Nonlin. Processes Geophys. Discuss., https://doi.org/10.5194/npg-2018-15
Manuscript under review for journal Nonlin. Processes Geophys.
Discussion started: 5 March 2018

Nonlinear Processes in Geophysics
Discussions
Open Access

Review article: Comparison of local particle filters and new implementations

Farchi Alban[1] and Bocquet Marc[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 weight degeneracy. This phenomenon is a consequence of the curse of dimensionality that prevents one from using PF methods for high-dimensional data assimilation. The use of local analyses to counteract the curse of dimensionality was suggested early on. However, implementing localisation in the PF is a challenge because there is no simple and yet consistent way of gluing 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.

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 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). 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 poor estimates of the model state. This phenomenon is a consequence 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 and localisation.

Importance sampling is at the heart of PF methods where the goal is to construct a representation 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 particles. However, importance sampling with a proposal density can lead to more costly algorithms that are not necessarily free 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, 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 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 of freedom is small compared to the others. Examples of assimilation using hybrid filters can be found when applied to geophysical low-order models (Chustagulprom et al., 2016) and in Lagrangian DA (Apte and Jones, 2013; Slivinski et al., 2015).

In most geophysical system, 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 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 are given in Sect. 10.

2 Background

2.1 The data assimilation filtering problem

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

The model can alternatively be described by

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

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

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

Let $\pi_{k|k}$ be the *analysis* (or *filtering*) density $\pi_{k|k} = p(\mathbf{x}_k|\mathbf{y}_{k:0})$, where $\mathbf{y}_{k:0}$ is the set $\{\mathbf{y}_l, l = 0\ldots k\}$ and let $\pi_{k+1|k}$ be the *prediction* (or *forecast*) density $\pi_{k+1|k} = p(\mathbf{x}_{k+1}|\mathbf{y}_{k:0})$ with $\pi_{0|-1}$ coinciding with $p(\mathbf{x}_0)$ by convention.

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

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

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

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

The DA filtering problem consists in estimating $\pi_{k|k}$ with given realisations of $\mathbf{y}_{k:0}$.

2.2 Particle filtering

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

$$\pi_{k|k}^{N_e} = \sum_{i=1}^{N_e} w_k^i \, \delta_{\mathbf{x}_k^i}, \tag{5}$$

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

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

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

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:

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

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

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

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

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

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

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

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 $N_e \to \infty$, $\pi_{k|k}^{N_e}$ 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 duplicated while particles with low probability are discarded. It is desirable that the selection of particles has a low impact on the empirical density $\pi_{k|k}^{N_e}$. 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 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.

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

The resampling step and the correction step 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 $N_x \times N_e$ matrix whose columns are the ensemble members $\mathbf{x}_k^i$. 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 $N_e \times N_e$ 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

$$\sum_{i=1}^{N_e} [\mathbf{T}]^{i,j} = 1, \quad j = 1 \dots N_e, \tag{13}$$

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

$$\sum_{j=1}^{N_e} [\mathbf{T}]^{i,j} = N_e w_k^i, \quad i = 1 \dots N_e. \tag{14}$$

In the classical 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 $\pi_{k|k}^{N_e}$ 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):

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

where $\mathcal{T}$ is the set of $N_e \times N_e$ transformation matrices satisfying Eqs. (13) and (14). In this way, the correlation between the prior and the updated ensembles is increased and $\pi_{k|k}^{N_e}$ 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 $q(\mathbf{x}_{k+1})$ be a density whose support is larger than that of $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ — i.e. $q(\mathbf{x}_{k+1}) > 0$ whenever $p(\mathbf{x}_{k+1}|\mathbf{x}_k) > 0$. The Chapman–Kolmogorov Eq. (3) can be written:

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

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:

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

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

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 $\mathbf{x}_k$ or $\mathbf{y}_{k+1}$ in order to reduce the variance of the importance weights.

The SIR algorithm is recovered with the *standard* proposal $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$, while the *optimal importance* proposal $p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{y}_{k+1})$ 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

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

It is remarkable that this formula does not depend on $\mathbf{x}_{k+1}$. Hence the optimal importance proposal is optimal in the sense that it minimises the variance of the weights over realisations of $\mathbf{x}_{k+1}^i$ — namely 0. Moreover, it can be shown that it also minimises the variance of the weights over realisations of the whole trajectory $\mathbf{x}^i_{k+1:0}$ among proposal densities that depend on $\mathbf{x}_k$ and $\mathbf{y}_{k+1}$ (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 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 to weight degeneracy (e.g., van Leeuwen, 2003; Zhou et al., 2006).

Bocquet et al. (2010) reproduce weight degeneracy in low-order models, for example in the Lorenz 1996 (L96, Lorenz and Emanuel, 1998) model in the standard configuration described in Appendix A3. They illustrate 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 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 surprising that, although MC method 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 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).

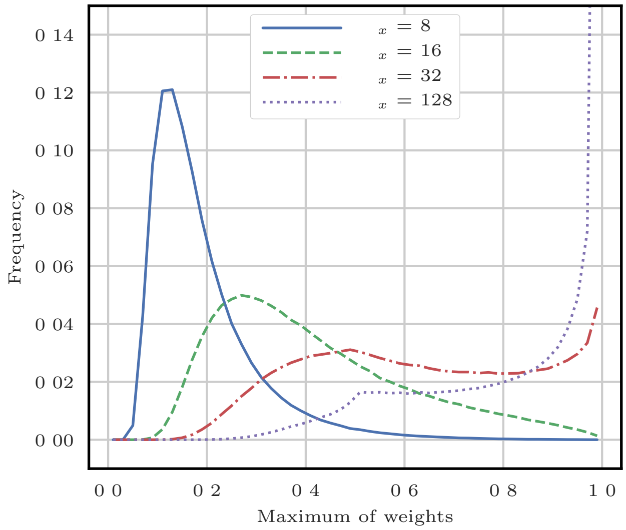

Figure 1. Empirical statistics of the maximum of the weights for one importance sampling step applied to the Gaussian linear model of Appendix A1. The model parameters are $p = 1$, $a = 1$, $h = 1$, the ensemble size is $N_e = 128$ and the system size varies from $N_x = 8$ (well balanced case) to $N_x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

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:

205 $$\tau^2 = \operatorname{var}\left[\ln\left(p\left(\mathbf{y}_k|\mathbf{x}_k\right)\right)\right], \tag{20}$$

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:

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

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

210 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 215 linear model of Appendix A1 with a SIR algorithm, for which:

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

Snyder et al. (2008) does 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 $p\left(\mathbf{x}_{k+1}|\mathbf{x}_k,\mathbf{y}_{k+1}\right)$ to sample $\mathbf{x}_k$ in the prediction and sampling step $S^{N_e} \circ P_k$, 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 $\mathbf{x}_{k+1}$.

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 come from the correction step C_k, but from the recursion in the PF and in particular 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:

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

and they found a good accuracy of Eq. (21) in the limit $N_e \ll \exp\left(\tau^2/2\right)$. 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 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 $\mathbf{y}_k$. 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. 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 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 $S^{N_e} \circ P_k$ 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. In EnKF methods, these issues are mitigated by using smooth functions to taper the influence of the observation components. 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.

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 $\mathbf{x}_k$, hereafter called *state variable* or *grid point*, is attached to a location;
- each component of $\mathbf{y}_k$, hereafter called *observation site* is attached to a location as well (local observations);
- 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 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 each grid point by using only the observation sites that influence this point. This leads to algorithms that are easy to define, to implement and to parallelise. However, there is no obvious relationship between grid points, which could be problematic with respect to the imbalance issue. We will call this approach *state–domain* (and later *state–block–domain*) localisation.

In the second approach, an analysis is performed for each observation site. When assimilating an observation of a site, we partition the state space: nearby grid points are updated while distant grid point remain unchanged. In this formalism, observation sites need to be assimilated sequentially, which makes the algorithms harder to define and to parallelise but may mitigate the imbalance issue. We will call this approach *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 $i \in \{1 \dots N_e\}$ is the member index, the subscript $n \in \{1 \dots N_x\}$ is the grid point index, the subscript $q \in \{1 \dots N_y\}$ is the observation site index, the subscript $b \in \{1 \dots N_b\}$ 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 $n = 1 \dots N_x$ is

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

whose localised version is

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

The weights w_n^i depend on the spatial position through the grid point 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 fix is to approximate the full density as the product of its marginals:

$$p(\mathbf{x}) = \prod_{n=1}^{N_x} \sum_{i=1}^{N_e} w_n^i \, \delta_{x_n^i}, \tag{26}$$

which is a weighted sum of the $N_e^{N_x}$ 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.

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 blocks* with the additional constraint that the weights should be constant over the blocks. The resampling is then performed 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$^{\mathrm{x}}$ algorithms, are characterised by

- the geometry of the local (state) blocks over which the weights are constant;
- 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 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 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 same principle can be applied to the SBD localisation formalism: the local domain of a block will be a circle whose center coincides with that of the block and whose radius will be a free parameter.

The terminology adopted here (circle, 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 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

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

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

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

where α is a constant that should be of the same order as $\max p(y|\mathbf{x})$, $d_{q,b}$ is the distance between the observation site q and the center of 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 formula for w_b^i directly inspired from local EnKF methods:

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

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$^{\mathrm{x}}$. 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 block b (i.e. the rows of $\mathbf{E}$ corresponding to state variables that are located within block b). $\mathbf{E}_{|b}$ is a $N_x/N_b \times N_e$ matrix.

An illustration of the definition of local blocks and 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 example of their block particle filter algorithm. In this algorithm, the distinction between 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.

Algorithm 1 Analysis step for a generic LPF$^{\text{x}}$

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

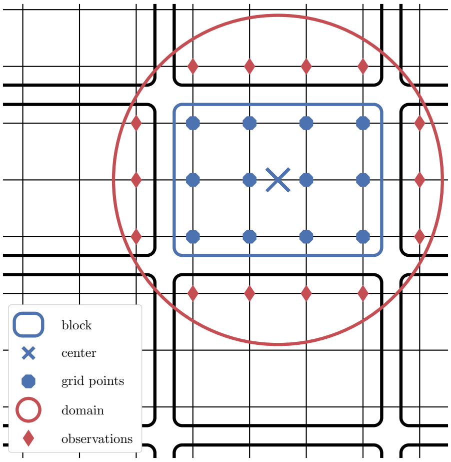

Figure 2. Example of geometry in the SBD localisation formalism for a two-dimensional space. The focus is on the block in the middle which gathers 12 grid points. The local domain is circumscribed by a circle around the block center with potential observation sites outside the local block.

The main mathematical result is that, under reasonable hypotheses, the error on the analysis density for this LPF can be bounded by the sum of a bias and a variance term. The bias term is related to the block boundaries and decrease exponentially with the diameter of the blocks. It is due to the fact that the correction is not Bayesian any more since only a subset of observation sites 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 $\exp(K)/\sqrt{N_e}$, 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. This implies that LPF algorithms can indeed beat the curse of dimensionality.

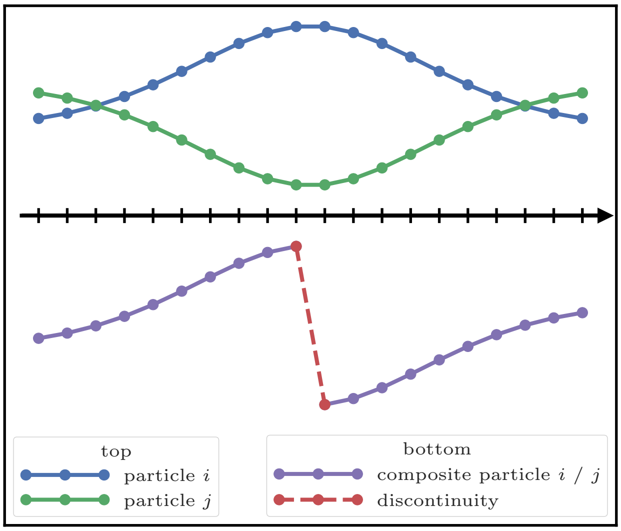

Figure 3. Example of concatenation of particle i on the left and particle j on the right. Top: the prior particles i and j. Bottom: the composite particle, concatenation of i and j. In this situation, a large and artificial discontinuity appears at the boundary.

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 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 concatenation is likely to lead to artificial discontinuities in the updated particles, regardless of the spatial smoothness of the analysis weights. More precisely, we want to build *composite particles*, that is when the i-th updated particle is the concatenation of the j-th particle on one block and the k-th particle on an adjacent block with $j \neq k$ — as shown by Fig. 3. There is no guarantee that the j-th and the k-th local particles are *close* and that their concatenation 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 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 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 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 $\mathbf{E}_b^{\mathrm{r}}$ be the matrix of the ensemble computed by applying the resampling method to the global ensemble weighted by the local weights w_b^i. $\mathbf{E}_b^{\mathrm{r}}$ is a $N_x \times N_e$ matrix different from the $N_x/N_b \times N_e$ matrix $\mathbf{E}_{|b}^{\mathrm{r}}$ implicitly defined by step 3 of Algorithm 1. We then define the smoothed ensemble matrix $\mathbf{E}^{\mathrm{s}}$ by

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

where $d_{n,b}$ is the distance between grid point n and the center of 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 the resampling map at block b, i.e. the map computed with the local weights w_b^i such that $\phi_b(i)$ is the index of the i-th resampled particle. $\mathbf{E}$ being the prior ensemble matrix, Eq. (30) becomes

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

Finally, the ensemble is updated as

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

where $\mathbf{E}^{\mathrm{r}}$ be the resampled ensemble matrix implicitly defined by step 5 of Algorithm 1, 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$^{\mathrm{x}}$ 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 of this algorithm.

The smoothing by weights step is an ad-hoc fix to reduce afterwards the artificial discontinuities introduced in the local resampling. 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$^\mathrm{x}$ with smoothing by weights

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

4.4.2 Refining the sampling algorithms

A sampling algorithm is said to be *balanced* if, for $i = 1 \dots N_e$ the number of copies of the i-th particle does not differ by more than one unity from $w^i N_e$. 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 $i \in \{1 \dots N_e\}$ 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*. The SU sampling and the multinomial resampling algorithms can be simply modified to yield adjustment-minimising resampling algorithms.

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 a stochastic source of artificial spatial discontinuities. Choosing balanced and adjustment-minimising resampling algorithms is an attempt to include some 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 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 discontinuities in them.

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

$$\min_{\mathbf{T}_b \in \mathcal{T}_b} \sum_{i,j=1}^{N_e} [\mathbf{T}_b]^{i,j} c_b^{i,j}, \tag{33}$$

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

$$\sum_{j=1}^{N_e} [\mathbf{T}_b]^{i,j} = N_e w_b^i, \quad i = 1 \ldots N_e. \tag{34}$$

In the ETPF, the coefficients $c^{i,j}$ were chosen as the squared L^2 distance between the whole 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 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$^\mathrm{x}$ that uses optimal ensemble coupling as local resampling algorithm.

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, the focus of Chustagulprom et al. (2016) was to include the ETPF algorithm in a hybridisation context. They use the local weights defined by Eq. (29), but they only consider blocks of size 1 grid point and local coefficients given by

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

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.

Algorithm 3 Analysis step for a generic LPF$^{\mathrm{x}}$ using optimal ensemble coupling for the local resampling

```
Require: Prior ensemble $\mathbf{x}^i$, $i = 1 \dots N_e$
for $b = 1$ to $N_b$ 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 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 $\mathbf{x}^i$, $i = 1 \dots N_e$
```

Using optimal ensemble coupling for the resampling step in a LPF$^{\mathrm{x}}$ 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 dimension, the OT problem is highly non-trivial and numerically very costly (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 n, we define the prior (marginal) pdf p_n^{f} as the empirical density of the unweighted prior ensemble and the analysis pdf p_n^{a} as the empirical density of the prior ensemble weighted by the analysis weights. We seek the map T_n that solves the following OT problem:

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

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

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

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

In one dimension, this transport map is also known as 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, 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 is to use the kernel density estimation (KDE) theory (Silverman, 1986; Musso et al., 2001). In this context, the prior density can be written

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

while the updated density is

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

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 $\left\{x_n^i, i = 1 \ldots N_e\right\}$ and the weighted ensemble $\left\{\left(x_n^i, w_n^i\right), i = 1 \ldots N_e\right\}$ 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 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 to a Gaussian, 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$^{\mathrm{x}}$ that uses anamorphosis as local resampling algorithm.

The local resampling algorithm using anamorphosis is, together with the algorithm using optimal ensemble coupling, a deterministic transformation. This means that potentially large artificial discontinuities due to different random realisations over the grid points are avoided. As explained by Poterjoy (2016), in such 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 discretisation is fine enough — and this kind of regularity is transferred in the updated ensemble.

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.

Algorithm 4 Analysis step for a generic LPF$^{\mathrm{x}}$ using anamorphosis for the local resampling

```
Require: Prior ensemble $\mathbf{x}^i, i = 1 \ldots N_e$
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 $N_e$ 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 $\mathbf{x}^i$, $i = 1 \ldots N_e$.
```

The refinements of the resampling algorithms suggested in Sect. 4.4.2 were designed to minimise the number of artificial discontinuities in the local resampling step. The goal of the smoothing by weights step is to reduce the impact of the discontinuities after they have been introduced. On the other hand, the local resampling algorithms based on OT are designed to reduce the 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.

4.4.5 Parameter choice and asymptotic limit

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.

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 LPF$^\text{x}$ algorithm, the local analyses should in general be much faster because there is no matrix inversion to perform. In this case, using bigger state blocks is a way to decrease the proportion of block borders, which are potential spots for artificial 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 blocks should be small — no more than a few grid points — only big ensemble could benefit from the use of larger state blocks.

The analysis step of LPF$^\text{x}$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$^\text{x}$s using optimal ensemble coupling is equivalent to the ETPF analysis step in the asymptotic limit $r \to \infty$ and $r_\text{d} \to \infty$ 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$^\text{x}$s using anamorphosis is asymptotically unbiased.

5 Numerical illustration of LPF$^\text{x}$ algorithms with the Lorenz-96 model

5.1 Model specifications

In this section, we illustrate the performance of LPF$^\text{x}$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 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 be entirely reflected in the RMSE. However, not only does the RMSE offers 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 C 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 and periodic boundary conditions consistent with the system size. Therefore, the local domain centered on grid point n is $\{n - \lfloor r \rfloor, \ldots, n + \lfloor r \rfloor\}$, 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 and representative of synoptic scale meteorology and the error distributions are close to Gaussians. 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.

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:

$$\mathbf{x}_{k+1}^{i} - \mathcal{M}\left(\mathbf{x}_{k}^{i}\right) = \mathbf{w}_{k} \sim \mathcal{N}\left(\mathbf{0}, q^{2}\mathbf{I}\right), \tag{42}$$

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. (42) 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:

$$\mathbf{x}_{k+1}^{i} \leftarrow \mathbf{x}_{k+1}^{i} + \mathbf{u}, \quad \mathbf{u} \sim \mathcal{N}\left(\mathbf{0}, s^{2}\mathbf{I}\right), \tag{43}$$

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 discontinuities. Somehow this method is similar to ensemble 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 Standard S(IR)$^{\mathrm{x}}$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 than the truth.[1] With 10^3 particles, the RMSE is around 0.6 and with 10^4 it is around 0.4.

We define the standard S(IR)$^{\mathrm{x}}$R algorithm — sampling, importance, resampling, regularisation, the x exponent meaning that steps in parentheses are performed locally on each state block — as the LPF$^{\mathrm{x}}$ 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. (42) 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. (43) 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)$^{\mathrm{x}}$R algorithm has five parameters: (N_e, N_b, r, q, s). All algorithms tested in this section are variants of this standard algorithm and are named S(I$_*$$*$)$^{\mathrm{x}}$$*R_*$ with the conventions detailed in Table 1.

5.3 Tuning the localisation radius

We first check that, in this configuration, localisation is working by testing the S(IR)$^{\mathrm{x}}$R algorithm with $N_b = 40$ blocks of size 1 grid point. We take $N_e = 10$ 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. 4. 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) 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 small (5 observation sites) 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)$^{\mathrm{x}}$R algorithm with $N_e = 10$ particles, $N_b = 40$ blocks of size 1 grid point and a localisation radius $r = 3$ grid points. The evolution of the RMSE as a function of the integration jitter q is shown in Fig. 5 and as a function of the regularisation jitter s in Fig. 6.

From these results, we can identify two regimes:

---

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

Table 1. Nomenclature conventions for the S($\alpha\beta$)$^{\mathrm{x}}\gamma\delta$ algorithms.

α	Local importance weights
$\mathrm{I_0}$	Eq. (28)
I	Eq. (29)

β	Local resampling algorithm
$\mathrm{R_{SU}}$	SU sampling algorithm
$\mathrm{R_d}$	adjustment-minimising SU sampling algorithm with the same random numbers over all blocks
R	adjustment-minimising SU sampling algorithm
$\mathrm{T_e}$	optimal transport in ensemble space
$\mathrm{T_s}$	optimal transport in state space

γ	Smoothing-by-weights
S	enabled
–	disabled

δ	Regularisation method (see Sect. 5.8)
R	white noise method
$\mathrm{R_a}$	coloured noise method

- 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 yields significantly better results. This indicates that the model integration indeed smoothes the jitter out and removes artificial discontinuities for the correction step. We observed the same tendency for most LPFs tested in this study.

In the rest of this section, we take zero integration jitter ($q = 0$) and the localisation radius r and the regularisation jitter s are systematically tuned to yield the lowest RMSE score.

5.5 Increasing the size of the local blocks

To illustrate the influence of the size of the local blocks, we compare the RMSEs obtained by the S(IR)$^{\mathrm{x}}$R algorithm 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. 7. For

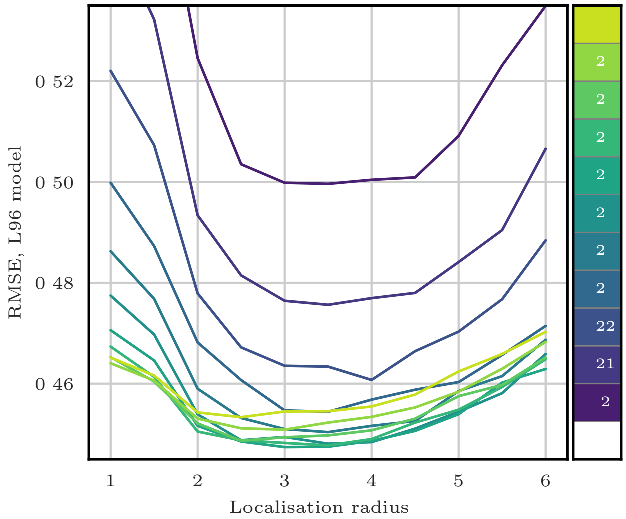

Figure 4. RMSE as a function of the localisation radius r for the S(IR)$^{\mathrm{x}}$R algorithm with $N_e = 10$ particles, $N_b = 40$ blocks of size 1 grid point and no integration jitter ($q = 0$). For each r, several values for the regularisation jitter s are tested as shown by the color scale.

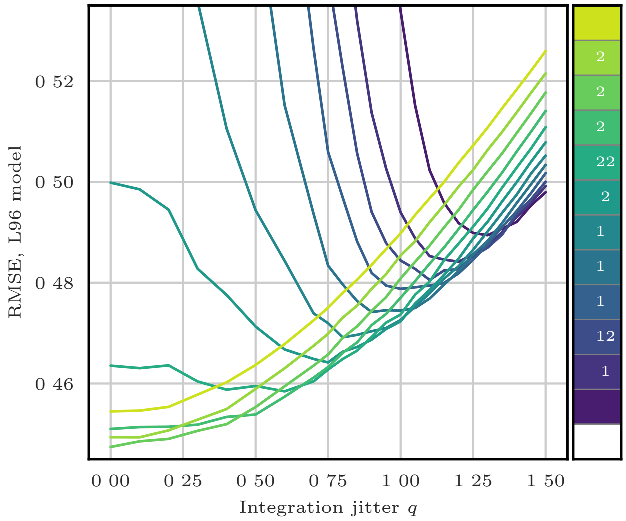

Figure 5. RMSE as a function of the integration jitter q for the S(IR)$^{\mathrm{x}}$R algorithm with $N_e = 10$ particles, $N_b = 40$ blocks of size 1 grid point and a localisation radius $r = 3$ grid points. For each q, several values for the regularisation jitter s are tested as shown by the color scale.

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 ensembles benefit from the use of large blocks as a consequence of the reduction of proportion of block boundaries, potential spots for artificial discontinuities.

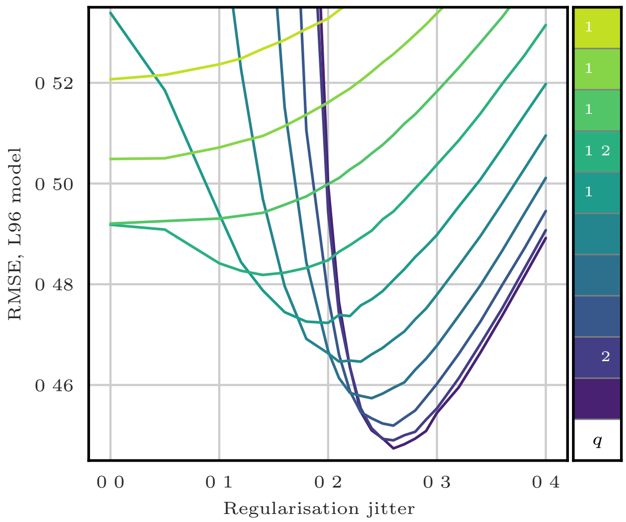

Figure 6. RMSE as a function of the regularisation jitter s for the S(IR)$^{\text{x}}$R algorithm with $N_e = 10$ particles, $N_b = 40$ blocks of size 1 grid point and a localisation radius $r = 3$ grid points. For each s, several values for the integration jitter q are tested as shown by the color scale.

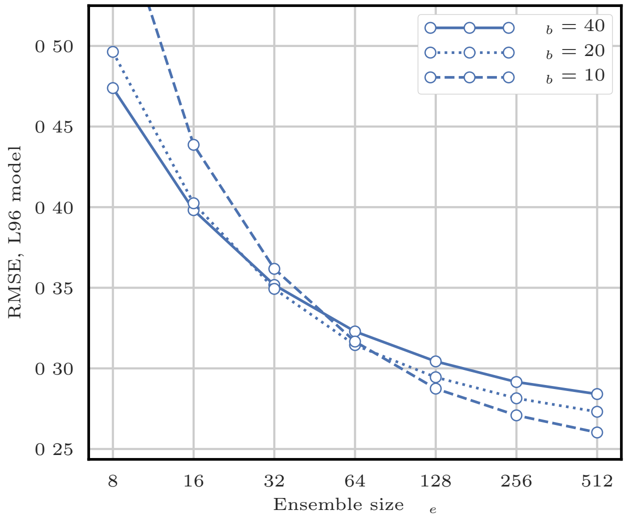

Figure 7. RMSE as a function of the ensemble size N_e for the S(IR)$^{\text{x}}$R algorithm with respectively $N_b = 40, 20, 10$ blocks of size $1, 2, 4$ grid points.

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

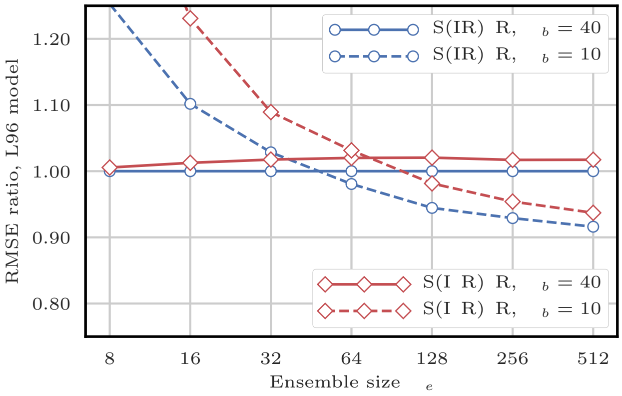

Figure 8. RMSE as a function of the ensemble size N_e for the S(IR)$^\mathrm{x}$R and the S(I$_0$R)$^\mathrm{x}$R algorithms with respectively $N_b = 40, 10$ blocks of size $1, 4$ grid points. The scores are displayed in units of the RMSE of the S(IR)$^\mathrm{x}$R algorithm with $N_b = 40$ blocks of size 1 grid point.

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)$^\mathrm{x}$R and the S(I$_0$R)$^\mathrm{x}$R algorithms. These two variants only differ by their definition of the local importance weights: the S(IR)$^\mathrm{x}$R algorithm uses the Gaussian tapering of observations influence defined by Eq. (29) while the S(I$_0$R)$^\mathrm{x}$R algorithm uses the non-Gaussian tapering given by Eq. (28).

Figure 8 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. In the following, we always use Eq. (29) to define the local weights.

5.7 Refining the stochastic universal sampling

Here, the sampling algorithms proposed in Sect. 4.4.2 are tested. To do this we compare, the RMSEs of the S(IR)$^\mathrm{x}$R algorithms with those of:

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

Figure 9 shows the evolution of the RMSE as a function of the ensemble size N_e. The higher RMSEs of the S(IR$_\mathrm{SU}$)$^\mathrm{x}$R algorithm, especially for large ensembles, show that the adjustment-minimising property is indeed an efficient way of reducing the number of artificial 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 discontinuities introduced when concatenating the locally updated particles. This is probably a consequence of the fact that the SU sampling algorithm only uses one random number to compute

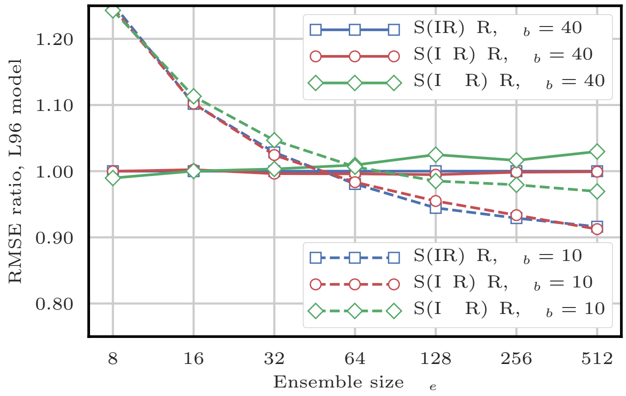

Figure 9. RMSE as a function of the ensemble size N_e for the S(IR)$^{\mathrm{x}}$R, the S(IR$_{\mathrm{d}}$)$^{\mathrm{x}}$R and the the S(IR$_{\mathrm{SU}}$)$^{\mathrm{x}}$R algorithms with respectively $N_b = 40, 10$ blocks of size $1, 4$ grid points. The scores are displayed in units of the RMSE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

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.

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

According to Eqs. (42) and (43), 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.

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

$$[\mathbf{\Sigma}]_{n,m} = \frac{\hat{h}}{1 - \sum\limits_{i=1}^{N_e} \left(w^i\right)^2} \sum_{i=1}^{N_e} w^i \left(x_n^i - \overline{x}_n\right)\left(x_m^i - \overline{x}_m\right), \tag{44}$$

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

$$\overline{x}_n = \frac{1}{N_e} \sum_{i=1}^{N_e} x_n^i. \tag{45}$$

In practice, the $N_x \times N_e$ anomaly matrix is defined by

$$[\mathbf{X}]_{n,i} = \sqrt{\frac{\hat{h}w^i}{1-\sum_{i=1}^{N_e}\left(w^i\right)^2}}\left(x_n^i - \overline{x}_n\right), \tag{46}$$

and the regularisation is added as

$$\mathbf{E} \leftarrow \mathbf{E} + \mathbf{XZ}, \tag{47}$$

with $\mathbf{Z}$ being a $N_e \times N_e$ 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 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 (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.

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)$^{\mathrm{x}}$R algorithm and its variants tested so far. We still want to investigate the potential gains of using colored jitters in LPF$^{\mathrm{x}}$s.

In the analysis step of LPF$^{\mathrm{x}}$s, there is a unique weight for each state variable n, therefore it is not possible to compute the covariance of the regularisation jitter with Eq. (44). 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. However, this approach systematically leads to higher RMSEs for the S(IR)$^{\mathrm{x}}$R algorithm (not shown here). This can be potentially explained by two factors. 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, as following:

$$[\mathbf{X}]_{n,i} = \sqrt{\frac{\hat{h}w_n^i}{1-\sum_{i=1}^{N_e}\left(w_n^i\right)^2}}\left(x_n^i - \overline{x}_n\right). \tag{48}$$

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

$$[\mathbf{\Sigma}]_{n,m} = \sum_{i=1}^{N_e} \frac{\hat{h}\sqrt{w_n^i w_m^i}\left(x_n^i - \overline{x}_n\right)\left(x_m^i - \overline{x}_m\right)}{\sqrt{\left(1-\sum_{i=1}^{N_e}\left(w_n^i\right)^2\right)\left(1-\sum_{i=1}^{N_e}\left(w_m^i\right)^2\right)}}, \tag{49}$$

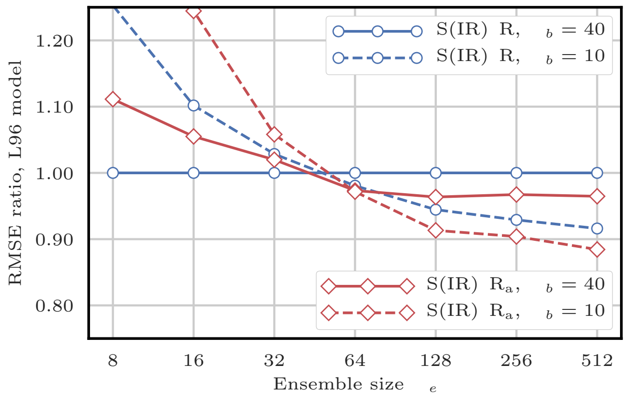

Figure 10. RMSE as a function of the ensemble size N_e for the S(IR)$^{\mathrm{x}}$R and the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithms with respectively $N_b = 40, 10$ blocks of size $1, 4$ grid points. The scores are displayed in units of the RMSE of the S(IR)$^{\mathrm{x}}$R algorithm with $N_b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

which is an extension of Eq. (44).

We then experiment with the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithm, in which the regularisation jitter is colourised as described by Eqs. (47) and (48). 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)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithm (not shown here) is very similar to the evolution of the RMSE as a function of s for the S(IR)$^{\mathrm{x}}$R algorithm. In the following, when using the coloured regularisation jitter method, $\hat{h}$ is systematically tuned to yield the lowest RMSE score.

Figure 10 shows the evolution of the RMSE as a function of the ensemble size N_e for the S(IR)$^{\mathrm{x}}$R and the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithms. These two variants only differ by the regularisation method. The S(IR)$^{\mathrm{x}}$R algorithm uses white regularisation jitter while the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithm uses coloured regularisation jitter. For small ensembles, the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithm yields higher RMSEs, but for big ensembles, the RMSEs are slightly better when the regularisation jitter is colourised. Depending on the block size, the transition between both regimes happens around $N_e = 32$ to 64 particles. The higher RMSEs of the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$ 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.

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)$^{\mathrm{x}}$SR and the S(IR)$^{\mathrm{x}}$SR$_{\mathrm{a}}$ algorithms. These algorithms only differ from the S(IR)$^{\mathrm{x}}$R and the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$ 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 11 shows the evolution of the RMSE as a

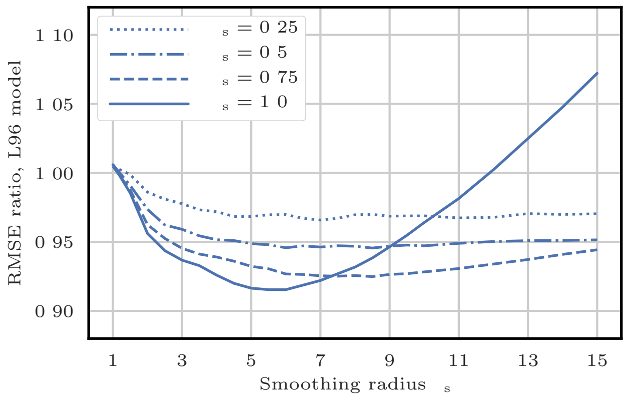

Figure 11. RMSE as a function of the smoothing radius r_s for the S(IR)$^\mathrm{x}$SR algorithms with $N_e = 16$ particles and $N_b = 40$ blocks of size 1 grid point for several values of the smoothing strength α_s. The scores are displayed in units of the RMSE of the S(IR)$^\mathrm{x}$R algorithm with $N_e = 16$ particles and $N_b = 40$ blocks of size 1 grid point.

function of the smoothing radius r_s for the S(IR)$^\mathrm{x}$SR with $N_e = 10$ particles, $N_b = 40$ 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 increases. It reaches a minimum and then increases again. In this example, the optimal smoothing radius r_s was found 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)$^\mathrm{x}$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)$^\mathrm{x}$SR and the S(IR)$^\mathrm{x}$SR$_\mathrm{a}$ algorithms with N_e ranging from 8 to 128 particles, we found 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;
- 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 12 shows the evolution of the RMSE as a function of the ensemble size N_e for the S(IR)$^\mathrm{x}$SR and the S(IR)$^\mathrm{x}$SR$_\mathrm{a}$ algorithms. The S(IR)$^\mathrm{x}$SR algorithm yields systematically lower RMSEs than the standard S(IR)$^\mathrm{x}$R. However, as the ensemble size N_e grows, the gain in RMSE score becomes very small. With $N_e = 512$ 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)$^\mathrm{x}$SR$_\mathrm{a}$ algorithm also yields lower

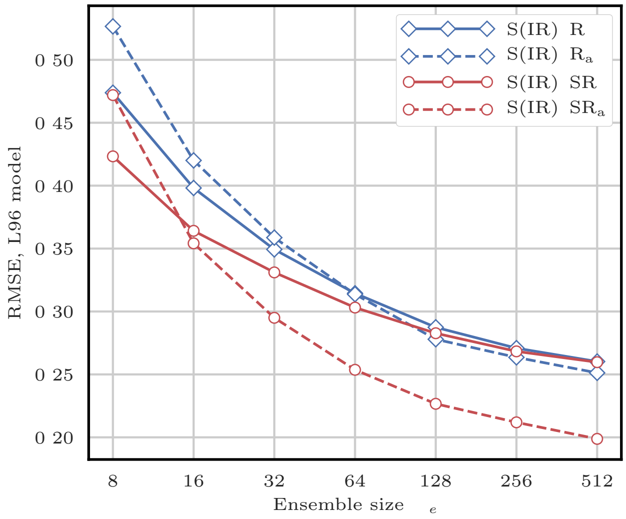

Figure 12. RMSE as a function of the ensemble size N_e for S(IR)$^{\mathrm{x}}$R, the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$, the S(IR)$^{\mathrm{x}}$SR and the S(IR)$^{\mathrm{x}}$SR algorithms.

RMSEs than the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithm. Yet, in this case, the gain in RMSE is still significant for large ensembles and with $N_e = 512$ particles, the RMSEs are even comparable to those of the EnKF.

From these results, we conclude that the smoothing by weights step is an efficient way of reduce afterwards the artificial discontinuities introduced when concatenating 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 the artificial discontinuities of the local resampling step by experimenting the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$R and the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithms. These algorithms only differ from the S(IR)$^{\mathrm{x}}$R and the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$ 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 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 investigated the influence of the parameters N_b and r_{d} by performing extensive tests of the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$R and the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$R$_{\mathrm{a}}$ 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 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

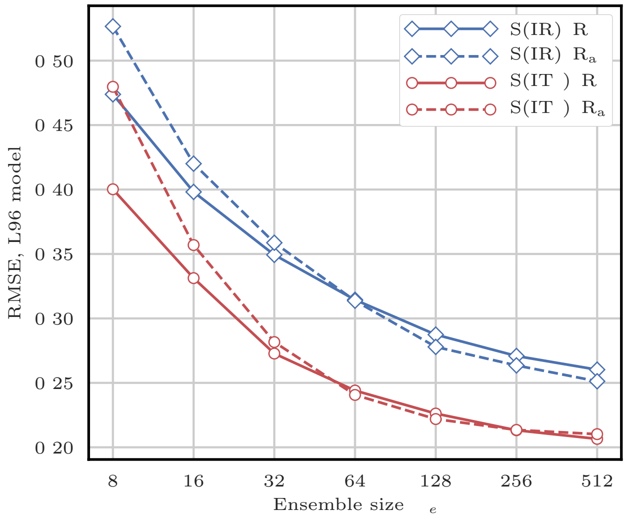

Figure 13. RMSE as a function of the ensemble size N_e for the S(IR)$^{\mathrm{x}}$R, the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$, the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$R and the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithms.

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 $N_b = 20$ blocks of size 2 grid points systematically yields higher RMSEs than using $N_b = 40$ 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 $N_b = 40$ blocks of size 1 grid point. Figure 13 shows the evolution of the RMSE as a function of the ensemble size N_e for the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$R and the S(IT$_{\mathrm{e}}$R)$^{\mathrm{x}}$R$_{\mathrm{a}}$ 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 also performed extensive tests with N_e ranging from 8 to 128 particles on the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$SR and the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$SR$_{\mathrm{a}}$ 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 ($N_e \leq 32$ particles), we obtained RMSEs barely smaller than those obtained with the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$R and the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$R$_{\mathrm{a}}$ 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 the artificial discontinuities inherent to the concatenation of the locally updated particles.

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 the artificial discontinuities of the local resampling step by experimenting the S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R and the S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithms. These algorithms only differ from the S(IR)$^{\mathrm{x}}$R and the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$ 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 S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R and the S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithms, we take $N_b = 40$ 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 investigated the influence of the regularisation bandwidth h by performing extensive tests of the S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R and the S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithms with N_e ranging from 8 to 128 particles (not shown here). For small ensembles ($N_e \leq 16$ particles), optimal values for h were found 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 14 shows the evolution of the RMSE as a function of the ensemble size N_e for the S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R and the S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R$_{\mathrm{a}}$ 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 also performed extensive tests with N_e ranging from 8 to 128 particles on the S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$SR algorithm, in which the anamorphosis resampling method is combined with the smoothing by weights method (not shown here). As for the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$SR and the S(IT$_{\mathrm{e}}$)$^{\mathrm{x}}$SR$_{\mathrm{a}}$ 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 S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R algorithm than for the S(IR)$^{\mathrm{x}}$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 the artificial discontinuities inherent to the concatenation of the locally updated particles.

5.12 Summary

As a summary, Fig. 15 shows the evolution of the RMSE as a function of the ensemble size N_e for the main LPF$^{\mathrm{x}}$s tested in this section. For small ensembles ($N_e \leq 32$ particles), the algorithms using OT-based resampling methods clearly yield lower RMSEs than the other algorithms. For large ensemble ($N_e \geq 128$ particles), combining the smoothing by weights with the coloured noise regularisation jitter methods yields equally good scores as the algorithms using OT. For $N_e = 512$ particles (the largest ensemble size tested in this study), the best RMSE scores obtained with LPF$^{\mathrm{x}}$s become comparable to those of the EnKF.

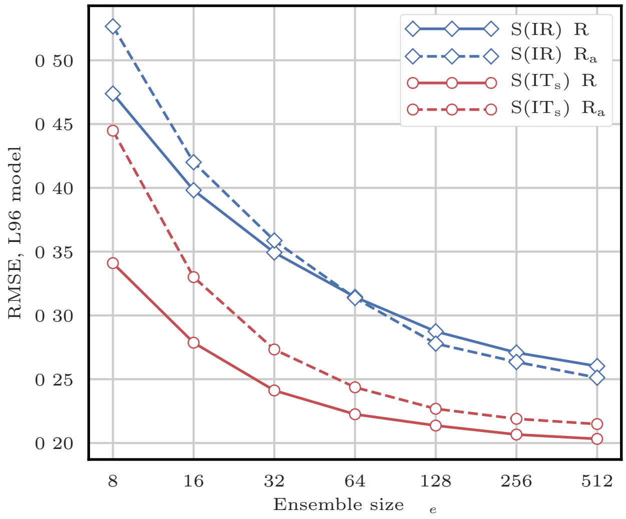

Figure 14. RMSE as a function of the ensemble size N_e for the S(IR)$^{\mathrm{x}}$R, the S(IR)$^{\mathrm{x}}$R$_{\mathrm{a}}$, the S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R and the S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R$_{\mathrm{a}}$ algorithms.

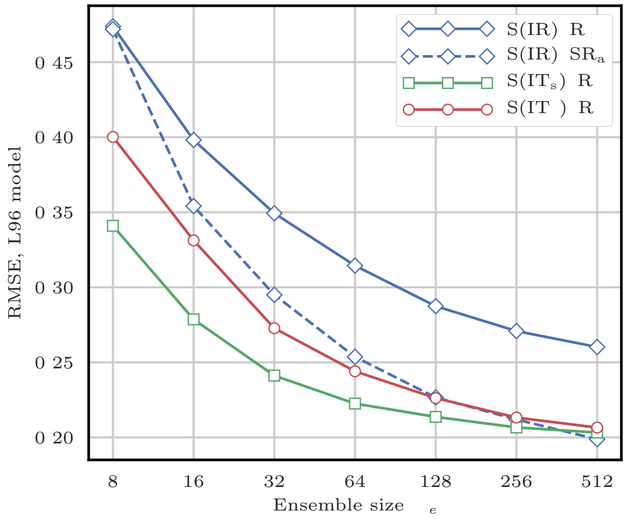

Figure 15. RMSE as a function of the ensemble size N_e for the main LPF$^{\mathrm{x}}$s tested in this section.

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

6 Numerical illustration of the LPF$^{\mathrm{x}}$ algorithms with a barotropic vorticity model

6.1 Model specifications

In this section, we illustrate the performance of LPF$^{\mathrm{x}}$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 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 $9\times10^3\Delta t$ 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 16 shows the evolution of the RMSE as a function of the ensemble size N_e for the (global) ETKF and the local ETKF (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). 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.

With $N_e \leq 1024$ particles, we could not find a set of tuning parameters with which the bootstrap filter or the ETPF yield RMSEs significantly lower than 1.

6.3 Scores for the LPF$^{\mathrm{x}}$ algorithms

In this section, we test the LPF$^{\mathrm{x}}$s with N_e ranging from 8 to 128 particles. For clarity, the naming conventions of the algorithms are the same as in Sect. 5.

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.

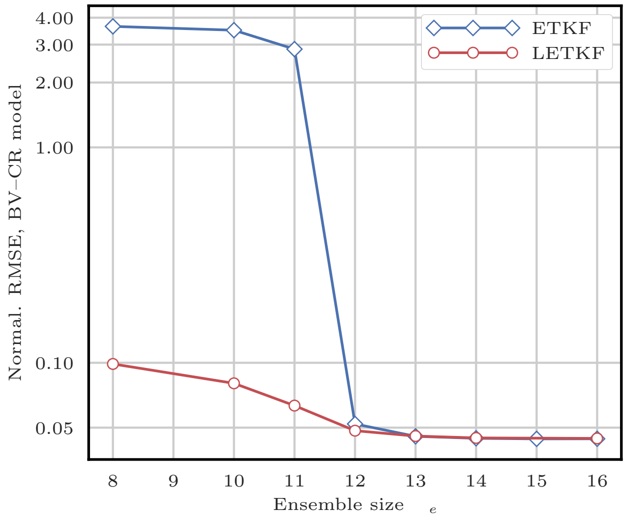

Figure 16. RMSE as a function of the ensemble size N_e for the ETKF and the LETKF. The scores are displayed in units of the observation standard deviation σ.

- For the algorithms using the SU sampling algorithm (i.e. the S(IR)$^{\mathrm{x}}**$ 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$_*$)$^{\mathrm{x}}*$ variants) we only test blocks of shape 1×1 grid point.
- 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 17 shows the evolution of the RMSE as a function of the ensemble size N_e for the main LPF$^{\mathrm{x}}$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.

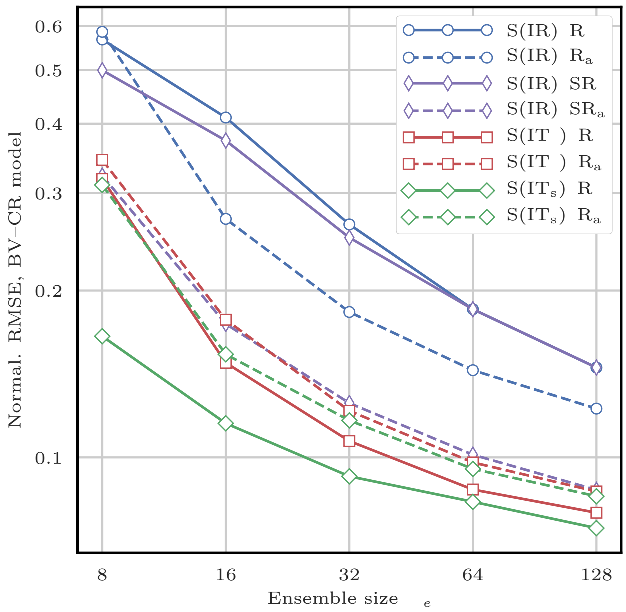

Figure 17. RMSE as a function of the ensemble size N_e for the main LPF$^{\text{x}}$s. The scores are displayed in units of the observation standard deviation σ.

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 the jitter. We observe indeed that the S(IR)$^{\text{x}}$R$_{\text{a}}$ and the S(IR)$^{\text{x}}$SR$_{\text{a}}$ algorithms yield significantly lower RMSEs than the S(IR)$^{\text{x}}$R and the S(IR)$^{\text{x}}$SR algorithms. Yet, the S(IT$_{\text{e}}$)$^{\text{x}}$R$_{\text{a}}$ and the S(IT$_{\text{s}}$)$^{\text{x}}$R$_{\text{a}}$ algorithms are clearly outperformed by both the S(IT$_{\text{e}}$)$^{\text{x}}$R and the S(IT$_{\text{s}}$)$^{\text{x}}$R algorithms in terms of RMSEs. This hints at the fact 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, $N_e \leq 128$ particles. In this configuration, the RMSE scores of LPF$^{\text{x}}$s are not yet comparable with those of the EnKF (shown in Fig. 16).

Finally, it should be noted that for the S(IT$_{\text{e}}$)$^{\text{x}}$R and the S(IT$_{\text{e}}$)$^{\text{x}}$R$_{\text{a}}$ algorithms with $N_e \geq 32$ particles, optimal values for the distance radius r_{d} were found between 3 and 6 grid points (not shown here) contrary to the results obtained with the L96 model, for which $r_{\text{d}} = 1$ grid point could be considered optimal. More generally for all LPF$^{\text{x}}$s, the optimal values for the localisation radius r (not shown here) are significantly larger 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 influence these grid points. In the sequential–observation (SO) localisation formalism, we use a different approach.

Observations sites are assimilated sequentially and assimilating one observation site should only update nearby grid points. LPF algorithms using the SO localisation formalism will be called LPF$^{\mathrm{y}}$ algorithms.

In this section, we set $q \in \{1 \dots N_y\}$ and we describe how the site y_q should be assimilated. 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 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:

$$[\mathbf{C}]_{m,n} = G\left(\frac{d_{m,n}}{r}\right), \tag{50}$$

where $d_{m,n}$ is the distance between 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 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 18 illustrates this UVW partition. We emphasise that both the LG and the UVW state partitions depends on the observation site y_q. They are fundamentally different from the 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 $\mathbf{x}_A$ be the restriction of vector $\mathbf{x}$ to A, i.e. the state variables of $\mathbf{x}$ that are located within A.

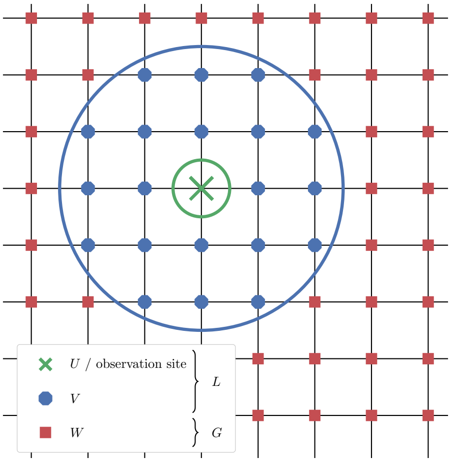

Figure 18. Example of the UVW partition for a two-dimensional space. The observation site y_q lies in the middle. The local parts U and V are circumscribed by the thick green and blue circles and respectively contain 1 and 20 grid points. The global region W contains all remaining grid points. In the case of the LG partition, the local region L gathers all 21 grid points in U and V.

7.2.1 With the LG partition

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

$$p(\mathbf{x}|y_q) = p(\mathbf{x}_L, \mathbf{x}_G|y_q) = p(\mathbf{x}_L|\mathbf{x}_G, y_q)\,p(\mathbf{x}_G|y_q)\,. \tag{51}$$

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

905 $$p(\mathbf{x}_G|y_q) = p(\mathbf{x}_G)\,, \tag{52}$$

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

$$\begin{aligned} p(\mathbf{x}_L|\mathbf{x}_G, y_q) &= \frac{p(y_q|\mathbf{x}_G, \mathbf{x}_L)\,p(\mathbf{x}_G, \mathbf{x}_L)}{p(\mathbf{x}_G, y_q)}, \\ &= \frac{p(y_q|\mathbf{x}_L)\,p(\mathbf{x}_G, \mathbf{x}_L)}{p(\mathbf{x}_G, y_q)}. \end{aligned} \tag{53}$$

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

Algorithm 5 Single analysis step for a generic LPF$^{\mathrm{y}}$ algorithm using the LG partition

```
Require: Prior ensemble $\mathbf{x}^i$, $i = 1 \dots N_e$ and observation site $\mathbf{y}_q$
Build the $LG$ partition as described in Sect. 7.1
for $i = 1$ to $N_e$ do
    Do not update $\mathbf{x}^i_G$
    Update $\mathbf{x}^i_L$ conditionally to $y_q$ and $\mathbf{x}^i_G$ as stated by Eq. (53)
end for
return Updated ensemble $\mathbf{x}^i$, $i = 1 \dots N_e$
```

7.2.2 With the UVW partition

With the UVW partition, the conditional density is factored as

$$\begin{aligned} p(\mathbf{x}|y_q) &= p(\mathbf{x}_U, \mathbf{x}_V, \mathbf{x}_W|y_q), \\ &= \frac{p(\mathbf{x}_U, \mathbf{x}_V, \mathbf{x}_W, y_q)}{p(y_q)}, \\ &= \frac{p(y_q|\mathbf{x})\, p(\mathbf{x}_V|\mathbf{x}_U, \mathbf{x}_W)\, p(\mathbf{x}_U, \mathbf{x}_W)}{p(y_q)}, \\ &= \frac{p(y_q|\mathbf{x}_U)\, p(\mathbf{x}_V|\mathbf{x}_U, \mathbf{x}_W)\, p(\mathbf{x}_U, \mathbf{x}_W)}{p(y_q)}. \end{aligned} \tag{54}$$

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

$$p(\mathbf{x}_U, \mathbf{x}_W) = p(\mathbf{x}_U)\, p(\mathbf{x}_W). \tag{55}$$

Finally, the conditional density is

$$p(\mathbf{x}|y_q) = p(\mathbf{x}_U|y_q)\, p(\mathbf{x}_V|\mathbf{x}_U, \mathbf{x}_W)\, p(\mathbf{x}_W). \tag{56}$$

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

7.2.3 The partition and the particle filter

The SO formalism is 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 $p(\mathbf{x}_G, \mathbf{x}_L)$ and $p(\mathbf{x}_V|\mathbf{x}_U, \mathbf{x}_W)$ in Eqs. (53) and (54) 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.

Algorithm 6 Single analysis step for a generic LPF$^{\text{y}}$ algorithm using the UVW partition

```
Require: Prior ensemble $\mathbf{x}^i$, $i = 1 \dots N_e$ and observation site $\mathbf{y}_q$
Build the $UVW$ partition as described in Sect. 7.1
for $i = 1$ to $N_e$ do
    Do not update $\mathbf{x}_W^i$
    Update $\mathbf{x}_U^i$ conditionally to $y_q$
    Update $\mathbf{x}_V^i$ conditionally to $\mathbf{x}_W^i$ and (the updated) $\mathbf{x}_U^i$ as stated by Eq. (54)
end for
return Updated ensemble $\mathbf{x}^i$, $i = 1 \dots N_e$
```

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:

$$p(\mathbf{x}|y) = p(x_1|y)\,p(x_2|x_1)\dots p(x_{n+1}|x_n,\dots,x_1)\dots \tag{57}$$

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.

7.3 Poterjoy's local particle filter

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.

7.3.1 Step 1: particle filter resampling

The global unnormalised importance weights are given by

$$w^i = p\left(y_q|\mathbf{x}^i\right). \tag{58}$$

Using theses weights, 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) and the update is propagated to all grid points n as

$$x_n^i = \overline{x}_n + \omega_n^{\mathrm{a}} \left(x_n^{\phi(i)} - \overline{x}_n \right) + \omega_n^{\mathrm{f}} \left(x_n^i - \overline{x}_n \right), \tag{59}$$

where $\overline{x}_n$ is the ensemble mean at 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 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. 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$^{\mathrm{x}}$ algorithms. Therefore, we did not include it in our presentation of Poterjoy's LPF.

Algorithm 7 Single analysis step for Poterjoy's LPF

```
Require: Prior ensemble $\mathbf{x}^i$, $i = 1 \ldots N_e$ and observation site $\mathbf{y}_q$
Compute the analysis weights using Eq. (58)
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 $N_e$ do
        Update $x_n^i$ using Eq. (59)
    end for
end for
return Updated ensemble $\mathbf{x}^i$, $i = 1 \ldots N_e$
```

7.4 An algorithm inspired from the EnKPF

The 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.

7.4.1 Preliminary: the covariance matrix

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

$$\mathbf{\Sigma}^{\mathrm{f}} = \mathrm{cov}\left(\mathbf{x}\right). \tag{60}$$

In a localisation context, it seems reasonable to use a tapered representation of the covariance, in such a way that the covariance matrix $\mathbf{\Sigma}$ of use will be

$$\mathbf{\Sigma} = \mathbf{C} \circ \mathbf{\Sigma}^{\mathrm{f}}, \tag{61}$$

where $\mathbf{C}$ is the valid tapering matrix mentioned in section 7.1 and $\circ$ means the Schur product for matrices.

7.4.2 Step 1: update in the U region

Each ensemble member is weighted by the global unnormalised importance weights Eq. (58). Using these weights, we resample the ensemble in the U region and compute the update $\Delta\mathbf{x}_U^i$. For this resampling step, any resampling algorithm can be used:

- an adjustment-minimising resampling algorithm can be used to minimise the number of updates $\Delta\mathbf{x}_U^i$ 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 the LPF$^{\mathrm{x}}$ 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

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

$$\Delta\mathbf{x}_V^i = \mathbf{\Sigma}_{VU}\mathbf{\Sigma}_U^{-1}\Delta\mathbf{x}_U^i, \tag{62}$$

where $\mathbf{\Sigma}_{VU}$ and $\mathbf{\Sigma}_U$ are submatrices of $\mathbf{\Sigma}$. The full derivation of Eq. (62) is available in Robert and Künsch (2017). Note that $\mathbf{\Sigma}$ is a $N_x \times N_x$ matrix but only the submatrices $\mathbf{\Sigma}_{VU}$ and $\mathbf{\Sigma}_U$ need to be computed.

Finally, the LEF$_{\mathrm{RK}}$ algorithm is summarised by Algorithm 8 in a generic context, with any resampling algorithm.

7.5 Pros and cons of the LPF$^{\mathrm{y}}$ algorithms

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$^{\mathrm{x}}$s, which is why the partition was introduced in the first place.

Algorithm 8 Single analysis step for a generic LEF$_{\mathrm{RK}}$ algorithm

```
Require: Prior ensemble $\mathbf{x}^i$, $i = 1 \dots N_e$, observation site $\mathbf{y}_q$
1: Build the $UVW$ partition as described in Sect. 7.1
2: Compute the prior covariance submatrices $\mathbf{\Sigma}_{VU}$ and $\mathbf{\Sigma}_U$
3: Compute the analysis weights using Eq. (58)
4: Resample the ensemble on region $U$
5: Compute the associated updates $\Delta\mathbf{x}_U^i$, $i = 1 \dots N_e$
6: for $i = 1$ to $N_e$ do
7:     Compute the update $\Delta\mathbf{x}_V^i$ using Eq. (62)
8:     Apply the update $\Delta\mathbf{x}_V^i$ on region $V$
9: end for
10: return Updated ensemble $\mathbf{x}^i$, $i = 1 \dots N_e$
```

However, the LPF$^{\mathrm{y}}$s are by construction non parallel: observation sites are assimilated sequentially. 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 are disjoint. That is to say, one can assimilate in parallel several observation 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 may overlap, and parallelisation is not possible.

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. Indeed, using order-2 moments to propagate the update introduces bias in the analysis. On the other hand, order-2 methods are in general less sensitive to the curse of dimensionality. Therefore, we expect the LEF$_{\mathrm{RK}}$ algorithm to be able to handle larger values for the localisation radius r than the LPF$^{\mathrm{x}}$s.

7.6 Gathering observation sites into blocks

The LPF$^{\mathrm{y}}$s can be extended to the case where observation sites are compounded into small blocks as follows:

- the unnormalised importance weights Eq. (58) are modified such that they account for all observation sites inside the block;
- any distance that needed to be computed relatively to the observation site y_q (for example for the ω^* weights for Poterjoy's LPF) is now computed relatively to the block center;
- in the LEF$_{\mathrm{RK}}$ algorithm, the UVW partition is modified: the U region now covers all grid points that directly influence all observation sites inside the block.

Table 2. Nomenclature conventions for the $\mathrm{S}(\mathrm{I}\alpha\mathrm{P}_{\beta})^{\mathrm{y}}\gamma$ algorithms

α	Local resampling algorithm
R	adjustment-minimising SU sampling algorithm
$\mathrm{T_e}$	optimal transport in ensemble space
$\mathrm{T_s}$	optimal transport in state space

β	Propagation method
P	Poterjoy's LPF (Algorithm 7)
RK	$\mathrm{LEF_{RK}}$ propagation (Algorithm 8)

γ	Regularisation method
R	white noise method
$\mathrm{R_a}$	coloured noise method

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 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$^{\mathrm{y}}$ algorithms

8.1 Experimental setup

In this section, we illustrate the performance of the LPF$^{\mathrm{y}}$s using twin simulations of the L96 and the BV models. The model specifications for this test series are the same as for the LPF$^{\mathrm{x}}$ 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$^{\mathrm{y}}$ algorithms are named $\mathrm{S}(\mathrm{I}{*}\mathrm{P}_{*})^{\mathrm{y}}\mathrm{R}_{*}$ — 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 2.

8.1.1 Regularisation jitter

For the same reasons as with LPF$^{\mathrm{x}}$s, jittering the LPF$^{\mathrm{y}}$s is necessary not to experience a fast collapse. As we eventually did for the LPF$^{\mathrm{x}}$s, 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. (43) 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 S(IRP$_\mathrm{P}$)$^\mathrm{y}$R algorithm and its variant

With the regularisation method described in Sect. 8.1.1, the S(IRP$_\mathrm{P}$)$^\mathrm{y}$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 LPF$^\mathrm{x}$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.

As mentioned in Sect. 7.3.2, the original algorithm designed by Poterjoy (2016) included another tuning parameter, the weights inflation, which serves the same purpose as the regularisation jitter. Based on extensive tests in the L96 model with 8 to 128 particles, we found that using weights inflation instead of regularisation jitter always yields higher RMSEs. Therefore, we did not include weight inflation in the S(IRP$_\mathrm{P}$)$^\mathrm{y}$R algorithm.

In the S(IRP$_\mathrm{P}$)$^\mathrm{y}$R$_\mathrm{a}$ 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$_\mathrm{RK}$)$^\mathrm{y}$R algorithm and its variants

With the regularisation method described in Sect. 8.1.1, the S(IRP$_\mathrm{RK}$)$^\mathrm{y}$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. (61);
- 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 variables in the region U are computed in the same way as for LPF$^\mathrm{x}$ 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$^\mathrm{y}$s with the L96 model is shown in Fig. 19. The RMSEs obtained with the S(IRP$_\mathrm{P}$)$^\mathrm{y}$R algorithm are comparable to those obtained with the S(IR)$^\mathrm{x}$R algorithm. When

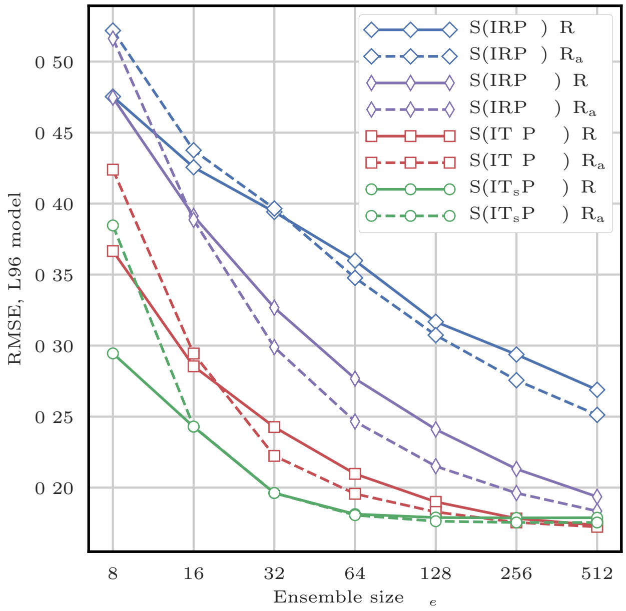

Figure 19. RMSE as a function of the ensemble size N_e for the main $\mathrm{LPF^y}$s.

using the $\mathrm{LEF_{RK}}$ propagation method, the RMSEs are, as expected, significantly lower. Thanks to the order-2 truncation, the algorithms are less sensitive to the curse of dimensionality than $\mathrm{LPF^x}$s: optimal values of the localisation radius r are significantly larger and less regularisation jitter s is required. As for the $\mathrm{LPF^x}$s, combining the $\mathrm{LEF_{RK}}$ propagation with OT-based resampling methods (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 $\mathrm{S(IT_sP_{RK})^yR}$ algorithm), the scores are even lower than those obtained with the reference EnKF implementation. 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 $\mathrm{LPF^x}$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 $\mathrm{LPF^y}$s with the BV model is shown in Fig. 20. Most of the conclusions drawn with the L96 model remain true with the BV model. However in this case, as the ensemble size N_e grows the RMSE decreases significantly more slowly for the $\mathrm{S(IRP_P)^yR}$ and the $\mathrm{S(IRP_P)^yR_a}$ algorithms than for the other algorithms. Finally, with an ensemble size $N_e \geq 64$ particles, the $\mathrm{S(IT_sP_{RK})^yR}$ and the $\mathrm{S(IT_sP_{RK})^yR_a}$ algorithms yield RMSEs almost equivalent to those of the reference LETKF implementation.

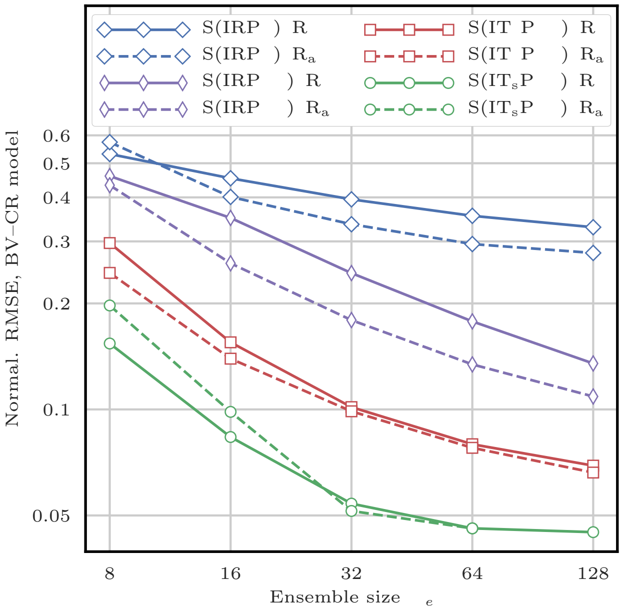

Figure 20. RMSE as a function of the ensemble size N_e for the main LPF^{y}s. The scores are displayed in units of the observation standard deviation σ.

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 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 3. 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:

$$\mathbf{x}_0^i = \mathbf{x}_0 + 0.5 \times \mathbf{u} + \mathbf{u}^i, \quad i = 1 \ldots N_e, \tag{63}$$

with $\mathbf{u}$ and the $\mathbf{u}^i$ 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.

9.3 RMSE time series

Figure 21 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 3. 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 observation network, 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 truncation (LETKF and $\mathrm{S(I{*}P_{RK})^{y}R}$ 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 $\mathrm{S(IR)^{x}R}$ and the $\mathrm{S(IRP_{P})^{x}R}$ 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, using OT-based local resampling methods yields significantly lower RMSEs. The RMSEs of the $\mathrm{S(IT_{s})^{x}R}$ algorithm, though not as good as those of the LETKF algorithm, show that the true state is reconstructed with an acceptable accuracy. The RMSEs of the $\mathrm{S(IT_{s}P_{RK})^{y}R}$ 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 $\mathrm{LPF^{x}}$s are faster than both the N_x local analyses of the LETKF and the N_y sequential assimilations of the $\mathrm{LPF^{y}}$s (Poterjoy's LPF and the $\mathrm{LEF_{RK}}$ algorithm). The $\mathrm{LEF_{RK}}$ algorithm is slower because of the linear algebra involved in the propagation method. Poterjoy's algorithm is slower because of the computation of the ω^{*} weights. The LETKF is slower because of the matrices inversion in the ensemble space. The difference between the $\mathrm{LPF^{x}}$s and the $\mathrm{LPF^{y}}$s is even more visible on our 24-core platform. The $\mathrm{LPF^{y}}$s are not parallel, that is why they are more than 70 times slower than the $\mathrm{LPF^{x}}$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 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 particles. Adequate solutions to these issues are not obvious, witness the few but unsimilar 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.

Table 3. Characteristics of the algorithms tested with the BV model in the HR configuration (Fig. 21). The LPF$^{\mathrm{x}}$s use zero integration jitter ($q = 0$) and $N_b = N_x$ blocks of size 1 grid point. The LPF$^{\mathrm{y}}$s also use zero integration jitter ($q = 0$). For the LETKF, the optimal multiplicative inflation is reported in the regularisation jitter column. 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$_{\mathrm{P}}$)$^{\mathrm{y}}$R	0.03	0.70	—	0.90	122.18	—
S(IR)$^{\mathrm{x}}$R	0.02	0.55	—	0.78	7.58*	0.54 (×14.04)
S(IRP$_{\mathrm{RK}}$)$^{\mathrm{y}}$R	0.07	0.25	—	0.46	52.97	—
S(IT$_{\mathrm{s}}$)$^{\mathrm{x}}$R	0.08	0.11	$h = 3$	0.33	13.94*	0.86 (×16.21)
S(IT$_{\mathrm{s}}$P$_{\mathrm{RK}}$)$^{\mathrm{y}}$R	0.20	0.01	$h = 1$	0.13	64.79	—
LETKF	0.35	1.04	—	0.10	103.90*	5.09 (×20.41)

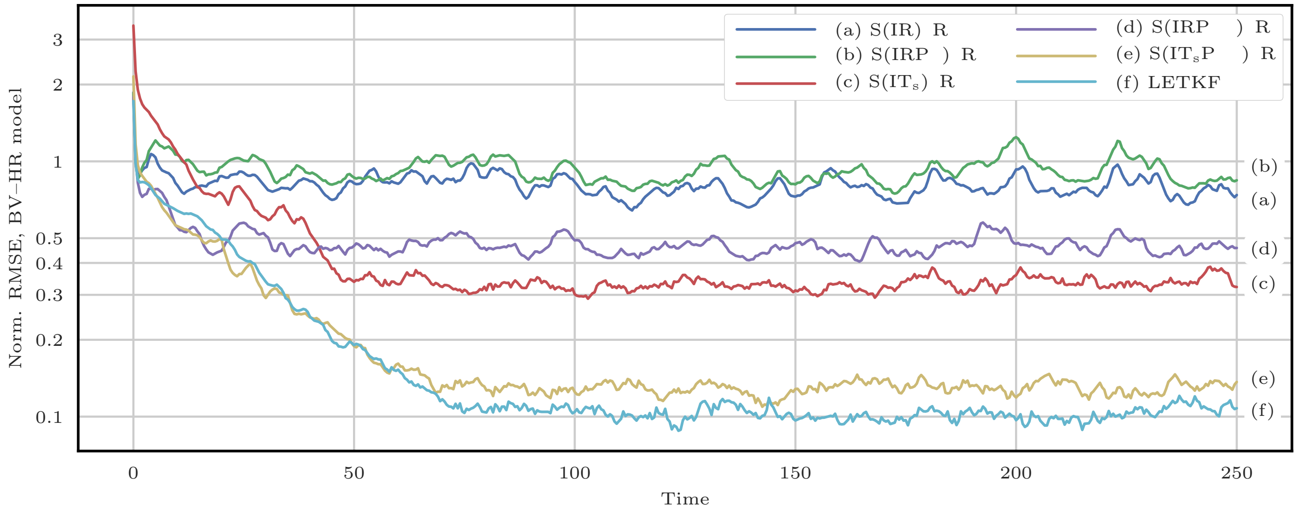

Figure 21. Instantaneous analysis RMSE for the selection of algorithms detailed in Table 3. The scores are displayed in units of the observation standard deviation σ.

With the LPF$^{\mathrm{x}}$ methods, the analysis is localised by allowing the analysis weights to vary over the state variables. We have shown that this leads to 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 locally updated particles. The quality of the updated ensemble directly depends on the regularity of the local resampling. This is related to the potential discontinuities in the concatenated particles. Therefore we have presented practical methods to improve the design of this local resampling step by reducing the discontinuities.

In the LPF$^{\mathrm{y}}$ 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 issues inherent to LPF$^{\mathrm{x}}$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 are by construction assimilated sequentially, which is a great disadvantage when the number of observation sites 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 S(IR)$^{\mathrm{x}}$R algorithm. In particular, with the BV model, the scores of the S(IR)$^{\mathrm{x}}$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 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. We have shown that using OT-based local resampling methods always yields important gains in RMSE scores. For the LPF$^{\mathrm{x}}$s, it is a consequence of the minimisation of the discontinuities introduced in the local resampling step. For the LPF$^{\mathrm{y}}$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 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

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

whose transition distribution is

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

and whose observation distribution is

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

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

A2 Generic model with Gaussian additive noise

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

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

and the observation distribution is:

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

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:

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

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 $F = 8$. The observations are given by

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

and the time interval between consecutive observations is $\Delta t = 0.05$ time unit, which represents 6 h of real time.

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. 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 (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, 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.

Nonlin. Processes Geophys. Discuss., https://doi.org/10.5194/npg-2018-15
Manuscript under review for journal Nonlin. Processes Geophys.
Discussion started: 5 March 2018

A4.1 Coarse resolution configuration

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

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

$$\xi = 1 \times 10^{-2}, \tag{A12}$$

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

the deterministic forcing is given by

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

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 discretisation is enough to allow a reasonable description of a few (typically five to ten) vortices inside the domain. The time 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. 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 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 $1 \times 10^4 \Delta t$. 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, but the number of vertices is approximately the same with much fewer details.

A4.2 High resolution configuration

For the high resolution configuration, the physical parameters are

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

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

$$\nu = 1 \times 10^{-6}; \tag{A21}$$

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 (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. At every cycle k, the observation at site $(q_1, q_2) \in \{1 \dots P/2\}^2$ is given by

$$y_{q_1,q_2} = x_{2q_1-1,2q_2-1} + v_{q_1,q_2}, \tag{A25}$$

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

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:

$$W = \sum_{i=1}^{N_e} w^i = \sum_{i=1}^{N_e} p\left(y_q|\mathbf{x}^i\right), \tag{B1}$$

$$c_n = \frac{\alpha N_e \left(1 - G\left(\frac{d_{q,n}}{r}\right)\right)}{W G\left(\frac{d_{q,n}}{r}\right)}, \tag{B2}$$

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

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

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 observation site q and grid point n, r is the localisation radius, $\overline{x}_n$ is the mean and σ_n the standard deviation of the weighted ensemble $\left\{\left(x_n^i, w^i\right), i = 1 \ldots N_e\right\}$. The particles are then updated using Eq. (59).

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 one update equation in Algorithm 1 of Poterjoy (2016) is incorrect (last equation on page 66).

Appendix C: 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 C1. 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. C1.

The histogram of the EnKF is quite flat in the middle, its edges reflect a small over dispersion. The histogram of the tuned S(IR)$^{\mathrm{x}}$R algorithm is characterised by a large hump, showing that the ensemble is over dispersive. 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 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)$^{\mathrm{x}}$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) and the localisation radius smaller (to avoid the filter divergence). Of course, the RMSE score for the untuned

Table C1. Rank histograms computed with the L96 model in the standard configuration (see Appendix C). 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. C1.

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

$\mathrm{S(IR)^xR}$ algorithm is higher than for its tuned version. Similar conclusions can be found with the histograms of the tuned and untuned $\mathrm{S(IT_s)^xR}$ algorithm. Note that in this case the histograms are significantly flatter than with the $\mathrm{S(IR)^xR}$ algorithm. 1290 Finally, the histogram of the (tuned) $\mathrm{S(IT_sP_{RK})^yR}$ is remarkably flat.

In summary, the rank histograms of the LPFs are in general rather flat. The ensemble are more or less over dispersive, 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.

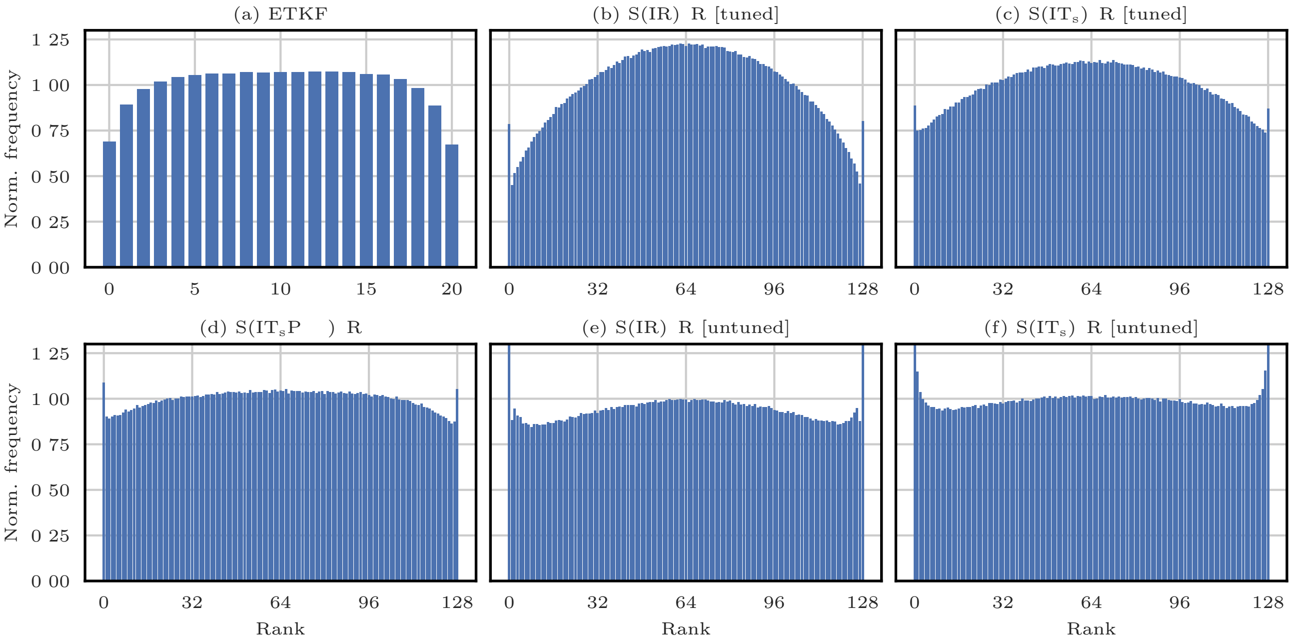

Figure C1. Rank histograms for the selection of algorithms detailed in Table C1. The frequency is normalised by $N_e + 1$ (the number of bins).

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.

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.

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.