RESAMPLED ENSEMBLE KALMAN INVERSION FOR
BAYESIAN PARAMETER ESTIMATION
WITH SEQUENTIAL DATA

JIANGQI WU AND LINJIE WEN

School of Mathematical Sciences, Shanghai Jiao Tong University
800 Dongchuan Rd, Shanghai 200240, China

JINGLAI LI∗

School of Mathematics, University of Birmingham
Edgbaston Birmingham, B15 2TT, UK

ABSTRACT. Many real-world problems require to estimate parameters of interest in a Bayesian framework from data that are collected sequentially in time. Conventional methods to sample the posterior distributions, such as Markov Chain Monte Carlo methods can not efficiently deal with such problems as they do not take advantage of the sequential structure. To this end, the Ensemble Kalman inversion (EnKI), which updates the particles whenever a new collection of data arrive, becomes a popular tool to solve this type of problems. In this work we present a method to improve the performance of EnKI, which removes some particles that significantly deviate from the posterior distribution via a resampling procedure. Specifically we adopt an idea developed in the sequential Monte Carlo sampler, and simplify it to compute an approximate weight function. Finally we use the computed weights to identify and remove those particles seriously deviating from the target distribution. With numerical examples, we demonstrate that, without requiring any additional evaluations of the forward model, the proposed method can improve the performance of standard EnKI in certain class of problems.

1. Introduction. The Bayesian inference is a popular method to estimate unknown parameters from data, largely due to its ability to quantify uncertainty in the estimation results [11]. In this work we consider a special type of Bayesian inference problems where data have to be collected in a sequential manner. A typical example of this type of problems is to estimate certain parameters, such as the initial states or the coefficients, in a dynamical system from observations related to the state vector at discrete times. Such problems arise from many real-world applications, ranging from weather prediction [1] to biochemical networks [13, 19]. It should be emphasized that, unlike the usual data assimilation problems that seek to estimate the time-dependent states in dynamical systems, the parameters that we want to estimate here do not vary in time. To distinguish the two types of problems, we refer to the former as the state estimation problems and the latter as the parameter estimation problems. Conventional methods to compute the posterior distributions, such as the Markov Chain Monte Carlo (MCMC) simulation [12],

∗ Corresponding author: Jinglai Li.
which use all the data in a single batch, are unable to take advantage of the sequential structure of the type of problems. On the other hand, the sequential methods utilize the sequential structure of the problems and update the posterior whenever a new collection of data become available, which makes them particularly convenient and efficient for the sequential inference problems.

The ensemble Kalman filter (EnKF) is a very popular method for sampling the posteriors in the dynamical state estimation problems [9]. The EnKF method was extended to estimate parameters in many practical problems, e.g., [1, 2], and more recently it was generically formulated as a derivative-free optimization based parameter estimation method in [15]. The new sequential parameter estimation method is termed as the Ensemble Kalman inversion (EnKI), and it was further developed and analyzed in [4, 14, 8, 18], etc. The basic idea of the EnKI method for parameter estimation is to construct an artificial dynamical system, turning the parameters of interest into the states of the constructed dynamical system, and then apply the standard EnKF procedure to estimate the states of the system. Since the particles in EnKI are updated based on the Gaussian assumption, some of the particles may deviate significantly from the actual target distribution, causing estimation errors. Moreover the error due to this Gaussian assumption, unlike the random sampling error, can not be reduced by increasing the sample size. In the EnKF context, such an error can be partially corrected by assigning appropriate weight to each particle [17, 10], and then resampling according to the weights can essentially eliminate those deviating particles, leading to more accurate estimates than the standard EnKF. However, as will be shown in Section 3, in the sequential inference framework, computing the weight directly becomes infeasible. To tackle the issue, we make use of a technique developed in the Sequential Monte Carlo sampler (SMCS) methods [6], a method to sample from a sequence of distribution. Namely, we adopt the idea in SMCS and make some approximations, which then allows us to compute the particle weight approximately. Based on this idea, we develop an EnKI algorithm with an additional step to resample according to the weight, and we demonstrate with numerical examples that the proposed method has an improved performance over the standard EnKF in a range of problems.

The remaining of the work is organized as follows. In Section 2 we present the generic setup of the sequential inverse problems that we consider in this work. In Sections 3 and 4 we respectively review the EnKI and the SMCS methods. In Section 5 we present the proposed resampled EnKI method and in Section 6 we provide several numerical examples to illustrate the performance of the proposed method. Finally Section 7 offers some concluding remarks.

2. Problem setup. We consider a sequential inference problem formulated as follows, and as is mentioned earlier such problems arise frequently in the field of biochemical networks [13]. Suppose that we want to estimate the parameter \( x \in \mathbb{R}^n_x \) from data \( y_1, ..., y_t, ..., y_T \) which become available sequentially in time. In particular the data \( y_t \in \mathbb{R}^n_y \) is related to the parameter of interest \( x \) via the follow model,

\[
y_t = G_t(x) + \eta_t, \quad t = 1...T,
\]

where each \( G_t(\cdot) \) is a mapping from \( \mathbb{R}^n_x \) to \( \mathbb{R}^n_y \), and the observation noise \( \eta_t \sim \mathcal{N}(0, R_t) \). It follows that the likelihood function can be written as,

\[
\pi(y_t|x) = \mathcal{N}(G_t(x), R_t), \quad t = 1...T.
\] (2.1)
It is important to note here that, the restriction that the error model has to be additive Gaussian as is in Eq. (2.1) is due to the use of EnKF. While acknowledging that relaxing such a restriction is desirable, we note here that additive Gaussian noise is a reasonable assumption for a wide range of practical problems. We now can write the posterior distribution in a sequential form:

\[
\pi_t(x) = \pi(x | y_1, \ldots, y_t) \propto \pi_0(x) \prod_{i=1}^{t} \pi(y_i | x) \propto \pi_{t-1}(x) \pi(y_t | x),
\]  

(2.2)

where \( \pi_0(x) \) is the prior distribution of \( x \), and our goal is to draw samples from \( \pi_t \) for any \( 0 < t \leq T \). It is important to mention here that in such problems the main computational cost lies in the evaluation of the likelihood function \( \pi(y_t | x) \), which involves the computation of the forward model \( G_t \).

It can also be seen from Eq. (2.2) that it is particularly expensive to evaluate the final posterior \( \pi_T(x) \) as each evaluation requires the computation of the forward model from \( G_1 \) to \( G_T \). To this end, these sequential inference problems pose serious computational challenges for MCMC methods which require to evaluate \( \pi_T(x) \) repeatedly, especially when the amount of data, i.e., \( T \), is large. In what follows, we discuss the sequential methods to address this type of problems.

3. Ensemble Kalman inversion. In this section we give a brief overview of the EnKI parameter estimation method proposed in [15], which essentially aims to compute a Gaussian approximation of \( \pi_t(x_t) \) in each time step \( t \) using an EnKF framework. To formulate the problem in an EnKF framework, we first construct an artificial dynamical system: at any time \( t \), we have the states \( u_t = [x_t, z_t]^T \) where \( z_t = G(x_t) \), and the dynamical model,

\[
u_t = F_t(u_{t-1}) : \quad x_t = x_{t-1}, \quad z_t = G_t(x_t).
\]  

(3.1)

The data is associated to the states as \( y_t = z_t + \eta_t \), or equivalently

\[
y_t = Hu_t + \eta_t = [\theta_{n_y \times n_x}, I_{n_y \times n_y}] u_t + \eta_t,
\]

where \( I_{n_y \times n_y} \) is a \( n_y \times n_y \) identity matrix and \( \theta_{n_y \times n_x} \) is a \( n_y \times n_x \) zero matrix. We emphasize here that once we have the posterior distribution \( \pi(u_t | y_{1:t}) \), we can obtain the posterior \( \pi_t(x_t) = \pi(x_t | y_{1:t}) \) by marginalizing \( \pi(u_t | y_{1:t}) \) over \( z_t \).

Now let us see how EnKF proceeds to compute a Gaussian approximation of the posterior distribution \( \pi(u_t | y_{1:t}) \). At time \( t \), suppose that the prior \( \pi(u_t | y_{1:t-1}) \) can be approximated by a Gaussian distribution with mean \( \hat{\mu}_t \) and covariance \( \hat{C}_t \). It follows that the posterior distribution \( \pi(u_t | y_{1:t}) \) is also Gaussian and its mean and covariance can be obtained analytically:

\[
\mu_t = \hat{\mu}_t + Q_t(y_t - H\hat{\mu}_t), \quad C_t = (I - Q_tH)\hat{C}_t,
\]  

(3.2)

where \( I \) is the identity matrix and

\[
Q_t = \hat{C}_t H^T (H\hat{C}_t H^T + R_t)^{-1}
\]  

(3.3)

is the so-called Kalman gain matrix.

In the EnKF method, one avoids computing the mean and the covariance directly in each step. Instead, both the prior and the posterior distributions are represented with a set of samples, known as an ensemble. Suppose that at time \( t - 1 \), we have an ensemble of particles \( \{u_{t-1}^m\}_{m=1}^M \) drawn according to the posterior distribution \( \pi(u_{t-1} | y_{1:t-1}) \), and we then can propagate the particles via the dynamical
model (3.1):

\[
\tilde{u}_m^t = F_t(u_{t-1}^m),
\]

for \(m = 1...M\), obtaining an ensemble \(\{\tilde{u}_m^t\}_{m=1}^{M}\) following the prior \(\pi(u_t|y_{1:t-1})\).

We can compute a Gaussian approximation \(N(\tilde{\mu}_t, \tilde{C}_t)\) of \(\pi(u_t|y_{1:t-1})\), where the mean and the covariance of \(\pi(u_t|y_{1:t-1})\) are estimated from the samples:

\[
\tilde{\mu}_t = \frac{1}{M} \sum_{m=1}^{M} \tilde{u}_t^m, \quad \tilde{C}_t = \frac{1}{M-1} \sum_{m=1}^{M} (\tilde{u}_t^m - \tilde{\mu}_t)(\tilde{u}_t^m - \tilde{\mu}_t)^T.
\]

Once \(\tilde{\mu}_t\) and \(\tilde{C}_t\) are obtained, one then can compute \(\mu_t\) and \(C_t\) directly from Eq. (3.2), and by design, the posterior distribution \(\pi(u_t|y_{1:t})\) is approximated by \(\mathcal{N}(\mu_t, C_t)\). Moreover it can be verified that the samples

\[
u_t^m = \tilde{u}_t^m + Q_t(y_t - (H\tilde{u}_t^m - \eta_t^m)), \quad \eta_t^m \sim N(0, R_t), \quad m = 1...M,
\]

with \(Q_t\) computed by Eq. (3.3), follow the distribution \(\mathcal{N}(\mu_t, C_t)\). That is, \(\{\nu_t^m\}_{m=1}^{M}\) are the approximate ensemble of \(\pi(u_t|y_{1:t})\), and consequently the associated \(\{x_t^m\}_{m=1}^{M}\) approximately follows distribution \(\pi_t(x_t) = \pi(x_t|y_{1:t})\).

As is mentioned earlier, the EnKI method can only provide a biased estimator of the actual posterior distribution. In the data assimilation literature, it is well known that the bias in EnKF can be corrected by assigning appropriate weight to each particle, as is done in the so-called weighted EnKF (WEnKF) [17]. Simply put, WEnKF computes the weight of each particle as follows [17],

\[
u_t^m = w_{t-1}^m \frac{\pi(y_t|u_t^m)\pi(u_t^m|u_{t-1}^m)}{\mathcal{N}(\tilde{u}_t^m + Q_t(y_t - H\tilde{u}_t^m), Q_tR_tQ_t^T)},
\]

One can see that this method does not apply directly to the present problem as in our setting, \(\pi(u_t^m|u_{t-1}^m)\) is a delta function: \(\pi(u_t^m|u_{t-1}^m) = 0\) if \(u_t^m \neq u_{t-1}^m\). To address the issue we shall borrow the idea from the sequential Monte Carlo sampler [6], which is introduced in the next section.

4. The sequential monte carlo sampler. We first give a brief introduction to the SMCS method for sampling the posterior distribution \(\pi_t(x_t)\), which is specifically tailored to fit our purpose. The key idea of SMCS is to construct a joint distribution \(\pi(x_{t-1}, x_t)\), the marginal of which is equal to the target distribution \(\pi_t(\cdot)\). One then applies importance sampling to draw weighted samples from \(\pi(x_{t-1}, x_t)\), which after being marginalized over \(x_{t-1}\) yields weighted samples from \(\pi_t(\cdot)\).

Let \(L_{t-1}(x_{t-1}|x_t)\) be an arbitrary conditional distribution and we can define a joint distribution of \(x_{t-1}\) and \(x_t\),

\[
\pi(x_{t-1}, x_t) = \pi_t(x_t)L_{t-1}(x_{t-1}|x_t),
\]

which is the key of SMCS. It is easy to see that the marginal distribution of \(\pi(x_{t-1}, x_t)\) over \(x_{t-1}\) is \(\pi_t(x_t)\). Now choose a conditional distribution \(K_t(x_t|x_{t-1})\) and we can construct an importance sampling (IS) distribution for \(\pi_t(x_{t-1}, x_t)\) in the form of

\[
q(x_{t-1}, x_t) = \pi_{t-1}(x_{t-1})K_t(x_t|x_{t-1}).
\]

Namely, we use \(q(x_{t-1}, x_t)\) as the IS distribution to draw weighted samples from \(\pi(x_{t-1}, x_t)\). Let \(\{x_m^m\}_{m=1}^{M}\) be an ensemble drawn from \(q(x_{t-1}, x_t)\), and it is
easy to see that the weighted ensemble \( \{(x_{t-1}, t, w^m_t)\} \) follows the distribution \( \pi(x_{t-1}, t) \), with weight \( w_t(x) \) computed as,

\[
w_t(x_{t-1}; t) = \frac{\pi(x_{t-1}, x_t)}{q(x_{t-1}, x_t)} = \frac{\pi_t(x_t) L_{t-1}(x_{t-1}|x_t)}{\pi_{t-1}(x_{t-1}) K_t(x_t|x_{t-1})},
\]

(4.3)

As can be seen here, once the two conditional distributions \( K_t \) and \( L_{t-1} \) (referred to as the forward and the backward kernels in the rest of the paper) are chosen, we can draw samples from Eq. (4.2) and compute the associated weights from Eq. (4.3), obtaining a weighted ensemble \( \{(x^m_t, x^m_t, w^m_t)\} \) from \( \pi(x_{t-1}, x_t) \). Now recall that \( \pi_t(x_t) \) is the marginal distribution of \( \pi(x_{t-1}, x_t) \), and by marginalizing the ensemble over \( x_{t-1} \), we obtain a weighted ensemble \( \{(x^m_t, w^m_t)\} \) of \( \pi_t(x_t) \).

As can be seen from the discussion above, the key in SMCS is to determine the two kernels. In principle optimal choices of these kernels are available. It is known that once \( K_t(x_t|x_{t-1}) \) is provided, one can derive that the optimal choice of \( L_{t-1}(x_{t-1}|x_t) \) is [6]:

\[
L_{t-1}^{opt}(x_{t-1}|x_t) = \frac{\pi_{t-1}(x_{t-1}) K_t(x_t|x_{t-1})}{q_t(x_t)} = \frac{\pi_{t-1}(x_{t-1}) K_t(x_t|x_{t-1})}{\int \pi_{t-1}(x_{t-1}) K_t(x_t|x_{t-1}) dx_{t-1}},
\]

(4.4)

where the optimality is in the sense of yielding the minimal estimator variance. We are offering two remarks here: first as is discussed earlier, the forward kernel \( K_t \) is given by EnKI, the derivation of which is provided in next section; second we do not employ the actual SMCS method in this work as it is considerably more expensive than EnKI (details will be discussed in the next section), and rather we adopt its idea to derive an approximate weight function for the EnKI particles.

5. Resampled ensemble Kalman inversion. We present the resampled EnKI method in this section. Now recall that the update scheme in EnKI is defined for the artificially expanded variable \( u_t \), and in what follows we shall discuss how to rewrite it in the form of a forward kernel \( K_t \) defined for \( x_t \). Once the kernel \( K_t \) is derived, it becomes possible to use Eq. (4.3) to compute the weight of each particle. Given \( u_t = [x_t, z_t]^T, H = [\theta_{n_y \times n_x}, 1_{n_y \times n_y}] \) and the propagation model \( x_t = x_{t-1}, \) we can derive from Eq. (3.6) that,

\[
x_t = x_{t-1} + Q^x_t(y_t - G(x_{t-1})) + Q^\eta_t \eta_t + \eta'_t, \quad \eta_t \sim \mathcal{N}(0, R_t), \quad \eta'_t \sim \mathcal{N}(0, \delta^2 \Sigma_t), \quad (5.1)
\]

where \( Q^x_t = Q_t[1 : n_x, 1 : n_y], \delta \) is a small constant and \( \Sigma_t \) is the covariance of \( \pi_{t-1} \) (the evaluation of \( \Sigma_t \) is provided in Eq. (5.5)). Eq. (5.1) can also be written as a conditional distribution:

\[
K_t(\cdot | x_{t-1}) = \mathcal{N}(\xi^K_t(x_{t-1}), \Sigma^K_t),
\]

(5.2a)

with

\[
\xi^K_t(x_{t-1}) = x_{t-1} + Q^x_t(y_t - G_t(x_{t-1})) \quad \text{and} \quad \Sigma^K_t = Q^x_t R_t (Q^x_t)^T + \delta^2 \Sigma_t.
\]

(5.2b)

Note that the purpose of introducing the small noise \( \eta'_t \) in Eq. (5.1) is to ensure that \( \Sigma^K_t \) is strictly positive definite and so \( K_t \) is a proper Gaussian conditional distribution. In all the numerical implementations performed in this work, \( \delta \) is set to be \( 10^{-2} \). In the SMCS framework, Eq. (5.2) provides a good forward Kernel that is used to draw new particles, and Eq. (4.3) are then used to update the weight of particles. However, computing the particle weights poses serious computational
challenge which makes it infeasible to perform the SMCS exactly. To understand
the challenge here, we re-write Eq. (4.3) as the following,
\[ w(x_{t-1}) = \frac{\pi_t(x_t)}{\pi_{t-1}(x_{t-1})} \times \frac{L_{t-1}(x_{t-1}|x_t)}{K_t(x_t|x_{t-1})}. \] (5.3)
The first issue that needs to be addressed is the computational cost for evaluating
\[ \frac{\pi_t(x_t)}{\pi_{t-1}(x_{t-1})} \] which, as mentioned earlier, involves computing the forward
models from \( G_1(x_t) \) all the way to \( G_t(x_t) \), and the associated computational cost
can be prohibitive. To reduce the computational cost, we propose the following
approximate method to evaluate Eq. (5.3). Namely we first write \( \pi_t(x_t)/\pi_{t-1}(x_{t-1}) \)
as,
\[ \frac{\pi_t(x_t)}{\pi_{t-1}(x_{t-1})} = \frac{\pi_{t-1}(x_t)}{\pi_{t-1}(x_{t-1})} \pi(y_t|x_t) \approx \frac{\hat{\pi}_{t-1}(x_t)}{\pi_{t-1}(x_{t-1})} \pi(y_t|x_t), \] (5.4)
where \( \hat{\pi}_{t-1} = \mathcal{N}(\xi_{t-1}, \Sigma_{t-1}) \) is the Gaussian approximation of \( \pi_{t-1} \). Here \( \xi_{t-1} \) and
\( \Sigma_{t-1} \) are the mean and the variance of \( \hat{\pi}_{t-1} \) respectively, and in practice they can
be estimated from the ensemble \( \{x_{m,t-1}\}_{m=1}^M \):
\[ \xi_{t-1} = \frac{1}{M} \sum_{m=1}^M x_{m,t-1}, \quad \Sigma_{t-1} = \frac{1}{M-1} \sum_{m=1}^M (x_{m,t-1} - \xi_{t-1})(x_{m,t-1} - \xi_{t-1})^T. \] (5.5)
We reinstate here that, with this approximation, the algorithm does not require any
more forward model evaluations than the standard EnKI. Next we shall evaluate
\( \hat{K}_t(x_{t-1}|x_t) \) and once again we shall derive an approximation of the term so that it
can be computed efficiently. The idea is to replace \( K_t(x_{t-1}|x_t) \) with an approximation of
it,
\[ \hat{K}_t(x_{t-1}|x_t) = \mathcal{N}(x_{t-1} + Q^*_t(y_t - \tilde{G}_t x_{t-1}), \Sigma^K_t), \] (5.6)
where \( \tilde{G}_t \) is a linearization of \( G_t \). Moreover, in this case if we take the backward kernel to be
\[ \hat{L}_{t-1}(x_t) = \mathcal{N}(\xi^{L}_{t-1}(x_t), \Sigma^{L}_{t-1}), \] (5.7a)
where
\[ \xi^{L}_{t-1}(x_t) = (I - Q^*_t \tilde{G}_t)^{-1}(x_t - Q^*_t y_t), \] (5.7b)
and
\[ \Sigma^{L}_{t-1} = (I - Q^*_t \tilde{G}_t)^{-1} \Sigma^K_t (I - Q^*_t \tilde{G}_t)^{-1}, \] (5.7c)
it then follows that \( \hat{L}_{t-1}(x_{t-1}|x_t) / \hat{K}_t(x_{t-1}|x_t) \) is a constant. By combining these two terms
together, we obtain an approximation of the weight function:
\[ w(x_{t-1}; x_t) \approx \frac{\hat{\pi}_{t-1}(x_t)}{\pi_{t-1}(x_{t-1})} \pi(y_t|x_t). \] (5.8)
Next since the EnKI scheme uses unweighted particles, following [17] we generate
unweighted samples by conducting a resampling procedure at the end of each
step. In this respect, one possible approach is to use the standard resampling tech-
niques that are well documented in the PF literature, e.g., [5, 7]. However, since
the weights computed here are only approximate, strictly resampling according to
these weights may not improve the estimation results. Alternatively we propose
to use the approximate weights to identify and eliminate the “bad” particles, i.e.,
those deviate significantly from the posterior distribution. Such particles are indi-
cated by very low weights. Specifically, given a prescribed elimination percentage \( \rho \)
(e.g., \( = 20\% \)), our resampling proceeds as follows: in each step the \( 100\rho\% \) particles
with the smallest weights are removed and replaced by particles randomly chosen from the rest of the particles. In practice the value of $\rho$ can be determined by conducting simulations with synthetic data in advance. Since this particle-eliminating resampling step is performed every step, we refer to the proposed method as re-sampled EnKI (REnKI). The complete procedure of REnKI method is described in Algorithm 1. Intuitively, the method can be understood as a modified version of EnKI which improves the sampling efficiency by eliminating particles that deviate significantly from the posterior distribution. In addition, it is easy to see that, the standard EnKI can be regarded as a special case of REnKI with $\rho = 0\%$.

**Algorithm 1** The REnKI algorithm

Initialization: draw sample $\{x^m_0\}_{m=1}^M$ from $\pi_0$;

for $t = 1$ to $T$ do

estimate $\xi_{t-1}$ and $\Sigma_{t-1}$ as the sample mean and variance of the ensemble $\{x^m_{t-1}\}_{m=1}^M$ using Eq. (5.5), and let $\tilde{\pi}_{t-1} = \mathcal{N}(\xi_{t-1}, \Sigma_{t-1})$;

let $\tilde{u}^m_t = [x^m_{t-1}, G(x^m_{t-1})]^T$ for $m = 1...M$;

evaluate $\tilde{\mu}_t$ and $\tilde{C}_t$ with Eq. (3.5), and compute $Q_t$ with Eq. (3.3);

draw $x^m_t \sim K_t(x_t | x^m_{t-1})$ for $m = 1...M$ with $K_t$ given by Eq. (5.2);

update the weights:

$$w^m_t = \frac{\tilde{\pi}_{t-1}(x^m_t) \pi(y_t | x^m_t)}{\tilde{\pi}_{t-1}(x^m_{t-1})}$$

and renormalize $\{w^m_t\}_{m=1}^M$ so that $\sum_{m=1}^M w^m_t = 1$;

let $M_\rho = [\rho M]$;

sort the samples ascendingly according to their weights: $\{x^1_t, ..., x^M_t\}$;

randomly draw $M_\rho$ samples from $\{x^{M_\rho+1}_t, ..., x^M_t\}$, denoted as $\{z^1, ..., z^{M_\rho}\}$;

let $x^m_t = z^m$ for $m = 1, ..., M_\rho$.

end

6. **Numerical examples.** We shall provide three examples to demonstrate the performance of the proposed REnKI algorithm. We emphasize here that in all these examples, we assume that the forward model $G_t$ is computationally intensive and thus the main computational cost arises from the simulation of $G_t$. As a result the main computational cost of all methods is measured by the number of forward model evaluations, which in all the methods used in this section depends only on the number of steps and that of the particles.

6.1. **The Bernoulli model.** Our first example is the Bernoulli equation,

$$\frac{dv}{d\tau} - v = -v^3, \quad v(0) = x,$$  \hspace{1cm} (6.1a)

which has an analytical solution,

$$v(\tau) = G(x, \tau) = x(x^2 + (1 - x^2)e^{-2\tau})^{-1/2}.$$  \hspace{1cm} (6.1b)

This model is an often-used benchmark problem for data assimilation methods as it exhibits strong non-Gaussian behavior [3]. Here we pose it as a sequential inverse problem. Namely, suppose that we can observe the solution of the equation, $v(\tau)$, at different times $\tau = t\Delta$ for $t = 1, ..., T$, and we aim to estimate the initial condition $x$ from the sequentially observed data. The observation noise is assumed to follow
Figure 1. The simulated data for $\sigma = 0.4$ (left) and $\sigma = 0.6$ (right). The lines show the simulated states in continuous time and the dots are the noisy observations.

Figure 2. The results for the case where noise variance is $0.4^2$. Left: the error for the simulation with 100 particles. Right: the error for simulation with 500 particles.

Figure 3. The results for the case where noise variance is $0.6^2$. Left: the error for the simulation with 100 particles. Right: the error for simulation with 500 particles.
a zero-mean Gaussian distribution with standard deviation $\sigma$. In this example, we take $T = 50$, and $\Delta = 0.3$, and moreover, we consider two different noise levels: $\sigma = 0.4$ and $\sigma = 0.6$. In the numerical experiments, we set the ground truth to be $x = 10^{-4}$ and the data is simulated from the model (6.1) in either case ($\sigma = 0.4$ and $\sigma = 0.6$), which are shown in Figs. 1. Moreover, in the experiments the prior distribution for $x$ is taken to be uniform: $U[-1,10]$. The main purpose of the example is to demonstrate the performance of the proposed method in the presence of strong non-Gaussianity.

We generate particles with EnKI and the proposed REnKI method, and in either method, we test two sample sizes $M = 100$ and $M = 500$, where the sample mean is used as an estimator of the sought parameters. In the proposed EnKI method, we use three different values of $\rho$: 50%, 70% and 90%. To evaluate the performance of the two methods, each test is repeated 100 times and we compute the average estimation error as follows:

$$\text{error} = \frac{1}{J} \sum_{j=1}^{J} | \hat{x}_j^t - x^* |, \quad (6.2)$$

where $x^*$ is the ground truth, $J$ is the total number of repeated trials (100 in this example), and $\hat{x}_j^t$ is the sample mean of the $j$-th trial.

In Fig. 2, we plot the average estimation error of both methods for the case where the noise variance is $0.4^2$, and in Fig. 3 we plot the same results but for the case where the noise variance is $0.6^2$. From the Fig. 2 one can see that, in the small noise case, the results indicate that REnKI yields evidently lower bias than EnKI. For the large noise case, as is shown by Fig. 3, the differences between the results of the two methods are quite substantial, especially for $\rho = 70\%$ and 90%. We note that the reason that we need to use these relative large values of $\rho$ is due to the strong non-Gaussianity in the problem; namely when strong non-Gaussianity is present, a rather large portion of samples drawn by EnKI deviate from the posterior and thus are removed by the REnKI algorithm.

6.2. Lorenz 96 model. Our second example is to estimate certain parameters in the Lorenz 96 model [16], which consists of a system of coupled differential equations with cyclic boundary conditions,

$$\frac{du_i}{d\tau} = (u_{i+1} - u_{i-2})u_{i-1} - u_i + F, \quad i = 1, ..., D, \quad (6.3a)$$

$$u_{-1} = u_{D-1}, \quad u_0 = u_D, \quad u_{D+1} = u_1, \quad (6.3b)$$

$$u_i(0) = x_i \quad i = 1, ..., D. \quad (6.3c)$$

In this example, we choose $D$ to be 40, and the constant $F = 8$. We then assume that we make observations of the system states $u_i$ sequentially at different times, and from these observations, we want to estimate the initial condition $x = (x_1, ..., x_D)$. Specifically, we assume that observations can only be made on odd dimensions; namely $\{u_{2j+1} | j = 0, ..., 19\}$ are the observed dimensions, and $\{u_{2j+2} | j = 0, ..., 19\}$ are the unobserved ones, and the observations are made sequentially at times $\tau_t = t\Delta$ for $t = 1, ..., T$. Moreover the observation noise is assumed to follow a zero-mean Gaussian distribution with standard deviation $\sigma$. In the numerical experiments we set $T = 50$ and $\sigma = 1$, and we consider two different values of $\Delta$: $\Delta = 0.05$ and $\Delta = 0.1$. Intuitively, the larger $\Delta$ is, the more nonlinear the model becomes and it then follows that the posterior differs more from Gaussian. Following [16] we set
the ground truth to be $x_i = 8$ for all $i \neq 20$ and $x_{20} = 8.01$, and the observation data are simulated by plugging the ground truth in the Lorenz system (6.3). In the inference the prior distribution is taken to be Gaussian distribution with mean 11, and standard deviation 1.

In either case ($\Delta = 0.05$ or $\Delta = 0.1$), we draw particles using EnKI and REnKI, and we use two different sample sizes $M = 2000$ and $M = 5000$. Moreover, in the REnKI method we use three different values of $\rho$: 10%, 20% and 30%. As before, the simulation is repeated 50 times and the average errors are calculated for each parameter. Finally we average the errors over the observed dimensions and and the unobserved ones separately. We plot the results for $\Delta = 0.05$ in Figs. 4, and those for $\Delta = 0.1$ in Figs. 5. In each set of figures, we show the results for $M = 2000$ and $M = 5000$. First one can see from the figures that, in the case $\Delta = 0.05$ where the posterior is closer to Gaussian, we can see that the results of both methods are quite similar while some setting $M = 5000$ and $\rho = 30\%$ even results in higher error than EnKI, suggesting that in this case the REnKI scheme does not improve the performance of standard EnKI. On the other hand, for $\Delta = 0.1$, we can see that the results of REnKI are evidently better than those of EnKI. It seems that the difference on the observed dimensions is smaller than that on the unobserved ones, possibly due to the fact that the posteriors of the unobserved dimensions are

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.png}
\caption{$\Delta = 0.05$. Top: the average estimation error of the simulation with 2000 particles. Bottom: the average estimation error of the simulation with 5000 particles. In both rows, the left figure shows the results of the observed dimensions and the right one shows the unobserved ones.}
\end{figure}
Figure 5.  $\Delta = 0.1$. Top: the average estimation error of the simulation with 2000 particles. Bottom: the average estimation error of the simulation with 5000 particles. In both rows, the left figure shows the results of the observed dimensions and the right one shows the unobserved ones.

typically subject to larger variance. For REnKI, the results of the three different values of $\rho$ are similar with those of $\rho = 30\%$ slightly better than the other two. We also observe that, the performance of EnKI is improved with the large sample size ($M = 5000$), but even in this case the performance of REnKI is still evidently better on the unobserved dimensions. In summary, we observe here that, when the posterior is well approximated by Gaussian, REnKI yields similar results as the standard EnKI, and when the posterior moves away from Gaussian, REnKI can evidently improve the performance of EnKI.

6.3. A nonlinear mathematical model. Our last example is the following nonlinear model,

$$G_t(x) = \sin(z^2 + (1 - z^2)(1 - \exp\left(-\frac{t}{\Delta}\right))), \quad t = 1, \ldots, T,$$

where $z = Ax$ and

$$A = \begin{bmatrix}
1 & -0.1 & 0 & \cdots & 0 \\
-0.1 & 1 & -0.1 & \cdots & 0 \\
0 & -0.1 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & -0.1 \\
0 & 0 & \cdots & 0 & 1
\end{bmatrix}.$$
Moreover we let  

\[ y_t = G_t(x) + \eta_t, \]

with \( \eta_t \sim \mathcal{N}(0, \sigma^2 I) \). In our numerical experiments, we let \( T = 10, \sigma = 0.05 \), and to compare the performance of the methods in high-dimensional setting, we set the dimensionality of \( x \) to be 100. It should be clear that the dimensionality of the data \( y_t \) is also 100. The ground truth for \( x \) used in this example is show in Fig. 6 (left). As before, here we use synthetic data simulated from the model 6.4, and for illustration purpose, in Fig. 6 (right) we plot both the noise-free and the noisy data at \( t = 3 \). The prior distribution is taken to be a Gaussian distribution with mean 2 and variance 3.

We perform both EnKI and REnKI methods to estimate the posterior distributions and with REnKI we use three different values of \( \rho \); 10\%, 30\% and 50\%. In both methods, we test two sample sizes; \( M = 2000 \) and \( M = 5000 \). As is in the previous examples, the simulations are repeated 50 times and the average estimation errors are calculated and plotted in Figs. 7. The left figure is the results for \( M = 2000 \) and the right one is those for \( M = 5000 \). First one can see from the figures that for both methods, 5000 particles generally produce better estimates than 2000 particles. When comparing the two methods, in the results of both sample sizes, the REnKI method yields evidently lower errors than EnKI. Moreover when comparing the performance of the three different values of \( \rho \), we can see that in this example \( \rho = 50\% \) has the best results, suggesting that due to the high nonlinearity a large portion of the particles need to be replaced.

6.4. Some remarks on the numerical results. In this section we want to provide some remarks regarding the numerical results. First, in all these examples, we have demonstrated that the proposed REnKI algorithm may have advantages over the standard EnKI in certain situations (likely those with strong non-Gaussianity), while in the others REnKI may produce results that are similar or even slightly worse than EnKI. In practice, whether REnKI should be used for a specific model can be determined in advance via simulated data. A related question is what is the appropriate value of the parameter \( \rho \) and similarly this can also be determined by conducting the computation with simulated data. To this end, the standard EnKI can be regarded as special case of REnKI where \( \rho \) is taken to be zero. Finally though the results show that the proposed REnKI method performs better when
Figure 7. The estimation error for the high dimensional nonlinear example. Left: the results for $M = 2000$; Right: the results for $M = 5000$.

for problems that are less Gaussian, we can not expect that it applies to strongly non-Gaussian problems. That is because we make a Gaussian approximation when computing the particle weights.

7. Conclusions. In summary, we consider in this work the Bayesian parameter estimation problems with sequential data and we propose a scheme to improve the performance of EnKI by removing particles that seriously deviate from the posterior. The method first compute an approximate weight for each particle and then removing the portion of particles with the smallest weights. As is demonstrated by examples, in certain class of problems the proposed REnKI method can improve the performance of standard EnKI without imposing any additional forward model evaluations that constitute the main computational cost in such problems. We expect that the proposed method can be useful in a range of problems, especially when the posteriors are not very close to Gaussian. Several issues related to the work deserve further studies. First this work does not provide convergence results of the algorithm or analysis of errors due to the approximation made in computing the particle weights. Such analysis is important for understanding the accuracy of REnKI as well as the applicability of the method and should be studied in the future. Moreover, since the proposed method only computes an approximate weight function, the particles obtained are still subject to approximation errors and do not follow the posterior exactly. To actually obtain samples from the posterior distribution, we may need to use the complete SMCS method, which, as has been discussed, may result in much higher computational cost. To this end, we also plan to investigate in the future the possibility of an efficient implementation of a complete SMCS method in the sequential parameter estimation problems.

REFERENCES

[1] J. D. Annan and J. C. Hargreaves, Efficient parameter estimation for a highly chaotic system, Tellus A: Dynamic Meteorology and Oceanography, 56 (2004), 520–526.
[2] J. D. Annan, D. J. Lunt, J. C. Hargreaves and P. J. Valdes, Parameter estimation in an atmospheric gcm using the ensemble kalman filter, In Nonlinear Processes in Geophysics, European Geosciences Union (EGU), 12 (2005), 363–371.
[3] A. Apte, M. Hairer, A. M. Stuart and J. Voss, Sampling the posterior: An approach to non-gaussian data assimilation, Physica D: Nonlinear Phenomena, 230 (2007), 50–64.
[4] A. Arnold, D. Calvetti and E. Somersalo, Parameter estimation for stiff deterministic dynamical systems via ensemble kalman filter, *Inverse Problems*, 30 (2014), 105008, 30pp.

[5] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, *IEEE Transactions on Signal Processing*, 50 (2002), 174–188.

[6] P. Del Moral, A. Doucet and A. Jasra, Sequential monte carlo samplers, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (2006), 411–436.

[7] A. Doucet and A. M. Johansen, A tutorial on particle filtering and smoothing: Fifteen years later, *Handbook of Nonlinear Filtering*, (2011), 656–704.

[8] O. G. Ernst, B. Sprungk and H.-J. Starkloff, Analysis of the ensemble and polynomial chaos kalman filters in bayesian inverse problems, *SIAM/ASA Journal on Uncertainty Quantification*, 3 (2015), 823–851.

[9] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*, Springer Science & Business Media, 2009.

[10] M. Frei and H. R. Künsch, Bridging the ensemble kalman and particle filters, *Biometrika*, 100 (2013), 781–800.

[11] A. Gelman, J. B. Carlin, H. S. Stern, D. Dunson, A. Vehtari and D. B. Rubin, *Bayesian Data Analysis (3rd edition)*. Texts in Statistical Science Series. CRC Press, Boca Raton, FL, 2014.

[12] W. R. Gilks, S. Richardson and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Interdisciplinary Statistics. Chapman & Hall, London, 1996.

[13] A. Golightly and D. J. Wilkinson, Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo, *Interface Focus*, 1 (2011), 807–820.

[14] M. A. Iglesias, A regularizing iterative ensemble kalman method for pde-constrained inverse problems, *Inverse Problems*, 32 (2016), 025002, 45pp.

[15] M. A Iglesias, K. J. H. Law and A. M. Stuart, Ensemble kalman methods for inverse problems, *Inverse Problems*, 29 (2013), 045001, 20pp.

[16] E. N. Lorenz and K. A. Emanuel, Optimal sites for supplementary weather observations: Simulation with a small model, *Journal of the Atmospheric Sciences*, 55 (1998), 399–414.

[17] N. Papadakis, É. Mémé, A. Cuzol and N. Gengembre, Data assimilation with the weighted ensemble kalman filter, *Tellus A: Dynamic Meteorology and Oceanography*, 62 (2010), 673–697.

[18] C. Schillings and A. M. Stuart, Analysis of the ensemble kalman filter for inverse problems, *SIAM Journal on Numerical Analysis*, 55 (2017), 1264–1290.

[19] X. Sun, L. Jin and M. Xiong, Extended kalman filter for estimation of parameters in nonlinear state-space models of biochemical networks, *PloS One*, 3 (2008), e3758.

Received January 2021; revised February 2021.

E-mail address: jiangqiw@sjtu.edu.cn
E-mail address: wenlinjie@sjtu.edu.cn
E-mail address: j.li.100@bham.ac.uk