



Research article

A Hybrid Monte Carlo Sampling Filter for Non-Gaussian Data Assimilation

Ahmed Attia^{1,*} and Adrian Sandu¹

¹ Computational Science Laboratory,
Department of Computer Science,
Virginia Polytechnic Institute and State University,
Blacksburg, Virginia, 24060, USA

* **Correspondence:** Email: attia@vt.edu; Tel: +1-540-231-6186.

Abstract: Data assimilation combines information from models, measurements, and priors to obtain improved estimates of the state of a dynamical system such as the atmosphere. Ensemble-based data assimilation approaches such as the Ensemble Kalman filter (EnKF) have gained wide popularity due to their simple formulation, ease of implementation, and good practical results. Many of these methods are derived under the assumption that the underlying probability distributions are Gaussian. It is well accepted, however, that the Gaussianity assumption is too restrictive when applied to large nonlinear models, nonlinear observation operators, and large levels of uncertainty. When the Gaussianity assumptions are severely violated, the performance of EnKF variations degrades. This paper proposes a new ensemble-based data assimilation method, named the “*sampling filter*”, which obtains the analysis by sampling directly from the posterior distribution. The sampling strategy is based on a Hybrid Monte Carlo (HMC) approach that can handle non-Gaussian probability distributions. Numerical experiments are carried out using the Lorenz-96 model and observation operators with different levels of non-linearity and differentiability. The proposed filter is also tested with shallow water model on a sphere with linear observation operator. Numerical results show that the sampling filter performs well even in highly nonlinear situations where the traditional filters diverge.

Keywords: Data assimilation; variational methods; ensemble filters; Markov chain; hybrid Monte-Carlo

1. Introduction

Data assimilation (DA) is the process of combining information from models, measurements, and priors - all with associated uncertainties - in order to better describe the the true state of a physical system. The ultimate goal of DA is to efficiently represent the posterior PDF that fully describes

the uncertainty in the analysis. Two families of methods, variational and ensemble based methods, have proved very successful in real applications. Variational methods, rooted in control theory, require costly developments of tangent linear and adjoint models [32]. Ensemble-based sequential DA schemes are rooted in statistical estimation theory. The ensemble Kalman Filter was introduced by Evensen [17] and has undergone considerable developments since then. EnKF formulations fall in one of two classes, namely stochastic or deterministic formulations [55]. In the stochastic approach, each ensemble member is updated using a perturbed version of the observation vector [11, 26]. In the deterministic formulation (which leads to square root ensemble filters [5, 9, 46, 55, 59]) no observation noise is added, but transformations of the covariance matrix are applied such as to recover the correct analysis statistics.

All variants of the EnKF work well in case of linear observations [19], however in real applications the observation operators are in general nonlinear. EnKF can accommodate nonlinear observation operators using linearization, in the spirit of the extended Kalman filter [60]. An alternative approach to handle the non-linearity of observation operators is to use the difference between nonlinear operators evaluated at two states instead of the linearized version; this approach can result in mathematical inconsistencies [60]. A different approach to deal with nonlinear observations is to pose a nonlinear estimation problem in a subspace spanned by the ensemble members, and to compute the maximum a posteriori estimate in that subspace. This leads to the maximum likelihood ensemble filter (MLEF) proposed by Zupanski [60]. MLEF minimizes a cost function that depends on nonlinear observation operators. MLEF doesn't require the observation operator to be differentiable and uses a difference approximation of the Jacobian of the observation operator. However, this approach may diverge if the observation operator is highly nonlinear or the observations are very sparse. In addition it is inherently assumed that the posterior distribution is Gaussian; the MLEF maximum a posteriori probability estimate may face difficulties in case of multimodal distributions. The iterative ensemble Kalman filter (IEnKF) [23, 47] handles nonlinearity in observations and models by repeating analyses steps with the same observations. Each iteration of IEnKF still assumes that the underlying probability distributions are Gaussian and the analysis state is the mode of the posterior. The "running in place" (RIP) EnKF scheme [33] uses a no-cost EnKS and repeatedly assimilates the observations over each assimilation window several times. Another nonlinear extension of EnKF incorporates a non-linear change of variables (anamorphosis function) [50] and executes the analysis step in a space where the distributions of the transformed variables are Gaussian.

Particle filters (PF) [15, 22, 34, 56] are another family of nonlinear and non-Gaussian methods. PF performs well with small dimensional systems. For large dimensional systems, or when the number of independent observations increases, PF is more likely to suffer from filter degeneracy [51], a situation in which only one particle gets the significant weight and the all the other weights become close to zero. Development of efficient particle filters for large scale nonlinear and non-Gaussian DA is an active area of research. For example, in the merging particle filter (MPF) [41], the posterior particles are formulated as linear combinations of the resampled particles from the prior at the measurement time, in an attempt to reduce the variance in the weights and to avoid filter degeneracy. Although the mean and the covariance of the posterior are preserved, this filter does not guarantee the shape of the posterior PDF is preserved in the filtered ensemble. As mentioned in [41] if the posterior PDF is significantly non-Gaussian, the MPF will likely produce rather poor estimates. The performance of EnKF, MLEF, and PF when observation operators are highly nonlinear was tested in [28, 29].

A promising approach for sequential Monte Carlo sampling from the posterior density is the implicit particle filter [12]. This algorithm directs the sampling towards the regions of high density areas in the posterior and in this way it controls the number of particles in case of very high dimensional state spaces. The implicit sampling filter, however, is expensive: it requires solving a set of large algebraic equations for each particle. Another enhancement over the standard particle filter is the equivalent-weights particle filter [1, 57, 58] which incorporates a relaxation proposal density to steer the particles towards observations and consequently towards high-probability regions of the posterior PDF, and also uses an equivalent-weights proposal density to insure that the retained particles are given similar relative weights thus avoiding filter degeneracy.

In addition to the need of fully non-Gaussian DA schemes, efficient ensemble replenishment methods are needed to support parallel implementations of the current ensemble-based DA algorithms. In parallel implementations of EnKF each ensemble member typically runs on a separate set of nodes, and when any of the nodes dies out the corresponding ensemble member is lost. The ensemble number of members is reduced and after several node failures the entire EnKF DA process is in jeopardy. To continue the EnKF process efficiently, it is necessary to replace the lost ensemble member with a new one that is drawn independently from the underlying probability distribution.

Markov Chain Monte Carlo (MCMC) is a family of algorithms designed to sample from a given probability distribution by generating a Markov chain whose invariant (stationary) distribution is the target PDF. The traditional wisdom considers MCMC sampling methods to be a gold standard [35] in DA, which is however impractical in case of high dimensions due the massive computational cost required to achieve convergence and explore the whole state space. Development of scalable MCMC algorithms to sample efficiently from high dimensional space is an active research area.

A very promising approach for large systems is Hybrid Monte Carlo (HMC), a variant of MCMC sampling that incorporates an auxiliary variable and takes advantage of the properties of Hamiltonian system dynamics [16]. HMC has been considered before in the context of DA. [6] solve a nonlinear inverse problem that finds a solution to the shallow water equations such as to minimize the residual between the trace of this solution and ill-posed boundary conditions. Once the minimum is reached posterior error statistics are approximated by sampling the nearby states using HMC. They use a simulated annealing framework, where at low temperatures a minimum is obtained, and at high temperatures a posterior sample is obtained. [19, Chapter 6] discusses the solution of weak-constraint 4D-Var problem using a gradient method and employing HMC to estimate the analysis errors. The use of simulated annealing with HMC random walk as an alternative to gradient-based minimization is discussed in [6, 19].

Alexander, Eiyink, and Restrepo [2] advocate the use of HMC for solving smoothing problems. They employ a generalized version of HMC aimed at shortening the decorrelation, i.e., the number of chain steps needed to ensure independence of the generated states. Generalized HMC uses the dynamics of a modified system that is no longer Hamiltonian. There are several differences between this work and [2]. In [2] the model error, represented by additive random noise, is the only source of uncertainty; here we consider the state of the system to be uncertain due to uncertainties in initial conditions and model equations. A leap-frog discretization is employed in [2], while here we consider high order symplectic time discretizations. Finally, they test their method on a one-dimensional system. We note that the generalized HMC method employed in [2] uses the dynamics of a modified system that is no longer Hamiltonian. Finally, the tests performed in [2] focus on a one-dimensional system,

while here we carry out the numerical experiments with a multidimensional nonlinear model.

The current advances in sampling algorithms make it feasible to solve inverse problems with large models by directly sampling from the posterior probability distribution. This work develops a sequential ensemble-based DA filtering technique that can accommodate non-Gaussian posterior distributions and can be efficiently applied in operational situations. Moreover, it can be used as an ensemble replenishment tool to support parallel implementations of other ensemble based DA schemes. The approach is based on applying HMC sampling in a recursive manner to solve the sequential filtering problem. The new fully nonlinear sampling filter can accommodate nonlinear observation operators and it does not require the target probability distribution to be Gaussian. The filter is open to further improvements such as the use of the generalized (nonlocal) version described in [2]. We note that a smoothing approach using MCMC to sample the posterior distribution is discussed in [14, 35], however that approach is not based on HMC.

The paper is organized as follows. An overview of the DA problem and widely-used solution strategies is given in Section 2. Sampling MCMC and HMC algorithms are summarized in Section 3. The proposed sampling filter is presented in Section 4. Numerical experiments, and a comparison of the sampling filter against the traditional EnKF, MLEF, and IEnKF methods, are given in Section 5. Conclusions are drawn in Section 6.

2. Data Assimilation

This section provides a brief overview of the DA problem and of several solution strategies, and highlights the motivation behind the present research.

2.1. Problem formulation

DA combines information from prior (background) knowledge, a numerical model, and observations, all with associated errors, to describe the posterior distribution of the state of a physical system.

The background represents the best estimate of the true state *prior* to any measurement being available. The background errors (uncertainties) are typically assumed to have a Gaussian distribution $\mathbf{x}^b - \mathbf{x}^{\text{true}} \in \mathcal{N}(0, \mathbf{B})$, where \mathbf{x}^b is the background state, \mathbf{x}^{true} is the unknown true state, and \mathbf{B} is the background error covariance matrix. The Gaussian assumption is widely used and we will follow it as well. As discussed in Section 4 this assumption can be inaccurate, due to the recursive application of the sampling filter, and better approximations to the prior distribution might be beneficial.

The numerical model propagates the initial model state (initial condition) $\mathbf{x}_0 \in \mathbb{R}^{\text{Nvar}}$ at time t_0 to future states $\mathbf{x}_k \in \mathbb{R}^{\text{Nvar}}$ at times t_k :

$$\mathbf{x}_k = \mathcal{M}_{t_0 \rightarrow t_k}(\mathbf{x}_0), \quad t_0 \leq t_k \leq t_F, \quad (1)$$

where t_0 and t_F are the beginning and the end points of the simulation time interval. The model solution operator \mathcal{M} represents, for example, a discrete approximation of the partial differential equations that govern the evolution of the dynamical system (e.g., the atmosphere). The state space is typically large, e.g., $\text{Nvar} \sim 10^6 - 10^9$ variables for atmospheric simulations.

Small perturbations $\delta\mathbf{x}$ of the state of the system evolve according to the tangent linear model:

$$\delta\mathbf{x}_k = \mathbf{M}_{t_0 \rightarrow t_k}(\mathbf{x}_0) \cdot \delta\mathbf{x}_0, \quad t_0 \leq t_k \leq t_F, \quad (2)$$

where $\mathbf{M} = \mathcal{M}$ is the linearized model solution operator.

Observations of the true state are available at discrete time instants t_k , $t_0 \leq t_k \leq t_F$,

$$\mathbf{y}_k = \mathbf{y}(t_k) = \mathcal{H}_k(\mathbf{x}_k) + \varepsilon_k, \quad k = 0, 1, \dots, N_{\text{obs}} - 1.$$

The observation operator \mathcal{H}_k maps the state space to the observation space at time t_k . The observations are corrupted by measurement and representativeness errors [13], which are also assumed to have a normal distribution, $\varepsilon_k \in \mathcal{N}(0, \mathbf{R}_k)$, where \mathbf{R}_k is the observation error covariance matrix at time t_k .

Data assimilation combines the background estimate, the measurements, and the model to obtain an improved estimate \mathbf{x}^a , called the “analysis” (or *posterior*), of the true state \mathbf{x}^{true} , together with the corresponding analysis probability distribution. Two approaches for solving the DA problem have gained widespread popularity, variational and ensemble-based methods. The sampling filter proposed in this paper belongs to the latter family.

Since EnKF variations are the most popular ensemble-based algorithms in practice, we compare the new methodology against EnKF, MLEF, IEnKF. These schemes are briefly reviewed in the following subsections.

2.2. The ensemble Kalman filter

Kalman filters (KF) [30, 31] are sequential DA methodologies, where measurements are incorporated at the time moment when they become available. Sequential DA algorithms proceed in two steps, namely, *forecast* and *analysis*. In the forecast step, the state of the system is propagated forward by the model equations (1) to the next time point where observations are available, producing a forecast of the state of the system, and a forecast error covariance matrix is presented to quantify the uncertainty of the forecast.

The ensemble Kalman filter (EnKF) [11, 17, 18, 26] takes a Monte-Carlo approach to representing the uncertainty. An ensemble of N_{ens} states ($\mathbf{x}_{k-1}^a(e)$, $e = 1, \dots, N_{\text{ens}}$) is used to sample the analysis probability distribution at time t_{k-1} . Each member of the ensemble is propagated to t_k using the nonlinear model (1) to obtain the “forecast” ensemble

$$\mathbf{x}_k^f(e) = \mathcal{M}_{t_{k-1} \rightarrow t_k}(\mathbf{x}_{k-1}^a(e)) + \eta_k(e), \quad e = 1, \dots, N_{\text{ens}}. \quad (3a)$$

To simulate the fact that the model is an imperfect representation of reality model errors are added. They are typically considered Gaussian random variables, $\eta_k \in \mathcal{N}(0, \mathbf{Q}_k)$. The ensemble mean and covariance approximate the background estimate and the background error covariance of the state at the next time point t_k :

$$\bar{\mathbf{x}}_k^f = \frac{1}{N_{\text{ens}}} \sum_{e=1}^{N_{\text{ens}}} \mathbf{x}_k^f(e), \quad (3b)$$

$$\mathbf{X}_k^f = [\mathbf{x}_k^f(1) - \bar{\mathbf{x}}_k^f, \dots, \mathbf{x}_k^f(N_{\text{ens}}) - \bar{\mathbf{x}}_k^f], \quad (3c)$$

$$\mathbf{B}_k = \left(\frac{1}{N_{\text{ens}} - 1} \left(\mathbf{X}_k^f (\mathbf{X}_k^f)^T \right) \right) \circ \rho. \quad (3d)$$

To reduce sampling error due to the small ensemble size, localization [24, 27, 59] is performed by taking the point-wise product of the ensemble covariance and a decorrelation matrix ρ .

Each member of the forecast (ensemble of forecast states $\{\mathbf{x}_k^f(e)\}_{e=1,\dots,N_{\text{ens}}}$) is analyzed separately using the KF formulas [11, 17]

$$\mathbf{x}_k^a(e) = \mathbf{x}_k^f(e) + \mathbf{K}_k \left(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k^f(e)) + \zeta_k(e) \right), \quad (4a)$$

$$\mathbf{K}_k = \mathbf{B}_k \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{B}_k \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1}. \quad (4b)$$

The stochastic (“perturbed observations”) version [11] of the ensemble Kalman filter adds a different realization of the observation noise $\zeta_k \in \mathcal{N}(0, \mathbf{R}_k)$ to each individual assimilation. The Kalman gain matrix \mathbf{K}_k makes use of the linearized observation operator $\mathbf{H}_k = \mathcal{H}'_k(\bar{\mathbf{x}}_k^f)$.

Square root versions (deterministic formulations) of EnKF [55] avoid adding random noise to observations, and thus avoid additional sampling errors. The mean of the forecast ensemble is updated to produce the analysis state (posterior mean), and the posterior ensemble is induced after updating a matrix of state deviations from the ensemble mean.

The main limitation of the EnKF is due to the Gaussianity assumption on which the Kalman updates are based. The filter is optimal only when both the forecast and the observation errors are Gaussian, and the observations are linearly related to system state.

2.3. The maximum likelihood ensemble filter

The maximum likelihood ensemble filter (MLEF) [60] seeks to alleviate the limitations of the Gaussian assumptions by computing the maximum likelihood estimate of the state in the ensemble space. Specifically, it maximizes the posterior probability density, or equivalently, minimizes the following nonlinear objective function over the ensemble subspace [37, 60]:

$$\mathbf{x}_k^{\text{opt}} = \arg \min_{\mathbf{x}} \mathcal{J}(\mathbf{x}), \quad (5a)$$

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2} (\mathbf{x} - \mathbf{x}_k^b)^T \mathbf{B}_k^{-1} (\mathbf{x} - \mathbf{x}_k^b) + \frac{1}{2} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}))^T \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x})), \quad (5b)$$

and then updates the analysis error covariance matrix based on the fact that it is approximately equal to the inverse of the Hessian matrix at the minimum [20]. An important advantage of the algorithm is that the observation operator is partly linearized only in the ensemble perturbations. Consequently MLEF can work efficiently with non-linear observation operators (without the requirement of differentiability and without using finite-difference approximations of the Jacobian of the observation operators) [60]. The cost function (5b) to minimize implicitly assumes that the posterior distribution is Gaussian. The method is unlikely to give good results when the posterior distributions are multimodal.

2.4. The iterative ensemble Kalman filter

As discussed in Section 2.3 MLEF follows an iterative minimization procedure of a penalty (cost) functional in the subspace spanned by the ensemble members. MLEF is specifically designed to handle nonlinear observations but may fail in the presence of strong nonlinearity in the governing model [47]. The ensemble iterative randomized maximum likelihood (EnRML) filter [23], and the iterative ensemble Kalman filter (IEnKF) [47] are both developed as extensions of MLEF. Following an iterative minimization of the cost functional with EnKF as linear solution, both EnRML and IEnKF aim at handling highly nonlinear models as well as nonlinear observation operators. A deterministic approach to the

EnKF is used as base for IEnKF, while the stochastic (perturbed observations) approach is followed in the case of EnRML. IEnKF also rescales the ensemble anomalies with the ensemble transform matrix from the previous iteration, rather than using a linear regression to estimate the sensitivities between the ensemble observations and ensemble anomalies [47]. Although IEnKF is very powerful in handling nonlinearity, it is not expected to perform well if the posterior is multimodal. The optimal solution obtained can be a local minimum of the cost functional resulting in an inaccurate representation of the true state of the system. Details of IEnKF with a simple, yet detailed, pseudo code can be found in [47]. We use the formulation of IEnKF presented in [47] to check the validity of the analysis produced by the sampling filter in the cases where nonlinear observation operators are used. The IEnKF parameters (inflation factor and tolerance) are carefully tuned in order to give the best possible analysis state with minimum RMSE. This requires more iterations of the IEnKF optimization step, but it will also allow us to check how close sampling filter estimate can approach the optimal solution.

3. Hybrid Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) algorithms [42], introduced by Metropolis *et. al* [40], can sample from distributions with complex probability densities $\pi(\mathbf{x})$. They generate a Markov chain $\{\mathbf{x}(i)\}_{i \geq 0}$ for which $\pi(\mathbf{x})$ is the invariant (stationary) distribution, given that $\pi(\mathbf{x})$ is known up to a multiplicative constant [42]. MCMC methods work by generating a random walk using a proposal PDF and an “*acceptance/rejection*” criterion to decide whether proposed samples should be accepted as part of the Markov chain or should just be rejected. These algorithms are generally powerful, but may take a long time to explore the whole state space or even to converge [54]. This section starts with a review of the Hamiltonian Monte-Carlo sampling (HMC) then presents the sampling filter algorithm for DA.

Hybrid Monte Carlo (HMC) methods, also known as Hamiltonian Monte Carlo, originated in the physics literature [16]. They attempt to handle the drawbacks of MCMC algorithms by incorporating an auxiliary variable such as to reduce the correlation between successive samples, to explore the entire space in very few steps, and to ensure high probability of acceptance for proposed samples in high dimensions [42].

3.1. Hamiltonian dynamics

Hamiltonian dynamical systems operate in a phase space of points $(\mathbf{p}, \mathbf{x}) \in \mathbb{R}^{2N_{\text{var}}}$, where the individual variables are the position $\mathbf{x} \in \mathbb{R}^{N_{\text{var}}}$ and the momentum $\mathbf{p} \in \mathbb{R}^{N_{\text{var}}}$. The total energy of the system is described by the Hamiltonian function $H(\mathbf{p}, \mathbf{x})$. The dynamics of the system in time is described by the following ordinary differential equations:

$$\frac{d\mathbf{x}}{dt} = \nabla_{\mathbf{p}} H, \quad \frac{d\mathbf{p}}{dt} = -\nabla_{\mathbf{x}} H. \quad (6)$$

The time evolution of the system (6) in state space is described by the flow [43, 49]

$$\mathcal{S}_T : \mathbb{R}^{2N_{\text{var}}} \rightarrow \mathbb{R}^{2N_{\text{var}}}; \quad \mathcal{S}_T(\mathbf{p}(0), \mathbf{x}(0)) = (\mathbf{p}(T), \mathbf{x}(T)), \quad (7)$$

which maps the initial state of the system $(\mathbf{p}(0), \mathbf{x}(0))$ to $(\mathbf{p}(T), \mathbf{x}(T))$, the state of the system at time T .

In practical computations the analytic flow \mathcal{S}_T is replaced by a numerical solution using a time reversible and symplectic numerical integration method [49, 48]. In this paper we use five different high

order symplectic integrators based on Strang's splitting formula: Verlet (Störmer, Leapfrog) algorithm (29) [2, 42], higher order integrators namely, two-stage (30), three-stage (31), and four-stage (32) position splitting integrators from [10], and the Hilbert space integrator (33) from [8]. The methods are summarized in B. To approximate \mathcal{S}_T the integrator at hand takes m steps of size $h = T/m$. With a slight abuse of notation we will also denote by \mathcal{S}_T the flow of the numerical solution.

3.2. HMC sampling algorithm

In order to draw samples $\{\mathbf{x}(e)\}_{e \geq 0}$ from a given probability distribution $\pi(\mathbf{x})$ HMC makes the following analogy with a Hamiltonian mechanical system (6). The state \mathbf{x} is viewed as a "position variable", and an auxiliary "momentum variable" \mathbf{p} is included. The Hamiltonian function of the system is:

$$\begin{aligned} H(\mathbf{p}, \mathbf{x}) &= \frac{1}{2} \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p} - \log(\pi(\mathbf{x})) \\ &= \frac{1}{2} \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p} + \mathcal{J}(\mathbf{x}). \end{aligned} \quad (8)$$

The negative logarithm of the target probability density $\mathcal{J}(\mathbf{x}) = -\log(\pi(\mathbf{x}))$ is viewed as the potential energy of the system. The kinetic energy of the system is given by the auxiliary momentum variable \mathbf{p} . The constant positive definite symmetric "mass matrix" \mathbf{M} is yet to be defined. Based on the Hamiltonian equations (6) the dynamics of the system is given by

$$\frac{d\mathbf{x}}{dt} = \mathbf{M}^{-1} \mathbf{p}, \quad \frac{d\mathbf{p}}{dt} = -\nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}). \quad (9)$$

The canonical probability distribution of the state of the system (\mathbf{p}, \mathbf{x}) in the phase space $\mathbb{R}^{2N_{\text{var}}}$ is, up to a constant, equal to

$$\begin{aligned} \exp(-H(\mathbf{p}, \mathbf{x})) &= \exp\left(-\frac{1}{2} \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p} - \mathcal{J}(\mathbf{x})\right) \\ &= \exp\left(-\frac{1}{2} \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}\right) \cdot \pi(\mathbf{x}). \end{aligned} \quad (10)$$

The product form of this joint probability distribution shows that the two variables \mathbf{p}, \mathbf{x} are independent. The marginal distribution of the momentum variable is Gaussian, $\mathbf{p} \sim \mathcal{N}(0, \mathbf{M})$, while the marginal distribution of the position variable is the target probability density, $\mathbf{x} \sim \pi$ [42].

The HMC sampling algorithm builds a Markov chain starting from an initial state $\mathbf{x}_0 = \mathbf{x}(0)$. Algorithm 1 summarizes the transition from the current Markov chain state \mathbf{x}_k to a new state \mathbf{x}_{k+1} [48]. Practical issues are related to the choice of the numerical integrator, the time step, and the choice of the function $\mathcal{J}(\mathbf{x})$ that represents the PDF we wish to sample from. The construction of the mass matrix \mathbf{M} does not impact the final distribution, but does affect the computational performance of the algorithm [21]. The mass matrix \mathbf{M} is symmetric and positive definite and is a parameter that is tuned by the user. It can be for example, a constant multiple of the identity [43], or a diagonal matrix whose entries are the background error variances [8, 36]. We found that the latter approach is more efficient for the current application and used it in all numerical experiments reported here. We note that the generalized HMC method employed in [2] uses the dynamics of a modified system that is no longer Hamiltonian.

Algorithm 1 HMC Sampling [48].

- 1: Draw a random vector $\mathbf{p}_k \sim \mathcal{N}(0, \mathbf{M})$.
- 2: Use a symplectic numerical integrator (from B) to advance the current state $(\mathbf{p}_k, \mathbf{x}_k)$ by a time increment T to obtain a *proposal* state $(\mathbf{p}^*, \mathbf{x}^*)$:

$$(\mathbf{p}^*, \mathbf{x}^*) = \mathcal{S}_T((\mathbf{p}_k, \mathbf{x}_k)), \quad (11)$$

where \mathcal{S} refers to the symplectic integrator at hand.

- 3: Evaluate the loss of energy based on the Hamiltonian function. For the standard Verlet (29), two-stage (30), three-stage (31), and four-stage (32) integrators [10, 42] the loss of energy is computed as:

$$\Delta H = H(\mathbf{p}^*, \mathbf{x}^*) - H(\mathbf{p}_k, \mathbf{x}_k). \quad (12)$$

For the Hilbert space integrator (33) [8] the loss of energy is computed as:

$$\begin{aligned} \Delta H &= \phi(\mathbf{x}^*) - \phi(\mathbf{x}_k) \\ &+ \frac{h^2}{8} \left(|\mathbf{M}^{-\frac{1}{2}}(-\nabla\phi(\mathbf{x}_k))|^2 - |\mathbf{M}^{-\frac{1}{2}}(-\nabla\phi(\mathbf{x}^*))|^2 \right) \\ &+ h \sum_{i=1}^{m-1} \left(\mathbf{p}_k^T (-\nabla\phi(\mathbf{x}_k)) \right) \\ &+ \frac{h}{2} \left(\mathbf{p}_k^T (-\nabla\phi(\mathbf{x}_k)) + (\mathbf{p}^*)^T (-\nabla\phi(\mathbf{x}^*)) \right), \end{aligned} \quad (13)$$

where $\phi(\mathbf{x}) = -\log(\pi(\mathbf{x}))$, and $h = T/m$ is the integration time step.

- 4: Calculate the probability:

$$a^{(k)} = 1 \wedge e^{-\Delta H}. \quad (14)$$

- 5: Discard both \mathbf{p}^* , \mathbf{p}_k .
 - 6: **(Acceptance/Rejection)** Draw a uniform random variable $u^{(k)} \sim \mathcal{U}(0, 1)$:
 - i- If $a^{(k)} > u^{(k)}$ accept the proposal as the next sample: $\mathbf{x}_{k+1} := \mathbf{x}^*$;
 - ii- If $a^{(k)} \leq u^{(k)}$ reject the proposal and continue with the current state: $\mathbf{x}_{k+1} := \mathbf{x}_k$.
 - 7: Repeat steps 1 to 6 until sufficiently many distinct samples are drawn.
-

4. The Sampling Filter for Data Assimilation

The goal of this filter is to replace the analysis step in the traditional EnKF with a resampling procedure that draws representative ensemble members from the posterior distribution $\pi(\mathbf{x}) = \mathcal{P}^a(\mathbf{x})$. Even if the posterior may in general be non-Gaussian we assume, as most of the current ensemble-based DA algorithms, that the posterior has the form:

$$\pi(\mathbf{x}) = \mathcal{P}^a(\mathbf{x}) \propto \exp(-\mathcal{J}(\mathbf{x})), \quad (15a)$$

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^b)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}^b) + \frac{1}{2}(\mathbf{y} - \mathcal{H}(\mathbf{x}))^T \mathbf{R}^{-1} (\mathbf{y} - \mathcal{H}(\mathbf{x})), \quad (15b)$$

where \mathbf{x}^b is the background state (forecast), \mathbf{y} is the observation vector, and \mathcal{H} is the observation operator that is generally non-linear. Here the prior distribution is assumed Gaussian. When applying the sampling filter sequentially, this assumption is likely to be violated because the prior is a result of propagating the last (non-Gaussian) posterior forward to the forecast time. In this work, for simplicity, we assume that the prior can be well-approximated by a Gaussian distribution whose moments are inferred from the forecast ensemble. The use of more complex distributions like Gaussian mixtures to describe the prior is the topic of on-going work.

For sampling at time t_k the corresponding $\mathcal{J}(\mathbf{x})$ is:

$$\begin{aligned} \mathcal{J}(\mathbf{x}) &= -\log(\mathcal{P}^a(\mathbf{x})) \\ &= \frac{1}{2}(\mathbf{x} - \mathbf{x}_k^b)^T \mathbf{B}_k^{-1} (\mathbf{x} - \mathbf{x}_k^b) + \frac{1}{2}(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}))^T \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x})), \end{aligned} \quad (16)$$

and its gradient has the form

$$\nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}) = \mathbf{B}_k^{-1} (\mathbf{x} - \mathbf{x}_k^b) - \mathbf{H}_k^T \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x})), \quad (17)$$

where $\mathbf{H}_k = \mathcal{H}'_k(\mathbf{x})$ is the linearized observation operator.

Algorithm 1 is used to generate N_{ens} ensemble members drawn from the posterior distribution $\{\mathbf{x}_k^a(e) \sim \mathcal{P}^a(\mathbf{x})\}_{e=1,2,\dots,N_{\text{ens}}}$. The mean of this ensemble is an estimate of the *analysis* state, and the ensemble covariance estimates the analysis error covariance matrix. Note that the proposed sampling filter is not restricted to a specific form of the posterior PDF, and the Gaussian assumption (16) can in principle be removed. The remaining issue is to represent non-Gaussian probability density functions and their logarithm. In the next section we describe the proposed sampling filter as an alternative to the EnKF.

The sampling filter is described in Algorithm 2. Like most of the ensemble-based sequential DA algorithms the sampling filter consists of two stages, namely, the *forecast* step and the *analysis* step.

Start with an ensemble $\{\mathbf{x}_{k-1}^a(e)\}_{e=1,\dots,N_{\text{ens}}}$ describing the analysis PDF at time t_{k-1} . In the forecast step each ensemble member is propagated by the full model to the next time t_k where observations are available, resulting in the forecast ensemble. In the analysis step the HMC algorithm is simply used to sample from the posterior PDF of the state, providing the new analysis ensemble $\{\mathbf{x}_k^a(e)\}_{e=1,\dots,N_{\text{ens}}}$.

As stated in step *ii*) of Algorithm 2, the explicit representation of the matrix \mathbf{B}_k is not necessary - one only needs to apply its inverse to a vector in (16), (17). Typically \mathbf{B}_k is formed as a linear combination

Algorithm 2 Sampling Filter

- 1: **Forecast step:** given an analysis ensemble $\{\mathbf{x}_{k-1}^a(e)\}_{e=1,2,\dots,N_{\text{ens}}}$ at time t_{k-1} ; generate the forecast ensemble by via the model \mathcal{M} :

$$\mathbf{x}_k^b(e) = \mathcal{M}_{t_{k-1} \rightarrow t_k}(\mathbf{x}_{k-1}^a(e)), \quad e = 1, 2, \dots, N_{\text{ens}}.$$

- 2: **Analysis step:** given the observation vector \mathbf{y}_k at time point t_k , follow the steps:

- i- Set the initial state \mathbf{x}_0 of the Markov Chain to be to the best estimate available, e.g., the mean of the forecast ensemble. One can use the EnKF analysis if the cost is acceptable, and this choice is expected to result in a faster convergence of the chain to the stationary distribution.
- ii- Calculate the ensemble-based forecast error covariance matrix \mathbf{B}_k (and possibly balance it by a fixed (or frequently updated) covariance matrix \mathbf{B}_0), and apply localization as in equation (3d). It is important to emphasize that explicitly building the full background error covariance matrix is not necessary for the current algorithm to work.
- iii- Choose a positive definite diagonal mass matrix \mathbf{M} . One choice that favors the performance of the sampling algorithm is the diagonal of the matrix \mathbf{B}_k^{-1} [43] which scales the components of the state vector vary. Ideally, \mathbf{M} should be set to the diagonal of the inverse posterior covariance matrix.
- iv- Apply Algorithm 1 with initial state \mathbf{x}_0 and generate N_{ens} ensemble members. In practice one starts accepting samples after a warm-up phase (of, say, 30 steps), to guarantee that selected members explore the entire state space.
- v- Use the generated samples $\{\mathbf{x}_k^a(e)\}_{e=1,2,\dots,N_{\text{ens}}}$ as an analysis ensemble and calculate the best estimate of the state (e.g. the mean), and the analysis error covariance matrix.

- 3: Increase time $k := k + 1$ and repeat steps 1 and 2.
-

between a fixed matrix \mathbf{B}_0 and the ensemble covariance. The calculation requires to evaluate the products

$$\begin{aligned} u &= \mathbf{B}_k^{-1} (\mathbf{x} - \mathbf{x}_k^b) \\ &= \left(\gamma \mathbf{B}_0 + \left(\frac{1 - \gamma}{N_{\text{ens}} - 1} \sum_{e=1}^{N_{\text{ens}}} \Delta \mathbf{x}(e) (\Delta \mathbf{x}(e))^T \right) \circ \rho \right)^{-1} (\mathbf{x} - \mathbf{x}_k^b), \end{aligned} \quad (18)$$

where $\Delta \mathbf{x}(e)$ is the deviation of the ensemble member $\mathbf{x}(e)$ from the mean of the ensemble, and ρ is the decorrelation matrix. The linear system

$$\left(\gamma \mathbf{B}_0 + \left(\frac{1 - \gamma}{N_{\text{ens}} - 1} \sum_{e=1}^{N_{\text{ens}}} \Delta \mathbf{x}(e) (\Delta \mathbf{x}(e))^T \right) \circ \rho \right) \cdot u = \mathbf{x} - \mathbf{x}_k^b, \quad (19)$$

can be solved without having to build the full matrix \mathbf{B}_k as discussed in [45]. The linear system (19) is solved only once for each assimilation cycle to obtain the background term.

In our numerical experiments we build flow-dependent background error covariance matrices \mathbf{B}_k at each time step. For experiments with Lorenz-96 model, a fully flow-dependent background error covariance matrix is used, i.e. $\gamma = 0$. In the experiments with the shallow water model on the sphere we chose $\gamma = 0.5$. We set \mathbf{M} to be equal to the diagonal of \mathbf{B}_k in case of Lorenz-96 model following [8, 36]. Taking \mathbf{M} equal to the diagonal of \mathbf{B}_k^{-1} leads to similar results for the Lorenz-96 model. For the shallow-water model on the sphere we set \mathbf{M} to be equal to the diagonal of \mathbf{B}_k^{-1} .

4.1. Computational considerations

The sampling filter calculations shown in Algorithm 2 are computationally expensive. It is important to keep in mind that the main goal here is not to replace the operational filters when they work (the majority of situations). Instead we seek to develop a fully non-Gaussian sequential MCMC-based filter that is robust and can solve DA problems when the operational filters fail, at the expense of an *affordable increase* in the computational cost.

Moreover, as pointed out before, the sampling methodology can be used as replenishment tool in conjunction with parallel implementations of any practical filter, where some ensemble members are lost due to hardware failures. The forward model is not required during the sampling process (or the analysis step).

The cost of the analysis step of the sampling filter is determined by the settings of the chain parameters: the number of burn-in steps, the internal step size and the number of internal steps of the Hamiltonian trajectory, the number of states dropped at stationarity, and the ensemble size. The number of burn-in steps, the number of dropped states at stationarity, along with the size of the ensemble, define the length of the Markov chain. The number of internal steps is different for each symplectic integrator, and consequently the computational cost of the filter also depends on the choice of the symplectic discretization method.

The total computational cost of the analysis step can be expressed in terms of linear algebra operations. Since the mass matrix is generally assumed to be diagonal, the computational bottleneck of the symplectic integrator is the evaluation(s) of the gradient of the potential energy. From Equation (17), assuming the observation operator is linear, three matrix-vector products (of dimension equal to the

number of observations) and one linear system solution (of dimension equal to the number of states) are required to evaluate the gradient.

The total number of gradient evaluations for each symplectic integrator depends on the Hamiltonian trajectory settings. Verlet (29) requires one evaluation of the gradient of the cost functional (17) per step. If the length of the Hamiltonian trajectory is $T = mh$, where m is the number of steps of size h , then $3m$ matrix-vector products and m linear systems are evaluated to generate a proposal. An additional evaluation of the cost functional (16) is carried out per proposal to evaluate the loss of energy (12). An evaluation of the cost functional (16) requires two additional dot-product evaluations. The evaluation of the kinetic energy term in the Hamiltonian requires matrix-vector multiplication by the diagonal mass matrix, which is the same as the cost of a vector dot-product. The cost of other integrators scales with the number of stages, i.e., is larger by a factor of two for (30), by a factor of three for (31), and by a factor of four for (30).

If the number of burn-in steps is b , and the number of states dropped at stationarity is $r - 1$ (each r^{th} proposal is retained), then the total number of proposal states is $(b + rN_{\text{ens}})$. The total cost of the analysis step is $(b + rN_{\text{ens}})$ times the cost per step.

In the forecast step the forward model is run once for each ensemble member. The total number of forward model runs per assimilation cycle is then equal the size of the ensemble N_{ens} . The linear system (19) is solved once in order to update the background term.

In addition to the above discussion of the computational complexity of the sampling filter, we report the CPU times of both the sampling filter and the traditional filters in Section 5.11.

5. Numerical Results

5.1. The Lorenz-96 model

Numerical tests are primarily performed using the 40-variables Lorenz-96 model [38] which is described by the equations:

$$\frac{dx_i}{dt} = x_{i-1}(x_{i+1} - x_{i-2}) - x_i + F, \quad (20)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_{40})^T \in \mathbb{R}^{40}$ is the state vector. The indices work in a circular fashion, e.g., $x_0 \equiv x_{40}$. The forcing parameter is set to $F = 8$ in our experiment. These settings make the system chaotic [39]. MATLAB implementations are used to carry out all the experiments reported in this paper. A fourth order explicit Runge-Kutta scheme is used to propagate the Lorenz-96 model forward in time. One time step is $dt = 0.01$ (units). A vector of equidistant components ranging from -2 to 2 was integrated forward in time for 1000 time steps and the final state was taken as a reference initial condition for the experiments. The reference solution (truth) was created by integrating the reference initial condition forward in time over the timespan of each experiment. The initial background error covariance matrix \mathbf{B}_0 was created based on an error level of 8% of the magnitude of the reference initial condition. The decorrelation distance is specified to be 4. The initial background state is obtained by adding normal random noise $r_0 \sim \mathcal{N}(0, B_0)$ to the reference initial condition. The initial ensemble was created by adding normal random noise $\sim \mathcal{N}(0, \mathbf{B}_0)$ to the initial background state. Synthetic observations are generated every 10 time steps by applying the observation operator to the reference trajectory and adding Gaussian noise such that the uncertainty level in observation is 5%. See Appendix A for details on how the initial background error and the observation error covariance matrices are

created. To study the behavior of the sampling filter with small ensemble, the number of ensemble members is chosen to be 30 for all experiments with Lorenz-96 model.

5.2. Observations and observation operators

We tested the performance of the sampling filter with several observation operators of different complexities and varying levels of non-linearity. In this section, we show the results of a linear, a discontinuous-quadratic, and an exponential observation operators. The discontinuous quadratic observation (quadratic with a threshold) operator used here was employed by Zupanski [60, 61] in the simple case of one dimensional state space. For experiments with more observation operators with different levels of discontinuity and nonlinearity see [3].

- a) **Linear observation operator:** in all the experiments conducted with Lorenz-96 model we assume that third components only are observed. The linear observation operator takes the form:

$$\mathcal{H}(\mathbf{x}) = \mathbf{H}\mathbf{x} = (x_1, x_4, x_7, \dots, x_{37}, x_{40})^T \in \mathbb{R}^{14}, \quad (21)$$

This operator is provided to compare compare the performance of the sampling filter with EnKF which is known to be optimal in this case.

- b) **Quadratic observation operator with a threshold:** this observation operator is similar to the simple version used by Zupanski et al in [61]. The observation vector is:

$$\mathcal{H}(\mathbf{x}) = (x'_1, x'_4, x'_7, \dots, x'_{37}, x'_{40})^T \in \mathbb{R}^{14}, \quad (22)$$

where

$$x'_i = \begin{cases} x_i^2 & : x_i \geq 0.5 \\ -x_i^2 & : x_i < 0.5. \end{cases} \quad (23)$$

This operator is non-linear and discontinuous.

- c) **Exponential observation operator:** this is a highly nonlinear, differentiable observation operator:

$$\mathcal{H}(\mathbf{x}) = (e^{r \cdot x_1}, e^{r \cdot x_4}, e^{r \cdot x_7}, \dots, e^{r \cdot x_{37}}, e^{r \cdot x_{40}})^T \in \mathbb{R}^{14}, \quad (24)$$

where $r \in \mathbb{R}$ is a scaling factor that controls the degree of nonlinearity.

5.3. Experimental setting

The sampling filter can be tuned to accommodate complex settings of the experiment in hand. The parameters of the Hamiltonian trajectory can be tuned empirically such as to achieve a specific acceptance rate. In general the step size should be chosen to ensure that the rejection rate falls between 25% and 30%, an approach we followed in our settings, and the number of steps should be large [43]. We chose the length of the Hamiltonian trajectory of the sampling filter empirically to be $T = 10$ with $h = 0.01$ and $m = 10$ for all symplectic integrators. This choice is made by trial and error such that the sampling filter with the standard position Verlet integrator yields seemingly satisfying results with the linear observation operator at moderate cost. More precisely, these settings lead to acceptance rates from 25% to 30% when a linear observation operator is used in the present experiments. In general, however, the time-stepping parameters of each symplectic integrator should be set individually to get

the best performance of the sampling filter. The goal here is to show that even with a very simple setting of the sampling filter parameters one can obtain promising results for this experimental setting. The mass matrix \mathbf{M} is set as a diagonal matrix which diagonal is the precisions of the background errors at each time step. To guarantee that the Markov chain reaches the stationary distribution before starting the sampling process a set of 50 steps are performed as burn-in stage. We noticed however that the chain always converges in a small number of burn-in steps, typically 10-20 steps. Stationarity tests will be given special attention in our future work. An alternative to the burn-in steps is to apply a three dimensional variational (3D-Var) step which will certainly result in a state belonging to the posterior distribution. That resulting state can be used to initialize the chain, and sampling can start immediately. After the burn-in stage an ensemble member is selected after each 10 generated states. dropping states between successive selection has the effect of decreasing correlation between generated ensemble members since the chain is not memoryless. The number of generated states (between successive selections) that are not retained during the sampling from stationarity will be referred to as the number of mixing steps. In our experiments the acceptance probability is high (usually over 0.9) with this sampling strategy. The number of mixing steps is a parameter that can be tuned by the user to control the performance of the sampling filter.

Stability requirements impose tight upper bounds on the step size h of the Verlet integrator. The step size should be decreased with the increasing dimension of the system in order to maintain $\mathcal{O}(1)$ acceptance probability [7]. On the other hand long Hamiltonian trajectories (large steps of the symplectic integrator) are needed in order to explore the space efficiently. There is no precise rule available to select the optimal step size values [48] and consequently h , and m should be tuned for each problem. The higher-order integrators (30), (31), (32) are expected to be more stable than Verlet for larger step sizes [10, 43] and a smaller number of steps.

To guarantee ergodicity of the Markov chain, which is a property required for the chain to converge to its invariant distribution, we follow [10, 43] and change the step length at the beginning of each Markov step (once at the beginning of the Hamiltonian trajectory) to $h = (1 + r)h_{\text{ref}}$ where h_{ref} is a reference step size and $r \sim \mathcal{U}(-0.2, 0.2)$ is a uniformly distributed random variable. Randomizing the step size of the symplectic integrator, in addition to other benefits, ensures that the results obtained are not entrusted with specific choice of the step size [43].

Each numerical experiment performs 100 realizations of the sampling filter. Each realization uses the same settings but the sequence of random numbers generated by the sampling filter, for both the potential variable and the acceptance/rejection rule, was different. The root mean squared error (RMSE) metric is used to compare the analyses against the reference solution at observation time points:

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{var}}} \sum_{i=1}^{N_{\text{var}}} (x_i - x_i^{\text{true}})^2}, \quad (25)$$

where \mathbf{x}^{true} is the reference state of the system and $N_{\text{var}} = 40$ in case of Lorenz-96 model. The RMSE is calculated at all assimilation time points along the trajectory over the time span of the experiment. The average of RMS errors over the 100 realizations of the sampling filter is plotted against RMS errors obtained from the traditional filters in Figures 1, 3, 5, 9, and 11. The RMS error statistics are presented in Tables 1, 2, and 3. In addition to RMSE, we use the rank histograms (Talagrand diagrams) [4, 53] to assess the quality of the generated ensembles.

5.4. Linear observation operator experiments

Figure 1 shows the average RMSE of the 100 instances of the sampling filter and the RMSE obtained by applying EnKF (with inflation factor 1.09). Statistics of RMSE results obtained by applying MLEF and IEnKF are summarized in Table 1. Rank histograms of the first two components of the state vector

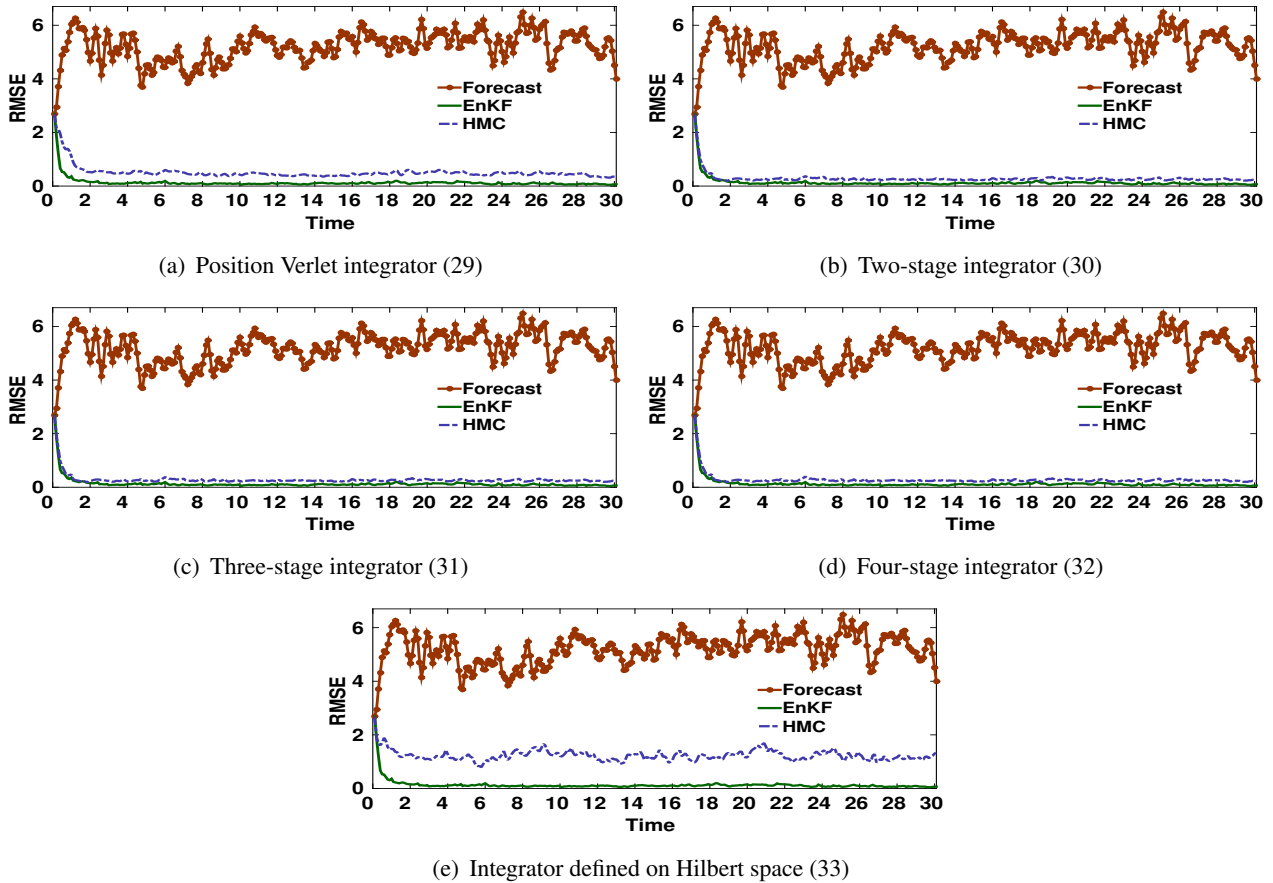


Figure 1. Data assimilation results with the Linear operator (21). The symplectic integrator used is indicated under each panel. The time step for all symplectic integrators is $T = 0.1$ with $h = 0.01$, $m = 10$, and 10 mixing steps. The RMSE reported for the sampling filter (HMC) is the average taken over the 100 realizations of the filter.

along the 300 assimilation cycles are given in Figure 2. We present rank histograms only for the first two components of the state vector because we noticed that rank histograms of observed components look very similar, and unobserved components nearly have similar rank histograms. Using the standard position Verlet as the symplectic integrator results in occasional failures (Figure 1). These failures are mostly due to the untuned step settings. On the other hand the higher order integrators show quite good performance. From Figure 2 we also see that using position Verlet the observed components (e.g. x_1) and unobserved components (e.g. x_2) tend to have uniform rank histogram while the use of higher order integrators results in overdispersed ensemble on average. Although not shown here, rank histograms resulting from three-stage and four-stage integrators are almost identical to those obtained by using two-stage integrator. The truth however lies in the middle of the ensemble and one can conclude that shorter Hamiltonian trajectory can be used in the case of higher-order integrators while the standard

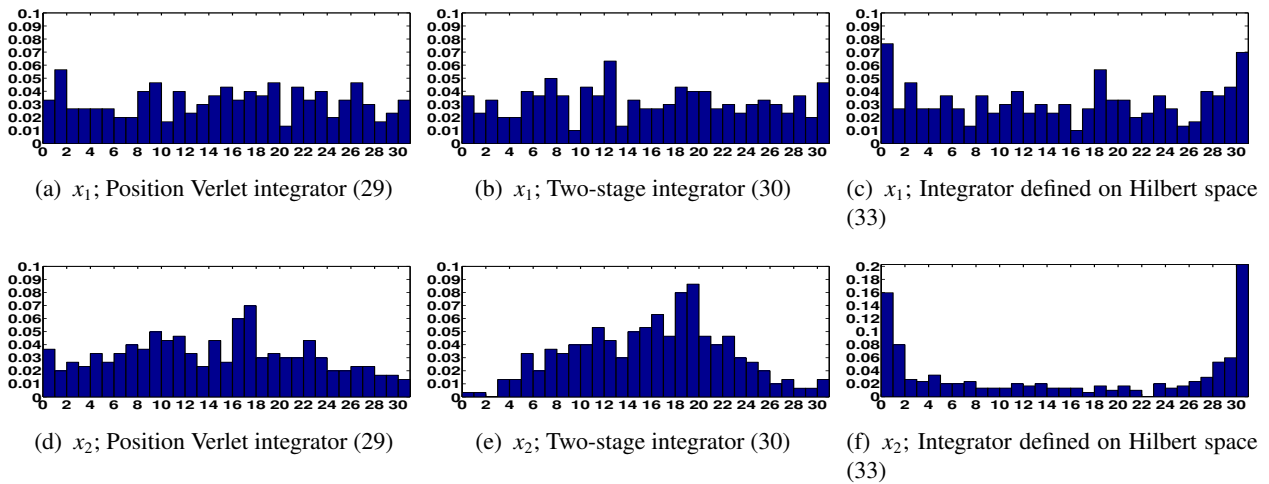


Figure 2. Data assimilation results with the Linear operator (21). The symplectic integrator used is indicated under each panel. The time step for all symplectic integrators is $T = 0.1$ with $h = 0.01$, $m = 10$, and 10 mixing steps. The rank histograms are shown for the first two components of the state vector over 300 assimilation cycles compared to the truth. The plotted component is indicated under each panel.

Verlet integrator requires smaller step size and larger number of steps in order to maintain stability and to reach distant points in state space respectively. For example, if the length of the Hamiltonian trajectory $T = 0.1$ is kept fixed, and the step size h is reduced to 0.005, the likelihood of filter diverging (see the maximum RMSE in Table 1 for this case) can be drastically reduced. This is confirmed by the results reported in Tables 1 and 2, where the standard deviation of RMSE is reduced from 0.546054 to 0.039302, and the RMSE average is slightly decreased as well. Further tuning of the filter parameters can lead to smaller RMSE, and consequently result in a better estimation of the true state. However, in nonlinear settings, one is concerned more about the distribution of the ensemble members under the posterior PDF, rather than estimating one specific state.

The use of the infinite dimensional integrator (33) (defined on Hilbert space) results in under-dispersive ensemble however it does not show signs of divergence. This is illustrated in Figures 1(e), 2(c), and 2(f).

5.5. Quadratic observation operator with threshold experiments

Figure 3 shows the RMSE results with quadratic observation operator (22) with threshold $a = 0.5$. In this case EnKF failed to converge, while both MLEF and IEnKF show very good behavior. However, we noticed that to get MLEF to produce good results inflation was needed. The performance of MLEF in the current setting is sensitive to both the observation frequency (the dimension of the observation space) and the noise level.

For clarity the figures only report results of the sampling filter (HMC) along with results obtained using MLEF (with inflation factor 1.25) and IEnKF (with inflation factor 1.09). Statistics of RMSE results for EnKF, MLEF, IEnKF and the sampling filter are given in Table 1. Given the current untuned parameter settings, we believe that the sampling filter using Verlet integrator fails due to the high non-linearity of the observation operator and/or the uncertainty levels. The high order integrators show

good results and the analysis RMSE is close to the level obtained in case of linear observation operator. The likelihood of outliers is small and decreases using higher-order integrators as can be seen from the maximum values of the RMSE shown in Table 1. The Hilbert integrator gives reasonable results. Further tuning of the filter parameters can result in much better results. Rank histograms (4), and

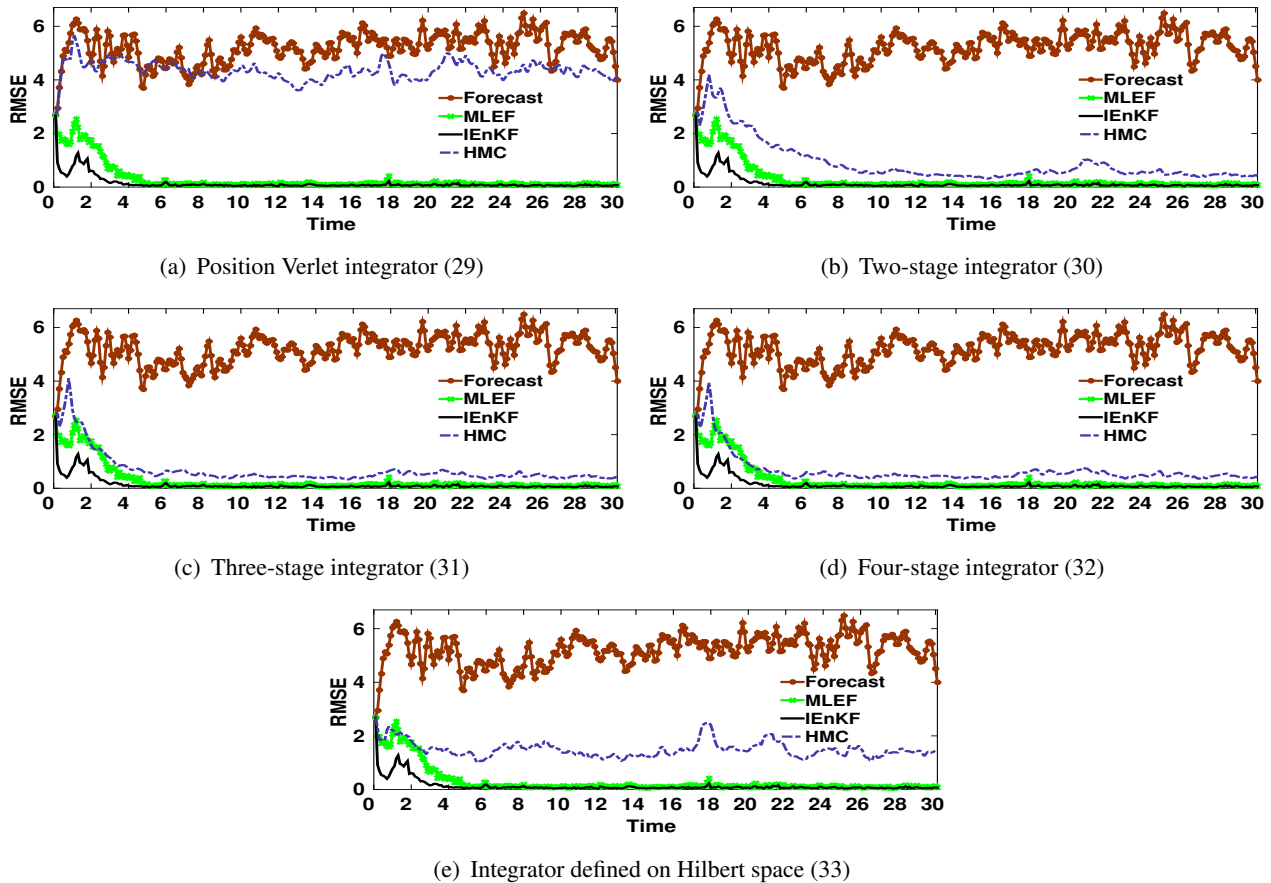


Figure 3. Data assimilation results with the quadratic observation operator (22) with a threshold $a = 0.5$. The symplectic integrator used is indicated under each panel. The time step for all symplectic integrators is $T = 0.1$ with $h = 0.01$, $m = 10$, and 10 mixing steps. The RMSE reported for the sampling filter (HMC) is the average taken over the 100 realizations of the filter.

standard deviations of RMS errors in Table 1, both show that higher order integrators, e.g. three-stage (Figures 4(a), 4(b)) produce acceptable spread around the truth. The integrator defined on Hilbert space continues to show underdispersive behaviour with larger variation of the RMSE. Careful tuning of the step size used in case of the position Verlet and the integrator defined on Hilbert space is required for the sampling filter to converge. For example, setting the length of the Hamiltonian trajectory to $T = 0.03$ with $h = 0.001$, $m = 30$ results in better performance as shown in Figures 5, 6, and in Table 2. The unobserved components have roughly uniform shapes except for the two spikes at the two extremes of the rank histogram in Figure 6(b). These results indicate that a longer Hamiltonian trajectory might be needed in order for the sampling filter to be able to sample from the tails of the posterior distribution.

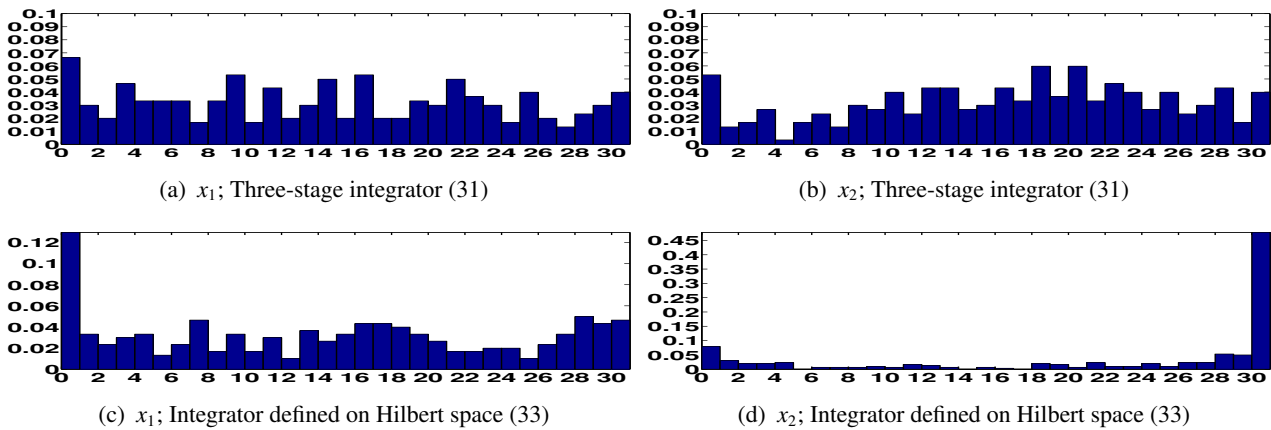


Figure 4. Data assimilation results with the quadratic observation operator (22) with a threshold $a = 0.5$. The symplectic integrator used is indicated under each panel. The time step for all symplectic integrators is $T = 0.1$ with $h = 0.01$, $m = 10$, and 10 mixing steps. The rank histograms are shown for the first two components of the state vector over 300 assimilation cycles compared to the truth. The plotted component is indicated under each panel.

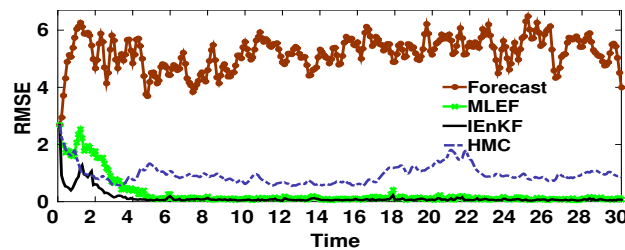


Figure 5. Data assimilation results with the quadratic observation operator (22) with a threshold $a = 0.5$. The symplectic integrator used is position Verlet with time step $T = 0.03$ with $h = 0.001$, $m = 30$, and 10 mixing steps. The RMSE reported for the sampling filter (HMC) is the average taken over the 100 realizations of the filter.

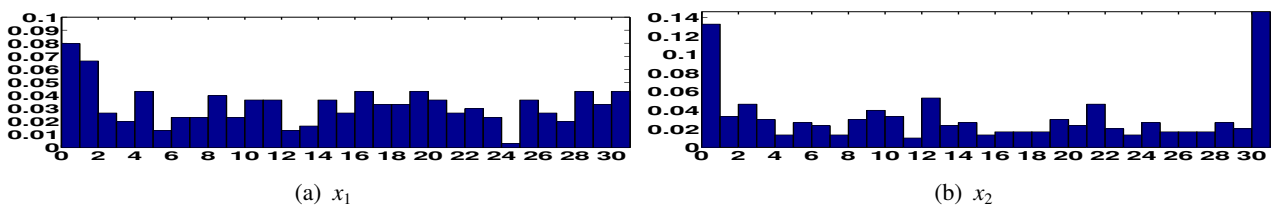


Figure 6. Data assimilation results with the quadratic observation operator (22) with a threshold $a = 0.5$. The symplectic integrator used is position Verlet with time step $T = 0.03$ with $h = 0.001$, $m = 30$, and 10 mixing steps. The rank histograms are shown for the first two components of the state vector over 300 assimilation cycles compared to the truth. The plotted component is indicated under each panel.

The ensemble size is taken to be $N_{\text{ens}} = 30$, however it is important to discuss the filter performance in the case where fewer ensemble members are desirable. Reducing the ensemble size definitely reduces the cost of the sampling filter but increases the sampling error introduced. The spread of the ensemble should not be destroyed where only few ensembles are used, otherwise the filter will be questionable. The sampling filter was tested in several settings where the ensemble size is varied. We pick the case where three-stage symplectic integrator (with $h = 0.01$, $m = 10$) is used in the presence of the discontinuous quadratic observation operator (22) to show results of the sampling filter with different ensemble sizes. As appreciated from Figure 7 the spread of the ensemble with 20, 10 members is preserved compared to results with 30 ensemble members shown in Figure 4(a), 4(b) giving a hope that in case of large dimensional systems, small number of ensemble members can produce sufficient information of the posterior distribution. When the ensemble size is varied, all other components of

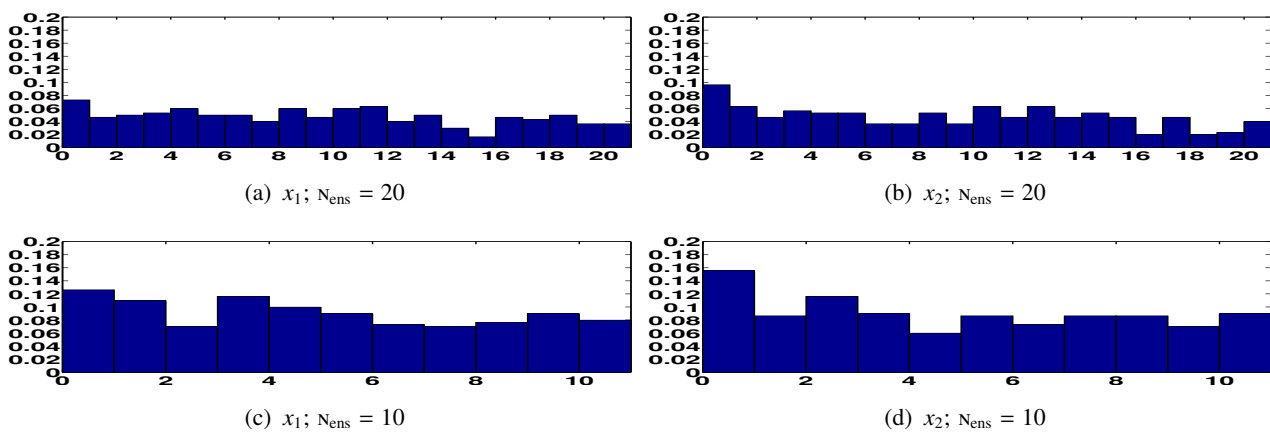


Figure 7. Data assimilation results with the quadratic observation operator (22) with a threshold $a = 0.5$. The symplectic integrator used is the three-stage (31) with time step $T = 0.1$ with $h = 0.01$, $m = 10$, and 10 mixing steps. Rank histograms of the first two components of the state vector are shown where ensemble size is varied. Plotted components and ensemble size are shown under each panel.

the state vector show similar behavior to the case where 30 ensemble members are used but we showed only the first two components to be consistent with results shown before.

The sampling filter can be used to replenish the lost ensemble members in parallel implementations of the traditional filters (e.g EnKF, IEnKF). The traditional approach to replenishing is to perturb the average of the remaining ensemble members with random noise generated from a Gaussian distribution with zero mean and covariance matrix approximated based on the remaining ensemble members. This approach leads to the new forecast states living in the subspace spanned by the remaining ensemble members, a fact that can lead to filter degradation in real applications. Moreover, this strategy is no longer valid if the Gaussianity assumption of the prior is violated.

One way to address this problem is to generate ensembles from the analysis in the previous cycle using HMC sampling filter, and to propagate them forward to the current time instance. In this case the sampling filter chain can be started from the average of the analysis ensemble and no burn-in states are required. We believe that the use of the sampling filter for replenishing dead ensemble members can be advantageous in the cases where few ensemble members are used or if the probability of node failure

is high. Generally speaking, replenishment using the sampling filter does not depend the probability of node failure.

To illustrate this, we perform a simulation with the Lorenz-96 model where each of the 30 ensemble members is killed with a (admittedly large) probability $P_f = 0.25$ (Figure 8). For clarity we removed the RMSE results for the case when the ensemble members are not replenished at all whenever any member dies out.

When the ensemble is not replenished, the IEnKF diverges after approximately 10 to 15 cycles. Replenishing the ensemble by perturbing the average of the remaining ensemble members with Gaussian perturbations with zero mean and covariance matrix \mathbf{B}_k still can lead to filter divergence, as shown in Figure 8. Even in this case the sampling filter successfully replenishes the ensemble and aids IEnKF to maintain its desired performance.

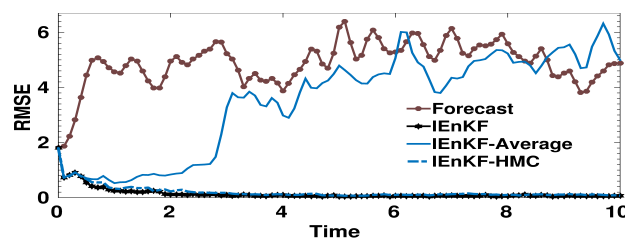


Figure 8. Data assimilation results with the quadratic observation operator (22) with a threshold $a = 0.5$. Probability of node failure is assumed to be 0.25 and the ensemble members are replenished by perturbing the ensemble average (IEnKF-Average), and using the sampling filter (IEnKF-HMC). IEnKF refers to RMSE results with no members' failures (i.e. $P_f = 0$). The two-stage symplectic integrator used with time step $T = 0.1$ with $h = 0.01$, $m = 10$, and 10 mixing steps.

5.6. Exponential observation operator (with factor $r = 0.2$) experiments

Figure 9 shows the RMSE results with the exponential observation operator (24) with factor $r = 0.2$. This observation operator is differentiable, however small perturbations in the state might result in relatively large changes in the measured values. Under strongly nonlinear conditions the sampling filter performs better than EnKF and is giving results comparable to results from MLEF and IEnKF. MLEF results are obtained by forcing inflation with factor 1.15 in this case. Based on RMSE results, it is quite clear that the best choice is the three-stage symplectic integrator for the given parameter settings (Figures 9(c), 10). The other integrators require longer Hamiltonian trajectories or smaller step size or both to sample the posterior properly.

5.7. A highly nonlinear observation operator

We have also tested the sampling filter capabilities in a very challenging setting: the exponential observation operator (24) is considered with a factor of $r = 0.5$. This factor leads to a large range of observation values (from $e^{-3.7}$ to $e^{6.2}$). In addition, small perturbations in the state variables cause very large changes in the corresponding measurements. EnKF, MLEF, and IEnKF all diverged when applied to this test, and consequently their results are not reported here.

This test problem is challenging for the sampling filter as well and the symplectic integration step

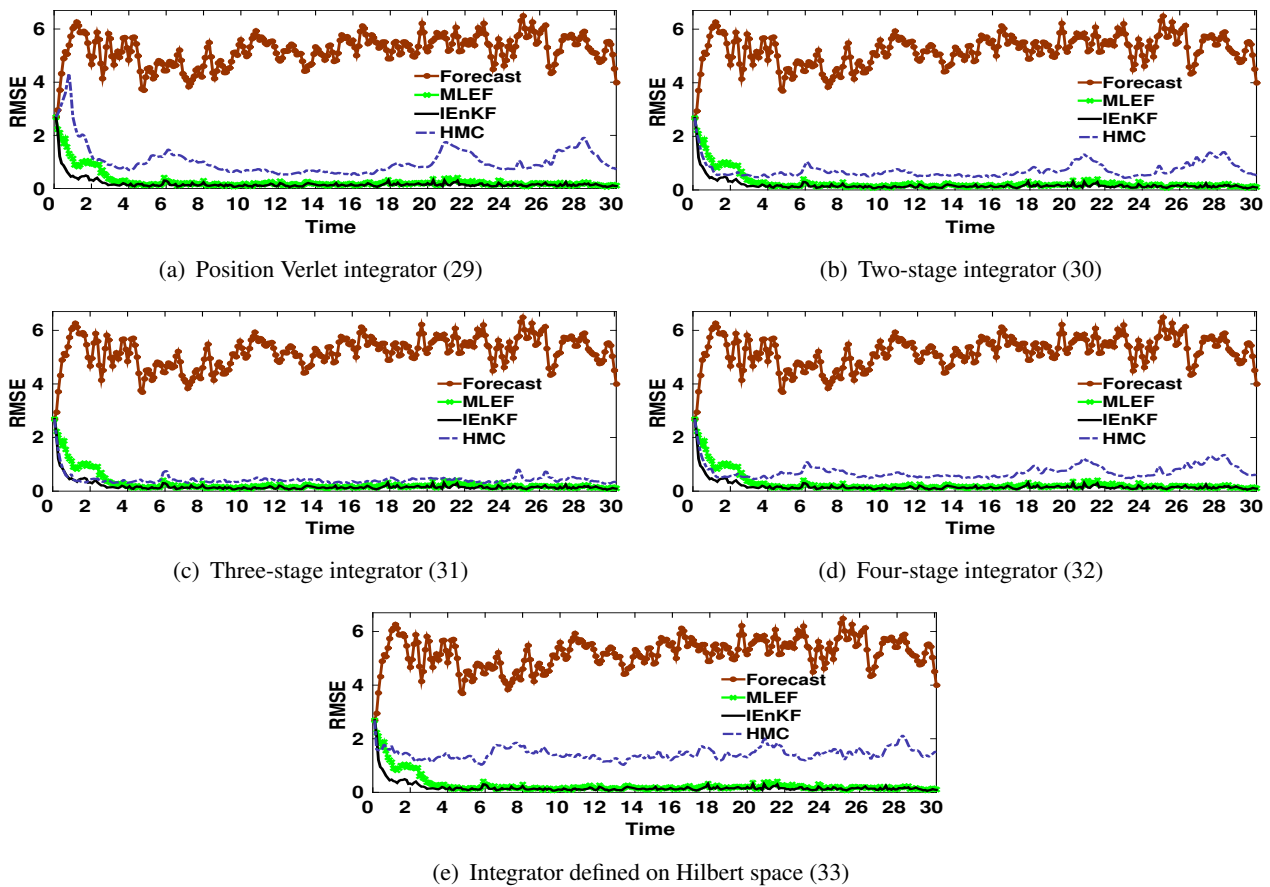


Figure 9. Data assimilation results with the exponential observation operator (24) with factor $r = 0.2$. The symplectic integrator used is indicated under each panel. The time step for all symplectic integrators is $T = 0.1$ with $h = 0.01$, $m = 10$, and 10 mixing steps. The RMSE reported for the sampling filter (HMC) is the average taken over the 100 realizations of the filter.

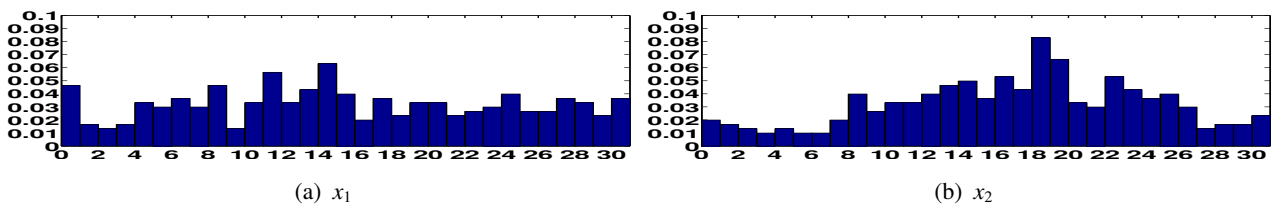


Figure 10. Data assimilation results with the exponential observation operator (24) with factor $r = 0.2$. The symplectic integrator used is indicated under each panel. The symplectic integrator used is the three-stage (31) with time step $T = 0.1$ with $h = 0.01$, $m = 10$, and 10 mixing steps. The rank histograms are shown for the first two components of the state vector over 300 assimilation cycles compared to the truth. The plotted component is indicated under each panel.

sizes need to be tuned to achieve convergence. For example, the number of steps taken by three-stage integrator had to be increased to $m = 60$ while keeping the step-size fixed to $h = 0.01$, to result in good performance as shown in Figure 11(a). The length of the trajectory of the Hamiltonian system has to be increased as well. For the infinite dimensional integrator, a shorter trajectory of the Hamiltonian system works well if the step size is sufficiently reduced, e.g., $h = 0.001$, and $m = 30$ as shown in Figure 11(b). Empirical tuning of the Verlet, two-stage, and four-stage integrators proved to be challenging with this observation operator. The computational cost associated with the tuned three-

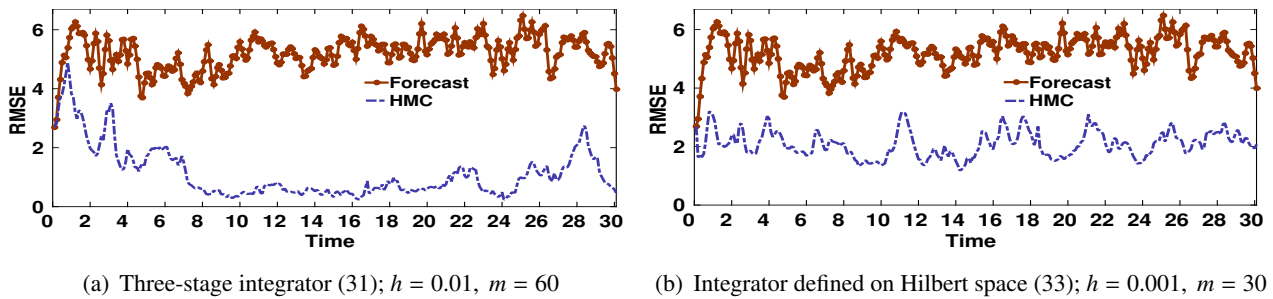


Figure 11. Data assimilation results with the exponential observation operator (24) with a factor $r = 0.5$. The symplectic integrator used is indicated under each panel. The step size h and the number of steps m are indicated under each panel. The number of mixing steps is 30. The RMSE reported for the sampling filter (HMC) is the average taken over the 100 realizations of the filter.

stage integrator in the present experiment is much higher than the cost resulting from the use of the infinite dimensional integrator. Despite the relatively large RMSE revealed by Figure 11(b), Figure 12 shows that the analysis produced by the HMC sampling filter using the integrator defined on Hilbert space follows the truth reasonably closely, which can justify the argument that this level of RMSE can be accepted if the traditional filters fail and the cost associated with the higher order integrators is prohibitive. The Hilbert integrator operates at a much lower cost, and can be used to periodically check the results obtained with the three-stage integrator to safeguard against outliers.

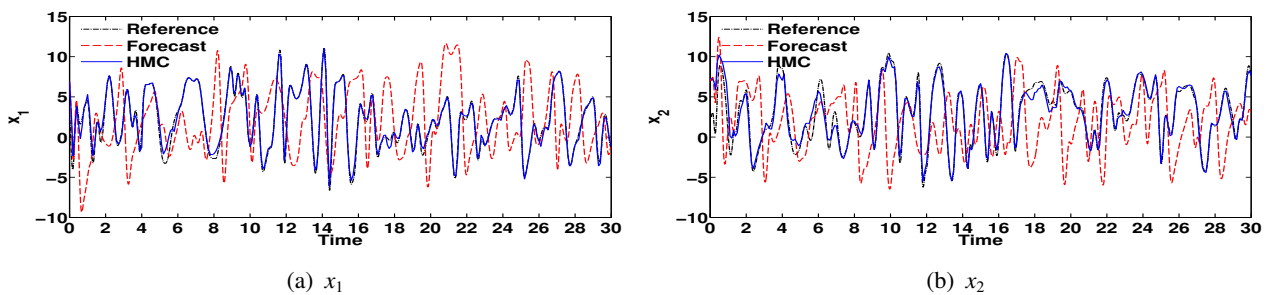


Figure 12. Data assimilation results with the exponential observation operator (24) with a factor $r = 0.5$. The symplectic integrator used is the integrator defined on Hilbert space (33) with time step $T = 0.03$ with $h = 0.001$, $m = 30$, and 30 mixing steps. The first two components of the state vector are plotted.

The statistics of the results with Lorenz-96 model are summarized in Tables 1, 2, and 3. Given the

Table 1. RMS error statistics of experiments for assimilation time points $24 \leq t \leq 30$ (after filter stabilizes). The time step for all symplectic integrators is $T = 0.1$ with $h = 0.01$, $m = 10$, and 10 mixing steps.

Observation Operator	Statistics	Symplectic integrator used with sampling filter					Traditional Filters		
		Verlet	Two-stage	Three-stage	Four-stage	Hilbert	EnKF	MLEF	IEnKF
Linear observation operator	Min	0.282158	0.225048	0.221871	0.22929	1.126023	0.04631	0.027045	0.027835
	Max	4.553241	0.331516	0.275494	0.292533	1.257923	0.13634	0.124409	0.197427
	Mean	0.433644	0.250042	0.249086	0.252403	1.179563	0.079809	0.069438	0.080403
Quadratic observation operator with threshold	Std	0.546054	0.014128	0.01077	0.012136	0.027446	0.020084	0.021752	0.027741
	Min	3.51313	0.362567	0.370246	0.364221	1.203196	2.223534	0.048766	0.033307
	Max	5.285517	2.525502	0.607215	1.123681	1.889033	8.452283	0.175145	0.116735
Exponential observation operator with $r = 0.2$	Mean	4.344827	0.476498	0.444522	0.462515	1.394191	3.949765	0.094103	0.06193
	Std	0.318406	0.218234	0.043667	0.086601	0.134929	1.184631	0.028429	0.017315
	Min	0.527899	0.478779	0.389757	0.500551	1.332802	3.598643	0.065779	0.05316
Exponential observation operator with $r = 0.2$	Max	2.406581	2.138528	0.596158	1.874854	1.840191	7.734853	0.239167	0.240487
	Mean	1.119951	0.902746	0.446232	0.887058	1.546304	5.381176	0.155157	0.132423
	Std	0.45629	0.374384	0.034703	0.35285	0.099781	0.956734	0.041048	0.036759

Table 2. RMS error statistics of experiments for assimilation time points $24 \leq t \leq 30$ (after filter stabilizes). Step parameters of the position Verlet symplectic integrator are tuned to give acceptable results.

Statistics	Observation operator and symplectic integrator used	
	Linear \mathcal{H} ;	Discontinuous quadratic \mathcal{H} ;
	Verlet: $h = 0.001$, $m = 30$	Verlet: $h = 0.005$, $m = 20$
Min	0.291365	0.580504
Max	0.569277	1.710029
Mean	0.362358	0.900346
Std	0.039302	0.370047

unified time step settings $h = 0.01$, $m = 10$ for all symplectic integrators, Table 1 summarizes results for the linear observation operator (21), the discontinuous quadratic observation operator (22), and the exponential observation operator (24) with factor $r = 0.2$. Table 2 summarizes results for the linear observation operator (21), and the discontinuous quadratic observation operator (22) where the step settings of the position Verlet integrator were tuned to give better results. Table 3 summarizes results for the exponential observation operator (24) with factor $r = 0.5$. The results shown are computed for 100 instances of the sampling filter, EnKF, MLEF, and IEnKF, over the last 20% of the assimilation cycles for each experiment.

Table 3. RMS error statistics of experiments for assimilation time points $8 \leq t \leq 10$ (after filter stabilizes). The exponential observation operator with factor $r = 0.5$ is used.

Statistics	Symplectic integrator used with sampling filter	
	Three-stage;	Hilbert;
	$h = 0.01$, $m = 60$	$h = 0.001$, $m = 30$
Min	0.304178	1.234498
Max	2.671971	2.350684
Mean	0.439776	1.699096
Std	0.274643	0.250088

5.8. Tuning the sampling filter parameters

Tuning the sampling filter parameters can prevent outliers (filter divergence) that can happen especially when nonlinear observation operators are used. In addition to selecting the mass matrix \mathbf{M} and the number of burn-in steps, there are two more parameters to be tuned. The first parameter is the length of the Hamiltonian trajectory (including step size and the number of steps of the symplectic integrator). The second parameter to be tuned is the number of steps skipped between selected states at stationarity (referred to as mixing steps). A common and a simple approach is to tune the parameters of the symplectic integrator by monitoring the acceptance rate in a preprocessing step. For simplicity, we followed this strategy in the present work. Automatically tuned versions of HMC have been proposed recently as well. No-U-Turn sampler (NUTS) [25] is a version of HMC capable of automatically tuning its parameters by prohibiting the sampler from retracing its steps along the constructed Hamiltonian trajectory. Riemann manifold HMC (RMHMC) [21] is another HMC sampler that tunes its parameters automatically using third-order derivative information. Tuning the symplectic integrator step settings can enhance the filter performance and reduce the of filter divergence, however, it might not be sufficient to completely prevent outliers. Tuning the number of mixing steps can in principle greatly enhance both the performance of the filter and the reliability of the results. The number of mixing steps in all experiments with Lorenz-96 is empirically set to 10 but it is not optimal. Increasing the number of mixing steps might be tempting, however, we cannot conclude that skipping more states will increase the chance of filter convergence [3]. A careful tuning of both the step size and the number of mixing steps in the chain may overcome the problem of outliers and lead to the desired performance of the filter. Generalized HMC with circular matrix can also be considered to shorten the decorrelation time [2] and hence reduce the number of mixing steps required by the sampling filter. From a statistical perspective, the former parameters can be tuned for example by inspecting the acceptance rate of the proposed states and the effective size of the ensemble.

In addition to controlling the time step settings of the integrator, and tuning the number of steps of the chain, we can use the Hilbert integrator (with tuned step size) to periodically validate the ensembles obtained using other integrators, since the Hilbert integrator suffers less from outliers. A simple solution is to run the assimilation process several times and exclude outlier states by creating a combined ensemble. Care must be exercised, however, to not change the probability density. These alternatives will be inspected in depth in future work in the context of more complex models.

5.9. Shallow water model on a sphere

As a first step towards large models we test the proposed sampling filter on the shallow water model on a sphere, using linear observation operator where all components are observed. The shallow water equations provide a simplified model of the atmosphere which describes the essential wave propagation mechanisms found in general circulation models (GCMs) [52]. The shallow water equations in spherical coordinates are given as

$$\frac{\partial u}{\partial t} + \frac{1}{a \cos \theta} \left(u \frac{\partial u}{\partial \lambda} + v \cos \theta \frac{\partial u}{\partial \theta} \right) - \left(f + \frac{u \tan \theta}{a} \right) v + \frac{g}{a \cos \theta} \frac{\partial h}{\partial \lambda} = 0, \quad (26a)$$

$$\frac{\partial v}{\partial t} + \frac{1}{a \cos \theta} \left(u \frac{\partial v}{\partial \lambda} + v \cos \theta \frac{\partial v}{\partial \theta} \right) + \left(f + \frac{u \tan \theta}{a} \right) u + \frac{g}{a} \frac{\partial h}{\partial \theta} = 0, \quad (26b)$$

$$\frac{\partial h}{\partial t} + \frac{1}{a \cos \theta} \left(\frac{\partial(hu)}{\partial \lambda} + \frac{\partial(hv \cos \theta)}{\partial \theta} \right) = 0. \quad (26c)$$

The Coriolis parameter is given by $f = 2\Omega \sin \theta$, where Ω is the angular speed of the rotation of the Earth, and θ is latitudinal direction. The longitudinal direction is λ . The height of the homogeneous atmosphere is represented by h , the zonal and meridional wind components are given by u and v respectively. The radius of the earth is a , and the gravitational constant is given by g . The space discretization follows the unstaggered Turkel-Zwas scheme [44]. The discretization has $nlon = 72$ nodes in longitudinal direction and $nlat = 36$ nodes in the latitudinal direction. The semi-discretization in space results in the following discrete model:

$$\mathbf{x}_{k+1} = \mathcal{M}_{t_k \rightarrow t_{k+1}}(\mathbf{x}_k, \theta), \quad k = 0, \dots, N; \quad \mathbf{x}_0 = \mathbf{x}_0(\theta). \quad (27)$$

The state space vector \mathbf{x} in (27) combines the zonal wind, the meridional wind, and the height variables into the vector $\mathbf{x} \in \mathbb{R}_{\text{var}}^N$ with $N_{\text{var}} = 3 \times nlat \times nlon = 7776$. The time integration is conducted using an adaptive time-stepping algorithm. A reference initial condition is used to generate a reference trajectory. Synthetic observations are created from the reference trajectory by adding Gaussian noise with zero mean and fixed standard deviation for each of the three components. The level of observation noise for height component is set to 1.5% of the average magnitude of the reference height component in the reference initial condition. The level of observation noise for wind components is set to 10% of the average magnitude of the reference wind component in the initial condition. Based on these uncertainty levels, the standard deviations of observation errors are $\sigma_h = 700$, $\sigma_u = \sigma_v = 3$ for the height, the zonal and the meridional wind components respectively. The initial background state is created by perturbing the reference initial condition by a Gaussian noise drawn from a modeled background error covariance matrix \mathbf{B}_0 . The standard deviation of the background errors for the height component is 2% of the average magnitude of the reference height component in the reference initial condition. The standard deviation of the background errors for the wind components is 15% of the average magnitude of the reference wind component in the reference initial condition.

The modeled version of the background error covariance, \mathbf{B}_0 , that accounts for correlations between state variables is created as follows:

- Start with a diagonal background error covariance matrix with uncertainty levels as mentioned previously.
- Apply the ensemble Kalman filter with 500 ensemble members for 48 hours. Synthetic initial ensemble is created by adding zero-mean Gaussian noise to the reference initial condition with covariances set to the initial (diagonal) background error covariance matrix.
- Decorrelate the ensemble-based covariances using a decorrelation matrix ρ with decorrelation distance $L = 1000 \text{ km}$.
- Calculate \mathbf{B}_0 by averaging the covariances over the last 6 hours.

This method of creating a synthetic initial background error covariance matrix is totally empirical, but we found that the resulting background error covariance matrix performs well for several algorithms including 4D-Var. Moreover, the quality of the background error covariance matrix can be enhanced by the flow-dependent versions obtained by the used filters.

5.10. Results for shallow water model with linear observations

The assimilation time interval of the current experiment is 22 hours with observations available each hour. The number of burn-in steps in the Markov chain is set to 50. The size of the ensemble is chosen to be $N_{\text{ens}} = 100$. Given the simple settings in this experiment, it is expected that Verlet (29) or any of the higher order symplectic integrators (30), (31), (32) will behave similarly with tuned parameters. Here we provide the results obtained using two-stage integrator with step size $h = 0.01$ and number of steps $m = 10$. The number of mixing steps is 5. Beginning with the second assimilation cycle, we noticed that the filter starts to show convergence in a very small number of steps, typically 5 to 10 steps. Statistical tests of convergence should be considered to start the sampling process as soon as convergence is achieved and to avoid discarding states generated from the posterior. We noticed that if the position Verlet is used instead, longer Hamiltonian trajectories are required while step sizes should be lowered in order to achieve both stability and fast space exploration. It is important to note that the quality of the background error covariance matrix and the way it is updated both have big impact on the performance of the filter. The flow-dependent background error covariance matrix is used to update the modelled background version. A hybrid version of the background error covariance matrix is created using (19) as a linear combination of the modeled version with the linear coefficient $\gamma = 0.5$. The

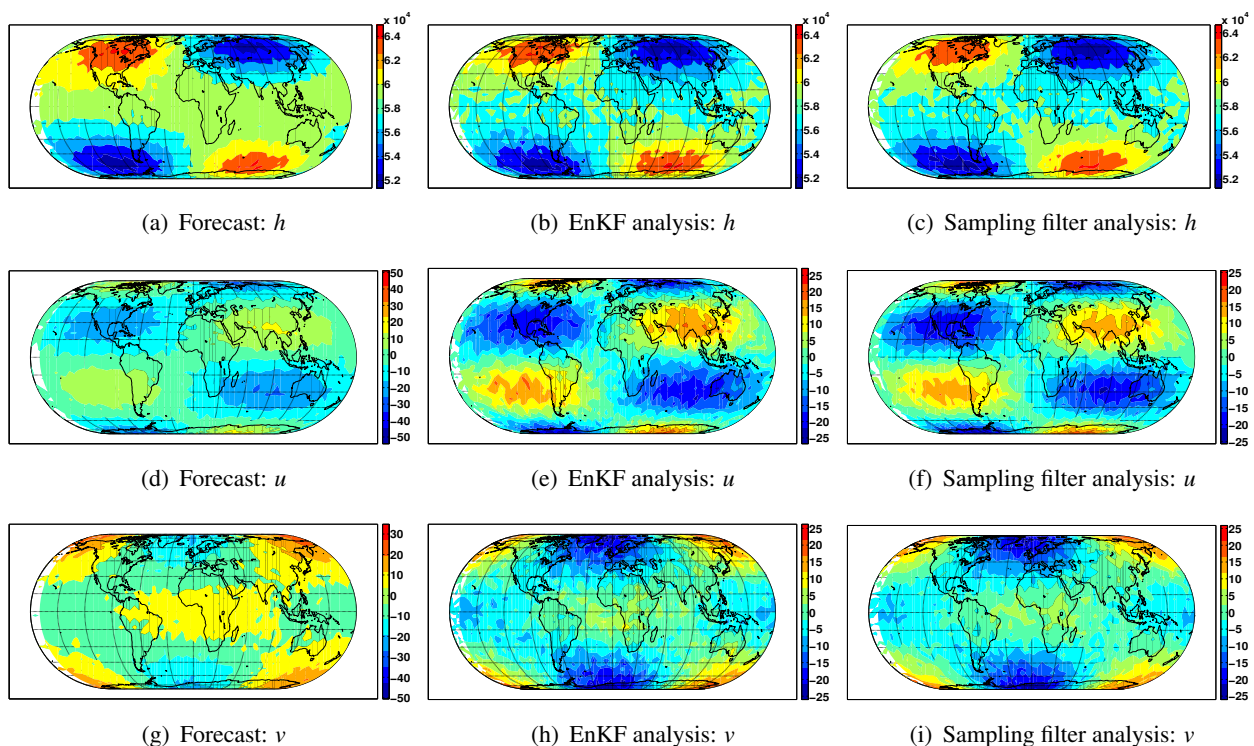


Figure 13. Data assimilation results for SWE on the sphere with linear observations where all components are observed. The plotted component of the state vector is indicated under each panel. The analysis shown is obtained after sequential assimilation of hourly observations for 22 hours. Only state at the last last assimilation cycle is plotted. The symplectic integrator used is the two-stage (30). The length of the Hamiltonian trajectory is $T = mh$, with $h = 0.01$, $m = 10$. The number of mixing steps is 5.

forecast and analysis states obtained by EnKF and the sampling filter at the last assimilation cycle are shown in Figure 13. The sampling filter results in an analysis state that is almost identical to the analysis obtained by EnKF. Figure 14 shows the RMSE of the analysis states, over the 22 cycles, obtained by both EnKF and the HMC sampling filter. Clearly, in the presence of linear observation operator, the sampling filter competes with EnKF which is known to be optimal given the current settings. It is clear from both Figures 14, and 13 that the HMC is capable of handling high-dimensional models as well as EnKF. Despite the additional cost, the performance of the sampling filter under the current settings encourage us to test the sampling filter with larger models under more challenging settings. A comparison between EnKF, MLEF, and PF applied to SWE on the sphere, in the presence of both linear and nonlinear observation operators, can be found in [29].

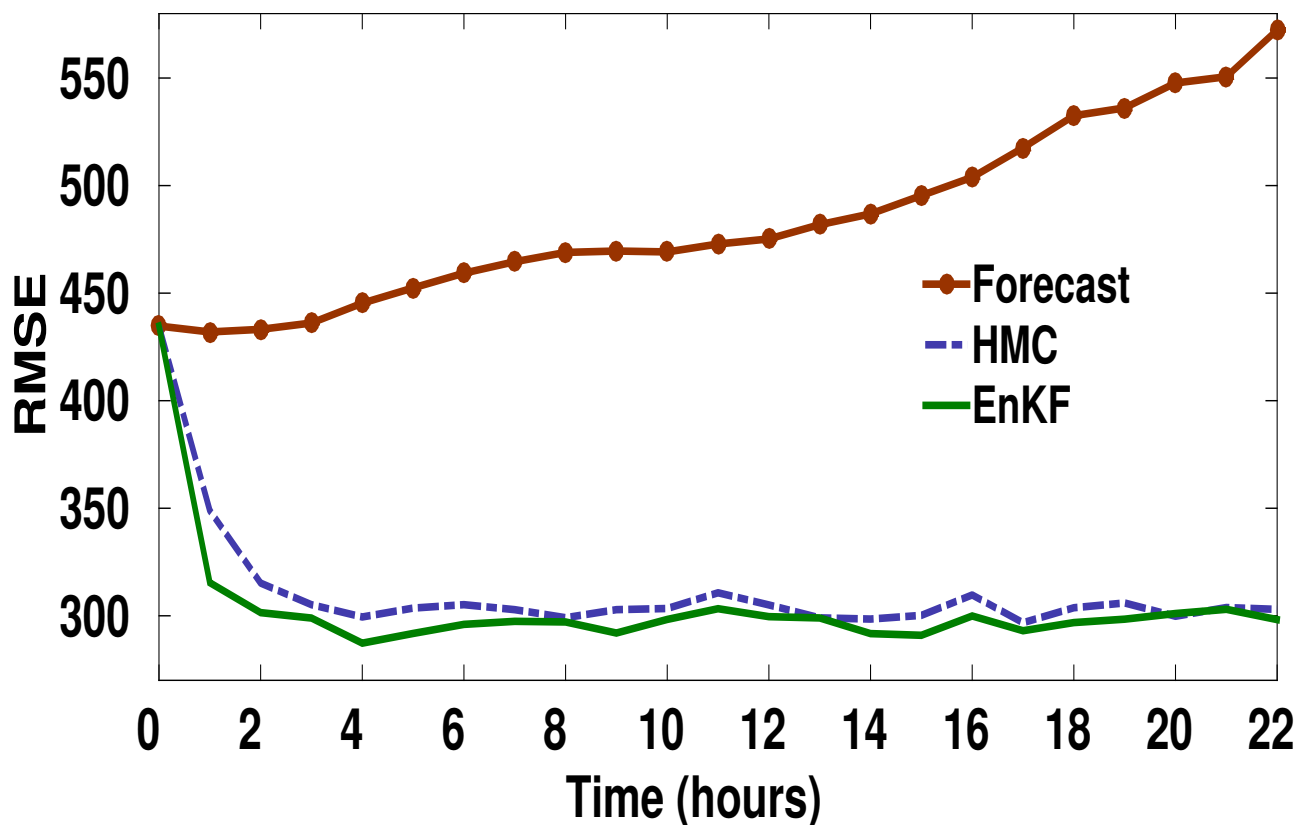


Figure 14. Data assimilation results for SWE on the sphere with linear observations where all components are observed. RMSE results of the sampling filter and EnKF are plotted against the Forecast(no assimilation) results. The symplectic integrator used is the two-stage (30). The length of the Hamiltonian trajectory is $T = 0.1$, with $h = 0.01$, $m = 10$. The number of mixing steps is 5.

Rank histograms of the uncorrelated components of the state vector along the 22 assimilation cycles are given in Figure 15. The first column of panels shows rank histograms where the ensemble size is $N_{\text{ens}} = 100$, while in the second column of panels the size of the ensemble is reduced to $N_{\text{ens}} = 20$. We see that even with 20 ensemble members, the rank histograms are approximately uniform. This gives the sampling filter a chance to work with high dimensional model where only small number of ensemble members is affordable.

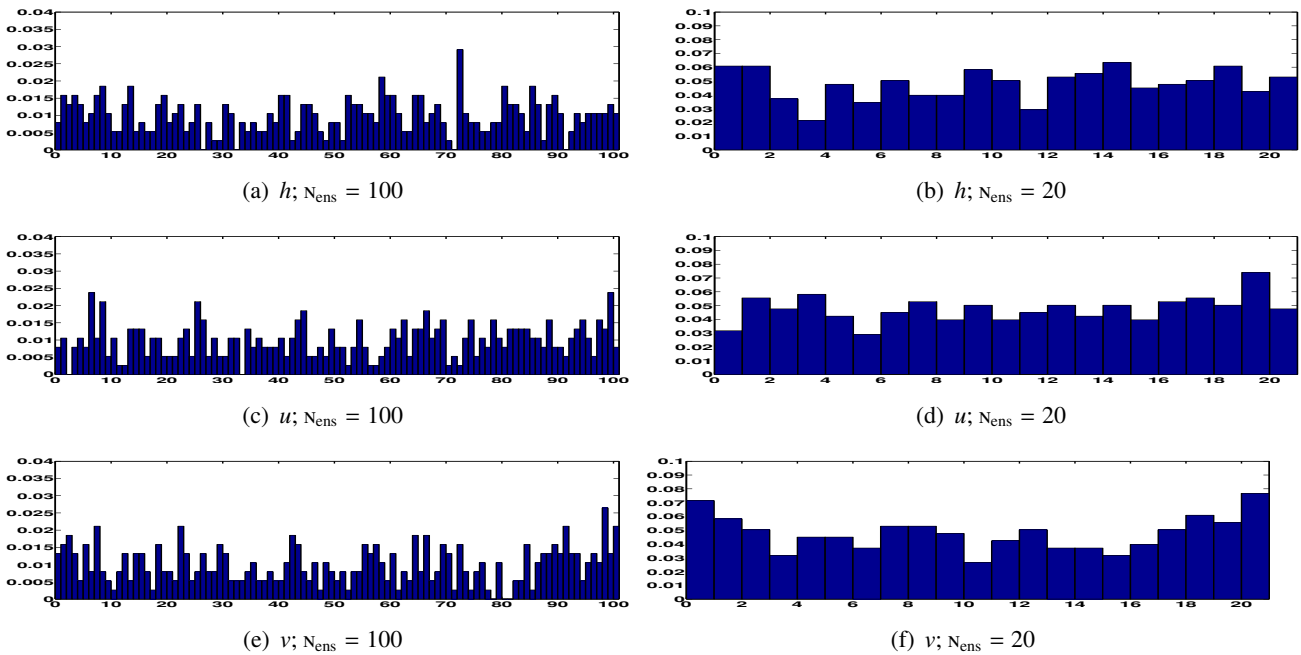


Figure 15. Data assimilation results for SWE on the sphere with linear observations where all components are observed. The rank histograms here consider only uncorrelated grid points of each of the three components. First row of panels 15(a), 15(c), 15(e), are obtained based on ensemble of 100 members. The following three panels 15(b), 15(d), 15(f) show rank histograms where the ensemble size is reduced to 20. The symplectic integrator used is the two-stage (30). The length of the Hamiltonian trajectory is $T = mh$, with $h = 0.01$, $m = 10$. The number of mixing steps is 5.

5.11. CPU-time usage

To complement the discussion provided in 4.1, we show the CPU usage, per assimilation cycle, of both the traditional filters and the HMC sampling filter.

The CPU-times for the experiments carried out using Lorenz-96 model are shown in Figure 16. As discussed in Section 4.1 the cost of the sampling filter depends on the settings of the Markov chain, and relies heavily on the choice of the parameters of the symplectic integrator of choice. Here we fix the parameters of all the symplectic integrators to $T = m \times h = 0.1$, with $m = 10$ and $h = 0.01$ (units) for both models. For experiments carried out using Lorenz-96 model, the size of the ensemble is 30 members, the number of burn-in steps is 50, and the number of mixing steps is 10. The blue parts of the bars report the CPU-time spent during the burn-in process, while the green parts show the time spent during sampling (collecting ensemble members after the Markov chain has reached stationarity). The reason behind the notable high CPU usage of IEnKF here is that we tune the IEnKF tolerance iteratively so that we guarantee the best possible results against which we can compare the HMC sampling results. While the average CPU-time of HMC sampling filter with Verlet integrator is roughly 4.7 times the CPU-time reported by EnKF, the cost of the HMC sampling filter is proportional to the number of stages of the symplectic integrator. This additional computational cost is justifiable in cases where traditional methods fail, or when ensemble member replenishment is needed. For a

linear observation operator we show the CPU-times for two cases with ensemble sizes of 100 and 20 members. The number of burn-in steps and the number of mixing steps are 50 and 5 respectively. With only 20 ensemble members collected the sampling filter with Verlet integrator takes approximately 2.16 longer than EnKF. As pointed out before, only the time spent during collecting sample points changes by varying the ensemble size, as seen by comparing results reported in Figures 17(b) and 17(a). To greatly reduce the time usage of the HMC sampler, one can run the burn-in process sequentially, to guarantee convergence to the right distribution, followed by a parallel run of the sampling process. The burn-in process itself can be replaced by a sub-optimal 3D-Var run to attain a local minimum of the negative-log of the posterior kernel.

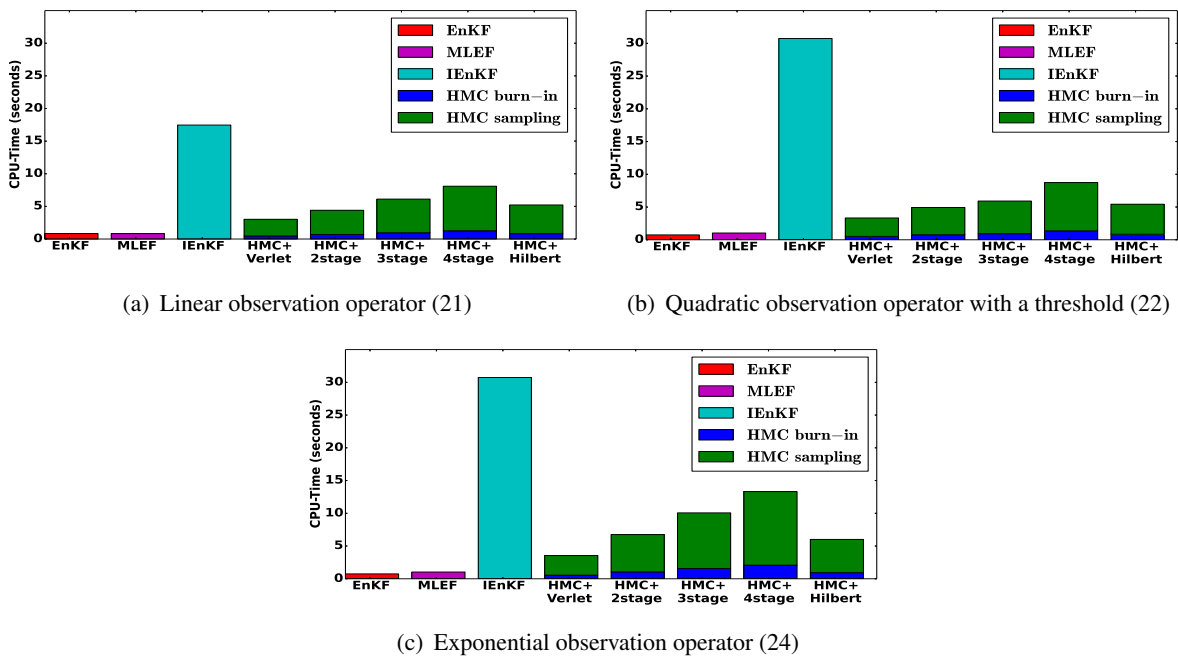


Figure 16. CPU-time per assimilation cycle of DA with the Lorenz-96 model. The time reported is the average CPU-time taken over 100 identical runs of each experiment. The ensemble size is fixed to 30 members for all experiments here. The algorithm and the symplectic integrators used in the HMC sampler are both indicated under each panel. The length of the Hamiltonian trajectory is fixed to $T = mh$, with $h = 0.01$, $m = 10$. The number of burn-in steps and the number of mixing steps are 50 and 10, respectively.

6. Conclusions and Future Work

This paper proposes a sampling filter for data assimilation where the analysis scheme is replaced by sampling directly from the posterior distribution. A Hamiltonian Monte-Carlo technique is employed to generate a representative analysis ensemble at each time. The sampling filter avoids the need to develop tangent linear or adjoint models of the model solution operator. The sampling filter can work with highly nonlinear observation operators and provides analysis ensembles that describe non-Gaussian posterior probability densities. The mean of the generated posterior ensemble provides the analysis (a minimum variance estimate of the state). The ensemble covariance offers an estimate

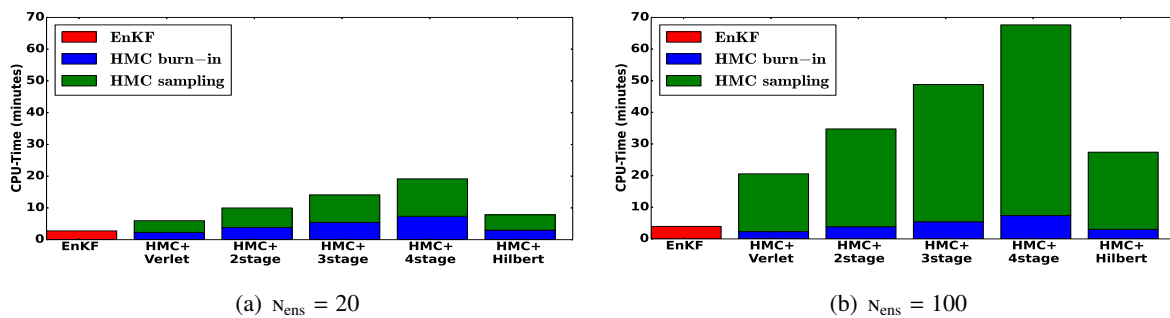


Figure 17. Average CPU-time per assimilation cycle of the numerical experiments carried out using SWE model on a sphere. The time reported is the average CPU-time taken over 100 identical runs of each experiment. The ensemble size is indicated under each panel. The length of the Hamiltonian trajectory is fixed to $T = mh$, with $h = 0.01$, $m = 10$. The number of burn-in steps and the number of mixing steps are set to 50 and 5, respectively.

of the analysis error covariance matrix and can be used to quantify the uncertainty associated with the analysis state. The implementation does not require the construction of full covariance matrices, which makes the method attractive for large scale data assimilation problems with operational models and complex observation operators.

Numerical experiments are carried out with the Lorenz-96 model with several observation operators with different levels of non-linearity and smoothness. The sampling filter provides results similar to EnKF for linear observations. For nonlinear observations the new filter outperforms EnKF, and its performance can be improved further with careful tuning of filter parameters. In addition, the sampling filter produces satisfactory results for test cases where EnKF, MLEF, and IEnKF fail.

Large scale ensemble filtering data assimilation problems are typically run on large parallel machines. One important challenge is the failure of subsets of nodes, which terminates some of the ensemble member runs, and leads to fewer ensemble members being available at the next time. Over several cycles the effective number of ensemble members can decrease considerably. The sampling strategy proposed herein can be used to replace dead ensemble members in any parallel implementation of the EnKF. In addition, the sampling filter can be used in combination with classical filters by building analysis ensembles that have members given by EnKF analysis (these members retain the history of the system) mixed with sampling members (which are consistent with the posterior probability density, but add new directions to explore and can therefore avoid filter divergence).

The computational performance of the sampling filter depends on tuning its parameters, especially the symplectic integration time step and the number of steps taken in the Markov chain between successive accepted ensemble members. Future work will focus on refining the strategies for parameter tuning in the context of large operational models at high-resolution, and on incorporating acceleration strategies for the HMC process. Use of non-Gaussian kernels such as Gaussian mixtures to represent the prior will also be investigated. A side by side comparison between the proposed sampling filter and the successful particle filters is planned. Balance is a concern for all filters including the sampling filter. A possible solution could be to restrict posterior PDF with physical information. This direction will be also investigated in future work.

A. Background and observation error covariances for Lorenz-96 experiments

The initial background error covariance matrix \mathbf{B}_0 , and the observations error covariance matrix \mathbf{R} used with Lorenz-96 model experiments are built as described below.

A.1. Initial background error covariance matrix \mathbf{B}_0

Let $\delta\mathbf{x} = \sigma_{\mathbf{x}_0} \mathbf{x}_0^{\text{ref}}$, where $\mathbf{x}_0^{\text{ref}}$ is the reference initial condition. $\sigma_{\mathbf{x}_0}$ is the noise level in initial background solution. The initial background error covariance matrix is given by:

$$\mathbf{B}_0 = 0.1 \times \mathbf{I} + (0.9 \times (\delta\mathbf{x} (\delta\mathbf{x})^T) \circ \rho), \quad (28)$$

where \mathbf{I} is the identity matrix, and ρ is the decorrelation matrix defined with localization radius $L = 4$. The first term is added to prevent zero or small variances from taking place. The perturbation vector $\delta\mathbf{x}$ is given by:

$$\begin{aligned} \delta\mathbf{x} = [& 0.2581, 0.2262, 0.2867, 0.4257, 0.6204, -0.0480, \\ & -0.0213, 0.4307, 0.2429, -0.3132, 0.1184, 0.3484, \\ & 0.6099, -0.1823, 0.1344, 0.3489, 0.6167, -0.3491, \\ & 0.5768, 0.1640, 0.0068, 0.4713, 0.3250, 0.0875, \\ & 0.3577, 0.6307, 0.4373, 0.1470, -0.0495, -0.1448, \\ & 0.0189, 0.5290, 0.2887, -0.1785, 0.2546, 0.5911, \\ & -0.1673, 0.2455, 0.6292, 0.7743]^T, \end{aligned}$$

where the noise level is set to $\sigma_{\mathbf{x}_0} = 0.08$.

A.2. Observation error covariance matrices \mathbf{R}

Observation error covariance matrices \mathbf{R} are assumed to be diagonal and are created based on the observation operator in hand. Diagonals of \mathbf{R} for the used observation operators are given as follows:

- Linear observation operator (21):

$$[0.0273, 0.0271, 0.0263, 0.0326, 0.0314, 0.0258, 0.0283, \\ 0.0273, 0.0323, 0.0287, 0.0294, 0.0340, 0.0223, 0.0281]^T.$$

- Quadratic observation operator (22) with a threshold $a = 0.5$:

$$[0.6901, 0.6022, 0.6442, 0.8984, 0.8009, 0.6371, 0.7297, \\ 0.6929, 1.0260, 0.7944, 0.8087, 1.1770, 0.5506, 0.7371]^T.$$

- Exponential observation operator (24) with factor $r = 0.2$:

$$[0.0093, 0.0090, 0.0094, 0.0109, 0.0106, 0.0092, 0.0095, \\ 0.0093, 0.0123, 0.0089, 0.0104, 0.0136, 0.0083, 0.0089]^T.$$

- Exponential observation operator (24) with factor $r = 0.5$:

$$[0.3096, 0.2065, 0.3227, 0.4626, 0.3911, 0.2820, 0.3281, \\ 0.3266, 0.7467, 0.4050, 0.4228, 1.1328, 0.3087, 0.3206]^T.$$

These observation error covariance matrices are created based on standard deviation of noise set to 5% of average magnitude of theoretical observations corresponding to the truth.

B. Symplectic numerical integrators

Here we present the five numerical integrators employed in this work. We start with the standard position Verlet integrator in B.1. The results of the standard Verlet are very sensitive to the choice of the time step. Three higher order integrators namely, two-stage (B.2), three-stage (B.3), and four-stage (B.4) position splitting integrators, are taken from [10]. These higher-order integrators lead to filters that are more stable and efficient than Verlet. The last integrator tested (B.5) is from [8] and is designed to work efficiently in infinite dimensional state spaces, and to avoid problems resulting from subtracting infinitely large numbers related to the total energy of the Hamiltonian system for infinite dimensional state spaces.

All the integrators are applied to a Hamiltonian system of the form (9) [48, 49].

B.1. Position Verlet integrator

One step of the position Verlet algorithm advances the solution of the Hamiltonian equations (9) from time t_k to time $t_{k+1} = t_k + h$ as follows [48]:

$$\begin{aligned} \mathbf{x}_{k+1/2} &= \mathbf{x}_k + \frac{h}{2} \mathbf{M}^{-1} \mathbf{p}_k, \\ \mathbf{p}_{k+1} &= \mathbf{p}_k - h \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_{k+1/2}), \\ \mathbf{x}_{k+1} &= \mathbf{x}_{k+1/2} + \frac{h}{2} \mathbf{M}^{-1} \mathbf{p}_{k+1}. \end{aligned}$$

The optimal time step h is $h \propto (1/N_{\text{var}})^{1/4}$ [7]. The experiments show that the step size should be small (close to zero) to make this integrator stable. It may still fail for high dimensionality and whenever complications are present in the target distributions. The weakness of this simple integrator is illustrated in our numerical experiments with highly nonlinear observation operators.

B.2. Two-stage integrator

One step of the two-stage algorithm advances the solution of the Hamiltonian equations (9) from time t_k to time $t_{k+1} = t_k + h$ as follows [10]:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_k + (a_1 h) \mathbf{M}^{-1} \mathbf{p}_k, & \mathbf{p}_1 &= \mathbf{p}_k - (b_1 h) \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_1), \\ \mathbf{x}_2 &= \mathbf{x}_1 + (a_2 h) \mathbf{M}^{-1} \mathbf{p}_1, & \mathbf{p}_{k+1} &= \mathbf{p}_1 - (b_1 h) \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_2), \\ \mathbf{x}_{k+1} &= \mathbf{x}_2 + (a_2 h) \mathbf{M}^{-1} \mathbf{p}_{k+1}, \end{aligned}$$

where $a_1 = 0.21132$, $a_2 = 1 - 2a_1$, and $b_1 = 0.5$.

Linear stability analysis using a standard harmonic oscillator test problem [10] shows that the stability of this symplectic integrator is achieved for time step that lies in the interval $0 < h\omega < 2.6321480259$, where ω is the frequency of the oscillator.

B.3. Three-stage integrator

One step of the three-stage algorithm advances the solution of the Hamiltonian equations (9) from time t_k to time $t_{k+1} = t_k + h$ by the set of equations [10]:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_k + (a_1 h) \mathbf{M}^{-1} \mathbf{p}_k, & \mathbf{p}_1 &= \mathbf{p}_k - (b_1 h) \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_1), \\ \mathbf{x}_2 &= \mathbf{x}_1 + (a_2 h) \mathbf{M}^{-1} \mathbf{p}_1, & \mathbf{p}_2 &= \mathbf{p}_1 - (b_2 h) \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_2), \\ \mathbf{x}_3 &= \mathbf{x}_2 + (a_2 h) \mathbf{M}^{-1} \mathbf{p}_2, & \mathbf{p}_{k+1} &= \mathbf{p}_2 - (b_1 h) \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_3), \\ \mathbf{x}_{k+1} &= \mathbf{x}_3 + (a_1 h) \mathbf{M}^{-1} \mathbf{p}_{k+1}, \end{aligned}$$

where: $a_1 = 0.11888010966548$, $a_2 = 0.5 - a_1$, $b_1 = 0.29619504261126$, and $b_2 = 1 - 2b_1$.

Following the same argument as in B.2, the stability interval of the time step associated with this time integrator is of length ≈ 4.67 , that is, h should be chosen such that $0 < h\omega < 4.67$, where ω is the frequency of the oscillator [10].

B.4. Four-stage integrator

One step of the four-stage algorithm advances the solution of the Hamiltonian equations (9) from time t_k to time $t_{k+1} = t_k + h$ as follows [10]:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_k + (a_1 h) \mathbf{M}^{-1} \mathbf{p}_k, & \mathbf{p}_1 &= \mathbf{p}_k - (b_1 h) \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_1), \\ \mathbf{x}_2 &= \mathbf{x}_1 + (a_2 h) \mathbf{M}^{-1} \mathbf{p}_1, & \mathbf{p}_2 &= \mathbf{p}_1 - (b_2 h) \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_2), \\ \mathbf{x}_3 &= \mathbf{x}_2 + (a_3 h) \mathbf{M}^{-1} \mathbf{p}_2, & \mathbf{p}_3 &= \mathbf{p}_2 - (b_2 h) \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_3), \\ \mathbf{x}_4 &= \mathbf{x}_3 + (a_2 h) \mathbf{M}^{-1} \mathbf{p}_3, & \mathbf{p}_{k+1} &= \mathbf{p}_3 - (b_1 h) \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_4), \\ \mathbf{x}_{k+1} &= \mathbf{x}_4 + (a_1 h) \mathbf{M}^{-1} \mathbf{p}_{k+1}, \end{aligned}$$

where: $a_1 = 0.071353913450279725904$, $a_2 = 0.268458791161230105820$, $a_3 = 1 - 2a_1 - 2a_2$, $b_1 = 0.1916678$, and $b_2 = 0.5 - b_1$.

Again, as discussed in B.2, and B.3, this integrator has a stability interval $0 < h\omega < 5.35$ [10].

The time here has unspecified units. Generally speaking, the high order integrators (30, 31, 32), provide more favorable and wider stability ranges for the time step. For more on the stability intervals of the time step settings of these high-order integrators, see [10].

B.5. General integrator defined on Hilbert space

One step of the Hilbert integrator advances the solution of the Hamiltonian equations (9) from time t_k to time $t_{k+1} = t_k + h$ as follows [8]:

$$\begin{aligned} \mathbf{p}_1 &= \mathbf{p}_k - \frac{h}{2} \mathbf{M}^{-1} \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_k), \\ \mathbf{x}_{k+1} &= \cos(h) \mathbf{x}_k + \sin(h) \mathbf{p}_1, \end{aligned}$$

$$\begin{aligned}\mathbf{p}_2 &= -\sin(h)\mathbf{x}_k + \cos(h)\mathbf{p}_1, \\ \mathbf{p}_{k+1} &= \mathbf{p}_2 - \frac{h}{2}\mathbf{M}^{-1}\nabla_{\mathbf{x}}\mathcal{J}(\mathbf{x}_{k+1}).\end{aligned}$$

As with the standard position Verlet integrator the selection criterion of step size is not precisely defined, however, it is designed to work with finite (non-zero) steps in infinite dimensional settings. Numerical results presented in Section 5 show that with careful tuning this integrator provides satisfactory results.

Acknowledgments

This work was supported in part by awards NSF CCF-1218454 and AFOSR FA9550-12-1-0293-DEF, and by the Computational Science Laboratory at Virginia Tech.

References

1. Ades M, Van Leeuwen PJ, (2015) The equivalent-weights particle filter in a high-dimensional system. *Quarterly J Royal Meteorological Society* Volume: 141, 484–503.
2. Alexander F, Eyink G, Restrepo J, (2005) Accelerated Monte Carlo for optimal estimation of time series. *J Statistical Physics* Volume: 119, 1331-1345.
3. Attia A, Sandu A, (2014) A Sampling Filter for Non-Gaussian Data Assimilation. *Cornell University, arXiv Preprint arXiv:1403.7137*. [Available from <http://arxiv.org/abs/1403.7137>.]
4. Anderson JL, (1996) A method for producing and evaluating probabilistic forecasts from ensemble model integrations. *J Climate* Volume: 9, 1518-1530.
5. Anderson JL, (2001) An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Rev* Volume: 129, 2884-2903.
6. Bennett A, Chua B, (1994) Open-ocean modeling as an inverse problem: the primitive equations. *Monthly Weather Rev* Volume: 122, 1326–1336.
7. Beskos A, Pillai N, Roberts G, Sanz-Serna JM, Stuart A, (2013) Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli* Volume: 19, 1501-1534.
8. Beskos A, Pinski FJ, Sanz-Serna JM, Stuart A, (2011) Hybrid Monte Carlo on Hilbert spaces. *Stochastic Processes Applications* Volume: 121.
9. Bishop CH, Etherton BJ, and Majumdar SJ, (2001) Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly Weather Rev* Volume: 129, 420-436.
10. Blanes S, Casas F, Sanz-Serna JM, (2014) Numerical integrators for the hybrid Monte Carlo method. *SIAM J on Scientific Computing* Volume: 36, A1556-A1580.
11. Burgers G, Van Leeuwen PJ, Evensen G, (1998) Analysis scheme in the ensemble Kalman filter. *Monthly Weather Rev* Volume: 126, 1719-1724.
12. Chorin A, Morzfeld M, Tu X, (2010) Implicit particle filters for data assimilation. *Communications in Applied Mathematics and Computational Sci* Volume: 5, 221-240.

13. Cohn SE, (1997) An introduction to estimation theory. *J Meteorological Society of Japan* Volume: 75, 257-288.
14. S. L. Cotter and M. Dashti and A. M. Stuart, (2012) Variational data assimilation using targeted random walks *Inter J numerical methods in fluids* Volume: 68, 403-421.
15. Doucet A, De Freitas N, Gordon NJ, (2001) An introduction to sequential Monte Carlo methods. *Series Statistics For Engineering and Information Sci.* Springer.
16. Duane S, Kennedy AD, Pendleton BJ, and Roweth D, (1987) Hybrid Monte Carlo. *Physics Letters B* Volume: 195, 216-222.
17. Evensen G, (1994) Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J Geophysical Res: Oceans* Volume: 99, 10143-10162.
18. Evensen G, (2003) The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics* Volume: 53, 343-367.
19. Evensen G, (2007) Data assimilation: The ensemble Kalman filter. Springer.
20. Fisher M, Courtier P, (1995) Estimating the covariance matrices of analysis and forecast error in variational data assimilation. *European Center for Medium-Range Weather Forecasts*
21. Girolami M, Calderhead B, (2011) Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J Royal Statistical Society: Series B (Statistical Methodology)* Volume: 73, 123-214.
22. Gordon Neil, Salmond D, Smith A, (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)* Volume: 140, 107-113.
23. Gu Y, Oliver D, (2001) *An iterative ensemble Kalman filter for multiphase fluid flow data assimilation.* Spe Journal, 12:438–446. Society of Petroleum Engineers.
24. Hamill TM, Whitaker JS, Snyder C, (2001) Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Rev* Volume: 129, 2776-2790.
25. Hoffman MD, Gelman A, (2014) The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *The J Machine Learning Res* Volume: 15(1), 1593–1623.
26. Houtekamer PL, Mitchell HL, (1998) Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Rev* Volume: 126, 796-811.
27. Houtekamer PL, Mitchell HL, (2001) A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Rev* Volume: 129, 123-137.
28. Jardak M, Navon IM, Zupanski M, (2010) Comparison of sequential data assimilation methods for the Kuramoto-Sivashinsky equation. *Inter J Numerical Methods in Fluids* Volume: 62, 374-402.
29. Jardak M, Navon IM, Zupanski M, (2013) Comparison of Ensemble Data Assimilation methods for the shallow water equations model in the presence of nonlinear observation operators. *Submitted to Tellus*
30. Kalman RE, (1960) A new approach to linear filtering and prediction problems. *J Fluids Engineering* Volume: 82, 35-45.

31. Kalman RE, Bucy RS, (1961) New results in linear filtering and prediction theory. *J Fluids Engineering* Volume: 83, 95-108.
32. Kalnay E, (2002) Atmospheric modeling, data assimilation and predictability. *Cambridge University Press*
33. Kalnay E, Yang S-C, (2010) Accelerating the spin-up of ensemble Kalman filtering. *Quarterly J Royal Meteorological Society* Volume: 136, 1644-1651.
34. Kitagawa G, (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J Computational and Graphical Statistics* Volume: 5, 1-25.
35. Law K and Stuart A, (2012) Evaluating data assimilation algorithms. *Monthly Weather Rev* Volume: 140, 3757-3782.
36. Liu JS, (2008) Monte Carlo strategies in scientific computing. Springer.
37. Lorenc AC, (1986) Analysis methods for numerical weather prediction. *Quarterly J Royal Meteorological Society* Volume: 112, 1177-1194.
38. Lorenz EN, (1996) Predictability: A problem partly solved. *Proc. Seminar on Predictability* Volume: 1.
39. Lorenz EN, Emanuel KA, (1998) Optimal sites for supplementary weather observations: Simulation with a small model. *J Atmospheric Sciences* Volume: 55, 399-414.
40. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E, (1953) Equation of state calculations by fast computing machines. *J Chemical Phys* Volume: 21, 1087-1092.
41. Nakano S, Ueno G, Higuchi T, (2007) Merging particle filter for sequential data assimilation. *Nonlinear Processes in Geophys* Volume: 14, 395-408.
42. Neal RM, (1993) Probabilistic inference using Markov chain Monte Carlo methods. *Department of Computer Science, University of Toronto Toronto, Ontario, Canada.*
43. Neal RM, (2011) MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo.*
44. Neta B, Giraldo FX, Navon IM, (1997) Analysis of the Turkel-Zwas scheme for the two-dimensional shallow water equations in spherical coordinates. *J Computational Phys* Volume: 133, 102-112.
45. Nino Ruiz ED, Sandu A, Anderson JL, (2014) An efficient implementation of the ensemble Kalman filter based on an iterative Sherman–Morrison formula. *Statistics and Computing* Volume: 1-17.
46. Ott E, Hunt BR, Szunyogh I, Zimin AV, Kostelich EJ, Corazza M, Kalnay E, Patil DJ, Yorke JA, (2004) A local ensemble kalman filter for atmospheric data assimilation. *Tellus A* Volume: 56, 415-428.
47. Sakov P, Oliver D, Bertino L. 2012. An iterative EnKF for strongly nonlinear systems. *Monthly Weather Rev* Volume: 140, 1988-2004.
48. Sanz-Serna JM, (2014) Markov chain Monte Carlo and numerical differential equations. *Current Challenges in Stability Issues for Numerical Differential Equations* Volume: 39-88.

49. Sanz-Serna JM , Calvo MP, (1994) Numerical Hamiltonian problems *Applied Mathematical Computation* Volume: 7. Chapman & Hall London.
50. Simon E, Bertino L, (2009) Application of the Gaussian anamorphosis to assimilation in a 3-D coupled physical-ecosystem model of the North Atlantic with the EnKF: a twin experiment. *Ocean Science* Volume: 5, 495-510.
51. Snyder C, Bengtsson T, Bickel P, Anderson J, (2008) Obstacles to high-dimensional particle filtering. *Monthly Weather Rev* Volume: 136, 4629-4640.
52. St-Cyr A, Jablonowski C, Dennis JM, Tufo HM, Thomas SJ, (2007) A comparison of two shallow water models with nonconforming adaptive grids. *Monthly Weather Rev* Volume: 136, 1898-1922.
53. Talagrand O, Vautard R, Strauss B, (1997) Evaluation of probabilistic prediction systems. *Proc. ECMWF Workshop on Predictability* Volume: 1-25.
54. Tierney L, (1994) Markov chains for exploring posterior distributions. *The Ann Statistics* Volume: 1701-1728.
55. Tippett MK, Anderson JL, Bishop CH, Hamill TM, Whitaker JS, (2003) Ensemble square root filters. *Monthly Weather Rev* Volume: 131, 1485-1490.
56. Van Leeuwen PJ, (2009) Particle filtering in geophysical systems. *Monthly Weather Rev* Volume: 137, 4089-4114.
57. Van Leeuwen PJ, (2010) Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly J Royal Meteorological Society* Volume: 136, 1991-1999.
58. Van Leeuwen PJ, (2011) Efficient nonlinear data-assimilation in geophysical fluid dynamics. *Computers & Fluids* Volume: 46, 52-58.
59. Whitaker JS, Hamill TM, (2002) Ensemble data assimilation without perturbed observations. *Monthly Weather Rev* Volume: 130, 1913-1924.
60. Zupanski M, (2005) Maximum likelihood ensemble filter: Theoretical aspects. *Monthly Weather Rev* Volume: 133, 1710-1726.
61. Zupanski M, Navon IM, Zupanski D, (2008) The maximum likelihood ensemble filter as a non-differentiable minimization algorithm. *Quarterly J Royal Meteorological Society* Volume: 134, 1039-1050.



©2015, Attia A., licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)