorithms. Yet, the S(IT$_\mathrm{R}$)^{x}R$_\mathrm{c}$ and the S(IT$_\mathrm{s}$)^{x}R$_\mathrm{c}$ algorithms are clearly outperformed by both the S(IT$_\mathrm{R}$)^{x}R and the S(IT$_\mathrm{s}$)^{x}R algorithms in terms of RMSEs. This ~~hints at the fact~~suggests that there is room for improvement in the design of regularisation jitter methods for PF algorithms.

Due to relatively high computation times, we restricted our study to reasonable ensemble sizes, $N_\mathrm{e}\leq 128$ particles. In this configuration, the RMSE scores of LPF^{x}s are not yet comparable with those of the EnKF (shown in Fig. 17).

Finally, it should be noted that for the S(IT$_\mathrm{R}$)^{x}R and the S(IT$_\mathrm{R}$)^{x}R$_\mathrm{c}$ algorithms with $N_\mathrm{e}\geq 32$ particles, optimal values for the distance radius r_d ~~were found~~lie between 3 and 6 grid points (not shown here) contrary to the results obtained with the L96 model, for which $r_\mathrm{d}=1$ grid point could be considered optimal. More generally for all LPF^{x}s, the optimal values for the localisation radius r (not shown here) are significantly larger (in number of grid points) for the BV model than for the L96 model.

7 Sequential–observation localisation for particle filters

In the SBD localisation formalism, each block of grid points is updated using the local domain of observation sites that ~~should~~may influence these grid points. In the sequential–observation (SO) localisation formalism, we use a different approach. Observations ~~sites~~ are assimilated sequentially and assimilating ~~one~~the observation at a site should only update nearby grid points. LPF algorithms using the SO localisation formalism will be called LPF^y algorithms[3].

In this section, we set $q\in\{1\ldots N_y\}$ and we describe how ~~the site y_q should be assimilated.~~to assimilate the observation y_q. In Sect. 7.1, we introduce the state space partitioning. The resulting decompositions of the conditional density are discussed in Sect. 7.2. Finally, practical algorithms using these principles are derived in Sects. 7.3 and 7.4.

These algorithms are designed to assimilate one observation ~~site~~ at a time. Hence, a full assimilation cycle requires N_y sequential iterations of these algorithms, during which the ensemble is gradually updated: the updated ensemble after assimilating ~~site~~ y_q will be the prior ensemble to assimilate ~~site~~ y_{q+1}.

7.1 Partitioning the state space

Following Robert and Künsch (2017) the state space $\mathbb{R}^{N_x}$ is divided into three regions:

1. the first region U covers all grid points that directly influence y_q — if $\mathcal{H}$ is linear, it is all columns of $\mathcal{H}$ that have non-zero entries on row q;
2. the second region V gathers all grid points that are deemed correlated to those in U;
3. the third region W contains all remaining grid points.

The meaning of "correlated" is to be understood as a prior hypothesis, where we define a valid tapering matrix $\mathbf{C}$ that

---

[3] The y exponent emphasises the fact that we perform one analysis per observation.

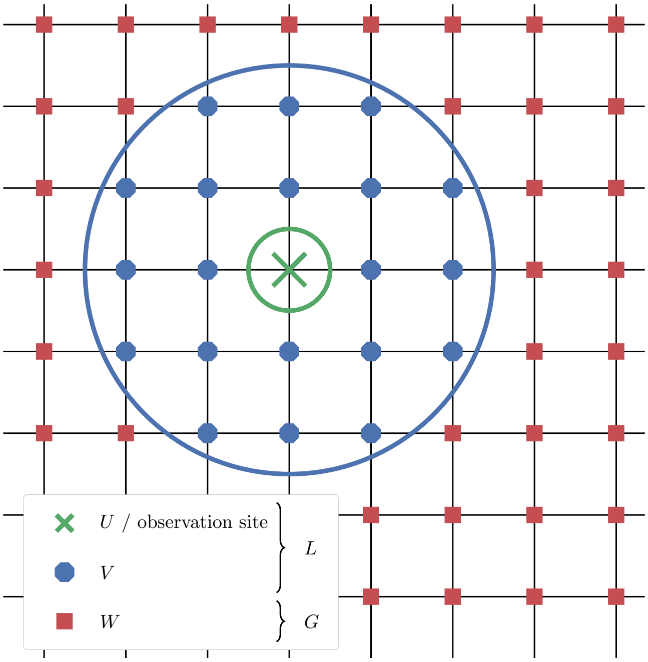


Figure 19. Example of the UVW partition for a two-dimensional space. The site of observation ~~site~~ y_q lies in the middle. The local ~~parts~~regions U and V are circumscribed by the thick green and blue circles and respectively contain 1 and 20 grid points. The global region W contains all remaining grid points. In the case of the LG partition, the local region L gathers all 21 grid points in U and V.

represents the decay of correlations. Non-zero elements of $\mathbf{C}$ should be located near the main diagonal and reflect the intensity of the correlation. A popular choice for $\mathbf{C}$ is the one obtained using the Gaspari–Cohn function:

$$[\mathbf{C}]_{m,n} = G\left(\frac{d_{m,n}}{r}\right), \tag{53}$$

where $d_{m,n}$ is the distance between the m-th and n-th grid points ~~m and n~~ and r is the localisation radius, a free parameter similar to the localisation radius defined in the SBD localisation formalism (see Sect. 4.2.2).

The UVW partition of the state space is ~~an extension~~a generalisation of the original LG partition introduced by Bengtsson et al. (2003) in which U and V are gathered into one region L, the *local* domain of y_q, and W is called G (for *global*). Figure 19 illustrates this UVW partition. We emphasise that both the LG and the UVW state partitions depends on the site of observation ~~site~~ y_q. They are fundamentally different from the ~~state~~(local state) block decomposition of Sect. 4.2.1 and therefore they shall simply be called "partition" to avoid confusion.

7.2 The conditional density

For any region A of the physical space, let $\boldsymbol{x}_A$ be the restriction of vector $\boldsymbol{x}$ to A, i.e. the state variables of $\boldsymbol{x}$ ~~that~~whose grid points are located within A.

7.2.1 With the LG partition

Without loss of generality, the conditional density is decomposed into:

$$p(\boldsymbol{x}|y_q) = p(\boldsymbol{x}_L, \boldsymbol{x}_G|y_q) = p(\boldsymbol{x}_L|\boldsymbol{x}_G, y_q)\,p(\boldsymbol{x}_G|y_q). \tag{54}$$

In a localisation context, it seems reasonable to assume that $\boldsymbol{x}_G$ and y_q are independent, that is:

$$p(\boldsymbol{x}_G|y_q) = p(\boldsymbol{x}_G), \tag{55}$$

and the conditional pdf of the L region can be written:

$$\begin{aligned} p(\boldsymbol{x}_L|\boldsymbol{x}_G, y_q) &= \frac{p(y_q|\boldsymbol{x}_G, \boldsymbol{x}_L)\,p(\boldsymbol{x}_G, \boldsymbol{x}_L)}{p(\boldsymbol{x}_G, y_q)}, \\ &= \frac{p(y_q|\boldsymbol{x}_L)\,p(\boldsymbol{x}_G, \boldsymbol{x}_L)}{p(\boldsymbol{x}_G, y_q)}. \end{aligned} \tag{56}$$

This yields an assimilation method for y_q described by Algorithm 5.

Algorithm 5 Single analysis step for a generic LPF^y algorithm using the LG partition

```
Require: Prior ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_\mathrm{e}$ and observation ~~site $y_q$~~$y_q$
1: Build the $LG$ partition as described in Sect. 7.1
2: for $i = 1$ to $N_\mathrm{e}$ do
3:     Do not update $\boldsymbol{x}^i_G$
4:     Update $\boldsymbol{x}^i_L$ conditionally to $y_q$ and $\boldsymbol{x}^i_G$ as stated by Eq. (56)
5: end for
6: return Updated ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_\mathrm{e}$
```

7.2.2 With the UVW partition

With the UVW partition, the conditional density is factored as

$$\begin{aligned} p(\boldsymbol{x}|y_q) &= p(\boldsymbol{x}_U, \boldsymbol{x}_V, \boldsymbol{x}_W|y_q), \\ &= \frac{p(\boldsymbol{x}_U, \boldsymbol{x}_V, \boldsymbol{x}_W, y_q)}{p(y_q)}, \\ &= \frac{p(y_q|\boldsymbol{x})\,p(\boldsymbol{x}_V|\boldsymbol{x}_U, \boldsymbol{x}_W)\,p(\boldsymbol{x}_U, \boldsymbol{x}_W)}{p(y_q)}, \\ &= \frac{p(y_q|\boldsymbol{x}_U)\,p(\boldsymbol{x}_V|\boldsymbol{x}_U, \boldsymbol{x}_W)\,p(\boldsymbol{x}_U, \boldsymbol{x}_W)}{p(y_q)}. \end{aligned} \tag{57}$$

If one assumes that the U and W regions are not only uncorrelated but also independent, then one can make the additional factorisation:

$$p(\boldsymbol{x}_U, \boldsymbol{x}_W) = p(\boldsymbol{x}_U)\,p(\boldsymbol{x}_W). \tag{58}$$

Finally, the conditional density is

$$p(\boldsymbol{x}|y_q) = p(\boldsymbol{x}_U|y_q)\, p(\boldsymbol{x}_V|\boldsymbol{x}_U,\boldsymbol{x}_W)\, p(\boldsymbol{x}_W)\,. \tag{59}$$

The assimilation method for y_q is now described by Algorithm 6.

Algorithm 6 Single analysis step for a generic LPF^y algorithm using the UVW partition

```
Require: Prior ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_e$ and observation ~~site $y_q$~~$y_q$
Build the $UVW$ partition as described in Sect. 7.1
for $i = 1$ to $N_e$ do
    Do not update $\boldsymbol{x}^i_W$
    Update $\boldsymbol{x}^i_U$ conditionally to $y_q$
    Update $\boldsymbol{x}^i_V$ conditionally to $\boldsymbol{x}^i_W$ and (the updated) $\boldsymbol{x}^i_U$ as stated by Eq. (57)
end for
return Updated ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_e$
```

7.2.3 The partition and the particle filter

~~The~~So far, the SO formalism ~~is~~looks elegant. The resulting assimilation schemes avoid the discontinuity issue inherent to the SBD formalism by using conditional updates of the ensemble.

However, this kind of update seems hopeless in a pure PF context. Indeed the factors $p(\boldsymbol{x}_G, \boldsymbol{x}_L)$ and $p(\boldsymbol{x}_V|\boldsymbol{x}_U,\boldsymbol{x}_W)$ in Eqs. (56) and (57) will be non-zero only if the updated particles are copies of the prior particles, which spoils the entire purpose of localising the assimilation. Hence potential solutions need to make approximations of the conditional density.

7.2.4 The multivariate rank histogram filter

Similar principles were used to design the multivariate rank histogram filter (MRHF) of Metref et al. (2014), with the main difference that the state space is entirely partitioned as follows. Assuming that y_q only depends on x_1, the conditional density can be written:

$$p(\boldsymbol{x}|y_q) = p(x_1|y_q)\, p(x_2|x_1) \dots p(x_{n+1}|x_n,\dots,x_1) \dots \tag{60}$$

In the MRHF analysis, the state variables are updated sequentially according to the conditional density $p(x_{n+1}|x_n,\dots,x_1)$. Zero factors in $p(x_{n+1}|x_n,\dots,x_1)$ are avoided by using a kernel representation *for the conditioning* on $x_n,\dots,x_1$, in a similar way as in Eqs. (40) and (41) with top hat functions for the regularisation kernel K. The resulting one-dimensional density along x_{n+1} is represented using histograms and the ensemble members are transformed using the same anamorphosis procedure as the one described in Sect. 4.4.4.

The MRHF could be used as a potential implementation of the SO localisation formalism. However, assimilating one observation ~~site~~ requires the computation of N_x different anamorphosis transformations.

7.2.5 Implementing the SO formalism

In the following sections, we introduce two different algorithms that implement the SO formalism (with the UVW partition) to assimilate one observation. Both algorithms are based on an "importance, resampling, propagation" scheme as follows. Global unnormalised importance weights are first computed as

$$w^i = p\left(y_q|\boldsymbol{x}^i\right). \tag{61}$$

Using these weights, we compute a resampling in the U region (essentially at the observation site). The update is then propagated to the V region using a dedicated propagation algorithm.

7.3 ~~Poterjoy's local particle filter~~A hybrid algorithm for the propagation

~~The LPF algorithm of Poterjoy (2016) (hereafter Poterjoy's LPF) is a hybrid scheme that mixes a (global) PF update and the prior in order to implement the UVW localisation formalism as described in this section.~~ The first algorithm that we introduce to implement the SO formalism using the "importance, resampling, propagation" scheme is the LPF of Poterjoy (2016) (hereafter Poterjoy's LPF). In this algorithm, the update is propagated using a hybrid scheme that mixes a (global) PF udpate with the prior ensemble.

7.3.1 Step 1: ~~particle filter resampling~~importance and resampling

~~The global unnormalised importance weights are given by [NB: Eq. (61) was here before] Using theses weights,~~Using the global unnormalised importance weights Eq. (61), we compute a resampling map ϕ, using for example the SU sampling algorithm.

7.3.2 Step 2: update and propagation

The resampling map ϕ is used to update the ensemble ~~at the observation site (i.e. in the U region)~~in the U region and the update is propagated to all grid points ~~n~~ as

$$x^i_n = \overline{x}_n + \omega^{\mathrm{a}}_n \left(x^{\phi(i)}_n - \overline{x}_n\right) + \omega^{\mathrm{f}}_n \left(x^i_n - \overline{x}_n\right), \tag{62}$$

where $\overline{x}_n$ is the ensemble mean at the n-th grid point~~ n~~, ω^{a} is the weight of the PF update and ω^{f} is the weight of the prior. If the resampling algorithm is adjustment-minimising, the number of updates that need to be propagated is minimal. Finally, the ω^* (either ω^{f} or ω^{a}) weights are chosen such that the updated ensemble yields correct ~~order-1 and order-2~~ statistics at the first and second orders.

At the observation site, $\omega^{\mathrm{a}}=1$ and $\omega^{\mathrm{f}}=0$, such that the update on the U region is the PF update and is Bayesian. Far from the observation site, $\omega^{\mathrm{a}}=0$ and $\omega^{\mathrm{f}}=1$, such that there is no update on the W region. Hence, the i-th updated particle is a composite particle between the i-th prior particle (in W) and the hypothetical i-th updated particle (in U) that would be obtained with a PF update. In between (in V) discontinuities are avoided by using a smooth transition for the ω^* weights, which involves the localisation radius r. A single analysis step according to Poterjoy's LPF is summarised by Algorithm 7.

The formulas for the ω^* weights are summarised in Appendix B. Their detailed derivation can be found in Poterjoy (2016), where ω^{a} and ω^{f} are called r_1 and r_2. Poterjoy (2016) included in his algorithm a weight inflation parameter that can be ignored to understand how the algorithm works. Moreover, the N_y sequential assimilations are followed by an optional KDDM step. As explained in Sect. 4.4.4, we found the KDDM step to be better suited for the local resampling step of LPF^x algorithms. Therefore, we ~~did not include~~have not included it in our presentation of Poterjoy's LPF.

Algorithm 7 Single analysis step for Poterjoy's LPF algorithm

```
Require: Prior ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_\mathrm{e}$ and observation ~~site $\boldsymbol{y}_q$~~$y_q$
Compute the analysis weights using Eq. (61)
Compute the resampling map $\phi$
for $n = 1$ to $N_x$ do
    Compute the weights $\omega_n^\mathrm{f}$ and $\omega_n^\mathrm{a}$
    for $i = 1$ to $N_\mathrm{e}$ do
        Update $x_n^i$ using Eq. (62)
    end for
end for
return Updated ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_\mathrm{e}$
```

7.4 ~~An algorithm inspired from the EnKPF~~A second order algorithm for the propagation

~~The ensemble Kalman particle filter (EnKPF) is a Gaussian mixture hybrid ensemble filter designed by Robert and Künsch (2017), in which the update is propagated from the observation site to all grid points using order-2 moments. The same ideas can be used to yield a LPF algorithm, hereafter called LEF$_{\mathrm{RK}}$, implementing the UVW localisation formalism as follows. The terminology "LEF" for local ensemble filter is favored over "LPF" to emphasise the order-2 truncation inherent to this algorithm.~~ The second algorithm that we introduce to implement the SO formalism using the "importance, resampling, propagation" scheme is based on the ensemble Kalman particle filter (EnKPF), a Gaussian mixture hybrid ensemble filter designed by Robert and Künsch (2017). In this algorithm, the updated is propagated using second order moments.

7.4.1 Preliminary: the covariance matrix

Since the update is propagated using ~~order-2~~second order moments, one first ~~need~~needs to compute the covariance matrix of the prior ensemble:

$$\boldsymbol{\Sigma}^{\mathrm{f}} = \operatorname{cov}(\boldsymbol{x}). \tag{63}$$

In a localisation context, it seems reasonable to use a tapered representation of the covariance~~, in such a way that the covariance matrix $\boldsymbol{\Sigma}$ of use will be~~. Therefore, we use the covariance matrix $\boldsymbol{\Sigma}$ defined by

$$\boldsymbol{\Sigma} = \mathbf{C} \circ \boldsymbol{\Sigma}^{\mathrm{f}}, \tag{64}$$

where $\mathbf{C}$ is the valid tapering matrix mentioned in section 7.1 (defined using the localisation radius r) and $\circ$ means the Schur product for matrices.

7.4.2 Step 1: ~~update in the U region~~importance and resampling

~~Each ensemble member is weighted by the global unnormalised importance weights Eq. (61). Using these weights~~Using the global unnormalised importance weights Eq. (61), we resample the ensemble in the U region and compute the update $\Delta\boldsymbol{x}_U^i$. For this resampling step, any resampling algorithm can be used:

- an adjustment-minimising resampling algorithm can be used to minimise the number of updates $\Delta\boldsymbol{x}_U^i$ that need to be propagated;
- the resampling algorithms based on OT in ensemble space or in state space, as derived in Sects. 4.4.3 and 4.4.4 can be used; as ~~in~~for the LPF^x methods, we expect them to create strong correlations between the prior and the updated ensembles.

7.4.3 Step 2: ~~propagate the update to the V region~~propagation

For each particle~~ i~~ the update of V, $\Delta\boldsymbol{x}_V^i$, depends on the update on U, $\Delta\boldsymbol{x}_U^i$, through the linear regression:

$$\Delta\boldsymbol{x}_V^i = \boldsymbol{\Sigma}_{VU}\boldsymbol{\Sigma}_U^{-1}\Delta\boldsymbol{x}_U^i, \tag{65}$$

where $\boldsymbol{\Sigma}_{VU}$ and $\boldsymbol{\Sigma}_U$ are submatrices of $\boldsymbol{\Sigma}$. The full derivation of Eq. (65) is available in Robert and Künsch (2017). Note that $\boldsymbol{\Sigma}$ is a $N_x \times N_x$ matrix but only the submatrices $\boldsymbol{\Sigma}_{VU}$ and $\boldsymbol{\Sigma}_U$ need to be computed.

~~Finally, the LEF$_{\mathrm{RK}}$ algorithm is summarised by Algorithm 8~~A single analysis step according to this second order algorithm is summarised by Algorithm 8 in a generic context, with any resampling algorithm.

Algorithm 8 Single analysis step for a generic ~~LEF$_{\mathrm{RK}}$ algorithm~~LPF^y using the second order propagation algorithm

```
Require: Prior ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_\mathrm{e}$, observation ~~site $\boldsymbol{y}_q$~~$y_q$
Build the $UVW$ partition as described in Sect. 7.1
Compute the prior covariance submatrices $\boldsymbol{\Sigma}_{VU}$ and $\boldsymbol{\Sigma}_U$
Compute the analysis weights using Eq. (61)
Resample the ensemble on region $U$
Compute the associated updates $\Delta\boldsymbol{x}_U^i$, $i = 1 \dots N_\mathrm{e}$
for $i = 1$ to $N_\mathrm{e}$ do
    Compute the update $\Delta\boldsymbol{x}_V^i$ using Eq. (65)
    Apply the update $\Delta\boldsymbol{x}_V^i$ on region $V$
end for
return Updated ensemble $\boldsymbol{x}^i$, $i = 1 \dots N_\mathrm{e}$
```

7.5 ~~Pros and cons of the LPF^y algorithms~~Summary for the LPF^y algorithms

7.5.1 Highlights

~~Both algorithms derived in this section include some spatial smoothness in the construction of the updated particles. In Poterjoy's LPF, the smoothness comes from the definition of the ω^* weights, whereas in the LEF$_{\mathrm{RK}}$ algorithm, it is a consequence of the use of correlations to propagate the update. Thus, we expect the discontinuity issues to be less critical with both algorithms than with LPF^{x}s, which is why the partition was introduced in the first place.~~ In this section, we have introduced a generic SO localisation framework, which we have used to define the LPF^{y}s, our second category of LPF methods. We have presented two algorithms, both based on an "importance, resampling, propagation" scheme:

1. The first algorithm is the LPF of Poterjoy (2016). It uses a hybrid scheme between a (global) PF update and the prior ensemble to propagate the update from the observation site to all grid points.
2. The second algorithm was inspired by the EnKPF of Robert and Künsch (2017). It uses tapered second order moments to propagate the update.

Both algorithms derived in this section include some spatial smoothness in the construction of the updated particles. In Poterjoy's LPF, the smoothness comes from the definition of the ω^* weights. In the second order propagation scheme, the smoothness comes from the prior correlations. Therefore, we expect the unphysical discontinuities to be less critical with these algorithms than with the LPF^x algorithms, which is why the partition was introduced in the first place.

7.5.2 Numerical complexity

Let N_U and N_V be the maximum number of grid points in U and V respectively and let $N_{UV} = N_U + N_V$. The complexity of assimilating one observation using Poterjoy's LPF is:

- $\mathcal{O}\left(N_\mathrm{e}\right)$ to compute the analysis weights Eq. (61) and the resampling map ϕ;
- $\mathcal{O}\left(N_\mathrm{e} N_{UV}\right)$ to compute the ω^* weights and to propagate the update to the U and V regions.

The complexity of assimilating one observation using the second order propagation algorithm is the sum of the complexity of computing the update on the U region, on the V region and of applying these updates to the ensemble. The complexity of computing the update on the U region is:

- $\mathcal{O}\left(N_\mathrm{e} N_U\right)$ when using the adjustment-minimising SU sampling algorithm;
- $\mathcal{O}\left(N_\mathrm{e}^2 N_x^\ell\left(r_\mathrm{d}\right) + N_\mathrm{e}^3 + N_\mathrm{e}^2 N_U\right)$ when using the optimal ensemble coupling derived in Sect. 4.4.3 with a distance radius r_d;
- $\mathcal{O}\left(N_U N_\mathrm{e} N_p\right)$ when using the anamorphosis derived in Sect. 4.4.4 with a fixed one-dimensional resolution of N_p points.

Using Eq. (65), the complexity of computing the update on the V region is:

- $\mathcal{O}\left(N_U^3\right)$ to compute $\boldsymbol{\Sigma}_U^{-1}$;
- $\mathcal{O}\left(N_\mathrm{e} N_U^2 + N_\mathrm{e} N_V N_U\right)$ to apply $\boldsymbol{\Sigma}_{VU}\boldsymbol{\Sigma}_U^{-1}$ to all $\Delta\boldsymbol{x}_U^i$, $i = 1 \dots N_\mathrm{e}$.

Finally, the complexity of applying the update on the U and V region is $\mathcal{O}\left(N_\mathrm{e} N_{UV}\right)$.

~~However, the LPF^{y}s are by construction non parallel: observation sites are assimilated sequentially.~~With LPF^y algorithms, observations are assimilated sequentially, which means that these algorithms are to be applied N_y times per assimilation cycle. This also means that the LPF^y algorithms are by construction non parallel. This issue was discussed by Robert and Künsch (2017): some level of parallelisation could be introduced in the algorithms, but only between observation sites for which the~~ domains~~ U and V regions are disjoint. That is to say, one can assimilate in parallel ~~several observation sites~~the observation at several sites as long as their domain of influence (in which an update is needed) do not overlap. This would require a preliminary geometric step to determine in which order observation sites are to be assimilated. This step would need to be performed again whenever the localisation radius r is changed. Moreover, when r is large enough, all U and V ~~domains~~regions may overlap, and parallelisation is not possible.

7.5.3 Asymptotic limit

~~The analysis step of Poterjoy's LPF is equivalent to a PF analysis step in the asymptotic limit $r \to \infty$. This is not the case for the LEF$_{\mathrm{RK}}$ algorithm.~~ By definition of the ω^*

weights, the single analysis step for Poterjoy's LPF is equivalent to the analysis step of the (global) PF for a single observation in the limit $r \to \infty$. This is not the case for the algorithm based on the second order propagation scheme. Indeed, using ~~order-2~~second order moments to propagate the update introduces a bias in the analysis. On the other hand, ~~order-2~~second order methods are in general less sensitive to the curse of dimensionality. Therefore, we expect ~~the $\mathrm{LEF}_{\mathrm{RK}}$~~this algorithm to be able to handle larger values for the localisation radius r than the LPF^xs.

7.6 Gathering observation sites into blocks

The LPF^ys can be extended to the case where observation sites are compounded into small blocks as follows:

- the unnormalised importance weights Eq. (61) are modified such that they account for all ~~observation~~ sites inside the block;
- any distance that ~~needed~~needs to be computed ~~relatively~~relative to the site of observation ~~site~~ y_q (for example for the ω^* weights for Poterjoy's LPF) is now computed relatively to the block center;
- in the ~~$\mathrm{LEF}_{\mathrm{RK}}$~~ algorithm based on the second order propagation scheme, the UVW partition is modified: the U region ~~now covers~~has to cover all grid points that directly influence ~~all observation sites~~every site inside the block.

Gathering observation sites into blocks reduces the number of sequential assimilations from N_y to the number of ~~observation sites~~ blocks, hence reducing the ~~computation~~computational time per cycle. However, it introduces an additional bias in the analysis. Therefore, we do not use this method in the numerical examples of Sects. 8 and 9.

8 Numerical illustration of the LPF^y algorithms

8.1 Experimental setup

In this section, we illustrate the performance of the LPF^y algorithms using twin simulations ~~of~~with the L96 and the BV models. The model specifications for this test series are the same as for the LPF^x test series: the L96 model is used in the standard configuration described in Appendix A3 and the BV model is used in the CR configuration described in Appendix A4.1. In a manner consistent with Sects. 5 and 6, the LPF^y algorithms are named ~~$\mathrm{S(I{*}P_{*})}^y\mathrm{R}_{*}$~~$\mathrm{S(I}\alpha\mathrm{P}_{\beta})^y\gamma$ — sampling, importance, resampling, propagation, regularisation, the y exponent meaning that steps in parentheses are performed locally for each observation ~~site~~ — with the conventions detailed in Table 3. Table 4 lists all LPF^y algorithms tested in this section and reports their characteristics according to the convention of Table 3.

Table 3. Nomenclature conventions for the $\mathrm{S(I}\alpha\mathrm{P}_{\beta})^y\gamma$ algorithms. Capital letters refer to the main algorithmic ingredients: "I" for importance, "R" for resampling or regularisation, "T" for transport, "P" for propagation. Subscripts are used to distinguish the methods in two different ways. Lower case subscripts refer to explicit concepts used in the method: "s" for state space, "c" for color; while upper case subscripts refer to the work that inspired the method: "P" for Poterjoy (2016) and "RK" for Robert and Künsch (2017). For simplicity, some subscripts are omitted: "amsu" for adjustment-minimising stochastic universal and "w" for white.

α	Local resampling algorithm
R	adjustment-minimising SU sampling algorithm
T_{R}	optimal transport in ensemble space (Sect. 4.4.3)
T_{s}	optimal transport in state space (Sect. 4.4.4)
β	Propagation method
P	Poterjoy's LPF (Algorithm 7)
RK	~~$\mathrm{LEF}_{\mathrm{RK}}$~~Second order propagation (Algorithm 8)
γ	Regularisation method (Sects. 5.2 and 5.8)
R	white noise method
R_{c}	coloured noise method

8.1.1 Regularisation jitter

For the same reasons as with LPF^xs, jittering the LPF^ys is necessary ~~not to experience~~to avoid a fast collapse. As we eventually did for the LPF^xs, the model is not perturbed (no integration jitter) and regularisation noise is added at the end of each assimilation cycle, either using the white noise method described by Eq. (46) or using the coloured noise method described in Sect. 5.8. With this latter method, the local weights required for the computation of the covariance matrix of the regularisation noise are computed with Eq. (29).

8.1.2 The $\mathrm{S(IRP_P)}^y\mathrm{R}$ algorithm and its variant

With the regularisation method described in Sect. 8.1.1, the $\mathrm{S(IRP_P)}^y\mathrm{R}$ has 3 parameters:

- the ensemble size N_e;
- the localisation radius r used to compute the ω^* weights (step 4 of Algorithm 7) as defined by Eqs. (B1) to (B4);
- the standard deviation s of the regularisation jitter, hereafter simply called "regularisation jitter" to be consistent with the LPF^xs.

For each value of the ensemble size N_e, the localisation radius r and the regularisation jitter s are systematically tuned to yield the lowest RMSE score.

As mentioned in Sect. 7.3.2, the original algorithm designed by Poterjoy (2016) included another tuning param-

Table 4. List of all LPF^y algorithms tested in this article. For each algorithm, the main characteristics are reported with appropriate references.

Algorithm	Resampling algorithm (Sect. 4.4)	Subsection	Propagation algorithm (Sects. 7.3 and 7.4) Algorithm 7 [Poterjoy's LPF]	Algorithm 8 [Second order]	Regularisation method (Sects. 5.2 and 5.8) Eq. (46) [white]	Eq. (50) [colour]
S(IRP$_{\mathrm{P}}$)^{y}R	adjustment-minimising SU sampling	4.4.2	✓		✓	
S(IRP$_{\mathrm{P}}$)^{y}R$_{\mathrm{c}}$	adjustment-minimising SU sampling	4.4.2	✓			✓
S(IRP$_{\mathrm{RK}}$)^{y}R	adjustment-minimising SU sampling	4.4.2		✓	✓	
S(IRP$_{\mathrm{RK}}$)^{y}R$_{\mathrm{c}}$	adjustment-minimising SU sampling	4.4.2		✓		✓
S(IT$_{\mathrm{R}}$P$_{\mathrm{RK}}$)^{y}R	optimal ensemble coupling	4.4.3		✓	✓	
S(IT$_{\mathrm{R}}$P$_{\mathrm{RK}}$)^{y}R$_{\mathrm{c}}$	optimal ensemble coupling	4.4.3		✓		✓
S(IT$_{\mathrm{s}}$P$_{\mathrm{RK}}$)^{y}R	anamorphosis	4.4.4		✓	✓	
S(IT$_{\mathrm{s}}$P$_{\mathrm{RK}}$)^{y}R$_{\mathrm{c}}$	anamorphosis	4.4.4		✓		✓

eter, the ~~weights~~weight inflation, which serves the same purpose as the regularisation jitter. Based on extensive tests in the L96 model with 8 to 128 particles (not shown here), we have found that using ~~weights~~weight inflation instead of regularisation jitter always yields higher RMSEs. Therefore, we ~~did not include~~have not included weight inflation in the S(IRP$_{\mathrm{P}}$)^{y}R algorithm.

In the S(IRP$_{\mathrm{P}}$)^{y}R$_{\mathrm{c}}$ algorithm, the regularisation jitter parameter s is replaced by $\hat{h}$ according to the coloured noise regularisation jitter method. The parameter tuning method is unchanged.

8.1.3 The S(IRP$_{\mathrm{RK}}$)^{y}R algorithm and its variants

With the regularisation method described in Sect. 8.1.1, the S(IRP$_{\mathrm{RK}}$)^{y}R has 3 parameters:

- the ensemble size N_{e};
- the localisation radius r used to define the valid tapering matrix $\mathbf{C}$ required for the computation of the prior covariance submatrices (step 2 of Algorithm 8) as defined by Eq. (64);
- the regularisation jitter s.

For each value of the ensemble size N_{e}, the localisation radius r and the regularisation jitter s are systematically tuned to yield the lowest RMSE score.

When using optimal ensemble coupling for the local resampling (step 4 of Algorithm 8), the local minimisation coefficients are computed using Eq. (35). This gives an additional tuning parameter, the distance radius r_{d}, which is also systematically tuned to yield the lowest RMSE score. When using anamorphosis for the local resampling step, the cdfs of the state variables in the region U are computed in the same way as for LPF^x algorithms, with a regularisation bandwidth $h = 1$. Finally, when using the coloured noise regularisation jitter method, the parameter s is replaced by $\hat{h}$ and the tuning method stays the same.

8.2 RMSE scores for the L96 model

The evolution of the RMSE as a function of the ensemble size N_{e} for the ~~main~~ LPF^y algorithms with the L96 model is shown in Fig. 20. The RMSEs obtained with the S(IRP$_{\mathrm{P}}$)^{y}R algorithm are comparable to those obtained with the S(IR)^{x}R algorithm. When using the ~~LEF$_{\mathrm{RK}}$~~second order propagation method, the RMSEs are, as expected, significantly lower. ~~Thanks to the order-2 truncation, the algorithms are~~The algorithm is less sensitive to the curse of dimensionality than the LPF^x algorithms: optimal values of the localisation radius r are significantly larger and less regularisation jitter s is required. ~~As for~~Similarly to the LPF^{x}s, combining the ~~LEF$_{\mathrm{RK}}$~~second order propagation method with OT-based resampling ~~methods~~algorithms (optimal ensemble coupling or anamorphosis) yields important gains in RMSE scores as a consequence of the minimisation of the update in the region U that needs to be propagated to the region V. With a reasonable number of particles (e.g. 64 for the S(IT$_{\mathrm{s}}$P$_{\mathrm{RK}}$)^{y}R algorithm), the scores are ~~even~~significantly lower than those obtained with the reference EnKF implementation (the ETKF). Finally, we observe that using the coloured noise regularisation jitter method improves the RMSEs for large ensembles when the local resampling step is performed with the SU sampling algorithm, in a similar way as for the LPF^{x}s. However when the local resampling step is performed with optimal ensemble coupling or with anamorphosis, the coloured noise regularisation jitter method barely improves the RMSEs.

8.3 RMSE scores for the BV model

The evolution of the RMSE as a function of the ensemble size N_{e} for the ~~main~~ LPF^y algorithms with the BV model is shown in Fig. 21. Most of the conclusions drawn with the L96 model remain true with the BV model. ~~However~~However, in this case, as the ensemble size N_{e} ~~grows~~grows, the RMSE decreases significantly more slowly for the S(IRP$_{\mathrm{P}}$)^{y}R and the S(IRP$_{\mathrm{P}}$)^{y}R$_{\mathrm{c}}$ algorithms than for

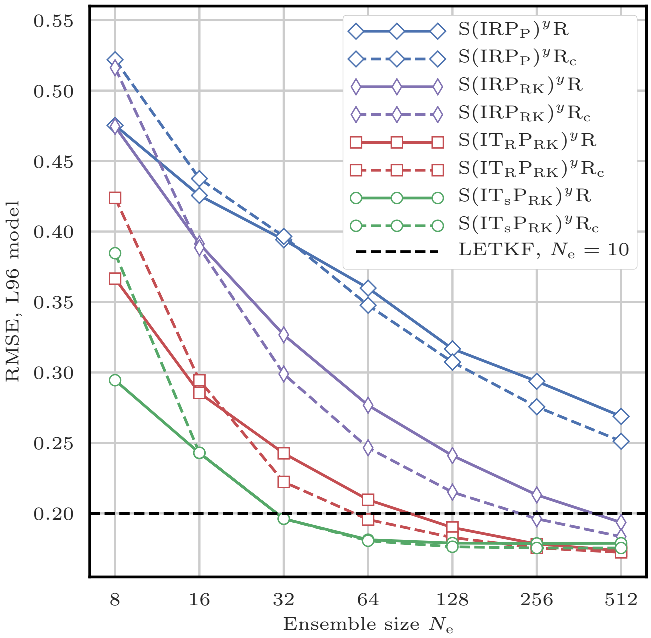


Figure 20. RMSE as a function of the ensemble size N_e for the ~~main~~ LPF^{y}s.

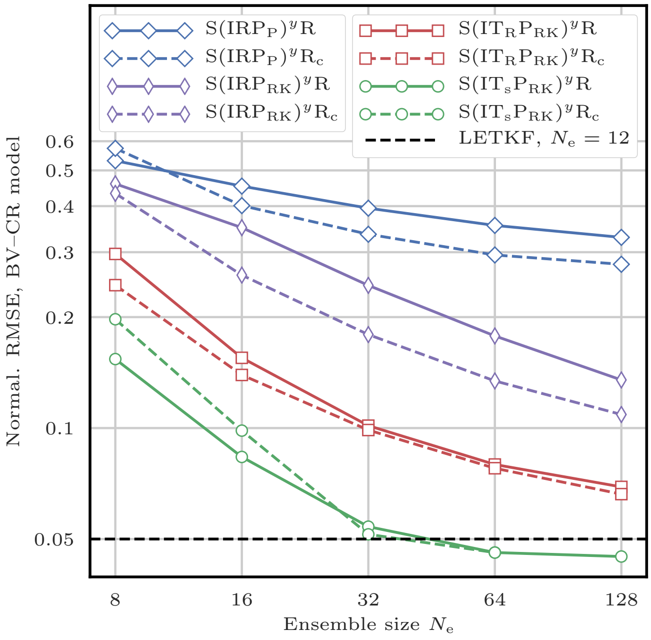


Figure 21. RMSE as a function of the ensemble size N_e for the ~~main~~ LPF^{y}s. The scores are displayed in units of the observation standard deviation σ.

the other algorithms. Finally, with an ensemble size $N_e \geq 64$ particles, the $S(IT_sP_{RK})^yR$ and the $S(IT_sP_{RK})^yR_c$ algorithms yield RMSEs almost equivalent to those of the reference LETKF implementation.

9 Numerical illustration with a high-dimensional barotropic vorticity model

9.1 Experimental setup

In this section, we illustrate the performance of a selection of LPF^{x}s and LPF^{y}s using twin simulations of the BV model in the high resolution (HR) configuration described in Appendix A4.2. Using this configuration yields a higher dimensional DA problem ($N_x = 65536$ and $N_y = 4096$) for which the analysis step is too costly to perform exhaustive tests. ~~Therefore~~Therefore, in this section, we take $N_e = 32$ ensemble members and we monitor the time evolution of the analysis RMSE during 501 assimilation steps.

As with the CR configuration, all geometrical considerations (~~local~~ blocks and domains, UVW partition...) use the Euclidean distance of the underlying physical space.

9.2 Algorithm specifications

For this test series, the selection of algorithms is listed in Table 5. ~~Approximate optimal values for the tuning parameters are found using several twin experiments with a few hundred assimilation cycles (not shown here).~~ Each algorithm uses the same initial ensemble obtained as follows:

$$\boldsymbol{x}_0^i = \boldsymbol{x}_0 + 0.5 \times \boldsymbol{u} + \boldsymbol{u}^i, \quad i = 1 \dots N_e, \tag{66}$$

with $\boldsymbol{u}$ and the $\boldsymbol{u}^i$ are random vectors whose coefficients are distributed according to a normal law. Such an ensemble is not very close to the truth (in terms of RMSE) and its spread is large enough to reflect the lack of initial information. The LPFs use zero integration jitter and $N_b = N_x$ blocks of size 1 grid point. Approximate optimal values for the localisation radius r and the regularisation jitter (s or $\hat{h}$ depending on the potential colourisation of the noise) are found using several twin experiments with a few hundred assimilation cycles (not shown here). The localisation radius r and the multiplicative inflation for the LETKF are found in a similar manner. When using OT in state space, we only test a few values for the regularisation bandwidth h. When using the smoothing by weights, we take the smoothing strength $\alpha_s = 1$ and the smoothing radius r_s is set to be equal to the localisation radius r.

9.3 RMSE time series

Figure 22 shows the evolution of the instantaneous analysis RMSE for the selected algorithms. Approximate optimal values for the tuning parameters, alongside with average analysis RMSE computed over the final 300 assimilation steps and wall-clock computation times are reported in Table 5. In terms of RMSE scores, the ranking of the methods is unchanged and most of the conclusions for this test series are the same as with the CR configuration.

Thanks to the ~~widespread~~uniformly distributed observation network, the posterior pdfs are close to Gaussian and therefore the LETKF algorithm can efficiently reconstruct a good approximation of the true state. As expected with this high-dimensional DA problem, the algorithms using ~~an order-2~~a second order truncation (the LETKF and the $\mathrm{S}(\mathrm{I}{*}\mathrm{P}_{\mathrm{RK}})^y\mathrm{R}$ algorithms) are more robust. Optimal values of the localisation radius are qualitatively large, which allows for a better reconstruction of the system dynamics.

For the $\mathrm{S}(\mathrm{IR})^x\mathrm{R}$ and the $\mathrm{S}(\mathrm{IRP}_{\mathrm{P}})^x\mathrm{R}$ algorithms, the optimal localisation radius r needs to be very small to counteract the curse of dimensionality. With such small values for r, the local domain of each grid point contains only 4 to 13 observation sites. This is empirically barely enough to reconstruct the true state with an RMSE score lower than the observation standard deviation σ. As ~~usual~~in the previous test series, using OT-based local resampling methods or the smoothing by weights step yields significantly lower RMSEs. The RMSEs of the $\mathrm{S}(\mathrm{IT}_{\mathrm{s}})^x\mathrm{R}$ ~~algorithm~~and the $\mathrm{S}(\mathrm{IR})^x\mathrm{S}_{\mathrm{PM}}\mathrm{R}_{\mathrm{c}}$ algorithms, though not as good as ~~those~~that of the LETKF algorithm, show that the true state is reconstructed with an acceptable accuracy. The RMSEs of the $\mathrm{S}(\mathrm{IT}_{\mathrm{s}}\mathrm{P}_{\mathrm{RK}})^y\mathrm{R}$ and the LETKF algorithms are almost comparable. Depending on the algorithm, the conditioning to the initial ensemble more or less quickly vanishes.

Without parallelisation, we observe that the N_x local analyses of the LPF^{x}s are almost always faster than both the N_x local analyses of the LETKF and the N_y sequential assimilations of the LPF^{y}s~~ (Poterjoy's LPF and the LEF$_{\mathrm{RK}}$ algorithm)~~. The ~~LEF$_{\mathrm{RK}}$~~second order propagation algorithm is slower because of the linear algebra involved in the ~~propagation ~~method. Poterjoy's propagation algorithm is slower because ~~of the computation of~~computing the ω^* weights is numerically expensive. The LETKF is slower because of the matrices inversion in ~~the ~~ensemble space. Finally the $\mathrm{S}(\mathrm{IR})^x\mathrm{S}_{\mathrm{PM}}\mathrm{R}_{\mathrm{c}}$ algorithm is even slower because, in this two-dimensional model, the smoothing by weights step is numerically very expensive.

The difference between the LPF^{x}s and the LPF^{y}s is even more visible on our 24-core platform. The LPF^{y}s are not parallel, that is why they are more than 70 times slower than the fastest LPF^{x}s.

10 Conclusions

The curse of dimensionality is a rather well-understood phenomenon in the statistical literature and it is the reason ~~for the failure of the application of PF methods~~why PF methods fail when applied to high-dimensional DA problems. We have recalled the main results related to weight degeneracy of PFs, and why the use of localisation can be used as a fix. Yet, implementing localisation in PF analysis raises two major issues: the gluing of locally updated particles and potential physical imbalance in the updated particles. Adequate solutions to these issues are not obvious, witness the few but ~~unsimilar~~dissimilar LPF algorithms developed in the geophysical literature. ~~In this article, we have reviewed the ideas related to localisation and particle filtering. Moreover, we have proposed a theoretical classification of LPFs into two categories.~~ In this article we have proposed a theoretical classification of LPF algorithms into two categories. For each category, we have presented the challenges of local particle filtering and we have reviewed the ideas that lead to practical implementation of LPFs. Some of them, already in the literature, have been detailed and sometimes generalised while others are new in this field and yield improvements in the design of LPF methods.

With the LPF^x methods, the analysis is localised by allowing the analysis weights to vary over the ~~state variables~~grid points. We have shown that this ~~leads to~~yields an analysis pdf from which only the marginals are known. The local resampling step is mandatory to reconstruct global particles, that are obtained ~~as the concatenation of~~by assembling the locally updated particles. The quality of the updated ensemble directly depends on the regularity of the local resampling. This is related to ~~the potential~~unphysical discontinuities in the ~~concatenated~~assembled particles. Therefore we have presented practical methods to improve ~~the design of this~~the local resampling step by reducing the unphysical discontinuities.

In the LPF^y methods, localisation is introduced more generally in the conditional density for one observation~~ site~~ by the means of a state partition. The goal of the partition is to build a framework for local particle filtering without the discontinuity issue inherent to LPF^{x}s. However, this framework is irreconcilable with algorithms based on pure "importance, resampling" methods. We have shown how two hybrid methods could yet be used as an implementation of this framework. Besides, we have emphasised the fact that with these methods, ~~observation sites~~observations are by construction assimilated sequentially, which is a great disadvantage when the number of ~~observation sites~~observations in the DA problem is high.

With localisation, a bias is introduced in the LPF analyses. We have shown that, depending on the localisation parametrisation, some methods can yield an analysis step equivalent to that of global PF methods which are known to be asymptotically Bayesian.

We have implemented and systematically tested the LPF algorithms with twin simulations of the L96 model and the BV model. A few observations could be made from these experiments. With these models, implementing localisation is simple and works as expected: the LPFs yield acceptable RMSE scores even with small ensembles, in regimes where global PF algorithms are degenerate. ~~Despite the fact that it was explicitly designed to avoid discontinuity issues, there is no clear advantage of using Poterjoy's LPF over the $\mathrm{S}(\mathrm{IR})^x\mathrm{R}$ algorithm. In particular, with the BV model, the scores of the $\mathrm{S}(\mathrm{IR})^x\mathrm{R}$ algorithm are slightly better than those of Poterjoy's LPF at a lower computational cost. As expected, using the LEF$_{\mathrm{RK}}$ propagation method yields the most efficient DA~~

Table 5. Characteristics of the algorithms tested with the BV model in the HR configuration (Fig. 22). The LPF^{x}s use zero integration jitter ($q = 0$) and $N_\mathrm{b} = N_x$ blocks of size 1 grid point. The LPF^{y}s also use zero integration jitter ($q = 0$). For the LETKF, the optimal multiplicative inflation is reported in the regularisation jitter column. For the S(IR)^{x}S$_\mathrm{PM}$R$_\mathrm{c}$ algorithm, the optimal regularisation jitter bandwidth $\hat{h}$ is reported in the regularisation jitter column as well. The average RMSE is computed over the final 300 assimilation steps and given in units of the observation standard deviation σ. The wall-clock computation time is the average time spent per analysis step. The simulations are performed on a *single* core of a double Intel Xeon E5-2680 platform (for a total of 24 cores). For comparison, the average time spent per forecast ($\Delta t = 0.5$) for the 32-member ensemble is 0.94 s. The * asterisk indicates that the local analyses can be carried out in parallel, allowing a theoretical gain in computation time of up to a factor 65536. For these algorithms, ~~the wall-clock computation time of~~ the average time spent per analysis step for the *parallelised* runs on this 24-core platform, as well as the acceleration factor, are reported in the last column.

Algorithm	Loc. radius r [in units of L]	Reg. jitter s	Other parameters	Average RMSE [in units of σ]	1-core wall-clock time [in s]	24-core wall-clock time [in s]
S(IRP$_\mathrm{P}$)^{y}R	0.03	0.70	–	0.90	122.18	–
S(IR)^{x}R	0.02	0.55	–	0.78	7.58*	0.54 (×14.04)
S(IRP$_\mathrm{RK}$)^{y}R	0.07	0.25	–	0.46	52.97	–
S(IR)^{x}S$_\mathrm{PM}$R$_\mathrm{c}$	0.05	1.0	$\alpha_\mathrm{s} = 1, r_\mathrm{s} = r$	0.38	226.20*	12.50 (×18.10)
S(IT$_\mathrm{s}$)^{x}R	0.08	0.11	$h = 3$	0.33	13.94*	0.86 (×16.21)
S(IT$_\mathrm{s}$P$_\mathrm{RK}$)^{y}R	0.20	0.01	$h = 1$	0.13	64.79	–
LETKF	0.35	1.04	–	0.10	103.90*	5.09 (×20.41)

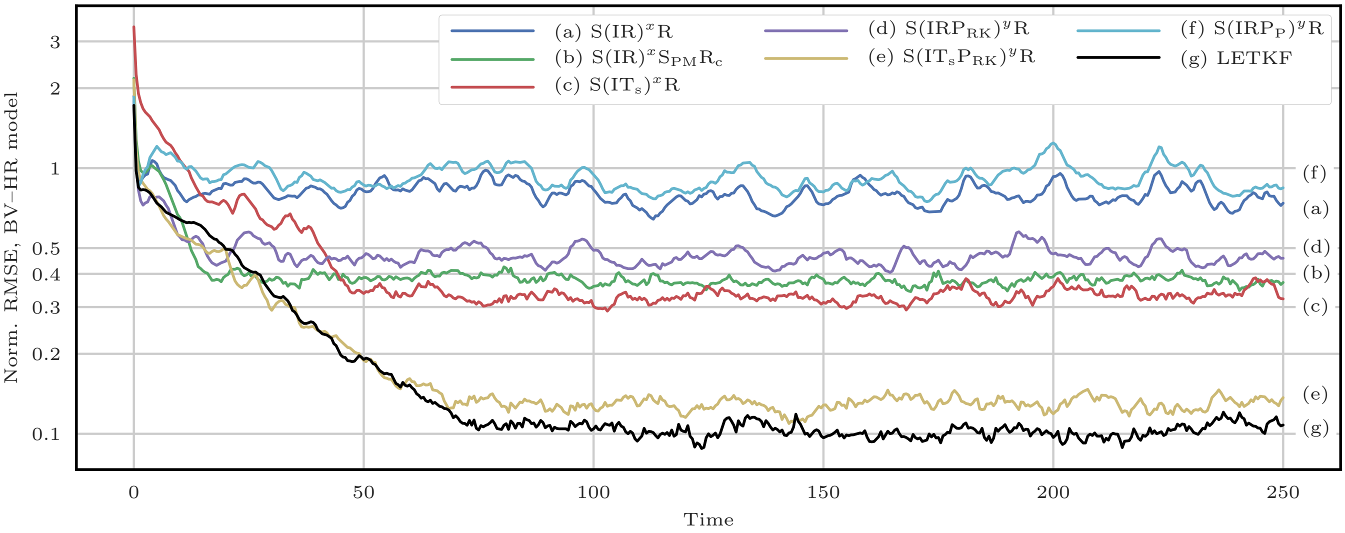


Figure 22. Instantaneous analysis RMSE for the selection of algorithms detailed in Table 5. The scores are displayed in units of the observation standard deviation σ.

~~method in terms of RMSE scores. This is a consequence of the fact that order-2 truncated methods are in general less sensitive to the curse of dimensionality.~~ In terms of RMSEs, there is no clear advantage of using Poterjoy's propagation method (designed to avoid unphysical discontinuities) over the (simpler) LPF^x algorithms, which have a lower computational cost. As expected, algorithms based on the second order propagation method are less sensitive to the curse of dimensionality and yields the lowest RMSE scores. We have shown that using OT-based local resampling methods always yields important gains in RMSE scores. For the LPF^{x}s, it is a consequence of ~~the minimisation of the~~mitigating the unphysical discontinuities introduced in the local resampling step. For the LPF^{y}s, it is a consequence of the minimisation of the update at the observation site that needs to be propagated to nearby grid points.

Finally, the successful application of the LPFs to DA problems with a perfect model is largely due to the use of regularisation jitter. Using regularisation jitter introduces an additional bias in the analysis alongside with an extra tuning parameter. For our numerical experiments, we have introduced two jittering methods: either using regularisation noise with fixed statistical properties (white noise) or by scaling the noise with the ensemble anomalies (coloured noise). We have discussed the relative performance of each method and concluded that there is room for improvement in the design of regularisation jitter methods for PFs. Ideally, the methods should be adaptative but this point is beyond the scope of this article. The conclusions could have been different when applying the LPFs to DA problems with stochastic models. However, the definition of a realistic model noise is non-trivial. Besides, the magnitude of such a model noise may be too small for the LPFs to perform well. That is why the use of regularisation jitter can be justified even with stochastic models.

Acknowledgements. The authors thank the Editor, Olivier Talagrand, and two reviewers, Stephen G. Penny and one anonymous reviewer, for their useful comments, suggestions and thorough reading of the manuscript. The authors are grateful to Patrick Raanes for enlightening debates. CEREA is a member of Institut Pierre–Simon Laplace (IPSL).

Appendix A: Numerical models

A1 The Gaussian linear model

The Gaussian linear model is the simplest model with size N_x whose prior distribution is

$$\boldsymbol{x}_0 \sim \mathcal{N}\left(\mathbf{0}, p^2\mathbf{I}\right), \tag{A1}$$

whose transition distribution is

$$\boldsymbol{x}_{k+1} - a\boldsymbol{x}_k = \boldsymbol{w}_k \sim \mathcal{N}\left(\mathbf{0}, q^2\mathbf{I}\right), \tag{A2}$$

and whose observation distribution is

$$\boldsymbol{y}_k - h\boldsymbol{x}_k = \boldsymbol{v}_k \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right), \tag{A3}$$

where $\mathcal{N}(\boldsymbol{v}, \boldsymbol{\Sigma})$ is the normal distribution with mean $\boldsymbol{v}$ and covariance matrix $\boldsymbol{\Sigma}$.

A2 Generic model with Gaussian additive noise

The Gaussian linear model can be generalised to include non-linearity in the model $\mathcal{M}$ and in the observation operator $\mathcal{H}$. In this case, the transition distribution is:

$$\boldsymbol{x}_{k+1} - \mathcal{M}(\boldsymbol{x}_k) = \boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \tag{A4}$$

and the observation distribution is:

$$\boldsymbol{y}_k - \mathcal{H}(\boldsymbol{x}_k) = \boldsymbol{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \tag{A5}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrices of the additive model and observation errors.

A3 The Lorenz 1996 model

The Lorenz 1996 model (Lorenz and Emanuel, 1998) is a low-order one-dimensional discrete chaotic model whose evolution is given by the following set of ODEs:

$$\frac{\mathrm{d}x_n}{\mathrm{d}t} = (x_{n+1} - x_{n-2})\,x_{n-1} - x_n + F, \quad n = 1 \ldots N_x, \tag{A6}$$

where the indices are to be understood with periodic boundary conditions: $x_{-1} = x_{N_x-1}$, $x_0 = x_{N_x}$, $x_1 = x_{N_x+1}$ and where the system size N_x can take arbitrary values. These ODEs are integrated using a fourth-order Runge–Kutta method with a time step of 0.05 time unit.

In the standard configuration, $N_x = 40$ and $F = 8$ which yields a chaotic dynamics with a doubling time around 0.42 time unit. The observations are given by

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \boldsymbol{v}_k, \quad \boldsymbol{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{A7}$$

and the time interval between consecutive observations is $\Delta t = 0.05$ time unit, which represents 6 h of real time and corresponds to a model autocorrelation around 0.967.

A4 The barotropic vorticity model

The barotropic vorticity model describes the evolution of the vorticity field of a two-dimensional incompressible homogeneous fluid in the $x_1 - x_2$ ~~plan~~plane. The time evolution of the unknown vorticity field q is governed by the scalar equation

$$\frac{\partial q}{\partial t} + \mathrm{J}(\psi, q) = -\xi q + \nu \Delta q + F, \tag{A8}$$

and q is related to the stream function ψ through

$$\Delta \psi = q. \tag{A9}$$

In these equations, $\mathrm{J}(\psi, q)$ is the advection of the vorticity by the stream, defined as

$$\mathrm{J}(\psi, q) = \frac{\partial \psi}{\partial x_1}\frac{\partial q}{\partial x_2} - \frac{\partial \psi}{\partial x_2}\frac{\partial q}{\partial x_1}, \tag{A10}$$

$\xi \in \mathbb{R}^+$ is the friction coefficient, $\nu \in \mathbb{R}^+$ is the diffusion coefficient and F is the forcing term, that may depend on x_1, x_2 and t. The system is characterised by homogeneous two-dimensional turbulence. The friction extracts energy at large scale, the diffusion dissipates vorticity at small scale and the forcing injects energy in the system. The number of degrees of freedom in this model can be roughly considered to be proportional to the number of ~~vertices~~vortices (Snyder, 2012, personal communication).

The equations are solved with P^2 grid points regularly distributed over the simulation domain $[0, L]^2$ with doubly periodic boundary conditions. Our time integration method is based on a semi-Lagrangian solver with a constant time step δt as follows:

1. At time t, solve Eq. (A9) for ψ.
2. At time t, compute the advection velocity with order-2 centered finite differences of the field ψ.
3. The advection of q during t and $t+\delta t$ is computed by applying a semi-Lagrangian method to the left-hand side of Eq. (A8). The solver cannot be more precise than

order-1 in time, since the value of ψ is not updated during this step. Therefore, our semi-Lagrangian solver uses the order-1 forward Euler time integration method. The interpolation method used is the cubic convolution interpolation algorithm, which yields an order-3 precision with respect to the spatial discretisation. In this step, the right-hand side of Eq. (A8) is ignored.

4. Integrate q from t to $t+\delta t$ by solving Eq. (A8) with an implicit order-1 time integration scheme, in which the advection term is the one computed in the previous step.

For the numerical experiments of this ~~study~~article, the spatial discretisation is fine enough such that the spatial interpolation error in the semi-Lagrangian step is negligible when compared to the time integration error. As a consequence, the overall integration method is order-1 in time. For the DA experiments with this model, we define and use two configurations.

A4.1 Coarse resolution configuration

The coarse resolution configuration is based on the following set of physical parameters:

$$L = 1, \tag{A11}$$

$$\xi = 10^{-2}, \tag{A12}$$

$$\nu = 5 \times 10^{-5}, \tag{A13}$$

the deterministic forcing is given by

$$F\left(x_1, x_2\right) = 0.25 \sin\left(4\pi x_1\right) \sin\left(4\pi x_2\right), \tag{A14}$$

and the space-time discretisation is

$$\delta t = 0.1, \tag{A15}$$

$$\delta x = \frac{L}{P} = \frac{1}{32}, \tag{A16}$$

which yields $N_x = (\delta x/L)^2 = 1024$. The ~~space~~spatial discretisation is enough to allow a reasonable description of a few (typically five to ten) vortices inside the domain. The ~~time~~temporal discretisation is empirically enough to ensure the stability of the integration method and allows a fast computation of the trajectory. The physical parameters are chosen to yield a proper time evolution of the vorticity q.

The initial true vorticity field for the DA twin experiments is the vorticity obtained after a run of 100 time units starting from a random, spatially correlated field. The system is partially observed on a regular square mesh with one observation site every 2 grid points in each direction, i.e. $N_y = 256$ observation sites for $N_x = 1024$ ~~state variables~~grid points. At every cycle k, the observation at site $(q_1, q_2) \in \{1 \dots P/2\}^2$ is given by

$$y_{q_1,q_2} = x_{2q_1-1,2q_2-1} + v_{q_1,q_2}, \tag{A17}$$

$$v_{q_1,q_2} \sim \mathcal{N}\left(0, \sigma^2\right), \tag{A18}$$

with $\sigma = 0.3$, about one tenth of the typical vorticity variability. The time interval between consecutive observations is $\Delta t = 0.5$ time unit, which was chosen to match approximately the model autocorrelation of 0.967 of the L96 model in the standard configuration.

We have checked that the vorticity flow remains stationary over the total simulation time of our DA twin experiments chosen to be $10^4 \Delta t$. Due to the forcing F, the flow remains uniformally and stationarily turbulent during the whole simulation. Compared to other experiments with the barotropic vorticity model (e.g. van Leeuwen and Ades, 2013; Ades and van Leeuwen, 2015; Browne, 2016), Δt is smaller and σ is ~~bigger~~larger, but the number of ~~vertices~~vortices is approximately the same with much fewer details.

A4.2 High resolution configuration

For the high resolution configuration, the physical parameters are

$$L = 1, \tag{A19}$$

$$\xi = 5 \times 10^{-5}, \tag{A20}$$

$$\nu = 10^{-6}; \tag{A21}$$

the deterministic forcing is given by

$$F\left(x_1, x_2\right) = 0.75 \sin\left(12\pi x_1\right) \sin\left(12\pi x_2\right), \tag{A22}$$

and the space-time discretisation is

$$\delta t = 0.1, \tag{A23}$$

$$\delta x = \frac{L}{P} = \frac{1}{256}, \tag{A24}$$

which yields $N_x = (\delta x/L)^2 = 65536$. Compared to the coarse resolution configuration, this set of parameters yields a vorticity field with more ~~vertices~~vortices (typically several dozens). The associated DA problem has therefore many more apparent or effective degrees of freedom. The initial true vorticity field for the DA twin experiments is the vorticity obtained after a run of 100 time units starting from a random, spatially correlated field. The system is partially observed on a regular square mesh with one observation site every 4 grid points in each direction, i.e. $N_y = 4096$ observation sites for $N_x = 65536$ ~~state variables~~grid points. At every cycle k, the observation at site $(q_1, q_2) \in \{1 \dots P/4\}^2$ is given by

$$y_{q_1,q_2} = x_{4q_1-1,4q_2-1} + v_{q_1,q_2}, \tag{A25}$$

$$v_{q_1,q_2} \sim \mathcal{N}\left(0, \sigma^2\right), \tag{A26}$$

and we keep the values $\Delta t = 0.5$ time units and $\sigma = 0.3$ from the coarse resolution configuration. We have checked that the vorticity flow remains stationary over the total simulation time of our DA twin experiments chosen to be $500\Delta t$. Due to the forcing F, the flow remains uniformally and stationarily turbulent during the whole simulation.

Appendix B: Update formulae of Poterjoy's LPF

Following Poterjoy (2016), we derived the following formulas for the ω^* weights required in the propagation step of Poterjoy's LPF described in Sect. 7.3.2:

$$W=\sum_{i=1}^{N_\mathrm{e}} w^i=\sum_{i=1}^{N_\mathrm{e}} p\left(y_q|\boldsymbol{x}^i\right), \tag{B1}$$

$$c_n=\frac{\alpha N_\mathrm{e}\left(1-G\left(\frac{d_{q,n}}{r}\right)\right)}{WG\left(\frac{d_{q,n}}{r}\right)}, \tag{B2}$$

$$\omega_n^\mathrm{a}=\sqrt{\frac{\sigma_n^2}{\frac{1}{N_\mathrm{e}-1}\sum\limits_{i=1}^{N_\mathrm{e}}\left\{x_n^{\phi(i)}-\overline{x}_n+c_n\left(x_n^i-\overline{x}_n\right)\right\}^2}}, \tag{B3}$$

$$\omega_n^\mathrm{f}=c_n\omega_n^\mathrm{a}, \tag{B4}$$

where W and c_n are ancillary variables, α is the constant used for the computation of the local weights (see Eq. (28)), G is the tapering function, $d_{q,n}$ is the distance between the q-th observation site ~~q~~ and the n-th grid point ~~n~~, r is the localisation radius, $\overline{x}_n$ is the mean and σ_n the standard deviation of the weighted ensemble $\left\{\left(x_n^i,w^i\right),i=1\dots N_\mathrm{e}\right\}$. The particles are then updated using Eq. (62).

In Poterjoy (2016), the pdfs are implicitly normalised, such that the constant α is 1. Therefore, our update Eqs. (B1) to (B4) are equivalent to the update Eqs. (A10), (A11), (A5) and (A3) derived by Poterjoy (2016). Note that there is a typing mistake ~~such that~~which renders one update equation in Algorithm 1 of Poterjoy (2016) ~~is~~ incorrect (last equation on page 66).

Appendix C: Nonlinear test series with the L96 model

As a complement to the mildly nonlinear test series of Sects. 5, 6, 8 and 9, we provide here a strongly nonlinear test series. We consider the L96 model in the standard configuration described in Appendix A3 with the only difference that the $N_y=N_x$ observations at each assimilation cycle are now given by

$$\forall n\in\{1\dots N_x\},\quad y_n=\ln|x_n|+v_n,\quad v_n\sim\mathcal{N}(0,1). \tag{C1}$$

This strongly nonlinear configuration has been used e.g. by Poterjoy (2016).

Similarly to the mildly nonlinear test series, the distance between the truth and the analysis is measured with the average analysis RMSE. The runs are $9\times10^3\Delta t$ long with an additional $10^3\Delta t$ spin-up period. Optimal values for the tuning parameters of each algorithms are found using the same method as for the mildly nonlinear test series. Figure C1 shows the evolution of the RMSE as a function of the ensemble size N_e for the LETKF and for the main LPF^x and LPF^y algorithms.

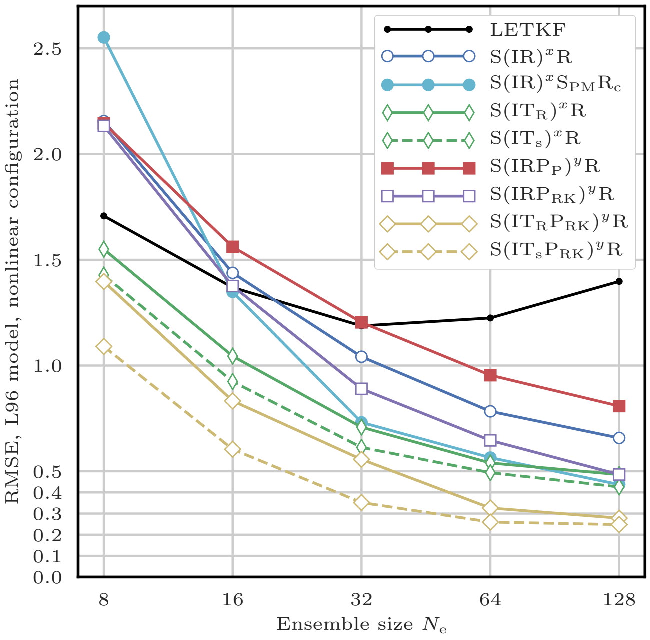


Figure C1. RMSE as a function of the ensemble size N_e for the LETKF and the main LPFs with the L96 model in the strongly nonlinear configuration. Note that the ultimate increase of the RMSE of the LETKF with the ensemble size could have been avoided by using random rotations in ensemble space.

As expected in this strongly nonlinear test series, the EnKF fails at accurately reconstructing the true state. By contrast, all LPFs yield at some point an RMSE under $\sigma=1$ (the observation standard deviation). Regarding the ranking of the methods, most conclusions from the mildly nonlinear case remain true. The best RMSE scores are obtained with algorithms using OT-based resampling methods. Combining the smoothing by weights with the coloured noise regularisation jitter methods yields almost equally good scores as the LPF^x algorithms using OT. Finally, using the second order propagation method yields the lowest RMSEs despite the non-Gaussian error distributions that result from nonlinearities.

Appendix D: Rank histograms for the L96 model

As a complement to the RMSE test series, we compute rank histograms of the ensembles (Anderson, 1996; Hamill, 2001). For this experiment, the DA problem is the same as the one in Sects. 5 and 8: the L96 model is used in its standard configuration.

Several algorithms are selected with characteristics detailed in Table D1. The histograms are obtained separately for each state variable by computing the rank of the truth in the unperturbed analysis ensemble (i.e. the analysis ensemble before the regularisation step for the LPFs). To ensure the convergence of the statistical indicators, the runs are $10^5\Delta t$ long with a $10^3\Delta t$ spin-up period. The mean histograms (averaged over the state variables) are reported in Fig. D1.

The histogram of the EnKF is quite flat in the middle, its edges reflect a small ~~over dispersion~~overdispersion. The histogram of the tuned S(IR)^{x}R algorithm is characterised by a large hump, showing that the ensemble is ~~over dispersive~~overdispersive. At the same time, the high frequencies at the edges show that the algorithm yields a poor representation of the distribution tails (as most PF methods). The ~~over dispersion~~overdispersion of the ensemble is a consequence of the fact that the parameters have been tuned to yield the best RMSE score, regardless of the flatness of the rank histogram. With a different set of parameter, the untuned S(IR)^{x}R algorithm yields a rank histogram much flatter. In this case, the regularisation jitter is lower (which explains the fact that the ensemble is less ~~over dispersive~~overdispersive) and the localisation radius smaller (to avoid the filter divergence). Of course, the RMSE score for the untuned S(IR)^{x}R algorithm is higher than for its tuned version. Similar conclusions can be found with the histograms of the tuned and untuned S(IT$_\mathrm{s}$)^{x}R algorithm. Note that in this case the histograms are significantly flatter than with the S(IR)^{x}R algorithm. Finally, the histogram of the (tuned) S(IT$_\mathrm{s}$P$_\mathrm{RK}$)^{y}R is remarkably flat.

In summary, the rank histograms of the LPFs are in general rather flat. The ensemble are more or less ~~over dispersive~~overdispersive, this is a consequence of the use of regularisation jitter, necessary to avoid the filter divergence. As most PF methods, the LPFs yield a poor representation of the distribution tails.

References

Acevedo, W., de Wiljes, J., and Reich, S.: Second-order accurate ensemble transform particle filters, SIAM J. Sci. Comput., 39, A1834–A1850, https://doi.org/10.1137/16M1095184, 2017.

Ades, M. and van Leeuwen, P. J.: The equivalent-weights particle filter in a high-dimensional system, Q. J. Roy. Meteor. Soc., 141, 484–503, https://doi.org/10.1002/qj.2370, 2015.

Anderson, J. L.: A Method for Producing and Evaluating Probabilistic Forecasts from Ensemble Model Integrations, J. Climate, 9, 1518–1530, https://doi.org/10.1175/1520-0442(1996)009<1518:AMFPAE>2.0.CO;2, 1996.

Apte, A. and Jones, C. K. R. T.: The impact of nonlinearity in Lagrangian data assimilation, Nonlinear Processes in Geophysics, 20, 329–341, https://doi.org/10.5194/npg-20-329-2013, 2013.

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T.: A tutorial on particle filters for online nonlinear non-Gaussian Bayesian Tracking, IEEE T. Signal Proces., 50, 174–188, https://doi.org/10.1109/78.978374, 2002.

Bengtsson, T., Snyder, C., and Nychka, D.: Toward a nonlinear ensemble filter for high-dimensional systems, J. Geophys. Res., 108, 8775, https://doi.org/10.1029/2002JD002900, 2003.

Bishop, C. H., Etherton, B. J., and Majumdar, S. J.: Adaptive Sampling with the Ensemble Transform Kalman Filter. Part I: Theoretical Aspects, Mon. Weather Rev., 129, 420–436, https://doi.org/10.1175/1520-0493(2001)129<0420:ASWTET>2.0.CO;2, 2001.

Bocquet, M. and Sakov, P.: An iterative ensemble Kalman smoother, Q. J. Roy. Meteor. Soc., 140, 1521–1535, https://doi.org/10.1002/qj.2236, 2014.

Bocquet, M., Pires, C. A., and Wu, L.: Beyond Gaussian statistical modeling in geophysical data assimilation, Mon. Weather Rev., 138, 2997–3023, https://doi.org/10.1175/2010MWR3164.1, 2010.

Bocquet, M., Raanes, P. N., and Hannart, A.: Expanding the validity of the ensemble Kalman filter without the intrinsic need for inflation, Nonlinear Proc. Geoph., 22, 645–662, https://doi.org/10.5194/npg-22-645-2015, 2015.

Browne, P. A.: A comparison of the equivalent weights particle filter and the local ensemble transform Kalman filter in application to the barotropic vorticity equation, Tellus A, 68, 30 466, https://doi.org/10.3402/tellusa.v68.30466, 2016.

Chen, Z.: Bayesian filtering: From Kalman filters to particle filters, and beyond, Statistics, 182, 1–69, https://doi.org/10.1080/02331880309257, 2003.

Chorin, A. J. and Tu, X.: Implicit sampling for particle filters, P. Natl. Acad. Sci. USA, 106, 17 249–17 254, https://doi.org/10.1073/pnas.0909196106, 2009.

Chorin, A. J., Morzfeld, M., and Tu, X.: Implicit particle filters for data assimilation, Comm. App. Math. Com. Sc., 5, 221–240, https://doi.org/10.2140/camcos.2010.5.221, 2010.

Chustagulprom, N., Reich, S., and Reinhardt, M.: A Hybrid Ensemble Transform Particle Filter for Nonlinear and Spatially Extended Dynamical Systems, SIAM/ASA Journal on Uncertainty Quantification, 4, 592–608, https://doi.org/10.1137/15M1040967, 2016.

Crisan, D. and Doucet, A.: A survey of convergence results on particle filtering methods for practitioners, IEEE T. Signal Proces., 50, 736–746, https://doi.org/10.1109/78.984773, 2002.

Dezső, B., Jüttner, A., and Kovács, P.: LEMON – an Open Source C++ Graph Template Library, Electronic Notes in Theoretical Computer Science, 264, 23–45, https://doi.org/10.1016/j.entcs.2011.06.003, 2011.

Doucet, A., Godsill, S., and Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering, Stat. Comput., 10, 197–208, https://doi.org/10.1023/A:1008935410038, 2000.

Doucet, A., de Freitas, N., and Gordon, N., eds.: Sequential Monte Carlo Methods in Practice, Springer-Verlag New York Inc., https://doi.org/10.1007/978-1-4757-3437-9, 2001.

Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res., 99, 10 143–10 162, https://doi.org/10.1029/94JC00572, 1994.

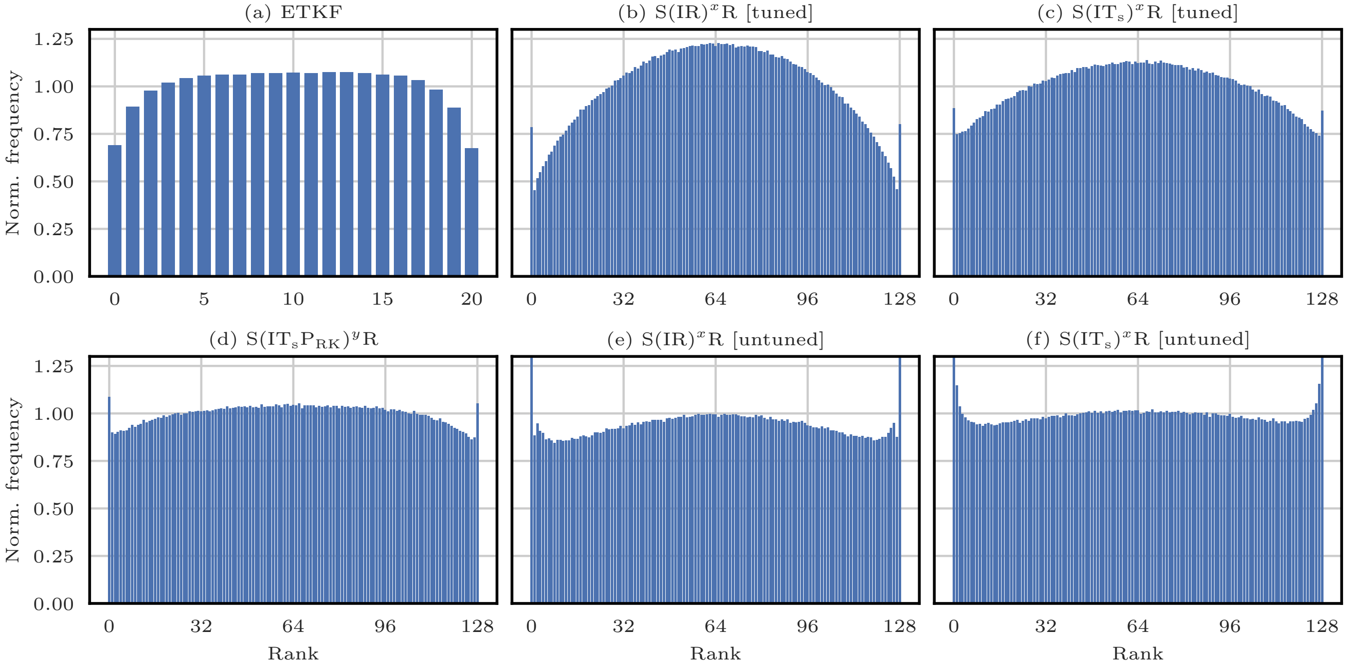
Evensen, G.: The Ensemble Kalman Filter: theoretical formulation and practical implementation, Ocean Dynam., 53, 343–367, https://doi.org/10.1007/s10236-003-0036-9, 2003.

Gaspari, G. and Cohn, S. E.: Construction of correlation functions in two and three dimensions, Q. J. Roy. Meteor. Soc., 125, 723–757, https://doi.org/10.1002/qj.49712555417, 1999.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proc. F, 140, 107–113, https://doi.org/10.1049/ip-f-2.1993.0015, 1993.

Table D1. Rank histograms computed with the L96 model in the standard configuration (see Appendix D). All LPFs use zero integration jitter ($q = 0$). The localisation radii are given in number of grid points. For the ETKF, the optimal multiplicative inflation is reported in the regularisation jitter column. The * asterisk in the RMSE column indicates that the algorithm parameters have been tuned to yield the lowest RMSE score. The first column indicates the corresponding panel in Fig. D1.

Panel	Algorithm	Ens. size N_e	Loc. radius r	Reg. jitter s	Other parameters	RMSE
(a)	ETKF	20	∞	1.02	—	0.188^*
(b)	$\mathrm{S(IR)}^x\mathrm{R}$	128	8	10.0×10^{-2}	$N_\mathrm{b} = 10$	0.289^*
(c)	$\mathrm{S(IT_s)}^x\mathrm{R}$	128	20	4.5×10^{-2}	$h = 1$	0.215^*
(d)	$\mathrm{S(IT_sP_{RK})}^y\mathrm{R}$	128	80	1.0×10^{-2}	$h = 1$	0.180^*
(e)	$\mathrm{S(IR)}^x\mathrm{R}$	128	5	8.0×10^{-2}	$N_\mathrm{b} = 40$	0.500
(f)	$\mathrm{S(IT_s)}^x\mathrm{R}$	128	10	3.0×10^{-2}	$h = 1$	0.228


(a) ETKF

(b) $\mathrm{S(IR)}^x\mathrm{R}$ [tuned]

(c) $\mathrm{S(IT_s)}^x\mathrm{R}$ [tuned]

(d) $\mathrm{S(IT_sP_{RK})}^y\mathrm{R}$

(e) $\mathrm{S(IR)}^x\mathrm{R}$ [untuned]

(f) $\mathrm{S(IT_s)}^x\mathrm{R}$ [untuned]

Figure D1. Rank histograms for the selection of algorithms detailed in Table D1. The frequency is normalised by $N_\mathrm{e} + 1$ (the number of bins).

Greybush, S. J., Kalnay, E., Miyoshi, T., Ide, K., and Hunt, B. R.: Balance and Ensemble Kalman Filter Localization Techniques, Mon. Weather Rev., 139, 511–522, https://doi.org/10.1175/2010MWR3328.1, 2011.

Hamill, T. M.: Interpretation of Rank Histograms for Verifying Ensemble Forecasts, Mon. Weather Rev., 129, 550–560, https://doi.org/10.1175/1520-0493(2001)129<0550:IORHFV>2.0.CO;2, 2001.

Hamill, T. M., Whitaker, J. S., and Snyder, C.: Distance-Dependent Filtering of Background Error Covariance Estimates in an Ensemble Kalman Filter, Mon. Weather Rev., 129, 2776–2790, https://doi.org/10.1175/1520-0493(2001)129<2776:DDFOBE>2.0.CO;2, 2001.

Houtekamer, P. L. and Mitchell, H. L.: A Sequential Ensemble Kalman Filter for Atmospheric Data Assimilation, Mon. Weather Rev., 129, 123–137, https://doi.org/10.1175/1520-0493(2001)129<0123:ASEKFF>2.0.CO;2, 2001.

Houtekamer, P. L., Mitchell, H. L., Pellerin, G., Buehner, M., Charron, M., Spacek, L., and Hansen, B.: Atmospheric data assimilation with an ensemble Kalman filter: Results with real observations, Mon. Weather Rev., 133, 604–620, https://doi.org/10.1175/MWR-2864.1, 2005.

Hunt, B. R., Kostelich, E. J., and Szunyogh, I.: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter, Physica D, 230, 112–126, https://doi.org/10.1016/j.physd.2006.11.008, 2007.

Kalnay, E. and Yang, S.-C.: Accelerating the spin-up of Ensemble Kalman Filtering, Q. J. Roy. Meteor. Soc., 136, 1644–1651, https://doi.org/10.1002/qj.652, 2010.

Kong, A., Liu, J. S., and Wong, W. H.: Sequential Imputations and Bayesian Missing Data Problems, J. Am. Stat. Assoc., 89, 278–288, https://doi.org/10.1080/01621459.1994.10476469, 1994.

Le Gland, F., Musso, C., and Oudjane, N.: An Analysis of Regularized Interacting Particle Methods for Nonlinear Filtering, in:

Proceedings of the 3rd IEEE European Workshop on Computer-Intensive Methods in Control and Signal Processing, pp. 167–174, 1998.

Lee, Y. and Majda, A. J.: State estimation and prediction using clustered particle filters, P. Natl. Acad. Sci. USA, 113, 14 609–14 614, https://doi.org/10.1073/pnas.1617398113, 2016.

Lorenz, E. N. and Emanuel, K. A.: Optimal Sites for Supplementary Weather Observations: Simulation with a Small Model, J. Atmos. Sci., 55, 399–414, https://doi.org/10.1175/1520-0469(1998)055<0399:OSFSWO>2.0.CO;2, 1998.

MacKay, D. J. C.: Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.

Metref, S., Cosme, E., Snyder, C., and Brasseur, P.: A non-Gaussian analysis scheme using rank histograms for ensemble data assimilation, Nonlinear Proc. Geoph., 21, 869–885, https://doi.org/10.5194/npg-21-869-2014, 2014.

Morzfeld, M., Tu, X., Atkins, E., and Chorin, A. J.: A random map implementation of implicit filters, J. Comput. Phys., 231, 2049–2066, https://doi.org/10.1016/j.jcp.2011.11.022, 2012.

Musso, C. and Oudjane, N.: Regularization schemes for branching particle systems as a numerical solving method of the nonlinear filtering problem, in: Proceedings of the Irish Signals and Systems Conference, Dublin, 1998.

Musso, C., Oudjane, N., and Le Gland, F.: Improving Regularised Particle Filters, in: Sequential Monte Carlo Methods in Practice, edited by Doucet, A., de Freitas, N., and Gordon, N., pp. 247–271, Springer-Verlag New York Inc., https://doi.org/10.1007/978-1-4757-3437-9_12, 2001.

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., Kalnay, E., Patil, D. J., and Yorke, J. A.: A local ensemble Kalman filter for atmospheric data assimilation, Tellus A, 56, 415–428, https://doi.org/10.1111/j.1600-0870.2004.00076.x, 2004.

Oudjane, N. and Musso, C.: Multiple model particle filter, in: Actes du dix-septième colloque GRETSI, Vannes, pp. 681–684, 1999.

Pele, O. and Werman, M.: Fast and robust Earth Mover's Distances, in: 2009 IEEE 12th International Conference on Computer Vision, pp. 460–467, IEEE, https://doi.org/10.1109/ICCV.2009.5459199, 2009.

Penny, S. G. and Miyoshi, T.: A local particle filter for high-dimensional geophysical systems, Nonlinear Proc. Geoph., 23, 391–405, https://doi.org/10.5194/npg-23-391-2016, 2016.

Pham, D. T.: Stochastic Methods for Sequential Data Assimilation in Strongly Nonlinear Systems, Mon. Weather Rev., 129, 1194–1207, https://doi.org/10.1175/1520-0493(2001)129<1194:SMFSDA>2.0.CO;2, 2001.

Poterjoy, J.: A Localized Particle Filter for High-Dimensional Nonlinear Systems, Mon. Weather Rev., 144, 59–76, https://doi.org/10.1175/MWR-D-15-0163.1, 2016.

Rebeschini, P. and van Handel, R.: Can local particle filters beat the curse of dimensionality?, Ann. Appl. Probab., 25, 2809—-2866, https://doi.org/10.1214/14-AAP1061, 2015.

Reich, S.: A nonparametric ensemble transform method for Bayesian inference, SIAM J. Sci. Comput., 35, A2013–A2014, https://doi.org/10.1137/130907367, 2013.

Reich, S. and Cotter, C.: Probabilistic Forecasting and Bayesian Data Assimilation, Cambridge University Press, https://doi.org/10.1017/CBO9781107706804, 2015.

Robert, S. and Künsch, H. R.: Localizing the Ensemble Kalman Particle Filter, Tellus A, 69, 1282 016, https://doi.org/10.1080/16000870.2017.1282016, 2017.

Sakov, P., Counillon, F., Bertino, L., Lisæter, K. A., Oke, P. R., and Korablev, A.: TOPAZ4: an ocean-sea ice data assimilation system for the North Atlantic and Arctic, Ocean Sci., 8, 633–656, https://doi.org/10.5194/os-8-633-2012, 2012a.

Sakov, P., Oliver, D. S., and Bertino, L.: An Iterative EnKF for Strongly Nonlinear Systems, Mon. Weather Rev., 140, 1988–2004, https://doi.org/10.1175/MWR-D-11-00176.1, 2012b.

Silverman, B. W.: Density Estimation for Statistics and Data Analysis, vol. 26 of *Monographs on Statistics & Applied Probability*, Chapman & Hall, London, 1986.

Slivinski, L., Spiller, E., Apte, A., and Sandstede, B.: A Hybrid Particle–Ensemble Kalman Filter for Lagrangian Data Assimilation, Mon. Weather Rev., 143, 195–211, https://doi.org/10.1175/MWR-D-14-00051.1, 2015.

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. L.: Obstacles to High-Dimensional Particle Filtering, Mon. Weather Rev., 136, 4629–4640, https://doi.org/10.1175/2008MWR2529.1, 2008.

Snyder, C., Bengtsson, T., and Morzfeld, M.: Performance Bounds for Particle Filters Using the Optimal Proposal, Mon. Weather Rev., 143, 4750–4761, https://doi.org/10.1175/MWR-D-15-0144.1, 2015.

van Leeuwen, P. J.: A Variance-Minimizing Filter for Large-Scale Applications, Mon. Weather Rev., 131, 2071–2084, https://doi.org/10.1175/1520-0493(2003)131<2071:AVFFLA>2.0.CO;2, 2003.

van Leeuwen, P. J.: Particle filtering in geophysical systems, Mon. Weather Rev., 137, 4089–4114, https://doi.org/10.1175/2009MWR2835.1, 2009.

van Leeuwen, P. J.: Nonlinear data assimilation in geosciences: an extremely efficient particle filter, Q. J. Roy. Meteor. Soc., 136, 1991–1999, https://doi.org/10.1002/qj.699, 2010.

van Leeuwen, P. J. and Ades, M.: Efficient fully nonlinear data assimilation for geophysical fluid dynamics, Computers & Geosciences, 55, 16–27, https://doi.org/https://doi.org/10.1016/j.cageo.2012.04.015, 2013.

Villani, C.: Optimal Transport, Old and New, vol. 338 of *Grundlehren der mathematischen Wissenschaften*, Springer-Verlag Berlin Heidelberg, 2009.

Zhou, Y., McLaughlin, D., and Entekhabi, D.: Assessing the Performance of the Ensemble Kalman Filter for Land Surface Data Assimilation, Mon. Weather Rev., 134, 2128–2142, https://doi.org/10.1175/MWR3153.1, 2006.

Zhu, M., van Leeuwen, P. J., and Amezcua, J.: Implicit equal-weights particle filter, Q. J. Roy. Meteor. Soc., 142, 1904–1919, https://doi.org/10.1002/qj.2784, 2016.

Zupanski, M.: Maximum Likelihood Ensemble Filter: Theoretical Aspects, Mon. Weather Rev., 133, 1710–1726, https://doi.org/10.1175/MWR2946.1, 2005.