



Implicit particle filtering for models with partial noise, and an application to geomagnetic data assimilation

M. Morzfeld¹ and A. J. Chorin^{1,2}

¹Lawrence Berkeley National Laboratory, Berkeley, CA, USA

²Department of Mathematics, University of California, Berkeley, CA, USA

Correspondence to: M. Morzfeld (mmo@math.lbl.gov)

Received: 27 September 2011 – Revised: 22 May 2012 – Accepted: 22 May 2012 – Published: 19 June 2012

Abstract. Implicit particle filtering is a sequential Monte Carlo method for data assimilation, designed to keep the number of particles manageable by focussing attention on regions of large probability. These regions are found by minimizing, for each particle, a scalar function F of the state variables. Some previous implementations of the implicit filter rely on finding the Hessians of these functions. The calculation of the Hessians can be cumbersome if the state dimension is large or if the underlying physics are such that derivatives of F are difficult to calculate, as happens in many geophysical applications, in particular in models with partial noise, i.e. with a singular state covariance matrix. Examples of models with partial noise include models where uncertain dynamic equations are supplemented by conservation laws with zero uncertainty, or with higher order (in time) stochastic partial differential equations (PDE) or with PDEs driven by spatially smooth noise processes. We make the implicit particle filter applicable to such situations by combining gradient descent minimization with random maps and show that the filter is efficient, accurate and reliable because it operates in a subspace of the state space. As an example, we consider a system of nonlinear stochastic PDEs that is of importance in geomagnetic data assimilation.

1 Introduction

The task in data assimilation is to use available data to update the forecast of a numerical model. The numerical model is typically given by a discretization of a stochastic differential equation (SDE)

$$\mathbf{x}^{n+1} = \mathbf{R}(\mathbf{x}^n, t^n) + \mathbf{G}(\mathbf{x}^n, t^n) \Delta \mathbf{W}^{n+1}, \quad (1)$$

where \mathbf{x} is an m -dimensional vector, called the state, t^n , $n = 0, 1, 2, \dots$, is a sequence of times, \mathbf{R} is an m -dimensional vector function, \mathbf{G} is an $m \times m$ matrix and $\Delta \mathbf{W}$ is an m -dimensional vector, whose elements are independent standard normal variates. The random vectors $\mathbf{G}(\mathbf{x}^n, t^n) \Delta \mathbf{W}^{n+1}$ represent the uncertainty in the system, however even for $\mathbf{G} = \mathbf{0}$ the state \mathbf{x}^n may be random for any n because the initial state \mathbf{x}^0 can be random. The data

$$\mathbf{z}^l = \mathbf{h}(\mathbf{x}^{q(l)}, t^{q(l)}) + \mathbf{Q}(\mathbf{x}^{q(l)}, t^{q(l)}) \mathbf{V}^l \quad (2)$$

are collected at times $t^{q(l)}$, $l = 1, 2, \dots$; for simplicity, we assume that the data are collected at a subset of the model steps, i.e. $q(l) = rl$, with $r \geq 1$ being a constant. In the above equation, \mathbf{z} is a k -dimensional vector ($k \leq m$), \mathbf{h} is a k -dimensional vector function, \mathbf{V} is a k -dimensional vector whose components are independent standard normal variates, and \mathbf{Q} is a $k \times k$ matrix. Throughout this paper, we will write $\mathbf{x}^{0:n}$ for the sequence of vectors $\mathbf{x}^0, \dots, \mathbf{x}^n$.

Data assimilation is necessary in many areas of science and engineering and is essential in geophysics, for example in oceanography, meteorology, geomagnetism or atmospheric chemistry (see e.g. the reviews Miller et al., 1994; Ide et al., 1997; Miller et al., 1999; van Leeuwen, 2009; Bocquet et al., 2010; Fournier et al., 2010). The assimilation of data in geophysics is often difficult because of the complicated underlying dynamics, which lead to a large state dimension m and a nonlinear function \mathbf{R} in Eq. (1).

If the model (1) as well as \mathbf{h} in Eq. (2) are linear in \mathbf{x} and if, in addition, the matrices \mathbf{G} and \mathbf{Q} are independent of \mathbf{x} , and $\Delta \mathbf{W}^n$ and \mathbf{V}^l in Eqs. (1) and (2) are Gaussian and independent, and if the initial state \mathbf{x}^0 is Gaussian, then the probability density function (pdf) of the state \mathbf{x}^n is Gaussian for any n

and can be characterized in full by its mean and covariance. The Kalman filter (KF) sequentially computes the mean of the model (1), conditioned on the observations and, thus, provides the best linear unbiased estimate of the state (Kalman, 1960; Kalman and Bucy, 1961; Gelb, 1974; Stengel, 1994). The ensemble Kalman filter (EnKF) is a Monte Carlo approximation of the Kalman filter, and can be obtained by replacing the state covariance matrix by the sample covariance matrix in the Kalman formalism (see Evensen, 2007). The state covariance is the covariance matrix of the pdf of the current state conditioned on the previous state which we calculate from the model (1) to be

$$p(\mathbf{x}^{n+1} | \mathbf{x}^n) \sim \mathcal{N}(\mathbf{R}(\mathbf{x}^n, t^n), \mathbf{G}(\mathbf{x}^n, t^n)\mathbf{G}(\mathbf{x}^n, t^n)^T), \quad (3)$$

where $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ denotes a Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix Σ . To streamline the notation we write for the state covariance:

$$\Sigma_x^n = \mathbf{G}(\mathbf{x}^n, t^n)\mathbf{G}(\mathbf{x}^n, t^n)^T, \quad (4)$$

where T denotes a transpose. In the EnKF, the sample covariance matrix is computed from an “ensemble”, by running the model (1) for different realizations of the noise process $\Delta \mathbf{W}$. The Monte Carlo approach avoids the computationally expensive step of updating the state covariance in the Kalman formalism. Both KF and EnKF have extensions to nonlinear, non-Gaussian models, however they rely on linearity and Gaussianity approximations (Julier and Uhlmann, 1997).

Variational methods (Zupanski, 1997; Tremolet, 2006; Talagrand, 1997; Courtier, 1997; Courtier et al., 1994; Bennet et al., 1993; Talagrand and Courtier, 1987) aim at assimilating the observations within a given time window by computing the state trajectory of maximum probability. This state trajectory is computed by minimizing a suitable cost function. In particular, 3-D-Var methods assimilate one observation at a time (Talagrand, 1997). Strong constraint 4-D-Var determines the most likely initial state \mathbf{x}_0 given the data z_1, z_2, \dots, z_l , a “perfect” model, i.e. $\mathbf{G} = \mathbf{0}$, and a Gaussian initial uncertainty, i.e. $\mathbf{x}^0 \sim \mathcal{N}(\boldsymbol{\mu}^0, \Sigma^0)$ (Talagrand, 1997; Courtier, 1997; Courtier et al., 1994; Talagrand and Courtier, 1987). Uncertain models with $\mathbf{G} \neq \mathbf{0}$ are tackled with a weak constraint 4-D-Var approach (Zupanski, 1997; Tremolet, 2006; Bennet et al., 1993). Many variational methods use an adjoint minimization method and are very efficient. To further speed up the computations, many practical implementations of variational methods, e.g. incremental 4-D-Var, use linearizations and Gaussian approximations.

For the remainder of this paper, we focus on sequential Monte Carlo (SMC) methods for data assimilation, called particle filters (Doucet et al., 2001; Weare, 2009; Moral, 1998; van Leeuwen, 2010; Moral, 2004; Arulampalam et al., 2002; Doucet et al., 2000; Chorin and Tu, 2009; Chorin et al., 2010; Gordon et al., 1993; Morzfeld et al., 2012). Particle filters do not rely upon linearity or Gaussianity assumptions and approximate the pdf of the state given the observations,

$p(\mathbf{x}^{0:q(l)} | \mathbf{z}^{1:l})$, by SMC. The state estimate is a statistic (e.g. the mean, median, mode etc.) of this pdf. Most particle filters rely on the recursive relation

$$p(\mathbf{x}^{0:q(l+1)} | \mathbf{z}^{1:l+1}) \propto p(\mathbf{x}^{0:q(l)} | \mathbf{z}^{1:l}) \times p(\mathbf{z}^{l+1} | \mathbf{x}^{q(l+1)})p(\mathbf{x}^{q(l)+1:q(l+1)} | \mathbf{x}^{q(l)}). \quad (5)$$

In the above equation $p(\mathbf{x}^{0:q(l+1)} | \mathbf{z}^{1:l+1})$ is the pdf of the state trajectory up to time $t^{q(l+1)}$, given all available observations up to time $t^{q(l+1)}$ and is called the target density; $p(\mathbf{z}^{l+1} | \mathbf{x}^{q(l+1)})$ is the probability density of the current observation given the current state and can be obtained from Eq. (2)

$$p(\mathbf{z}^{l+1} | \mathbf{x}^{q(l+1)}) \sim \mathcal{N}(\mathbf{h}(\mathbf{x}^{q(l)}, t^{q(l)}), \Sigma_z^n), \quad (6)$$

with

$$\Sigma_z^n = \mathbf{Q}(\mathbf{x}^n, t^n)\mathbf{Q}(\mathbf{x}^n, t^n)^T. \quad (7)$$

The pdf $p(\mathbf{x}^{q(l)+1:q(l+1)} | \mathbf{x}^{q(l)})$ is the density of the state trajectory from the previous assimilation step to the current observation, conditioned on the state at the previous assimilation step, and is determined by the model (1).

A standard version of the sequential importance sampling with resampling (SIR) particle filter (also called bootstrap filter, see e.g. Doucet et al., 2001) generates, at each step, samples from $p(\mathbf{x}^{q(l)+1:q(l+1)} | \mathbf{x}^{q(l)})$ (the prior density) by running the model. These samples (particles) are weighted by the observations with weights $w \propto p(\mathbf{z}^{l+1} | \mathbf{x}^{q(l+1)})$, to yield a posterior density that approximates the target density $p(\mathbf{x}^{0:q(l+1)} | \mathbf{z}^{1:l+1})$. One then removes particles with a small weight by “resampling” (see e.g. Arulampalam et al., 2002 for resampling algorithms) and repeats the procedure when the next observation becomes available. This SIR filter is straightforward to implement, however the catch is that many particles have small weights because the particles are generated without using information from the data. If many particles have a small weight, the approximation of the target density is poor and the number of particles required for a good approximation of the target density can grow catastrophically with the dimension of the state (Snyder et al., 2008; Bickel et al., 2008). Various methods, e.g. different prior densities and weighting schemes (see e.g. Doucet et al., 2001; van Leeuwen, 2010, 2009; Weare, 2009), have been invented to ameliorate this problem, but a rigorous analysis of how the number of particles scales with the dimension of the state space has not been reported for any of these methods.

The basic idea of implicit particle filters (Chorin and Tu, 2009; Chorin et al., 2010; Morzfeld et al., 2012) is to use the available observations to find regions of high probability in the target density and look for samples within this region. This implicit sampling strategy generates a thin particle beam within the high probability domain and, thus, keeps the number of particles required manageable, even if the state dimension is large. The focussing of particles is achieved by finding

the regions of high probability through a particle-by-particle minimization and then setting up an underdetermined algebraic equation that depends on the model (1) as well as on the data (2), and whose solution generates a high probability sample of the target density. We review the implicit filter in the next section, and it will become evident that the construction assumes that the state covariance Σ_x^n in Eq. (4) is non-singular. This condition is often not satisfied. If, for example, one wants to assimilate data into a stochastic partial differential equation (SPDE) driven by spatially smooth noise, then the continuous-time noise process can be represented by a series with rapidly decaying coefficients, leading to a singular or ill-conditioned state covariance Σ_x^n in discrete time and space (see Sects. 3.1 and 4, as well as Lord and Rougemont, 2004; Chueshov, 2000; Jentzen and Kloeden, 2009). A second important class of models with partial noise are uncertain dynamic equations supplemented by conservation laws (e.g. conservation of mass) with zero uncertainty. Such models often appear in data assimilation for fluid dynamics problems (Kurapov et al., 2007). A similar situation occurs when second-order (in time) equations are formulated as systems of first-order equations, e.g. in robotics.

The purpose of the present paper is two-fold. First, in Sect. 2, we present a new implementation of the implicit particle filter. Most previous implementations of the implicit filter (Chorin et al., 2010; Morzfeld et al., 2012) rely in one way or another on finding the Hessians of scalar functions of the state variables. For systems with very large state vectors and considerable gaps between observations, memory constraints may forbid a computation of these Hessians. Our new implementation combines gradient descent minimization with random maps (Morzfeld et al., 2012) to avoid the calculation of Hessians, and thus reduces the memory requirements.

The second objective is to consider models with a singular or ill-conditioned state covariance Σ_x^n where previous implementations of the implicit filter, as described in Chorin and Tu (2009); Chorin et al. (2010); Morzfeld et al. (2012), are not applicable. In Sect. 3, we make the implicit filter applicable to models with partial noise and show that our approach is then particularly efficient, because the filter operates in a space whose dimension is determined by the rank of Σ_x^n , rather than by the model dimension. We compare the new implicit filter to SIR, EnKF and variational methods.

In Sect. 4, we illustrate the theory with an application in geomagnetic data assimilation and consider two coupled nonlinear SPDEs with partial noise. We observe that the implicit filter gives good results with very few (4–10) particles, while EnKF and SIR require hundreds to thousands of particles for similar accuracy.

2 Implicit sampling with random maps

We first follow Morzfeld et al. (2012) closely to review implicit sampling with random maps. Suppose we are given a

collection of M particles $X_j^{q(l)}$, $j = 1, 2, \dots, M$, whose empirical distribution approximates the target density at time $t^{q(l)}$, where $q(l) = rl$, and suppose that an observation z^{l+1} is available after r steps at time $t^{q(l+1)} = t^{r(l+1)}$. From Eq. (5) we find, by repeatedly using Bayes’ theorem, that, for each particle,

$$\begin{aligned}
 p(X_j^{0:q(l+1)} | z^{1:l+1}) &\propto p(X_j^{0:q(l)} | z^{1:l}) p(z^{l+1} | X_j^{q(l+1)}) \\
 &\quad \times p(X_j^{q(l+1)} | X_j^{q(l+1)-1}) p(X_j^{q(l+1)-1} | X_j^{q(l+1)-2}) \\
 &\quad \vdots \\
 &\quad \times p(X_j^{q(l)+1} | X_j^{q(l)}). \tag{8}
 \end{aligned}$$

Implicit sampling is a recipe for computing high-probability samples from the above pdf. To draw a sample we define, for each particle, a function F_j by

$$\begin{aligned}
 \exp(-F(X_j)) &= p(X_j^{q(l+1)} | X_j^{q(l+1)-1}) \dots p(X_j^{q(l)+1} | X_j^{q(l)}) \\
 &\quad \times p(z^{l+1} | X_j^{q(l+1)}) \tag{9}
 \end{aligned}$$

where X_j is shorthand for the state trajectory $X_j^{q(l)+1:q(l+1)}$. Specifically, we have

$$\begin{aligned}
 F_j(X_j) &= \frac{1}{2} (X_j^{q(l)+1} - R_j^{q(l)})^T (\Sigma_{x,j}^{q(l)})^{-1} (X_j^{q(l)+1} - R_j^{q(l)}) \\
 &\quad + \frac{1}{2} (X_j^{q(l)+2} - R_j^{q(l+1)})^T (\Sigma_{x,j}^{q(l+1)})^{-1} (X_j^{q(l)+2} - R_j^{q(l+1)}) \\
 &\quad \vdots \\
 &\quad + \frac{1}{2} (X_j^{q(l+1)} - R_j^{q(l+1)-1})^T (\Sigma_{x,j}^{q(l+1)-1})^{-1} (X_j^{q(l+1)} - R_j^{q(l+1)-1}) \\
 &\quad + \frac{1}{2} (\mathbf{h}(X_j^{q(l+1)}) - z^{l+1})^T (\Sigma_{z,j}^{l+1})^{-1} (\mathbf{h}(X_j^{q(l+1)}) - z^{l+1}) \\
 &\quad + Z_j, \tag{10}
 \end{aligned}$$

where R_j^n is shorthand notation for $R(X_j^n, t^n)$ and where Z_j is a positive number that can be computed from the normalization constants of the various pdfs in the definition of F_j in Eq. (9). Note that the variables of the functions F_j are $X_j = X_j^{q(l)+1:q(l+1)}$, i.e. the state trajectory of the j -th particle from time $t^{q(l)+1}$ to $t^{q(l+1)}$. The previous position of the j -th particle at time $t^{q(l)}$, $X_j^{q(l)}$, is merely a parameter (which varies from particle to particle). The observation z^{l+1} is the same for all particles. The functions F_j are thus similar to one another. Moreover, each F_j is similar to the cost function of weak constraint 4-D-Var, however, the state at time $t^{q(l)}$ is “fixed” for each F_j , while it is a variable of the weak constraint 4-D-Var cost function.

The high probability region of the target density corresponds, by construction, to the neighborhood of the minima of the F_j ’s. We can thus identify the regions of high probability by minimizing F_j for each particle. We then map the high probability region of a reference variable, say ξ , to the high probability region of the target density. For a Gaussian

reference variable $\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbf{I})$, this can be done by solving the algebraic equation

$$F_j(\mathbf{X}_j) - \phi_j = \frac{1}{2} \boldsymbol{\xi}_j^T \boldsymbol{\xi}_j, \quad (11)$$

where $\boldsymbol{\xi}_j$ is a realization of the reference variable and where

$$\phi_j = \min F_j. \quad (12)$$

Note that a Gaussian reference variable does not imply linearity or Gaussianity assumptions and other choices are possible. What is important here is to realize that a likely sample $\boldsymbol{\xi}_j$ leads to a likely \mathbf{X}_j , because a small $\boldsymbol{\xi}_j$ leads to a \mathbf{X}_j which is in the neighborhood of the minimum of F_j and, thus, in the high probability region of the target pdf.

We find solutions of Eq. (11) by using the random map

$$\mathbf{X}_j = \boldsymbol{\mu}_j + \lambda_j \mathbf{L}_j \boldsymbol{\eta}_j, \quad (13)$$

where λ_j is a scalar, $\boldsymbol{\mu}_j$ is an rm -dimensional column vector which represents the location of the minimum of F_j , i.e. $\boldsymbol{\mu}_j = \operatorname{argmin} F_j$, \mathbf{L}_j is a deterministic $rm \times rm$ matrix we can choose, and $\boldsymbol{\eta}_j = \boldsymbol{\xi}_j / \sqrt{\boldsymbol{\xi}_j^T \boldsymbol{\xi}_j}$, is uniformly distributed on the unit rm -sphere. Upon substitution of Eq. (13) into Eq. (11), we can find a solution of Eq. (11) by solving a single algebraic equation in the variable λ_j . The weight of the particle can be shown to be

$$w_j^{q(l+1)} \propto w_j^{q(l)} \exp(-\phi_j) |\det \mathbf{L}_j| \rho_j^{1-rm/2} \left| \lambda_j^{rm-1} \frac{\partial \lambda_j}{\partial \rho_j} \right|, \quad (14)$$

where $\rho_j = \boldsymbol{\xi}_j^T \boldsymbol{\xi}_j$ and $\det \mathbf{L}_j$ denotes the determinant of the matrix L_j (see Morzfeld et al. (2012) for details of the calculation). An expression for the scalar derivative $\partial \lambda_j / \partial \rho_j$ can be obtained by implicit differentiation of Eq. (11):

$$\frac{\partial \lambda_j}{\partial \rho_j} = \frac{1}{2(\nabla F_j) \mathbf{L}_j^T \boldsymbol{\eta}_j}, \quad (15)$$

where ∇F_j denotes the gradient of F_j (an rm -dimensional row vector).

The weights are normalized so that their sum equals one. The weighted positions \mathbf{X}_j of the particles approximate the target pdf. We compute the mean of \mathbf{X}_j with weights w_j as the state estimate, and then proceed to assimilate the next observation.

2.1 Implementation of an implicit particle filter with gradient descent minimization and random maps

An algorithm for data assimilation with implicit sampling and random maps was presented in Morzfeld et al. (2012). This algorithm relies on the calculation of the Hessians of the F_j 's because these Hessians are used for minimizing the F_j 's with Newton's method and for setting up the random map. The calculation of the Hessians, however, may not be easy

in some applications because of a very large state dimension, or because the second derivatives are hard to calculate, as is the case for models with partial noise (see Sect. 3). To avoid the calculation of Hessians, we propose to use a gradient descent algorithm with line-search to minimize the F_j 's (see e.g. Nocedal and Wright, 2006), along with simple random maps. Of course other minimization techniques, in particular quasi-Newton methods (see e.g. Nocedal and Wright, 2006; Fletcher, 1987), can also be applied here. However, we decided to use gradient descent to keep the minimization as simple as possible.

For simplicity, we assume that \mathbf{G} and \mathbf{Q} in Eqs. (1)–(2) are constant matrices and calculate the gradient of F_j from Eq. (10):

$$\nabla F = \left(\frac{\partial F}{\partial \mathbf{X}^{q(l)+1}}, \frac{\partial F}{\partial \mathbf{X}^{q(l)+2}}, \dots, \frac{\partial F}{\partial \mathbf{X}^{q(l+1)-1}}, \frac{\partial F}{\partial \mathbf{X}^{q(l+1)}} \right), \quad (16)$$

with

$$\begin{aligned} \left(\frac{\partial F}{\partial \mathbf{X}^k} \right)^T &= \Sigma_x^{-1} (\mathbf{X}^k - \mathbf{R}^{k-1}) \\ &\quad - \left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{X}^k} \right)^T \Sigma_x^{-1} (\mathbf{X}^{k+1} - \mathbf{R}^k), \end{aligned} \quad (17)$$

for $k = q(l) + 1, q(l) + 2, \dots, q(l+1) - 1$, where \mathbf{R}^n is shorthand for $\mathbf{R}(\mathbf{X}^n, t^n)$, and where

$$\begin{aligned} \left(\frac{\partial F}{\partial \mathbf{X}^{q(l+1)}} \right)^T &= \Sigma_x^{-1} (\mathbf{X}^{q(l+1)} - \mathbf{R}^{q(l+1)-1}) \\ &\quad + \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{X}^{q(l+1)}} \right)^T \Sigma_z^{-1} (\mathbf{h}(\mathbf{X}^{q(l+1)}) - \mathbf{z}^{l+1}). \end{aligned} \quad (18)$$

Here, we dropped the index j for the particles for notational convenience. We initialize the minimization using the result of a simplified implicit particle filter (see next subsection). Once the minimum is obtained, we substitute the random map (13) with $\mathbf{L}_j = \mathbf{I}$, where \mathbf{I} is the identity matrix, into Eq. (11) and solve the resulting scalar equation by Newton's method. The scalar derivative we need for the Newton steps is computed numerically. We initialize this iteration with $\lambda_j = 0$. Finally, we compute the weights according to Eq. (14). If some weights are small, as indicated by a small effective sample size (Arulampalam et al., 2002)

$$M_{\text{Eff}} = 1 / \left(\sum_{j=1}^M (w_j^{q(l+1)})^2 \right), \quad (19)$$

we resample using algorithm 2 in Arulampalam et al. (2002). The implicit filtering algorithm with gradient descent minimization and random maps is summarized in pseudo-code in algorithm 1.

This implicit filtering algorithm shares with weak constraint 4-D-Var that a "cost function" (here F_j) is minimized by gradient descent. The two main differences between 4-D-Var and algorithm 1 are (i) weak constraint 4-D-Var does

Algorithm 1 Implicit Particle Filter with Random Maps and Gradient Descent Minimization

```

{Initialization,  $t = 0$ }
for  $j = 1, \dots, M$  do
    • sample  $X_j^0 \sim p_o(X)$ 
end for

{Assimilate observation  $z^l$ }
for  $j = 1, \dots, M$  do
    • Set up and minimize  $F_j$  using gradient descent to compute  $\phi_j$  and  $\mu_j$ 
    • Sample reference density  $\xi_j \sim \mathcal{N}(0, \mathbf{I})$ 
    • Compute  $\rho_j = \xi_j^T \xi_j$  and  $\eta_j = \xi_j / \sqrt{\rho_j}$ 
    • Solve (11) using the random map (13) with  $\mathbf{L}_j = \mathbf{I}$ 
    • Compute weight of the particle using (14)
    • Save particle  $X_j$  and weight  $w_j$ 
end for

    • Normalize the weights so that their sum equals 1
    • Compute state estimate from  $X_j$  weighted with  $w_j$  (e.g. the mean)
    • Resample if  $M_{\text{Eff}} < c$ 
    • Assimilate  $z^{l+1}$ 
    
```

not update the state sequentially, but the implicit particle filter does and, thus, reduces memory requirements; (ii) weak constraint 4-D-Var computes the most likely state, and this state estimate can be biased; the implicit particle filter approximates the target density and, thus, can compute other statistics as state estimates, in particular the conditional expectation which is, under wide conditions, the optimal state estimate (see e.g. Chorin and Hald, 2009). A more detailed exposition of the implicit filter and its connection to variational data assimilation is currently under review (Atkins et al., 2012).

2.2 A simplified implicit particle filtering algorithm with random maps and gradient descent minimization

We wish to simplify the implicit particle filtering algorithm by reducing the dimension of the function F_j . The idea is to do an implicit sampling step only at times $t^{q(l+1)}$, i.e. when an observation becomes available. The state trajectory of each particle from time $t^{q(l)}$ (the last time an observation became available) to $t^{q(l+1)-1}$ is generated using the model Eq. (1). This approach reduces the dimension of F_j from rm to m (the state dimension). The simplification is thus very attractive if the number of steps between observations, r , is large. However, difficulties can also be expected for large r : the state trajectories up to time $t^{q(l+1)-1}$ are generated by the model alone and, thus, may not have a high probability with respect to the observations at time $t^{q(l+1)}$. The focussing effect of implicit sampling can be expected to be less emphasized and the number of particles required may grow as the

gap between observations becomes larger. Whether or not the simplification we describe here can reduce the computational cost is problem dependent and we will illustrate advantages and disadvantages in the examples in Sect. 4.

Suppose we are given a collection of M particles $X_j^{q(l)}$, $j = 1, 2, \dots, M$, whose empirical distribution approximates the target density at time $t^{q(l)}$ and the next observation, z^{l+1} , is available after r steps at time $t^{q(l+1)}$. For each particle, we run the model for $r - 1$ steps to obtain $X_j^{q(l)+1}, \dots, X_j^{q(l+1)-1}$. We then define, for each particle, a function F_j by

$$\begin{aligned}
 F_j(X_j) = & \frac{1}{2} \left(X_j^{q(l+1)} - R_j^{q(l+1)-1} \right)^T \left(\Sigma_{x,j}^{q(l+1)-1} \right)^{-1} \\
 & \times \left(X_j^{q(l+1)} - R_j^{q(l+1)-1} \right) \\
 & + \frac{1}{2} \left(h \left(X_j^{q(l+1)} \right) - z^{l+1} \right)^T \left(\Sigma_{z,j}^{q(l+1)} \right)^{-1} \\
 & \left(h \left(X_j^{q(l+1)} \right) - z^{l+1} \right) \\
 & + Z_j,
 \end{aligned} \tag{20}$$

whose gradient is given by Eq. (18). The algorithm then proceeds as algorithm 1 in the previous section: we find the minimum of F_j using gradient descent and solve Eq. (11) with the random map (13) with $\mathbf{L}_j = \mathbf{I}$. The weights are calculated by Eq. (14) with $r = 1$ and the mean of X_j weighted by w_j is the state estimate at time $t^{q(l+1)}$.

This simplified implicit filter simplifies further if the observation function is linear, i.e. $h(x) = \mathbf{H}x$, where \mathbf{H} is a $k \times m$ matrix. One can show (see Morzfeld et al., 2012) that the minimum of F_j is

$$\phi_j = \frac{1}{2} (z^{l+1} - \mathbf{H}R_j^{q(l+1)-1})^T \mathbf{K}_j^{-1} (z^{l+1} - \mathbf{H}R_j^{q(l+1)-1}), \tag{21}$$

with

$$\mathbf{K}_j = \mathbf{H} \Sigma_{x,j}^{q(l+1)-1} \mathbf{H}^T + \Sigma_{z,j}^{l+1}. \tag{22}$$

The location of the minimum is

$$\mu_j = \Sigma_j \left(\left(\Sigma_{x,j}^{q(l+1)-1} \right)^{-1} R_j^{q(l+1)-1} + \mathbf{H}^T \left(\Sigma_{z,j}^{q(l+1)} \right)^{-1} z^{l+1} \right), \tag{23}$$

with

$$\Sigma_j^{-1} = \left(\Sigma_{x,j}^{q(l+1)-1} \right)^{-1} + \mathbf{H}^T \left(\Sigma_{z,j}^{l+1} \right)^{-1} \mathbf{H}. \tag{24}$$

A numerical approximation of the minimum is thus not required (one can use the above formula), however an iterative minimization may be necessary if the dimension of the state space is so large that storage of the matrices involved in Eqs. (21)–(24) causes difficulties.

To obtain a sample, we can solve Eq. (11) by computing the Cholesky factor \mathbf{L}_j of Σ_j , and using $X_j = \mu_j + \mathbf{L}_j \xi_j$. The weights in Eq. (14) then simplify to

$$w_j^{n+1} \propto w_j^n \exp(-\phi_j) |\det \mathbf{L}_j|. \tag{25}$$

For the special case of a linear observation function and observations available at every model step ($r = 1$), the simplified implicit filter is the full implicit filter and reduces to a version of optimal importance sampling (Arulampalam et al., 2002; Bocquet et al., 2010; Morzfeld et al., 2012; Chorin et al., 2010).

3 Implicit particle filtering for equations with partial noise

We consider the case of a singular state covariance matrix Σ_x in the context of implicit particle filtering. We start with an example taken from Jentzen and Kloeden (2009), to demonstrate how a singular state covariance appears naturally in the context of SPDEs driven by spatially smooth noise. The example serves as a motivation for more general developments in later sections.

Another class of models with partial noise consists of dynamical equations supplemented by conservation laws. The dynamics are often uncertain and thus driven by noise processes, however there is typically zero uncertainty in the conservation laws (e.g. conservation of mass), so that the full model (dynamics and conservation laws) is subject to partial noise (Kurapov et al., 2007). This situation is similar to that of handling second-order (in time) SDEs, for example in robotics. The second-order equation is often converted into a set of first-order equations, for which the additional equations are trivial (e.g. $du/dt = du/dt$). It is unphysical to inject noise into these augmenting equations, so that the second-order model in a first-order formulation is subject to partial noise.

3.1 Example of a model with partial noise: the semi-linear heat equation driven by spatially smooth noise

We consider the stochastic semi-linear heat equation on the one-dimensional domain $x \in [0, 1]$ over the time interval $t \in [0, 1]$

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \Gamma(u) + \frac{\partial W_t}{\partial t}, \quad (26)$$

where Γ is a continuous function, and W_t is a cylindrical Brownian motion (BM) (Jentzen and Kloeden, 2009). The derivative $\partial W_t / \partial t$ in Eq. (26) is formal only (it does not exist in the usual sense). Equation (26) is supplemented by homogeneous Dirichlet boundary conditions and the initial value $u(x, 0) = u_o(x)$. We expand the cylindrical BM W_t in the eigenfunctions of the Laplace operator

$$W_t = \sum_{k=1}^{\infty} \sqrt{2q_k} \sin(k\pi x) \beta_t^k, \quad (27)$$

where β_t^k denote independent BMs and where the coefficients $q_k \geq 0$ must be chosen such that, for $\gamma \in (0, 1)$,

$$\sum_{k=1}^{\infty} \lambda_k^{2\gamma-1} q_k < \infty, \quad (28)$$

where λ_k are the eigenvalues of the Laplace operator (Jentzen and Kloeden, 2009). If the coefficients q_k decay fast enough, then, by Eq. (27) and basic properties of Fourier series, the noise is smooth in space and, in addition, the sum (28) remains finite as is required. For example one may be interested in problems where

$$q_k = \begin{cases} e^{-2k}, & \text{if } k \leq c, \\ 0, & \text{if } k > c, \end{cases} \quad (29)$$

for some $c > 0$.

The continuous equation must be discretized for computations and here we consider the Galerkin projection of the SPDE into an m -dimensional space spanned by the first m eigenfunctions e_k of the Laplace operator

$$dU_t^m = (\mathbf{A}_m U_t^m + \mathbf{\Gamma}_m(U_t^m))dt + dW_t^m, \quad (30)$$

where U_t^m , $\mathbf{\Gamma}_m$ and W_t^m are m -dimensional truncations of the solution, the function $\mathbf{\Gamma}$ and the cylindrical BM W_t , respectively, and where \mathbf{A}_m is a discretization of the Laplace operator. Specifically, from Eqs. (27) and (29), we obtain:

$$dW_t^m = \sum_{k=1}^c \sqrt{2} e^{-k} \sin(k\pi x) d\beta_t^k. \quad (31)$$

After multiplying Eq. (30) with the basis functions and integrating over the spatial domain, we are left with a set of m stochastic ordinary differential equations

$$dx = f(x)dt + g dW, \quad (32)$$

where x is an m -dimensional state vector, f is a nonlinear vector function, W is a BM. In particular, we calculate from (31):

$$g = \frac{1}{\sqrt{2}} \text{diag} \left(\left(e^{-1}, e^{-2}, \dots, e^{-c}, 0, 0, \dots, 0 \right) \right), \quad c < m, \quad (33)$$

where $\text{diag}(a)$ is a diagonal matrix whose diagonal elements are the components of the vector a . Upon time discretization using, for example, a stochastic version of forward Euler with time step δ (Kloeden and Platen, 1999), we arrive at Eq. (1) with

$$R(x) = x^n + \delta f(x^n), \quad G(x) = \sqrt{\delta} g. \quad (34)$$

It is now clear that the state covariance matrix $\Sigma_x = \mathbf{G}\mathbf{G}^T$ is singular for $c < m$.

A singular state covariance causes no problems for running the discrete time model (1) forward in time. However problems do arise if we want to know the pdf of the current state given the previous one. For example, the functions F_j in the implicit particle filter algorithms (either those in Sect. 2, or those in Chorin and Tu, 2009; Chorin et al., 2010; Morzfeld et al., 2012) are not defined for singular Σ_x . If $c \geq m$, then Σ_x is ill-conditioned and causes a number of numerical issues in the implementation of these implicit particle filtering algorithms and, ultimately, the algorithms fail.

3.2 Implicit particle filtering of models with partial noise, supplemented by densely available data

We start with deriving the implicit filter for models with partial noise by considering the special case in which observations are available at every model step ($r = 1$). For simplicity, we assume that the noise is additive, i.e. $\mathbf{G}(\mathbf{x}^n, t^n)$ in Eq. (1) is constant and that \mathbf{Q} in Eq. (2) is also a constant matrix. Under these assumptions, we can use a linear coordinate transformation to diagonalize the state covariance matrix and rewrite the model (1) and the observations (2) as

$$\mathbf{x}^{n+1} = \mathbf{f}(\mathbf{x}^n, \mathbf{y}^n, t^n) + \Delta \mathbf{W}^{n+1}, \quad \Delta \mathbf{W}^{n+1} \sim \mathcal{N}(0, \hat{\Sigma}_x), \quad (35)$$

$$\mathbf{y}^{n+1} = \mathbf{g}(\mathbf{x}^n, \mathbf{y}^n, t^n), \quad (36)$$

$$\mathbf{z}^{n+1} = \mathbf{h}(\mathbf{x}^{n+1}, \mathbf{y}^{n+1}) + \mathbf{Q}\mathbf{V}^{n+1}, \quad (37)$$

where \mathbf{x} is a p -dimensional column vector, $p < m$ is the rank of the state covariance matrix Eq. (4), and where \mathbf{f} is a p -dimensional vector function, $\hat{\Sigma}_x$ is a non-singular, diagonal $p \times p$ matrix, \mathbf{y} is a $(m - p)$ -dimensional vector, and \mathbf{g} is a $(m - p)$ -dimensional vector function. For ease of notation, we drop the hat above the “new” state covariance matrix $\hat{\Sigma}_x$ in Eq. (35) and, for convenience, we refer to the set of variables \mathbf{x} and \mathbf{y} as the “forced” and “unforced variables” respectively.

The key to filtering this system is observing that the unforced variables at time t^{n+1} , given the state at time t^n , are not random. To be sure, \mathbf{y}^n is random for any n due to the nonlinear coupling $\mathbf{f}(\mathbf{x}, \mathbf{y})$ and $\mathbf{g}(\mathbf{x}, \mathbf{y})$, but the conditional pdf $p(\mathbf{y}^{n+1} | \mathbf{x}^n, \mathbf{y}^n)$ is the delta-distribution. For a given initial state $\mathbf{x}^0, \mathbf{y}^0$, the target density is

$$p(\mathbf{x}^{0:n+1}, \mathbf{y}^{0:n+1} | \mathbf{z}^{1:n+1}) \propto p(\mathbf{x}^{0:n}, \mathbf{y}^{0:n} | \mathbf{z}^{1:n}) \times p(\mathbf{z}^{n+1} | \mathbf{x}^{n+1}, \mathbf{y}^{n+1}) p(\mathbf{x}^{n+1} | \mathbf{x}^n, \mathbf{y}^n). \quad (38)$$

Suppose we are given a collection of M particles, $\mathbf{X}_j^n, \mathbf{Y}_j^n$, $j = 1, 2, \dots, M$, whose empirical distribution approximates the target density $p(\mathbf{x}^{0:n}, \mathbf{y}^{0:n} | \mathbf{z}^{1:n})$ at time t^n . The pdf for each particle at time t^{n+1} is thus given by Eq. (38) with the substitution of \mathbf{X}_j for \mathbf{x} and \mathbf{Y}_j for \mathbf{y} . In agreement with the definition of F_j in previous implementations of the implicit

filter, we define F_j for models with partial noise by

$$\exp(-F_j(\mathbf{X}_j^{n+1})) = p(\mathbf{z}^{n+1} | \mathbf{X}_j^{n+1}, \mathbf{Y}_j^{n+1})p(\mathbf{X}_j^{n+1} | \mathbf{X}_j^n, \mathbf{Y}_j^n). \quad (39)$$

More specifically,

$$F_j(\mathbf{X}_j^{n+1}) = \frac{1}{2}(\mathbf{X}_j^{n+1} - \mathbf{f}_j^n)^T \Sigma_x^{-1}(\mathbf{X}_j^{n+1} - \mathbf{f}_j^n) + \frac{1}{2}(\mathbf{h}(\mathbf{X}_j^{n+1}, \mathbf{Y}_j^{n+1}) - \mathbf{z}^{n+1})^T \times \Sigma_z^{-1}(\mathbf{h}(\mathbf{X}_j^{n+1}, \mathbf{Y}_j^{n+1}) - \mathbf{z}^{n+1}) + Z_j, \quad (40)$$

where \mathbf{f}_j^n is shorthand notation for $\mathbf{f}(\mathbf{X}_j^n, \mathbf{Y}_j^n, t^n)$. With this F_j , we can use algorithm 1 to construct the implicit filter. For this algorithm we need the gradient of F_j :

$$(\nabla F_j)^T = \Sigma_x^{-1}(\mathbf{X}_j^{n+1} - \mathbf{f}_j^n) + \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{X}_j^{n+1}}\right)^T \times \Sigma_z^{-1}(\mathbf{h}(\mathbf{X}_j^{n+1}, \mathbf{Y}_j^{n+1}) - \mathbf{z}^{n+1}). \quad (41)$$

Note that \mathbf{Y}_j^{n+1} is fixed for each particle, if the previous state, $(\mathbf{X}_j^n, \mathbf{Y}_j^n)$, is known, so that the filter only updates \mathbf{X}_j^{n+1} when the observations \mathbf{z}^{n+1} become available. The unforced variables of the particles, \mathbf{Y}_j^{n+1} , are moved forward in time using the model, as they should be, since there is no uncertainty in \mathbf{y}^{n+1} given $\mathbf{x}^n, \mathbf{y}^n$. The data are used in the state estimation of \mathbf{y} indirectly through the weights and through the nonlinear coupling between the forced and unforced variables of the model. If one observes only the unforced variables, i.e. $\mathbf{h}(\mathbf{x}, \mathbf{y}) = \mathbf{h}(\mathbf{y})$, then the data is not used directly when generating the forced variables, \mathbf{X}_j^{n+1} , because the second term in Eq. (40) is merely a constant. In this case, the implicit filter is equivalent to a standard SIR filter, with weights $w_j^{n+1} = w_j^n \exp(-\phi_j)$.

The implicit filter is numerically effective for filtering systems with partial noise, because the filter operates in a space of dimension p (the rank of the state covariance matrix), which is less than the state dimension (see the example in Sect. 4). The use of a gradient descent algorithm and random maps further makes the often costly computation of the Hessian of F_j unnecessary.

If the state covariance matrix is ill-conditioned, a direct implementation of algorithm 1 is not possible. We propose to diagonalize the state covariance and set all eigenvalues below a certain threshold to zero so that a model of the form Eqs. (35)–(37) can be obtained. In our experience, such approximations are accurate and the filter of this section can be used.

3.3 Implicit particle filtering for models with partial noise, supplemented by sparsely available data

We extend the results of Sect. 3.2 to the more general case of observations that are sparse in time. Again, the key is to realize that \mathbf{y}^{n+1} is fixed given $\mathbf{x}^n, \mathbf{y}^n$. For simplicity, we assume additive noise and a constant \mathbf{Q} in Eq. (2). The target density is

$$\begin{aligned} p(\mathbf{x}^{0:q(l+1)}, \mathbf{y}^{0:q(l+1)} | \mathbf{z}^{1:l+1}) &\propto p(\mathbf{x}^{0:q(l)}, \mathbf{y}^{0:q(l)} | \mathbf{z}^{1:l}) \\ &\quad \times p(\mathbf{z}^{l+1} | \mathbf{x}^{q(l+1)}, \mathbf{y}^{q(l+1)}) \\ &\quad \times p(\mathbf{x}^{q(l+1)} | \mathbf{x}^{q(l+1)-1}, \mathbf{y}^{q(l+1)-1}) \\ &\quad \times p(\mathbf{x}^{q(l+1)-1} | \mathbf{x}^{q(l+1)-2}, \mathbf{y}^{q(l+1)-2}) \\ &\quad \vdots \\ &\quad \times p(\mathbf{x}^{q(l)+1} | \mathbf{x}^{q(l)}, \mathbf{y}^{q(l)}). \end{aligned}$$

Given a collection of M particles, $\mathbf{X}_j^n, \mathbf{Y}_j^n$, $j = 1, 2, \dots, M$, whose empirical distribution approximates the target density $p(\mathbf{x}^{0:q(l)}, \mathbf{y}^{0:q(l)} | \mathbf{z}^{1:l})$ at time $t^{q(l)}$, we define, for each particle, the function F_j by

$$\begin{aligned} \exp(-F_j(\mathbf{X}_j)) &= p(\mathbf{z}^{l+1} | \mathbf{X}_j^{q(l+1)}, \mathbf{Y}_j^{q(l+1)}) \\ &\quad \times p(\mathbf{X}_j^{q(l+1)} | \mathbf{X}_j^{q(l+1)-1}, \mathbf{Y}_j^{q(l+1)-1}) \\ &\quad \vdots \\ &\quad \times p(\mathbf{X}_j^{q(l)+1} | \mathbf{X}_j^{q(l)}, \mathbf{Y}_j^{q(l)}), \end{aligned} \quad (42)$$

where \mathbf{X}_j is shorthand for $\mathbf{X}_j^{q(l)+1, \dots, q(l+1)}$, so that

$$\begin{aligned} F_j(\mathbf{X}_j) &= \frac{1}{2} (\mathbf{X}_j^{q(l)+1} - \mathbf{f}_j^{q(l)})^T \Sigma_x^{-1} (\mathbf{X}_j^{q(l)+1} - \mathbf{f}_j^{q(l)}) \quad (43) \\ &\quad + \frac{1}{2} (\mathbf{X}_j^{q(l)+2} - \mathbf{f}_j^{q(l+1)})^T \Sigma_x^{-1} (\mathbf{X}_j^{q(l)+2} - \mathbf{f}_j^{q(l+1)}) \\ &\quad \vdots \\ &\quad + \frac{1}{2} (\mathbf{X}_j^{q(l+1)} - \mathbf{f}_j^{q(l+1)-1})^T \Sigma_x^{-1} (\mathbf{X}_j^{q(l+1)} - \mathbf{f}_j^{q(l+1)-1}) \\ &\quad + \frac{1}{2} (\mathbf{h}(\mathbf{X}_j^{q(l+1)}, \mathbf{Y}_j^{q(l+1)}) - \mathbf{z}^{l+1})^T \Sigma_z^{-1} \\ &\quad \times (\mathbf{h}(\mathbf{X}_j^{q(l+1)}, \mathbf{Y}_j^{q(l+1)}) - \mathbf{z}^{l+1}) + Z_j. \end{aligned} \quad (44)$$

At each model step, the unforced variables of each particle depend on the forced and unforced variables of the particle at the previous time step, so that $\mathbf{Y}_j^{q(l+1)}$ is a function of $\mathbf{X}_j^{q(l)}, \mathbf{X}_j^{q(l)+1}, \dots, \mathbf{X}_j^{q(l+1)-1}$ and $\mathbf{f}_j^{q(l+1)}$ is a function of $\mathbf{X}_j^{q(l)+1}, \mathbf{X}_j^{q(l)+2}, \dots, \mathbf{X}_j^{q(l+1)}$. The function F_j thus depends on the forced variables only. However, the appearances of the unforced variables in F_j make it rather difficult to compute derivatives. The implicit filter with gradient descent minimization and random maps (see algorithm 1) is thus a good filter for this problem, because it only requires computation of the first derivatives of F_j , while previous implementations

(see Chorin et al., 2010; Morzfeld et al., 2012) require second derivatives as well.

The gradient of F_j is given by the rp -dimensional row vector

$$\nabla F_j = \left(\frac{\partial F_j}{\partial \mathbf{X}_j^{q(l)+1}}, \frac{\partial F_j}{\partial \mathbf{X}_j^{q(l)+2}}, \dots, \frac{\partial F_j}{\partial \mathbf{X}_j^{q(l+1)}} \right). \quad (45)$$

with

$$\begin{aligned} \frac{\partial F_j}{\partial \mathbf{X}_j^k}{}^T &= \Sigma_x^{-1} (\mathbf{X}_j^k - \mathbf{f}_j^{k-1}) \\ &\quad + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_k \right)^T \Sigma_x^{-1} (\mathbf{X}_j^{k+1} - \mathbf{f}_j^k) \\ &\quad + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}} \Big|_{k+1} \frac{\partial \mathbf{y}^{k+1}}{\partial \mathbf{X}_j^k} \right)^T \Sigma_x^{-1} (\mathbf{X}_j^{k+2} - \mathbf{f}_j^{k+1}) \\ &\quad + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}} \Big|_{k+2} \frac{\partial \mathbf{y}^{k+2}}{\partial \mathbf{X}_j^k} \right)^T \Sigma_x^{-1} (\mathbf{X}_j^{k+3} - \mathbf{f}_j^{k+2}) \\ &\quad \vdots \\ &\quad + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}} \Big|_{q(l)-1} \frac{\partial \mathbf{y}^{q(l)-1}}{\partial \mathbf{X}_j^k} \right)^T \Sigma_x^{-1} (\mathbf{X}_j^{q(l+1)} - \mathbf{f}_j^{q(l)-1}) \\ &\quad + \left(\frac{\partial \mathbf{h}}{\partial \mathbf{y}} \Big|_k \frac{\partial \mathbf{y}^{q(l)}}{\partial \mathbf{X}_j^k} \Big|_{k-1} \right)^T \\ &\quad \times \Sigma_z^{-1} (\mathbf{h}(\mathbf{X}_j^{q(l+1)}, \mathbf{Y}_j^{q(l+1)}) - \mathbf{z}^{l+1}) \end{aligned} \quad (46)$$

for $k = q(l) + 1, \dots, q(l+1) - 1$ and where $(\cdot) |_k$ denotes ‘‘evaluate at time t^k .’’ The derivatives $\partial \mathbf{y}^i / \partial \mathbf{X}_j^k$, $i = k + 1, \dots, q(l)$, can be computed recursively while constructing the sum, starting with

$$\frac{\partial \mathbf{y}^{k+1}}{\partial \mathbf{X}_j^k} = \frac{\partial}{\partial \mathbf{X}_j^k} (\mathbf{g}(\mathbf{X}_j^k, \mathbf{Y}_j^k)) = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \Big|_k, \quad (47)$$

and then using

$$\frac{\partial \mathbf{y}^{k+i}}{\partial \mathbf{X}_j^k} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \Big|_{i-1} \frac{\partial \mathbf{y}^{i-1}}{\partial \mathbf{X}_j^k} \Big|_{i-1}, \quad i = k + 2, \dots, q(l). \quad (48)$$

The minimization of F_j for each particle is initialized with a free model run for r steps, with initial conditions given by the final position of the j -th particle at the previous assimilation step. With this initial guess we compute the gradient using Eqs. (45)–(48) and, after a line search and one step of gradient descent, obtain a new set of forced variables. We use this result to update the unforced variables by the model, and proceed to the next iteration. Once the minimum ϕ_j and its location $\boldsymbol{\mu}_j$ are found, we use the random map (13) with $\mathbf{L}_j = \mathbf{I}$ to compute $\mathbf{X}_j^{q(l)+1}, \dots, \mathbf{X}_j^{q(l+1)}$ for this particle and then use these forced variables to compute $\mathbf{Y}_j^{q(l)+1, \dots, q(l+1)}$.

We do this for all particles, and compute the weights from Eq. (14) with $m = p$, then normalize the weights so that their sum equals one and thereby obtain an approximation of the target density. We resample if the effective sample size M_{Eff} is below a threshold and move on to assimilate the next observation. The implicit filtering algorithm is summarized with pseudo code in algorithm 2.

Algorithm 2 Implicit Particle Filter with Random Maps and Gradient Descent Minimization for Models with Partial Noise

```

{Initialization,  $t = 0$ }
for  $j = 1, \dots, M$  do
    • sample  $X_j^0 \sim p_o(X)$ 
end for

{Assimilate observation  $z^l$ }
for  $j = 1, \dots, M$  do
    • Set up and minimize  $F_j$  using gradient descent:
    Initialize minimization with a free model run
    while Convergence criteria not satisfied do
        Compute gradient by (45)
        Do a line search
        Compute next iterate by gradient descent step
        Use results to update unforced variables using the model
        Check if convergence criteria are satisfied
    end while
    • Sample reference density  $\xi_j \sim \mathcal{N}(0, \mathbf{I})$ 
    • Compute  $\rho_j = \xi_j^T \xi_j$  and  $\eta_j = \xi_j / \sqrt{\rho_j}$ 
    • Solve (11) using random map (13) with  $\mathbf{L}_j = \mathbf{I}$  to compute  $X_j$ 
    • Use this  $X_j$  and the model to compute corresponding  $Y_j$ 
    • Compute weight of the particle using (14)
    • Save particle  $(X_j, Y_j)$  and weight  $w_j$ 
end for

    • Normalize the weights so that their sum equals 1
    • Compute state estimate from  $X_j$  weighted with  $w_j$  (e.g. the mean)
    • Resample if  $M_{\text{Eff}} < c$ 
    • Assimilate  $z^{l+1}$ 

```

Note that all state variables are computed by using both the data and the model, regardless of which set of variables (the forced or unforced ones) is observed. The reason is that, for sparse observations, the F_j 's depend on the observed and unobserved variables due to the nonlinear coupling \mathbf{f} and \mathbf{g} in Eqs. (35)–(37). It should also be noted that the function F_j is a function of rp variables (rather than rm), because the filter operates in the subspace of the forced variables. If the minimization is computationally too expensive, because p or r is extremely large, then one can easily adapt the “simplified” implicit particle filter of Sect. 2.2 to the situation of partial noise using the methods we have just described. The simplified filter then requires a minimization of a p -dimensional function for each particle.

3.4 Discussion

We wish to point out similarities and differences between the implicit filter and three other data assimilation methods. In particular, we discuss how data are used in the computation of the state estimates.

It is clear that the implicit filter uses the available data as well as the model to generate the state trajectories for each particle, i.e. it makes use of the nonlinear coupling between forced and unforced variables. The SIR and EnKF make less direct use of the data. In SIR, the particle trajectories are generated using the model alone and only later weighted by the observations. Data thus propagate to the SIR state estimates indirectly through the weights. In EnKF, the state trajectories are generated by the model and the states at times $t^{q(l)}$ (when data are available) are updated by data. Thus, EnKF uses the data only to update its state estimates at times for which data are actually available.

A weak constraint 4-D-Var method is perhaps closest in spirit to the implicit filter. In weak constraint 4-D-Var, a cost function similar to F_j is minimized (typically by gradient descent) to find the state trajectory with maximum probability given data and model. This cost function depends on the model as well as the data, so that weak constraint 4-D-Var makes use of the model and the data to generate the state trajectories. In this sense, weak constraint 4-D-Var is similar to the implicit filter (see Atkins et al., 2012 for more details).

4 Application to geomagnetism

Data assimilation has been recently applied to geomagnetic applications and there is a need to find out which data assimilation technique is most suitable (Fournier et al., 2010). Thus far, a strong constraint 4-D-Var approach (Fournier et al., 2007) and a Kalman filter approach (Sun et al., 2007; Aubert and Fournier, 2011) have been considered. Here, we apply the implicit particle filter to a test problem very similar to the one first introduced by Fournier and his colleagues in Fournier et al. (2007). The model is given by two SPDEs

$$\partial_t u + u \partial_x u = b \partial_x b + v \partial_x^2 u + g_u \partial_t W_u(x, t), \quad (49)$$

$$\partial_t b + u \partial_x b = b \partial_x u + \partial_x^2 b + g_b \partial_t W_b(x, t), \quad (50)$$

where, v, g_u, g_b are scalars, and where W_u and W_b are independent stochastic processes (the derivative here is formal and may not exist in the usual sense). Physically, u represents the velocity field and b represents the magnetic field. We consider the above equations on the strip $0 \leq t \leq T, -1 \leq x \leq 1$ and with boundary conditions

$$u(x, t) = 0, \text{ if } x = \pm 1, \quad u(x, 0) = \sin(\pi x) + 2/5 \sin(5\pi x), \quad (51)$$

$$b(x, t) = \pm 1, \text{ if } x = \pm 1, \quad b(x, 0) = \cos(\pi x) + 2 \sin(\pi(x+1)/4). \quad (52)$$

The stochastic processes in Eqs. (49) and (50) are given by

$$W_u(x, t) = \sum_{k=0}^{\infty} \alpha_k^u \sin(k\pi x) w_k^1(t) + \beta_k^u \cos(k\pi/2x) w_k^2(t), \quad (53)$$

$$W_b(x, t) = \sum_{k=0}^{\infty} \alpha_k^b \sin(k\pi x) w_k^3(t) + \beta_k^b \cos(k\pi/2x) w_k^4(t). \quad (54)$$

where $w_k^1, w_k^2, w_k^3, w_k^4$ are independent BMs and where

$$\alpha_k^u = \beta_k^u = \alpha_k^b = \beta_k^b = \begin{cases} 1, & \text{if } k \leq 10, \\ 0, & \text{if } k > 10, \end{cases} \quad (55)$$

i.e. the noise processes are independent, identically distributed, but differ in magnitude (on average) due to the factors g_u and g_b in Eqs. (49) and (50) (see below). The stochastic process represents a spatially smooth noise which is zero at the boundaries. Information about the spatial distribution of the uncertainty can be incorporated by picking suitable coefficients α_k and β_k .

We study the above equations with $\nu = 10^{-3}$ as in Fournier et al. (2007), and with $g_u = 0.01, g_b = 1$. With this choice of parameters, we observe that the random disturbance to the velocity field u is on the order of 10^{-5} , and that the disturbance to the magnetic field b is on the order of 10^{-1} . While the absolute value of the noise on u is quite small, its effect is dramatic because the governing equation is sensitive to perturbations, because ν is small. An illustration of the noise process and its effect on the solution is given in Fig. 1. The upper left figure shows a realization of the noise process W and illustrates that the noise is smooth in space. The upper right part of Fig. 1 shows two realizations of the solution and, since the two realizations are very different, illustrates the need for data assimilation. The lower two panels of Fig. 1 show a typical snapshot of the noise on u (right) and b (left).

We chose the parameters g_u and g_b as large as possible and the parameter ν as small as possible without causing instabilities in our discretization (see below). For larger values of g_u and smaller values of ν , a more sophisticated discretization is necessary. However, the model itself (independent of the choice of parameters) is a dramatic simplification of more realistic three-dimensional dynamo models, so that the value of studying Eqs. (49) and (50) for larger g_b, g_u or smaller ν is limited. Our results should be interpreted as “proof of concept,” that implicit sampling can be used to improve the forecast and analysis of the hidden velocity field u by assimilating observations of the magnetic field b .

4.1 Discretization of the dynamical equations

We follow Fournier et al. (2007) in the discretization of the dynamical equations, however we present details here to explain how the noise process W comes into play.

For both fields, we use Legendre spectral elements of order N (see e.g. Canuto et al., 2006; Deville et al., 2006), so that

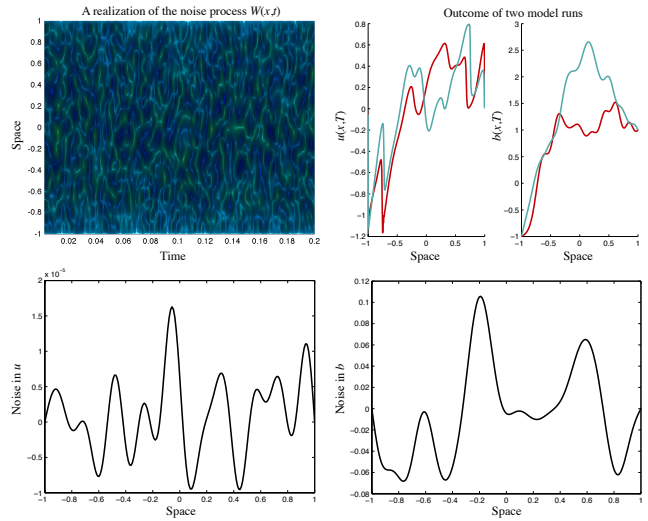


Fig. 1. The noise process $W(x, t)$ and its effects on the solution u and b . Upper left: the noise process $W(x, t)$ is plotted as a function of x and t . Upper right: two realizations of the solution at $t = T = 0.2$. Lower left: a snapshot of the noise on u . Lower right: a snapshot of the noise on b .

$$u(x, t) = \sum_{j=0}^N \hat{u}_j(t) \psi_j(x) = \sum_{j=1}^{N-1} \hat{u}_j(t) \psi_j(x),$$

$$b(x, t) = \sum_{j=0}^N \hat{b}_j(t) \psi_j(x) = -\psi_0(x) + \psi_N(x) + \sum_{j=1}^{N-1} \hat{b}_j(t) \psi_j(x),$$

$$W(x, t) = \sum_{j=0}^N \hat{W}_j(t) \psi_j(x) = \sum_{j=1}^{N-1} \hat{W}_j(t) \psi_j(x),$$

where ψ_j are the characteristic Lagrange polynomials of order N , centered at the j -th Gauss-Lobatto-Legendre (GLL) node ξ_j . We consider the weak form of Eqs. (49) and (50) without integration by parts because the solutions are smooth enough to do so. This weak form requires computation of the second derivatives of the characteristic Lagrange polynomials at the nodes, which can be done stably and accurately using recursion formulas. We substitute the series expansions into the weak form of Eqs. (49) and (50) and evaluate the integrals by Gauss-Lobatto-Legendre quadrature

$$\int_{-1}^1 p(x) dx \sim \sum_{j=0}^N p(\xi_j) w_j,$$

where w_j are the corresponding weights. Making use of the orthogonality of the basis functions, $\psi_j(\xi_k) = \delta_{j,k}$, we obtain

the set of SDEs

$$\mathbf{M}d_t \hat{\mathbf{u}} = \mathbf{M} \left(\hat{\mathbf{b}} \circ \mathbf{D}\hat{\mathbf{b}} - \hat{\mathbf{u}} \circ \mathbf{D}\hat{\mathbf{u}} + \nu \mathbf{D}^2 \hat{\mathbf{u}} + \Psi_{\mathbf{x}}^{\mathbf{B}} \hat{\mathbf{b}} + g_u \partial_t \hat{\mathbf{W}} \right),$$

$$\mathbf{M}d_t \hat{\mathbf{b}} = \mathbf{M} \left(\hat{\mathbf{b}} \circ \mathbf{D}\hat{\mathbf{u}} - \hat{\mathbf{u}} \circ \mathbf{D}\hat{\mathbf{b}} + \mathbf{D}^2 \hat{\mathbf{b}} - \Psi_{\mathbf{x}}^{\mathbf{B}} \hat{\mathbf{u}} + \Psi_{\mathbf{xx}}^{\mathbf{B}} + g_b \partial_t \hat{\mathbf{W}} \right),$$

where \circ denotes the Hadamard product $((\hat{\mathbf{u}} \circ \hat{\mathbf{b}})_k = \hat{u}_k \hat{b}_k)$, $\hat{\mathbf{u}}, \hat{\mathbf{b}}, \hat{\mathbf{W}}$ are $(N - 2)$ -dimensional column vectors whose components are the coefficients in the series expansions of u, b, W_u and W_b , respectively, and where $\Psi_{\mathbf{x}}^{\mathbf{B}} = \text{diag}((\partial_x \psi_j(\xi_1), \dots, \partial_x \psi_j(\xi_{N-1})))$ and $\Psi_{\mathbf{xx}}^{\mathbf{B}} = (\partial_{xx} \psi_2(\xi_1), \dots, \partial_{xx} \psi_{N-1}(\xi_{N-1}))^T$ is a diagonal $(N - 2) \times (N - 2)$ matrix and an $(N - 2)$ -dimensional column vector, respectively, which make sure that our approximation satisfies the boundary conditions. In the above equations, the $(N - 2) \times (N - 2)$ matrices \mathbf{M}, \mathbf{D} and \mathbf{D}^2 are given by

$$\mathbf{M} = \text{diag}((w_1, \dots, w_{N-1})), \quad \mathbf{D}_{j,k} = \partial_x \psi_j(\xi_k),$$

$$\mathbf{D}^2_{j,k} = \partial_{xx} \psi_j(\xi_k).$$

We apply a first-order implicit-explicit method with time step δ for time discretization and obtain the discrete-time and discrete-space equations

$$(\mathbf{M} - \delta \nu \mathbf{M} \mathbf{D}^2) \mathbf{u}^{n+1} =$$

$$\mathbf{M} \left(\mathbf{u}^n + \delta \left(\mathbf{b}^n \circ \mathbf{D} \mathbf{b}^n - \mathbf{u}^n \circ \mathbf{D} \mathbf{u}^n + \Psi_{\mathbf{x}}^{\mathbf{B}} \mathbf{b}^n \right) \right) + \Delta \mathbf{W}_u^n,$$

$$(\mathbf{M} - \delta \mathbf{M} \mathbf{D}^2) \mathbf{b}^{n+1} =$$

$$\mathbf{M} \left(\mathbf{b}^n + \delta \left(\mathbf{b}^n \circ \mathbf{D} \mathbf{u}^n - \mathbf{u}^n \circ \mathbf{D} \mathbf{b}^n - \Psi_{\mathbf{x}}^{\mathbf{B}} \mathbf{u}^n + \Psi_{\mathbf{xx}}^{\mathbf{B}} \right) \right) + \Delta \mathbf{W}_b^n,$$

where

$$\Delta \mathbf{W}_u \sim \mathcal{N}(\mathbf{0}, \Sigma_u), \quad \Delta \mathbf{W}_b \sim \mathcal{N}(\mathbf{0}, \Sigma_b), \tag{56}$$

and

$$\Sigma_u = g_u^2 \delta \mathbf{M} \left(\mathbf{F}_s \mathbf{C} \mathbf{C}^T \mathbf{F}_s^T + \mathbf{F}_c \mathbf{C} \mathbf{C}^T \mathbf{F}_c^T \right) \mathbf{M}^T, \tag{57}$$

$$\Sigma_b = g_b^2 \delta \mathbf{M} \left(\mathbf{F}_s \mathbf{C} \mathbf{C}^T \mathbf{F}_s^T + \mathbf{F}_c \mathbf{C} \mathbf{C}^T \mathbf{F}_c^T \right) \mathbf{M}^T, \tag{58}$$

$$\mathbf{C} = \text{diag}((\alpha_1, \dots, \alpha_n)), \tag{59}$$

$$\mathbf{F}_s = (\sin(\pi), \sin(2\pi), \dots, \sin(m\pi))(\xi_1, \xi_2, \dots, \xi_m)^T, \tag{60}$$

$$\mathbf{F}_c = (\cos(\pi/2), \cos(3\pi/2), \dots, \cos(m\pi/2))(\xi_1, \xi_2, \dots, \xi_m)^T. \tag{61}$$

For our choice of α_k, β_k in Eq. (55), the state covariance matrices Σ_u and Σ_b are singular if $N > 12$. To diagonalize the state covariances we solve the symmetric eigenvalue problems Parlett (1998)

$$(\mathbf{M} - \delta \nu \mathbf{M} \mathbf{D}^2) \mathbf{v}_u = \Sigma_u \mathbf{v}_u \lambda^u,$$

$$(\mathbf{M} - \delta \mathbf{M} \mathbf{D}^2) \mathbf{v}_b = \Sigma_b \mathbf{v}_b \lambda^b,$$

and define the linear coordinate transformations

$$\mathbf{u} = \mathbf{V}_u(\mathbf{x}_u, \mathbf{y}_u)^T, \quad \mathbf{b} = \mathbf{V}_b(\mathbf{x}_b, \mathbf{y}_b)^T, \tag{62}$$

where the columns of the $(N - 2) \times (N - 2)$ -matrices \mathbf{V}_u and \mathbf{V}_b are the eigenvectors of $\mathbf{v}_u, \mathbf{v}_b$, respectively. The discretization using Legendre spectral elements works in our

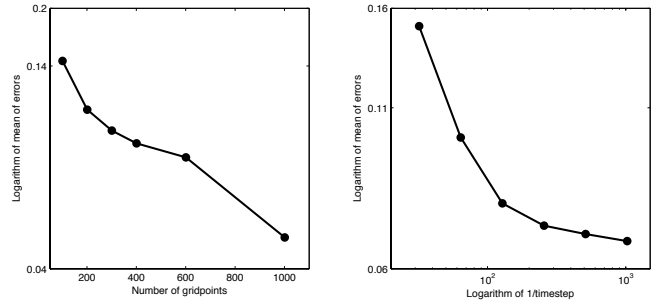


Fig. 2. Convergence of discretization scheme for geomagnetic equations. Left: Convergence in the number of spatial grid-points (log-linear scale). Right: Convergence in the time step (log-log scale).

favor here, because the matrices \mathbf{M} and \mathbf{D}^2 are symmetric so that we can diagonalize the left hand side simultaneously with the state covariance matrix to obtain

$$\mathbf{x}_u^{n+1} = \mathbf{f}_u(\mathbf{x}_u^n, \mathbf{y}_u^n, \mathbf{x}_b^n, \mathbf{y}_b^n) + \Delta \hat{\mathbf{W}}_u^n,$$

$$\mathbf{y}_u^{n+1} = \mathbf{g}_u(\mathbf{x}_u^n, \mathbf{y}_u^n, \mathbf{x}_b^n, \mathbf{y}_b^n),$$

$$\mathbf{x}_b^{n+1} = \mathbf{f}_b(\mathbf{x}_u^n, \mathbf{y}_u^n, \mathbf{x}_b^n, \mathbf{y}_b^n) + \Delta \hat{\mathbf{W}}_b^n,$$

$$\mathbf{y}_b^{n+1} = \mathbf{g}_b(\mathbf{x}_u^n, \mathbf{y}_u^n, \mathbf{x}_b^n, \mathbf{y}_b^n),$$

where $\mathbf{f}_u, \mathbf{f}_b$ are 10-dimensional vector functions, $\mathbf{g}_u, \mathbf{g}_b$ are $((N - 2) - 10)$ -dimensional vector functions and where

$$\hat{\mathbf{W}}_u^n \sim \mathcal{N}(\mathbf{0}, \text{diag}((\lambda_1^u, \lambda_2^u, \dots, \lambda_{10}^u))),$$

$$\hat{\mathbf{W}}_b^n \sim \mathcal{N}(\mathbf{0}, \text{diag}((\lambda_1^b, \lambda_2^b, \dots, \lambda_{10}^b))).$$

We test the convergence of our approximation as follows. To assess the convergence in the number of grid-points in space, we define a reference solution using $N = 2000$ grid-points and a time step of $\delta = 0.002$. We compute another approximation of the solution, using the same (discrete) BM as in the reference solution, but with another number of grid-points, say $N = 500$. We compute the error at $t = T = 0.2$,

$$e_x = \|(\mathbf{u}_{500}(x, T)^T, \mathbf{b}_{500}(x, T)^T) - (\mathbf{u}_{\text{Ref}}(x, T)^T, \mathbf{b}_{\text{Ref}}(x, T)^T)\|,$$

where $\|\cdot\|$ denotes the Euclidean norm, and store it. We repeat this procedure 500 times and compute the mean of the error norms and scale the result by the mean of the norm of the solution. The results are shown in the left panel of Fig. 2.

We observe a straight line, indicating super algebraic convergence of the scheme (as is expected from a spectral method).

Similarly, we check the convergence of the approximation in the time step by computing a reference solution with $N_{\text{Ref}} = 1000$ and $\delta_{\text{Ref}} = 2^{-12}$. Using the same BM as in the reference solution, we compute an approximation with time step δ and compute the error at $t = T = 0.2$, $e_t = \|(\mathbf{u}_\delta(x, T)^T, \mathbf{b}_\delta(x, T)^T) -$

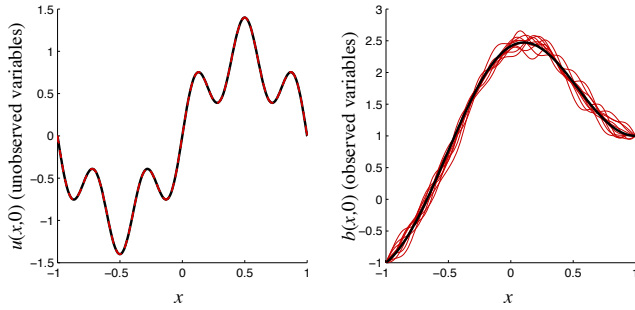


Fig. 3. Uncertainty in the initial state. Left: $u(x, 0)$ (unobserved). Right: $b(x, 0)$ (observed). Black: mean. Red: 10 realizations of the initial state.

$(\mathbf{u}_{\text{Ref}}(x, T)^T, \mathbf{b}_{\text{Ref}}(x, T)^T)$, and store it. We repeat this procedure 500 times and then compute the mean of these error norms, divided by the mean of the norm of the solution. The results are shown in the right panel of Fig. 2. We observe a first order decay in the error for time steps larger than $\delta = 0.02$ as is expected. The error has converged for time steps smaller than $\delta = 0.002$, so that a higher resolution in time does not improve the accuracy of the approximation.

Here we are satisfied with an approximation with $\delta = 0.002$ and $N = 300$ grid-points in space as in Fournier et al. (2007). The relatively small number of spatial grid-points is sufficient because the noise is very smooth in space and because the Legendre spectral elements accumulate nodes close to the boundaries and, thus, represent the steep boundary layer, characteristic of Eqs. (49)–(50), well even if N is small (see also Fournier et al., 2007).

4.2 Filtering results

We apply the implicit particle filter with gradient descent minimization and random maps (see algorithm 2 in Sect. 3), the simplified implicit particle filter (see Sect. 2.2) adapted to models with partial noise, a standard EnKF (without localization or inflation), as well as a standard SIR filter to the test problem Eqs. (49)–(50). The numerical model is given by the discretization described in the previous section with a random initial state. The distribution of the initial state is Gaussian with mean $u(x, 0)$, $b(x, 0)$ as in Eqs. (51)–(52) and with a covariance Σ_u , Σ_b given by Eqs. (57)–(58). In Fig. 3, we illustrate the uncertainty in the initial state and plot 10 realizations of the initial state (grey lines) along with its mean (black lines). We observe that the uncertainty in u_0 is small compared to the uncertainty in b_0 .

The data are the values of the magnetic field b , measured at k equally spaced locations in $[-1, 1]$ and corrupted by noise:

$$\mathbf{z}^l = \mathbf{H}\mathbf{b}^{q(l)} + s\mathbf{V}^l, \quad (63)$$

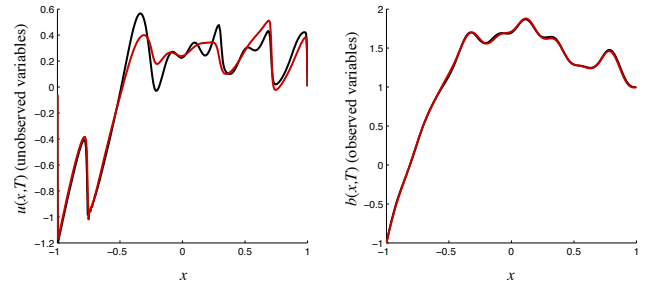


Fig. 4. Outcome of a twin experiment. Black: true state $u(x, 0.2)$ (left) and $b(x, 0.2)$ (right). Red: reconstruction by implicit particle filter with 4 particles.

where $s = 0.001$ and where \mathbf{H} is a $k \times m$ -matrix that maps the numerical approximation \mathbf{b} (defined at the GLL nodes) to the locations where data is collected. We consider data that are dense in time ($r = 1$) as well as sparse in time ($r > 1$). The data are sparse in space and we consider two cases: (i) we collect the magnetic field b at 200 equally spaced locations; and (ii) we collect the magnetic field b at 20 equally spaced locations. The velocity u is unobserved and it is of interest to study how the various data assimilation techniques make use of the information in b to update the unobserved variables u (Fournier et al., 2007, 2010).

To assess the performance of the filters, we ran 100 twin experiments. A twin experiment amounts to (i) drawing a sample from the initial state and running the model forward in time until $t = T = 0.2$ (one fifth of a magnetic diffusion time Fournier et al., 2007), (ii) collecting the data from this free model run, and (iii) using the data as the input to a filter and reconstructing the state trajectory. Figure 4 shows the result of one twin experiment for $r = 4$.

For each twin experiment, we calculate and store the error at $t = T = 0.2$ in the velocity, $e_u = \|\mathbf{u}(x, T) - \mathbf{u}_{\text{Filter}}(x, T)\|$, and in the magnetic field, $e_b = \|\mathbf{b}(x, T) - \mathbf{b}_{\text{Filter}}(x, T)\|$. After running the 100 twin experiments, we calculate the mean of the error norms (not the mean error) and the variance of the error norms (not the variance of the error) and scale the results by the mean of the norm of u and b , respectively. All filters we tested were “untuned”, i.e. we have not adjusted or inserted any free parameters to boost the performance of the filters.

Figure 5 shows the results for the implicit particle filter, the EnKF as well as the SIR filter for 200 measurement locations and for $r = 10$.

The figure indicates that the implicit particle filter requires only very few particles (~ 4 – 10) to yield accurate state estimates with less than 1% error in the observed variables b and less than 15% error in the unobserved velocity u . The SIR filter with 1000 particles gives significantly larger errors (about 10% in the observed variable b and 20% in the unobserved variable u) and much larger variances in the errors.

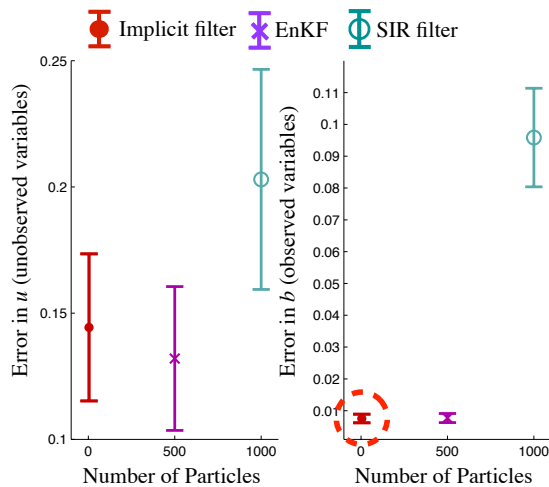


Fig. 5. Filtering results for data collected at a high spatial resolution (200 measurement locations). The errors at $t = 0.2$ of the implicit particle filter (red), EnKF (purple) and SIR filter (green) are plotted as a function of the number of particles. The error bars represent the mean of the errors and mean of the standard deviations of the errors.

The EnKF requires about 500 particles to achieve the accuracy of the implicit filter with only 4 particles.

In the experiments, we observed that the minimization in implicit particle filtering typically converged after 4–10 steps (depending on r , the gap in time between observations). The convergence criterion was to stop the iteration when the change in F_j was less than 10%. A more accurate minimization did not improve the results significantly, so that we were satisfied with a relatively crude estimate of the minimum in exchange for a speed-up of the algorithm. We found λ by solving Eq. (11) with Newton’s method using $\lambda^0 = 0$ as initial guess and observed that it converged after about eight steps. The convergence criterion was to stop the iteration if $|F(\lambda) - \phi - \rho| \leq 10^{-3}$, because the accurate solution of this scalar equation is numerically inexpensive. We resampled using algorithm 2 in Arulampalam et al. (2002), if the effective sample size M_{Eff} in Eq. (19) divided by the number of particles M is less than 90% of the number of particles.

To further investigate the performance of the filters, we run more numerical experiments and vary the availability of the data in time, as well as the number of particles. Figure 6 shows the results for the implicit particle filter, the simplified implicit particle filter, the EnKF and the SIR filter for 200 measurement locations and for $r = 1, 2, 4, 10$.

We observe from Fig. 6, that the error statistics of the implicit particle filter have converged, so that there is no significant improvement when we increase the number of particles to more than 10. In fact, the numerical experiments suggest that no more than 4 particles are required here. Independent of the gap between the observations in time, we observe an error of less than 1% in the observed variable b . The error in

the unobserved variable u , however, depends strongly on the gap between observations and, for a large gap, is about 15%.

The reconstructions of the observed variables by the simplified implicit particle filter are rather insensitive to the availability of data in time and, with 20 particles, the simplified filter gives an error in the observed quantity b of less than 1%. The errors in the unobserved quantity u depend strongly on the gap between the observations and can be as large as 15%. The error statistics in Fig. 6 have converged and only minor improvements can be expected if the number of particles is increased to more than 20.

The SIR filter required significantly more particles, than the implicit filter or simplified implicit filter. Independent of the gap between observations, the errors and their variances are larger than for the implicit and simplified implicit filter, even if the number of particles for SIR is set to 1000. The EnKF performs well and, for about 500 particles, gives results that are comparable to those of the implicit particle filter. The EnKF may give similarly accurate results at a smaller number of particles if localization and inflation techniques are implemented.

The errors in the reconstructions of the various filters are not Gaussian, so that an assessment of the errors based on the first two moments is incomplete. In the two panels on the right of Fig. 7, we show histograms of the errors of the implicit filter (10 particles), simplified implicit filter (20 particles), EnKF (1000 particles) and SIR filter (1000 particles) for $r = 10$ model steps between observations.

We observe that the errors of the implicit filter, simplified implicit filter and EnKF are centered to the left of the diagrams (at around 10% in the unobserved quantity u and about 1% for the observed quantity b) and show a considerably smaller spread than the errors of the SIR filter, which are centered at much larger errors (20% in the unobserved quantity u and about 9% for the observed quantity b). A closer look at the distribution of the errors thus confirms our conclusions we have drawn from an analysis based on the first two moments.

We further assess the performance of the filters by considering their effective sample size (19), which measures the quality of the particles ensemble Doucet et al. (2001). A large effective sample size indicates a good ensemble, i.e. the samples are independent and each of them contributes significantly to the approximation of the conditional mean; a small effective sample size indicates a “bad ensemble”, i.e. most of the samples carry only a small weight. We computed the effective sample size for the implicit particle filter, the simplified implicit particle filter and the SIR filter after each assimilation, and compute the average after each of 100 twin experiments. In Table 1, we show the average effective sample size (averaged over all 100 twin experiments and scaled by the number of particles) for a gap of $r = 10$ model steps between observations.

We observe that the effective sample size of the implicit filter is about 10 times larger than the effective sample size

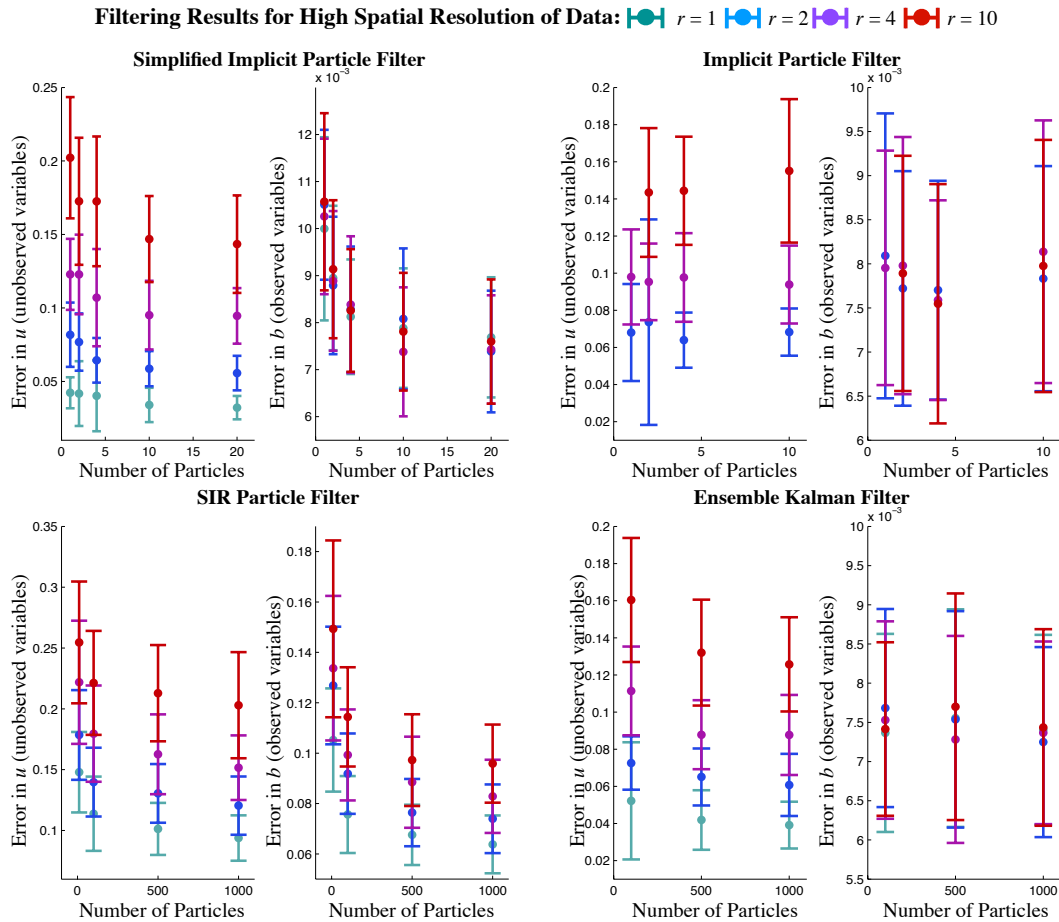


Fig. 6. Filtering results for data collected at a high spatial resolution (200 measurement locations). The errors at $t = 0.2$ of the simplified implicit particle filter (upper left), implicit particle filter (upper right), SIR filter (lower left) and EnKF (lower right) are plotted as a function of the number of particles and for different gaps between observations in time. The error bars represent the mean of the errors and mean of the standard deviations of the errors.

Table 1. Effective sample size of the simplified implicit filter, the implicit filter and the SIR filter.

	Simplified implicit filter	Implicit filter	SIR filter
M_{Eff}/M	0.20	0.19	0.02

of the SIR filter. This result indicates that the particles of the implicit filter are indeed focussed towards the high probability region of the target pdf.

Next, we decrease the spatial resolution of the data to 20 measurement locations and show filtering results from 100 twin experiments in Fig. 8.

The results are qualitatively similar to those obtained at a high spatial resolution of 200 data points per observation. The two panels on the right of Fig. 7, show histograms of the errors of the implicit filter (10 particles), simplified implicit filter (20 particles), EnKF (1000 particles) and SIR

filter (1000 particles) for $r = 10$ model steps between observations. Again, the results are qualitatively similar to the results we obtained at a higher spatial resolution of the data.

We observe for the implicit particle filter that the errors in the unobserved quantity are insensitive to the spatial resolution of the data, while the errors in the observed quantity are determined by the spatial resolution of the data and are rather insensitive to the temporal resolution of the data. These observations are in line with those reported in connection with a strong 4-D-Var algorithm in Fournier et al. (2007). All other filters we have tried show a dependence of the errors in the observed quantity on the temporal resolution of the data.

The reason for the accurate state estimates of the implicit particle filter, obtained at a low number of particles, is its direct use of the data: the implicit particle filter uses the information from the model, as well as from the data to search for the high probability region of the target pdf. This search is performed by the particle-by-particle minimization of the functions F_j . The implicit filter then generates samples

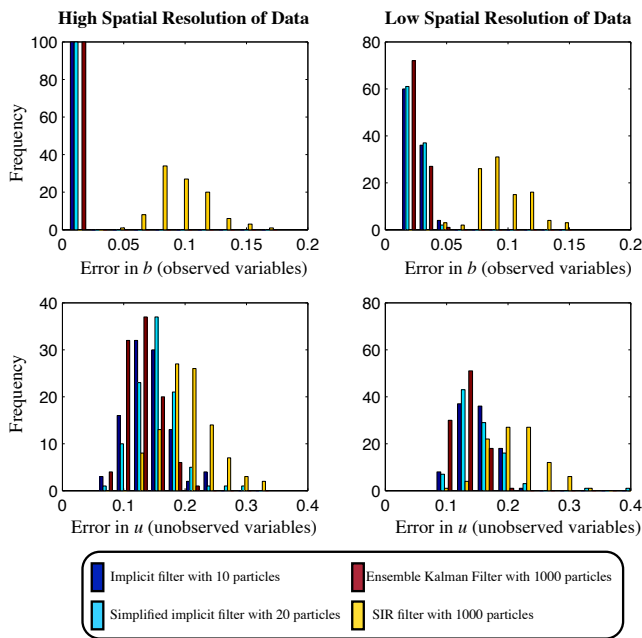


Fig. 7. Histogram of errors at $t = 0.2$ of the implicit filter, simplified implicit filter, EnKF and SIR filter. Left: data are available at a high spatial resolution (200 measurement locations) and every $r = 10$ model steps. Right: data are available at a low spatial resolution (20 measurement locations) and every $r = 10$ model steps.

within the high probability region by solving Eq. (11). Because the implicit filter focusses attention on regions of high probability, only a few samples are required for a good accuracy of the state estimate (the conditional mean). The information in the observations of the magnetic field b propagates to the filtered updates of the unobserved velocity u via the nonlinear coupling in Eqs. (49)–(50).

The EnKF on the other hand uses the data only at times when an observation is available. The state estimates at all other times are generated by the model alone. Moreover, the nonlinearity, and thus the coupling of observed and unobserved quantities, is represented only in the approximation of the state covariance matrix, so that the information in the data propagates slowly to the unobserved variables. The situation is very similar for the simplified implicit filter.

The SIR filter requires far more particles than the implicit filter because it samples the low probability region of the target pdf with a high probability. The reason is that the overlap of the pdf generated by the model alone and the target pdf becomes smaller and smaller as the data becomes sparser and sparser in time. For that reason, the SIR filter must generate far more samples to at least produce a few samples that are likely with respect to the observations. Moreover, the data is only used to weigh samples that are generated by the model alone; it does not use the nonlinear coupling between observed and unobserved quantities, so that the information in

the data propagates very slowly from the observed to the unobserved quantities.

In summary, we observe that the implicit particle filter yields the lowest errors with a small number of particles for all examples we considered, and performs well and reliably in this application. The SIR and simplified implicit particle filters can reach the accuracy of the implicit particle filter, at the expense that the number of particles is increased significantly. The very small number of particles required for a very high accuracy make the implicit filter the most efficient filter for this problem. Note that the partial noise works in our favor here, because the dimension of the space the implicit filter operates in is 20, rather than the state dimension 600.

Finally, we wish to compare our results with those in Fournier et al. (2007), where a strong constraint 4-D-Var algorithm was applied to the deterministic version of the test problem. Fournier and his colleagues used “perfect data,” i.e. the observations were not corrupted by noise, and applied a conjugate-gradient algorithm to minimize the 4-D-Var cost function. The iterative minimization was stopped after 5000 iterations. With 20 observations in space and a gap of $r = 5$ model steps between observations, an error of about 1.2% in u and 4.7% in b was achieved. With the implicit filter, we can get to a similar accuracy at the same spatial resolution of the data, but with a larger gap of $r = 10$ model steps between observations. However, the 4-D-Var approach can handle larger uncertainties and errors in the velocity field. The reason is that the initial conditions are assumed to be known (at least roughly) when we assimilate data sequentially. This assumption is of course not valid in “real” geomagnetic data assimilation (the velocity field is unknown), however a strong 4-D-Var calculation can be used to obtain approximate and uncertain initial conditions to then start assimilating new data with a filter. The implicit particle filter then reduces the memory requirements because it operates in the 20-dimensional subspace of the forced variables and assimilates the data sequentially. Each minimization is thus not as costly as a 600-dimensional strong constraint 4-D-Var minimization. Alternatively, one could extend the implicit particle filter presented here to include the initial conditions as variables of the F_j s. This set up would allow for larger uncertainties in the initial conditions than what we presented here.

5 Conclusions

We have considered implicit particle filters for data assimilation. Previous implementations of the implicit particle filter rely on finding the Hessians of functions F_j of the state variables. Finding these Hessians can be expensive if the state dimension is large and can be cumbersome if the second derivatives of the F_j s are hard to calculate. We presented a new implementation of the implicit filter combining gradient descent minimization with random maps. This

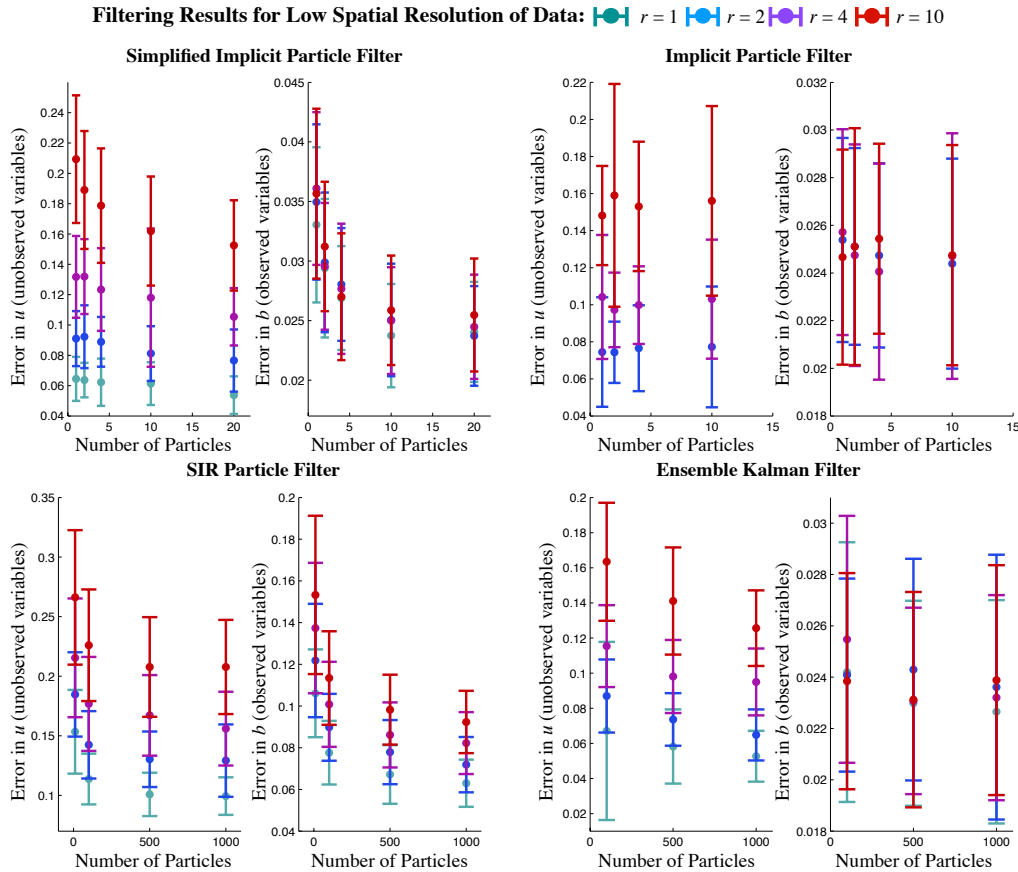


Fig. 8. Filtering results for data collected at a low spatial resolution (20 measurement locations). The errors at $T = 0.2$ of the simplified implicit particle filter (upper left), implicit particle filter (upper right), SIR filter (lower left) and EnKF (lower right) are plotted as a function of the number of particles and for different gaps between observations in time. The error bars represent the mean of the errors and mean of the standard deviations of the errors.

new implementation avoids the often costly calculation of the Hessians and, thus, reduces the memory requirements compared to earlier implementations of the filter.

We have considered models for which the state covariance matrix is singular or ill-conditioned. This happens often, for example, in geophysical applications in which the noise is smooth in space or if the model includes conservation laws with zero uncertainty. Previous implementations of the implicit filter are not applicable here and we have shown how to use our new implementation in this situation. The implicit filter is found to be more efficient than competing methods because it operates in a space whose dimension is given by the rank of the state covariance matrix rather than the dimension of the state space.

We applied the implicit filter in its new implementation to a test problem in geomagnetic data assimilation. The implicit filter performed well in comparison to other data assimilation methods (SIR, EnKF and 4-D-Var) and gave accurate state estimates with a small number of particles and at a low computational cost. We have studied how the various data assimilation techniques use the available data to propagate in-

formation from observed to unobserved quantities and found that the implicit particle filter uses the data in a direct way, propagating information to unobserved quantities faster than competing methods. The direct use of the data is the reason for the small errors in reconstructions of the state.

Acknowledgements. We thank the three reviewers for their interesting questions and helpful comments. We would like to thank our collaborators Ethan Atkins at UC Berkeley, and Professors Robert Miller, Yvette Spitz, and Brad Weir at Oregon State University, for their comments and helpful discussion. We thank Robert Saye for careful proofreading of early versions of this manuscript. This work was supported in part by the Director, Office of Science, Computational and Technology Research, US Department of Energy under Contract No. DE-AC02-05CH11231, and by the National Science Foundation under grants DMS-0705910 and OCE-0934298.

Edited by: O. Talagrand

Reviewed by: A. Fournier, C. Snyder, and P. J. van Leeuwen

References

- Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Trans. Signal Process.*, 50, 174–188, 2002.
- Atkins, E., Morzfeld, M., and Chorin, A. J.: The implicit particle filter and its connection to variational data assimilation, in review, 2012.
- Aubert, J. and Fournier, A.: Inferring internal properties of Earth's core dynamics and their evolution from surface observations and a numerical geodynamo model, *Nonlin. Processes Geophys.*, 18, 657–674, doi:10.5194/npg-18-657-2011, 2011.
- Bennet, A., Leslie, L., Hagelberg, C., and Powers, P.: A Cyclone prediction using a barotropic model initialized by a general inverse method, *Mon. Weather Rev.*, 121, 1714–1728, 1993.
- Bickel, P., Li, B., and Bengtsson, T.: Sharp failure rates for the bootstrap particle filter in high dimensions, *IMS Collections: Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, 3, 318–329, 2008.
- Bocquet, M., Pires, C., and Wu, L.: Beyond Gaussian statistical modeling in geophysical data assimilation, *Mon. Weather Rev.*, 138, 2997–3023, 2010.
- Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A.: *Spectral Methods. Fundamentals in Single Domains*, Springer, Berlin, Germany, 2006.
- Chorin, A. and Hald, O.: *Stochastic Tools in Mathematics and Science*, Springer, 2nd Edn., 2009.
- Chorin, A. J. and Tu, X.: Implicit sampling for particle filters, *Proc. Natl. Aca. Sci.*, 106, 17249–17254, 2009.
- Chorin, A. J., Morzfeld, M., and Tu, X.: Implicit particle filters for data assimilation, *Comm. Appl. Math. Comput. Sci.*, 5, 221–240, 2010.
- Chueshov, I.: Gevrey regularity of random attractors for stochastic reaction-diffusion equations, *Random Operators and Stochastic Equations*, 8, 143–162, 2000.
- Courtier, P.: Dual formulation of four-dimensional variational assimilation, *Q. J. Roy. Meteorol. Soc.*, 123, 2449–2461, 1997.
- Courtier, P., Thepaut, J., and Hollingsworth, A.: A strategy for operational implementation of 4D-Var, using an incremental approach, *Q. J. Roy. Meteorol. Soc.*, 120, 1367–1387, 1994.
- Deville, M. O., Fischer, P. F., and Mund, E. H.: *Higher-Order Methods for Incompressible Flow*, Cambridge University Press, Oxford, UK, 2006.
- Doucet, A., Godsill, S., and Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering, *Stat. Comput.* 10, 197–208, 2000.
- Doucet, A., de Freitas, N., and Gordon, N.: *Sequential Monte Carlo methods in practice*, Springer, 2001.
- Evensen, G.: *Data assimilation*, Springer, 2007.
- Fletcher, R.: *Practical Methods of Optimization*, Wiley, 2nd Edn., 1987.
- Fournier, A., Eymin, C., and Alboussière, T.: A case for variational geomagnetic data assimilation: insights from a one-dimensional, nonlinear, and sparsely observed MHD system, *Nonlin. Processes Geophys.*, 14, 163–180, doi:10.5194/npg-14-163-2007, 2007.
- Fournier, A., Hulot, G., Jault, D., Kuang, W., Tangborn, A., Gillet, N., Canet, E., Aubert, J., and Lhuillier, F.: An introduction to data assimilation and predictability in geomagnetism, *Space Sci. Rev.*, 155, 247–291, 2010.
- Gelb, A.: *Applied optimal estimation*, MIT Press Cambridge, 1974.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEEE Proc. Radar Signal Proc.*, 140, 107–113, 1993.
- Ide, K., Courtier, P., Ghil, M., and Lorenc, A.: Unified notation for data assimilation: operational, sequential and variational, *J. Meteorol. Soc. Jpn.*, 75, 181–189, 1997.
- Jentzen, A. and Kloeden, P. E.: Overcoming the order barrier in the numerical approximation of stochastic partial differential equations with additive space-time noise, *Proc. Roy. Soc. A*, 465, 649–667, 2009.
- Julier, S. J. and Uhlmann, J. K.: A new extension of the Kalman filter to nonlinear systems, *International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 3, 182–193, 1997.
- Kalman, R. E.: A New Approach to Linear Filtering and Prediction Theory, *J. Basic Eng. – T. ASME*, 82, 35–48, 1960.
- Kalman, R. E. and Bucy, R. S.: New Results in Linear Filtering and Prediction Theory, *J. Basic Eng. – T. ASME*, 83, 95–108, 1961.
- Kloeden, P. E. and Platen, E.: *Numerical solution of stochastic differential equations*, Springer, 1999.
- Kurapov, A. L., Egbert, G. D., Allen, J. S., and Miller, R. N.: Representer-based variational data assimilation in a nonlinear model of nearshore circulation, *J. Geophys. Res.*, 112, C11019, doi:10.1029/2007JC004117, 2007.
- Lord, G. J. and Rougemont, J.: A numerical scheme for stochastic PDEs with Gevrey regularity, *J. Num. Analysis*, 24, 587–604, 2004.
- Miller, R. N., Ghil, M., and Gauthiez, F.: Advanced data assimilation in strongly nonlinear dynamical systems, *J. Atmos. Sci.*, 51, 1037–1056, 1994.
- Miller, R. N., E. F. Carter, J., and Blue, S. T.: Data assimilation into nonlinear stochastic models, *Tellus*, 51, 167–194, 1999.
- Moral, P. D.: Measure-valued processes and interacting particle systems. Application to nonlinear filtering problems, *Ann. Appl. Probability*, 8, 438–495, 1998.
- Moral, P. D.: *Feynman-Kac Formulae*, Springer, NY, 2004.
- Morzfeld, M., Tu, X., Atkins, E., and Chorin, A.: A random map implementation of implicit filters, *J. Comput. Phys.*, 213, 2049–2066, 2012.
- Nocedal, J. and Wright, S. T.: *Numerical Optimization*, Springer, 2nd Edn., 2006.
- Parlett, B. N.: The symmetric eigenvalue problem, *Classics in Applied Mathematics*, Vol. 20, Society for Industrial and Applied Mathematics, Philadelphia, 1998.
- Snyder, C., Bengtsson, T., Bickel, P., and Andersson, J.: Obstacles to high-dimensional particle filtering, *Mon. Weather Rev.*, 136, 4629–4640, 2008.
- Stengel, R.: *Optimal control and estimation*, Dover Publications, 1994.
- Sun, Z., Tangborn, A., and Kuang, W.: Data assimilation in a sparsely observed one-dimensional modeled MHD system, *Nonlin. Processes Geophys.*, 14, 181–192, doi:10.5194/npg-14-181-2007, 2007.
- Talagrand, O.: *Assimilation of Observations, an Introduction*, *J. Meteorol. Soc. Jpn.*, 75, 191–209, 1997.
- Talagrand, O. and Courtier, P.: Variational assimilation of meteorological observations with the adjoint vorticity equation, I: The-

- ory, Q. J. Roy. Meteorol. Soc., 113, 1311–1328, 1987.
- Tremolet, Y.: Accounting for an imperfect mode in 4D-Var, Q. J. Roy. Meteorol. Soc., 132, 2483–2504, 2006.
- van Leeuwen, P.: Particle filtering in geophysical systems, Mon. Weather Rev., 137, 4089–4114, 2009.
- van Leeuwen, P.: Nonlinear data assimilation in geosciences: an extremely efficient particle filter, Q. J. Roy. Meteorol. Soc., 136, 1991–1999, 2010.
- Weare, J.: Particle filtering with path sampling and an application to a bimodal ocean current model, J. Comput. Phys., 228, 4312–4331, 2009.
- Zupanski, D.: A General Weak Constraint Applicable to Operational 4DVAR Data Assimilation systems, Mon. Weather Rev., 125, 2274–2292, 1997.