Comput Geosci (2014) 18:579–595
DOI 10.1007/s10596-014-9402-6

ORIGINAL PAPER

An iterative version of the adaptive Gaussian mixture filter

Andreas S. Stordal · Rolf J. Lorentzen

Received: 7 December 2012 / Accepted: 8 January 2014 / Published online: 2 March 2014

Abstract The adaptive Gaussian mixture filter (AGM) was introduced as a robust filter technique for large-scale applications and an alternative to the well-known ensemble Kalman filter (EnKF). It consists of two analysis steps, one linear update and one weighting/resampling step. The bias of AGM is determined by two parameters, one adaptive weight parameter (forcing the weights to be more uniform to avoid filter collapse) and one predetermined bandwidth parameter which decides the size of the linear update. It has been shown that if the adaptive parameter approaches one and the bandwidth parameter decreases, as an increasing function of the sample size, the filter can achieve asymptotic optimality. For large-scale applications with a limited sample size, the filter solution may be far from optimal as the adaptive parameter gets close to zero depending on how well the samples from the prior distribution match the data. The bandwidth parameter must often be selected significantly different from zero in order to make large enough linear updates to match the data, at the expense of bias in the estimates. In the iterative AGM we introduce here, we take advantage of the fact that the history matching problem is usually estimation of parameters and initial conditions. If the prior distribution of initial conditions and parameters is close to the posterior distribution, it is possible to match the historical data with a small bandwidth parameter and an adaptive weight parameter that gets close to one. Hence, the bias of the filter solution is small. In order to obtain this scenario, we iteratively run the AGM throughout the data history with a very small bandwidth to create a new prior distribution from the updated samples after each iteration. After a few iterations, nearly all samples from the previous iteration match the data, and the above scenario is achieved. A simple toy problem shows that it is possible to reconstruct the true posterior distribution using the iterative version of the AGM. Then a 2D synthetic reservoir is revisited to demonstrate the potential of the new method on large-scale problems.

Keywords History matching · Data assimilation · Gaussian mixture filters · Ensemble Kalman filter · Iterative importance sampling

A. S. Stordal (✉) · R. J. Lorentzen
IRIS, Bergen, Norway
e-mail: Andreas.S.Stordal@iris.no

1 Introduction

History matching of petroleum reservoirs is an area of great interest both in the industrial and the scientific community. In this paper, we take the Bayesian approach to history matching and assume that the solution is given by the posterior probability density, uniquely defined by the prior probability density the model under consideration and the observations. Due to the complexity of the posterior density, even in small dimensions, analytical solutions are intractable, and one has to suffice with sample (ensemble)-based approximations via Monte Carlo methods. For large-scale models such as oceanography and petroleum reservoirs, the most popular method within the Monte Carlo framework is the EnKF first introduced by [1] and further developed by [2]. However, recent studies have shown that the class of Gaussian mixture filters [3–5] might be more suited for nonlinear models. Our aim of this paper is to introduce an iterative version of the AGM [6]. The AGM has already been applied to petroleum reservoirs for comparison with EnKF [7, 8]. Note that by construction, the AGM is always capable of performing at least as well as the EnKF.

The advantage of the AGM compared to EnKF is that it relaxes the implicit Gaussian assumption of the EnKF with a more flexible Gaussian mixture. In addition, it takes advantage of some of the nonlinear information contained in the importance weights as is standard in other sequential Monte Carlo methods [9]. Therefore, it is possible to show that the AGM has better asymptotic properties than the EnKF for nonlinear models [10]. There is, however, a strong connection between the AGM and the EnKF. They both have a linear integration step to update the samples from the observations sequentially in time.

We start Section 2 by giving a brief introduction to Bayesian data assimilation and discuss the similarities and differences of the AGM and the EnKF. In Section 3, we review importance sampling and iterative importance sampling, and in Section 4, we introduce the iterative version of the AGM before we demonstrate the methodology on a simple toy problem and a synthetic reservoir in Section 5. The paper ends with conclusions and discussions.

2 The Bayesian approach to data assimilation

We study a general hidden Markov system defined by an initial condition, a state equation and a measurement equation:

$$X_0 \sim p(x_0, \theta), \tag{1}$$

$$X_t = \mathcal{M}(X_{t-1}, \theta, \xi_t), \quad 1 \le t \le T \tag{2}$$

$$Y_t = \mathcal{H}(X_t) + \epsilon_t, \quad 1 \le t \le T \tag{3}$$

where ξ_t and ϵ_t are independent noise processes, θ is a vector of unknown parameters, and $p(x_0, \theta)$ is the prior probability density of x_0 and θ. For the remainder of the paper, we assume that ϵ_t is zero mean Gaussian with covariance matrix $\mathbb{R}$, which may be time-dependent, but we omit the time index here. Also, we omit the subindex of all probability density functions and denote p to be the density of its arguments as long as there is no chance of confusion, e.g., $p(x|y) = p_{X|Y}(x|y)$, etc.

Data assimilation, or filtering, in the Bayesian framework consists of describing the probability density of x_t and the vector of parameters, θ, conditioned on sequence of measurements, $y_{1:T} = (y_1, \ldots, y_T)$. This probability density is referred to as the posterior probability density function and can be found by applying the Bayes rule using the known prior density $p(x_0, \theta)$, the known Markov transitions $p(x_t|x_{t-1}, \theta)$, and the known likelihood functions $p(y_t|x_t, \theta)$ in the following way:

$$p(x_{0:T}, \theta|y_{1:T}) = C^{-1} p(x_0, \theta) \prod_{t=1}^{T} p(y_t|x_t, \theta) p(x_t|x_{t-1}, \theta), \tag{4}$$

where C is the normalizing constant. For nonlinear systems with continuous state space, the posterior density is impossible to obtain analytically due to the complexity of computing the normalizing constant. Therefore, the focus changes from computing the posterior density to drawing random samples from it. Unfortunately, due to the complexity and/or large dimensionality of geophysical models, Monte Carlo methods with the correct asymptotic behavior such as particle filters [9] or Markov chain Monte Carlo (MCMC) methods [11] can not be applied directly. In fact, it is proven [12] that as the dimension, q, of the state space and, d, of the measurement space goes to infinity with $q = d$, the largest weight in a particle filter converges to one in probability provided that $\frac{\log N}{d}$ goes to zero in the Gaussian prior-Gaussian likelihood case or $\frac{\log N}{d^{1/2}}$ goes to zero without the Gaussian restriction, but with some additional assumptions. Here, N is the number of particles, and, by Gaussian prior-Gaussian likelihood, we mean that all the densities $p(y_t|x_t, \theta)$ and $p(x_t|x_{t-1}, \theta)$ are Gaussian. For this reason, more robust Monte Carlo methods such as the EnKF has been developed for large-scale applications.

Contrary to filter theory, the state equation is usually deterministic in data assimilation problems; hence, it is sufficient to draw a random sample from the posterior density of the parameters and initial conditions. In order to obtain a random sample from the posterior density of the states, one can simply apply the model operator $\mathcal{M}$ with the random sample of parameters and initial conditions inserted. Thus, the Bayesian solution to data assimilation can be viewed as Bayesian parameter estimation. If we assume that the parameters and initial conditions follow a prior probability density $p(\theta)$, where we include x_0 in the θ vector for notational purposes, the system is described by the following:

$$\begin{aligned} &\theta \sim p(\theta) \\ &Y_t = \mathcal{H}(\theta) + \epsilon_t \quad 0 \le t \le T. \end{aligned} \tag{5}$$

Given a sequence of independent measurements, $y_{1:T}$, the posterior density in (4) is now reduced to

$$p(\theta|y_{1:T}) = C^{-1} p(\theta) p(y_{1:T}|\theta) = C^{-1} p(\theta) \prod_{t=1}^{T} p(y_t|\theta), \tag{6}$$

where C is the normalizing constant. Even if we assume that the measurement space is low dimensional so that particle filters do not suffer directly from the curse of dimensionality, the parameter space in geophysical models is extremely large, and the high likelihood area of the posterior density is extremely small compared to that of the prior. Hence, naive Monte Carlo approaches such as MCMC, particle filters, or even importance sampling would require a huge amount of

samples to obtain results that are not variance-prone. Unfortunately, it is time-consuming to apply the model operator $\mathcal{M}$ to each sample. This naturally puts an upper limit on the number of samples that can be used. As a consequence, one has to look for more robust Monte Carlo methods at the expense of precision. In the next section, we describe two methods that fall into this category.

3 EnKF and AGM

We give a brief description of EnKF and AGM and point out the similarities and differences. Let the vector X_t include parameters, initial conditions, dynamic variables, and simulated measurements and let $\mathbb{H}$ be the matrix that selects the simulated measurements from X_t. Assume that we at time $t-1$ have a random sample $\{\widehat{X}^i_{t-1}\}_{i=1}^N$ that is conditioned on $y_{1:t-1}$. The forecast (prediction) step is identical for the EnKF and the AGM:

$$X^i_t = \left[\hat{\theta}^i_{t-1} \quad \mathcal{M}\left(\widehat{X}^i_{t-1}\right) \quad \mathcal{H}\left(\mathcal{M}\left(\widehat{X}^i_{t-1}\right)\right)\right]^T,$$

and for the parameters-only case, we have

$$X^i_t = \left[\hat{\theta}^i_{t-1} \; \mathcal{H}\left(\hat{\theta}^i_{t-1}\right)\right]^T.$$

Let $\mathbb{P}_t$ be the sample covariance matrix of $\{X^i_t\}_{i=1}^N$ and let $\mathbb{H}$ be the binary matrix such that $\mathbb{H}X^i_t = \mathcal{H}\left(\hat{\theta}^i_{t-1}\right), i = 1,\ldots,N$. EnKF and AGM both have a linear update step to integrate y_t; however, the AGM update is a dampened version of the EnKF update:

$$\begin{aligned}
&\text{EnKF: } \widehat{X}^i_t = X^i_t + \mathbb{P}_t\mathbb{H}^T\left(\mathbb{H}\mathbb{P}_t\mathbb{H}^T+\mathbb{R}\right)^{-1}\left(y_t - \mathbb{H}X^i_t+\epsilon^i_t\right),\\
&\text{AGM: } \widehat{X}^i_t = X^i_t + h^2\mathbb{P}_t\mathbb{H}^T\left(h^2\mathbb{H}\mathbb{P}_t\mathbb{H}^T+\mathbb{R}\right)^{-1}\left(y_t - \mathbb{H}X^i_t+\epsilon^i_t\right),
\end{aligned} \tag{7}$$

where $h \in [0,1]$. It was shown in [13] that the linear update is a monotone function of h with no update for $h = 0$ and the same update as EnKF when $h = 1$. In addition, the AGM uses some of the nonlinear information contained in the (approximate) importance weights; however, to avoid a weight collapse, these are adaptively shrunk towards uniform weights, as is used in the EnKF, hence

$$\begin{aligned}
&\text{EnKF:} \quad W^i_t = N^{-1},\\
&\text{AGM:} \quad W^i_t = \alpha_t\bar{W}^i_t + (1-\alpha_t)N^{-1},
\end{aligned} \tag{8}$$

where

$$\begin{aligned}
\bar{W}^i_t &= \frac{\widehat{W}^i_t}{\sum_j \widehat{W}^j_t}\\
\widehat{W}^i_t &= \Phi\left(y_t - HX^i_t, h^2\mathbb{H}\mathbb{P}_t\mathbb{H}^T + \mathbb{R}\right)W^i_{t-1},
\end{aligned} \tag{9}$$

where $\Phi(x-\mu, P)$ denotes a multivariate Gaussian density with mean μ and covariance matrix P. The AGM update and weight equation can be derived from Bayes theorem when the prior is approximated by a Gaussian Mixture with bandwidth parameter h, and the likelihood is Gaussian with a linear measurement operator (usually obtained by augmenting the state vector). It can be viewed as a smoothed version of a particle filter where a sum of dirac measures is used to approximate the prior. The linear update equation is derived using the exact same calculations as in the Kalman filter. The weight reduction parameter α_t is an ad hoc parameter introduced in [6] to avoid a too skewed distribution of the weights eventually leading to a degeneracy.

As in importance sampling, the weights are changing the distribution of the sample, taking into account that we are not sampling from the target distribution. Since the weights are given by the ratio of the target and the proposal, they would be uniform if we could sample from the target. Since EnKF is derived from a linear Gaussian model assumption, the weights are uniform by definition.

The weight reduction in (9) depends on α_t which is computed from a bias variance trade-off argument [6] as the normalized estimated effective sample size

$$\alpha_t = N^{-1}\widehat{N}_{\text{eff}} = \left(N\sum_{i=1}^N\left(\bar{W}^i_t\right)^2\right)^{-1}. \tag{10}$$

Note that the correct importance weights in this setting is $\Phi(y_t - HX^i_t, \mathbb{R})W^i_{t-1}$ and in view of (7), (8), and (9), we see that the AGM is reduced to the standard SIR filter [14] when $h = 0$ and $a_t = 1$, and it is shown in [10] that the AGM has the correct asymptotic properties if h and α_t are decreasing and increasing fast enough as a function of the sample size. This is, however, not the case for the selection of α_t in (10) since α_t do not converge to 1 as N goes to infinity for this choice [6].

There is of course a distinct difference between the asymptotic properties in theory and the finite sample distribution in practice. Previous studies [7, 8] have shown that in addition to the sample size, the bandwidth h depends on the dimension of the state space, the nonlinearity of the problem, and the uncertainty in the prior distribution (in the parameter only case). Our main focus in this paper is on the parameter estimation, and for these cases, we can avoid testing the filter with different values of h in lack of a selection criteria by observing the following: The bias of AGM is a function of h^2 and $(1-\alpha_t)$, $1 \le t \le T$. If the initial sample is good in the sense that it predicts data that are close to the measurements, small linear corrections are needed (small h value), and the estimated effective ensemble size will be relatively large, resulting in α_t values that are significantly different from zero. In other words, the bias of the AGM

depends heavily on the initial sample. Our aim is to show how we can iterate with AGM to draw initial samples that resembles a sample from the posterior, thus reducing the bias. Although the iterative filter is more time-consuming, it has the advantage that instead of rerunning the filter with an increased h value if the results are not satisfactory, one may simply use this sample to define a new prior density and rerun the filter with a better prior—better in the sense that it already contains some information from the observations. Before we define the iterative AGM, we describe the iterative importance sampling algorithm in the next section.

4 Importance sampling and iterative importance sampling

For ease of notation, we denote in this section X to be the random vector of interest, which in the state space model would be $X = (\theta, x_0 \ldots, x_T)$. Further, we denote by $\pi(x)$ the target density, i.e., the density from which we want to draw random samples from. In the state space model, we have $\pi(x) = p(\theta, x_{0:T}|y_{1:T})$. Let $q(x)$ be another density which contains the support of $\pi(x)$ and which we know how to draw random samples from. Assume that $\pi(x)$ is a complicated density, known only up to a proportional constant, i.e., $\pi(x) = C^{-1}\tilde{\pi}(x)$ where $\tilde{\pi}(x)$ is known, and suppose we want to draw a sample $\{X^i\}_{i=1}^N$ from π to estimate an integral of the form

$$I = \int f(x)\pi(x)\,dx, = C^{-1}\int f(x)\tilde{\pi}(x)\,dx, \tag{11}$$

where $f(x)$ is an integrable function. The importance sampling estimator is of the form

$$\hat{I}_N = \sum_{i=1}^{N} \frac{f(X^i)w(X^i)}{\sum_{j=1}^{N} w(X^j)}, \tag{12}$$

where $\{X^i\}_{i=1}^N$ is a sample from a density q and

$$w(x) = \frac{\tilde{\pi}(x)}{q(x)}. \tag{13}$$

The general idea is that if the shape of q is close to the shape π, the estimate of I improves. This is naturally f-dependent, for instance, if $f(x) = 1_{(a,\infty]}(x)$ where a is in the tail of π, then the estimator improves if one sample from a distribution with much heavier tails than π. However, we assume that we want to estimate integrals over the entire domain of π, and we are therefore interested in finding an importance function q that is as close to π as possible. Iterative importance sampling, sometimes called adaptive importance sampling (see [15] and references therein), is an attempt to improve the estimate of I by iterating the importance sampling changing the importance function using current and past information about the target distribution.

Assume that we have performed importance sampling to obtain a weighted sample $\{W^i, X^i\}_{i=1}^N$ where each X^i is a random sample from g and $W^i \overset{\text{def}}{=} w(X^i) \propto \tilde{\pi}(X^i)/q(X^i)$. We then define a new density

$$q_1^N(x) = \sum_{i=1}^{N} K_h(x - X^i)W^i, \tag{14}$$

where K_h is a symmetric kernel with finite second-order moment. From the theory of importance sampling [11] and kernel density estimation [16], it is known that q_i gets closer to pi as the sample size increases if h decreases as a function of N. Hence, we have a density function which is closer to π than the original importance function q (for sufficiently large sample sizes). For a general function f, it is therefore natural to believe that the estimate in (12) improves if one samples from q_1^N instead of q. The idea of iterative (or adaptive) importance sampling [15] is to perform sequential importance sampling several times and, each time, update the importance function using the samples from the previous iteration to incorporate as much information about π in the importance function as possible. The algorithm for iterative importance sampling is described below.

Algorithm 1 Iterative importance sampling

if $j = 0$ **then**
  Set $q_0 = q$
  for $i = 1 \to N$ **do**
    Sample $\left\{X_0^i\right\}_{i=1}^N \sim q_0$
    Compute $W^i \propto \tilde{\pi}\left(X_0^i\right)/q_0\left(X_0^i\right)$
  end for
else
  Compute $q_j^N = \sum_{i=1}^N K_h\left(x - X_{j-1}^i\right) W_{j-1}^i$
  for $i = 1 \to N$ **do**
    Sample $\left\{X_j^i\right\}_{i=1}^N \sim q_j^N$
    Compute $W_j^i \propto \tilde{\pi}\left(X_j^i\right)/q_j^N\left(X_j^i\right)$
  end for
end if
$j \leftarrow j + 1$

The importance function changes rapidly, and typically, only a few iterations are needed, usually two or three iterations are sufficient (see [15] where the convergence is also studied as a direct result of the theory in [17]). Next, we move our focus towards the Bayesian inversion problem.

4.1 Iterative importance sampling in the Bayesian framework

Assume that we have a model described by (5) and that we want to draw a sample from the posterior (and target) distribution $\pi(\theta) \stackrel{\text{def}}{=} p(\theta|y_{1:T})$ given by (6). This scenario can be directly implemented in the algorithm described in the previous section, e.g., let $q_0(\theta) = p(\theta)$ and then iterate. However, if T is large, the weights are going to suffer from the curse of dimensionality [12], that is, all but one sample have significant weight. This can, however, be avoided. If we assume that the dimension of each y_t is relatively small, then we may incorporate the iterative importance sampling in the sequential importance resampling framework [18], also known as sequential Monte Carlo or particle filters [9]. The iterative importance sampling algorithm is simply extended to iterative sequential importance sampling. For each j, we compute $W_t^i \propto W_{t-1}^i p(y_t|\theta^i)$, $i = 1, \ldots, N$, $t = 1, \ldots, T$ and perform resampling if the distribution of the weights are too skewed at a given time t. This is simply a consequence of the fact that we can decompose the weights as follows:

$$W_T = \frac{p(\theta|y_{1:T})}{q(\theta)} \propto \frac{p(\theta)}{q(\theta)} \prod_{t=1}^{T} p(y_t|\theta) = W_{T-1} p(y_T|\theta).$$

In other words, we can apply the iterative importance sampling algorithm sequentially in time, including resampling if T is large. However, if the dimension of the parameter space and/or the measurement space is large, there is no way around the curse of dimensionality except increasing the sample size vastly. For large-scale systems, the computational cost of running the numerical model usually puts an upper bound on the sample size, hence approximative Monte Carlo schemes, like AGM, are applied instead. Since AGM is an approximation of sequential importance resampling filters, we are motivated to introduce the iterative AGM.

5 Iterative adaptive Gaussian mixture filter in high-dimensional systems

The goal of this section is to describe an improved approximation of the posterior distribution of parameters and initial conditions with a limited sample size in a large dimensional state space. We assume that the system under consideration is highly influenced by an initial condition and/or a parameter vector which we denote by θ. A typical example is reservoir modeling, where the dynamics (and hence measurements) are governed by a set of partial differential equations where the solution is (mainly) determined by rock properties (parameters) and initial conditions.

Assume for a moment that the dimension of the measurement space is sufficiently small, so that a weight collapse due to the dimension of measurement space is not an issue. The reason why α_t from (10) in general will be small and h must be selected relatively large (increasing h is also shrinking the weights towards uniform weights) is because the prior sample space is huge compared to the solution space and the sample size. Almost all samples from the prior will give a statistical mismatch to the data. This is exactly why an MCMC or a standard particle filter cannot be applied, the number of samples needed is simply too large compared to the computational capacity we have at hand. Our approach to deal with this problem is to perform an iterative adaptive Gaussian mixture filter by iteratively using the filter solution of θ at the previous iteration as prior distribution in the next iteration as in the iterative importance sampling algorithm described in the previous section.

The new approach, IAGM, runs AGM iteratively with a small bandwidth h, since smaller linear updates significantly reduce the bias in nonlinear models (from a purely theoretical point of view). Then, to avoid overfitting the data when used several times, we compensate for the updated sample by using weights. This is also confirmed by Figs. 1, 2, 3, and 4. Of course in high-dimensional complex models, one has to be careful of overfitting since the weights are shrunk towards uniform weights and due to the fact that after several Kalman updates, the Gaussian mixture computed from the ensemble may no longer contain the support of the posterior.

What we really do is to construct a new improved proposal distribution for the parameters and initial conditions. From this point of view, it should be clear that no overfitting of the data occurs since we are allowed to condition the proposal density to the data when applying importance sampling. For instance, if the posterior, which is conditioned on all the data, is used as the proposal density, all the weights are equal. Of course, one could use the results from EnKF in this procedure, but as mentioned above, the bias introduced could be severe, and we claim that it is better to slowly change the proposal distribution using small linear updates. For the reservoir example in the next section, the IAGM starts performing resampling after a few iterations which is a consequence of the forecast samples matching the data statistically according to the likelihood function. This shows that the convergence is quite fast for this model. Of course, for larger realistic cases, increasing h in the first few iterations could speed up the algorithm.

At the start of a new iteration, j, we sample random vectors $\{\theta_j^i\}_{i=1}^N$ from the Gaussian mixture defined by the samples at the previous iteration.

$$\theta_j^i \sim q_j^N(\theta) \stackrel{\text{def}}{=} \sum_{i=1}^{N} W_{j-1}^i \Phi\left(\theta - \theta_{j-1}^i, h^2 \mathbb{P}_j\right) \tag{15}$$

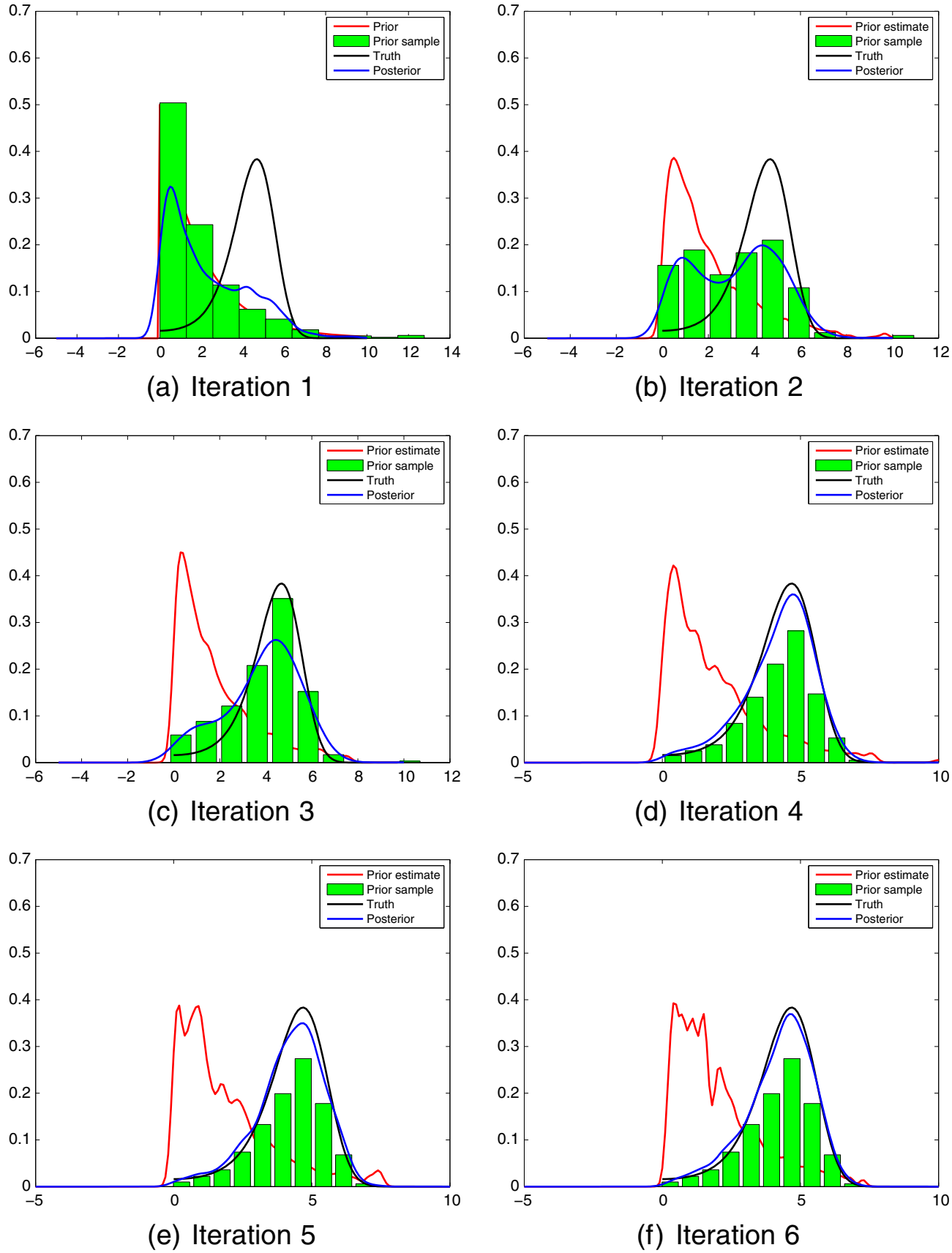


Fig. 1 Results of the IAGM with $h = 0.1$ and $N = 1{,}000$

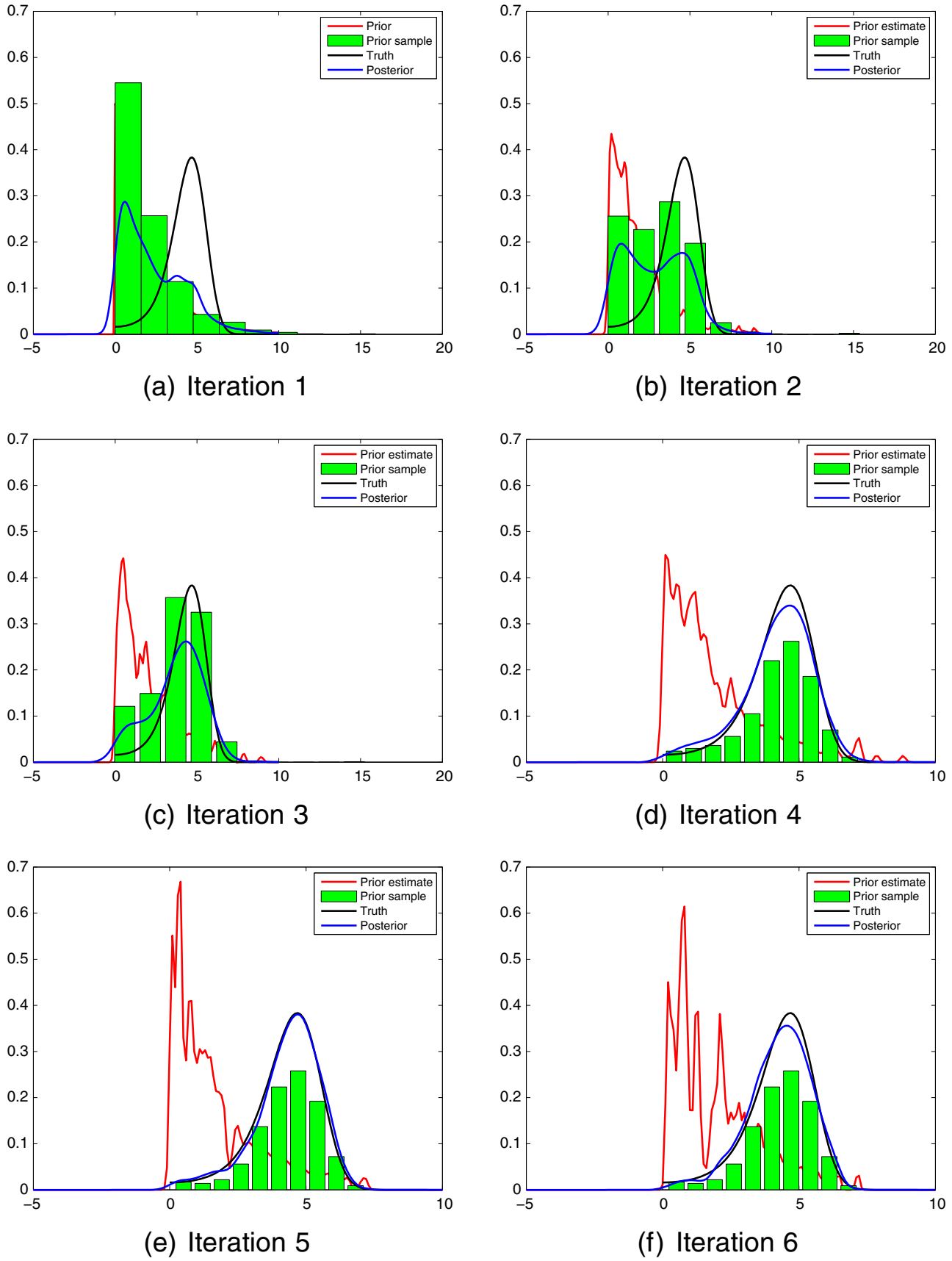


Fig. 2 Results of the IAGM with $h = 0.05$ and $N = 1{,}000$

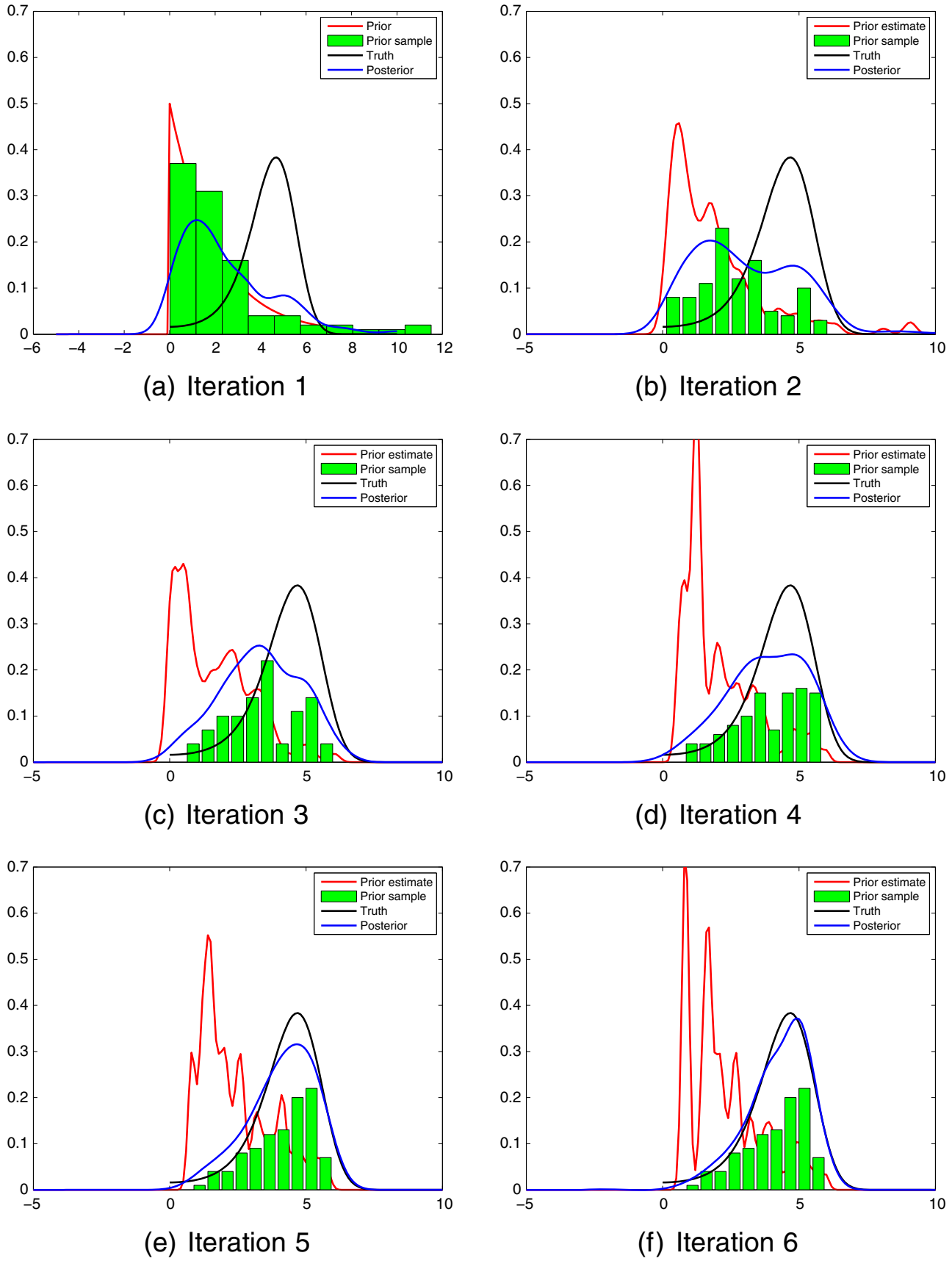


Fig. 3 Results of the IAGM with $h = 0.1$ and $N = 100$

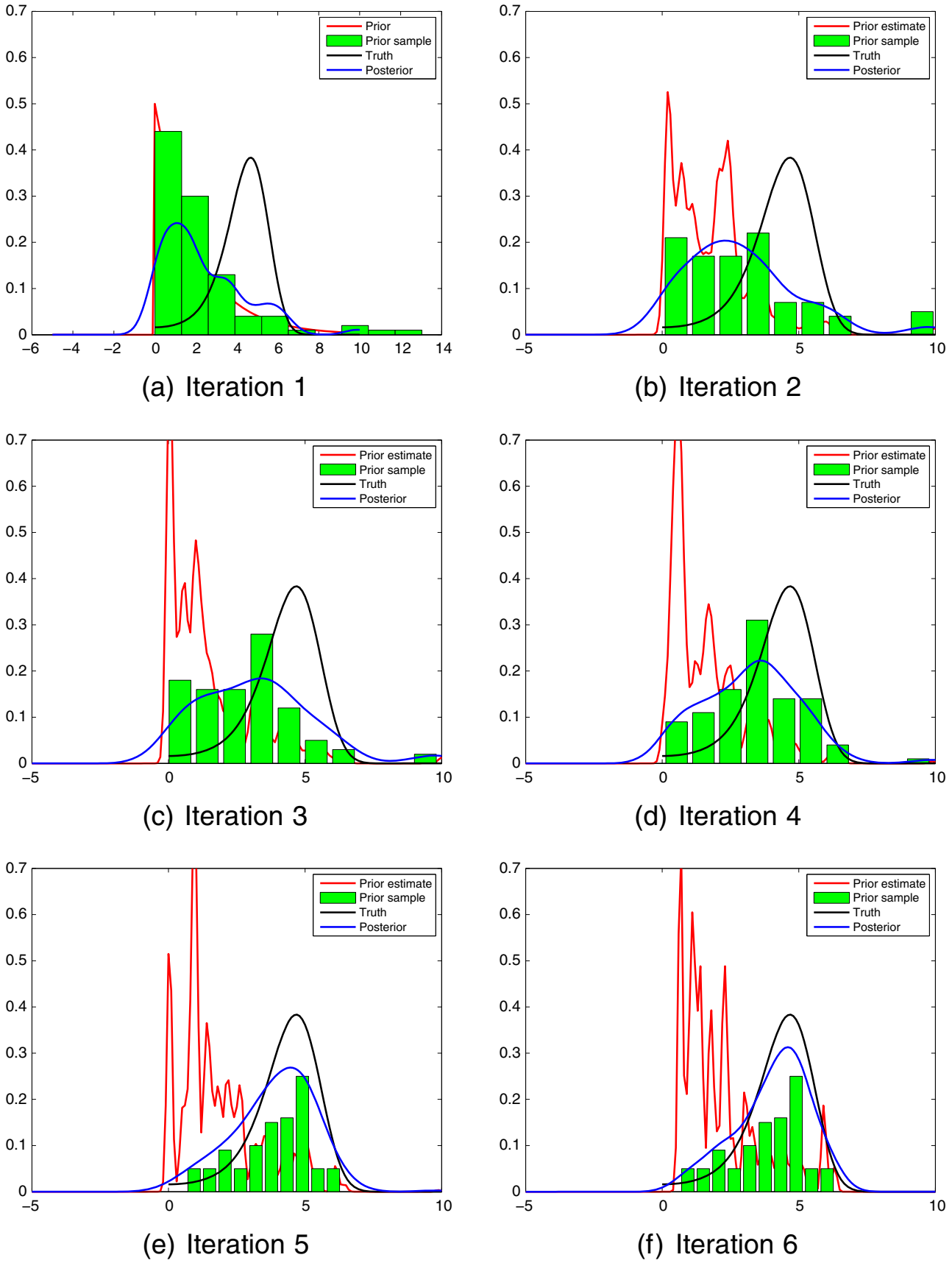


Fig. 4 Results of the IAGM with $h = 0.05$ and $N = 100$

and correct with weights

$$W_j^i = \frac{p\left(\theta_j^i\right)}{q_j^N\left(\theta_j^i\right)}. \tag{16}$$

Note that if the dimension of state space is large, computing these initial weights is time-consuming, and if the sample size is not sufficiently large, $\alpha_t \approx 1$. If this is the case, the initial weights are set to N^{-1} since the information content from the weights is negligible in the AGM setting.

Remark 1 To avoid resampling in the ensemble subspace, one could replace $\mathbb{P}_j$ with $\mathbb{C}_p$, the prior covariance matrix of θ in (15) if this is available in square root form.

Next, we run AGM sequentially through all time steps and update the weights accordingly. If we let $\alpha = 1$ and $h = 0$ in the final iteration J, we may compute the weights sequentially for the filter as follows (with possible resampling and resetting of the weights):

$$\frac{p\left(\theta_J^i\right)}{q_J^N\left(\theta_J^i\right)} \prod_{t=1}^{T} f\left(y_t|\theta_J^i\right) \tag{17}$$

and our algorithm will then have optimal asymptotic behavior, i.e., it reproduces the posterior distribution if we let the number of samples go to infinity. However, we do not believe that it is possible to go all the way to a particle filter with a small sample size and a large dimensional state space; hence, we suggest running a few iterations where h is relatively small (typically 0.1 or 0.05). However, we will have $h > 0$ at the final iteration as well. Some care has to be taken as we cannot claim that the new initial sample is a sample from the prior unless the tails of the density q_j^N is heavier than the original prior density for all j. We note that the idea of this method is similar to an EnKF/MCMC approach [19] where the EnKF is used to construct an improved proposal distribution for an MCMC run. However, in that method, the joint distribution from EnKF, and not the numerical model, was used to sample simulated measurements. Hence, the MCMC samples lies in the same subspace as the samples from EnKF.

6 Examples

We start our simulation studies with a simple toy problem where we can compute the true posterior and run the IAGM several times for comparison. Then, a 2D synthetic reservoir is studied where we run IAGM with two different values for h.

Experiments in non-Gaussian toy problem

To demonstrate the accuracy of the IAGM algorithm, we construct a simple toy model with a skewed prior and posterior distribution. Our simple model consists of one parameter and one measurement taken as a nonlinear function of the parameter

$$\begin{aligned} y &= 0.2\theta^2 + 0.3\theta + \epsilon \\ \theta &\sim \exp(\lambda) \\ \epsilon &\sim \mathcal{N}(\mu, \sigma). \end{aligned} \tag{18}$$

In the simulations, we have chosen $\lambda = 2$, $\mu = 0$, and $\sigma = 2$.

We sample two random variables, one from the prior exponential distribution and one from the Gaussian distribution of the measurement error, and we use these values in the example so that $\theta = 4.608$, and the measurement $y = 6.7023$. For this particular problem, we can compute the true posterior distribution using Bayes' theorem with numerical integration of the normalizing constant. We run the IAGM with 1,000 and 100 ensemble members and $h = 0.1$ and 0.05. With 1,000 ensemble members, we reproduce the true posterior with good precision for both values of h, while with 100 members, it seems like $h = 0.05$ converges too slowly as seen in Figs. 1, 2, 3, and 4. Note that the green histogram shows the prior sample at each iteration that is the green bars is the histogram of the sample drawn from the Gaussian mixture obtained at the previous iteration. Initially, it is a sample from the exponential prior. We see that for each iteration, the prior sample is closer to the posterior sample. The red density is an estimate of the true prior density using the prior sample and new weights computed from (16). That is a Gaussian mixture using the same samples as the green histogram but with weights computed as the ratio of the true prior and the Gaussian mixture from the previous iteration. Initially, this ratio is one since we sample from the prior. This is the weight correction step that avoids overfitting the data. Thus, even if we construct samples that are in regions of higher likelihood of the posterior than a sample from the prior, our weighted sample is a sample from the prior. The sample is then updated and weighted using (7) and (8), and the blue line is a density estimate computed from this weighted sample. The black line is the true posterior.

Obviously, we do not expect to have anywhere near the same precision for large-scale models with only 100

Table 1 Mean squared data mismatch values for the different filters. Rerun from time zero

Iteration	1	2	3	4	5
Initial	1.1×10^3	NA	NA	NA	NA
EnKF	3.64	NA	NA	NA	NA
IAGM h01	5.64	2.03	1.49	1.43	NA
IAGM h005	7.06	3.50	2.50	2.37	2.09

ensemble members, but it is satisfactory to see that with iterations, the asymptotic bias of the AGM is of the order h^2, and h is a parameter that we can choose.

6.1 Revisiting a 2D synthetic reservoir model

Our main test case is a synthetic two-phase 2D synthetic reservoir [5]. The reservoir has 900 grid cells, four oil producers, and one water injector. Each grid cell is $100 \times 200 \times 50$ m. The unknown parameters are the porosity (ϕ) and the log permeability ($\ln k$), and we assume a Gaussian prior with an isotropic spherical variogram and a correlation length of ten grid blocks. The mean and standard deviation for the porosity is 0.2 and 0.05, and 2 and $\sqrt{2}$ for the log permeability. The correlation between ϕ and $\ln k$ is set to 0.8. The reference porosity and log permeability fields are generated from this prior and run for 40 time steps where the wells are controlled by bottomhole pressure. At each time step, nine observations are collected in terms of oil and water production at each producer and water injection at the injector. The measurements are then perturbed with zero mean Gaussian noise and a standard deviation of 5 % of the true value.

In [5], the AGM was compared with EnKF for different values of h. In terms of history matching the data, the optimal value for h was case dependent; however, it seemed like smaller h values were uniformly better in terms of keeping the prior geology and hence better preserving the prior distribution. In the Bayesian framework, we naturally want to achieve both a low data mismatch and preserve the prior, and therefore, we iterate with the AGM with h values of 0.05 and 0.1.

To evaluate the results here, we compute the squared normalized data mismatch, D, after a rerun of the final permeability and porosity estimates which we denote by θ. Let y be the observations; X, the simulated measurements from the final estimates of perm and poro; $\mathbb{R}$, the covariance matrix of the observations; and N_d, the total number of measurements. Then,

$$D = \sqrt{N_d^{-1} \sum_{i=1}^{N} W^i (y - X^i)^T \mathbb{R}^{-1} (y - X^i)}. \tag{19}$$

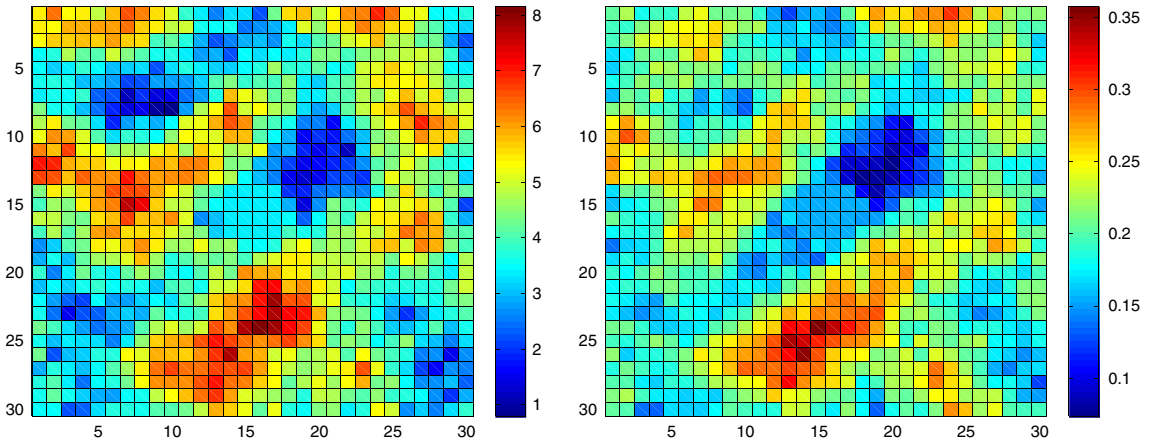
The results are shown in Table 1. We also compute the normalized objective function for each ensemble:

$$O_N(\theta^i) = \frac{2O(\theta^i)}{N_d}, \tag{20}$$

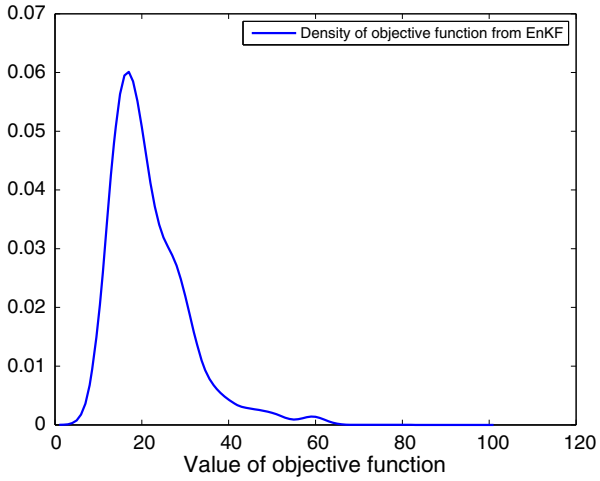
where N_d is the number of measurements and

$$O(\theta^i) = \left(\frac{1}{2} (\theta^i - \mu_p)^T C_p^{-1} (\theta^i - \mu_p) + \frac{1}{2} (y - x^i)^T \mathbb{R}^{-1} (y - x^i) \right). \tag{21}$$

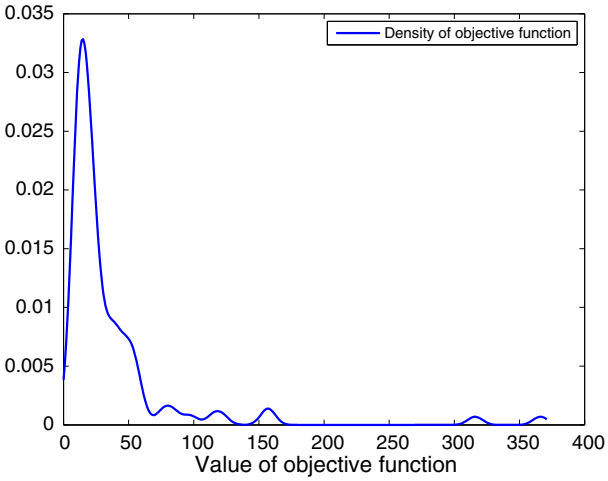
The distribution of the objective function for each filter is plotted in Figs. 5 and 6, and some statistics are summarized



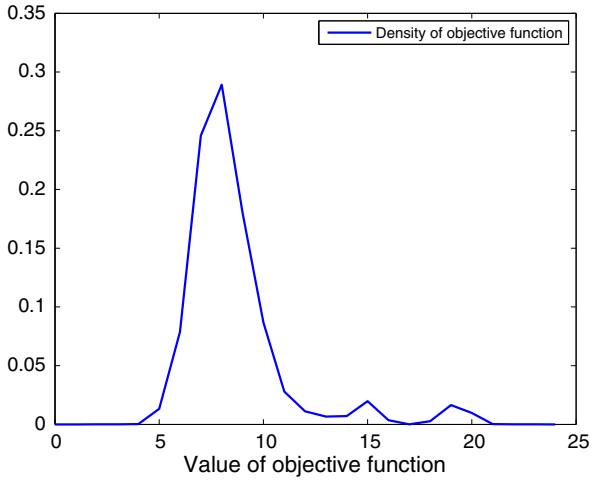
(a) logPerm reference (b) porosity reference



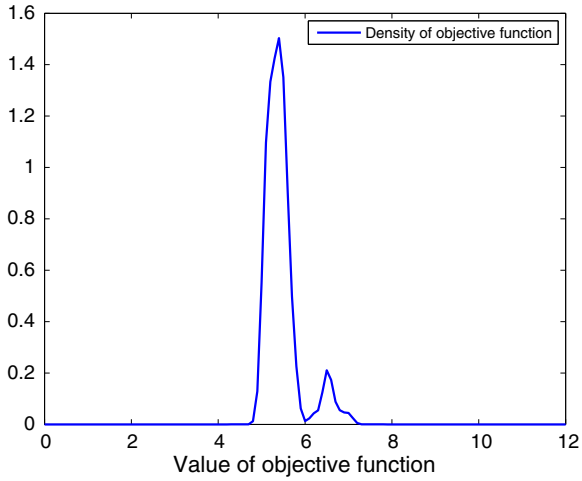
(a) EnKF



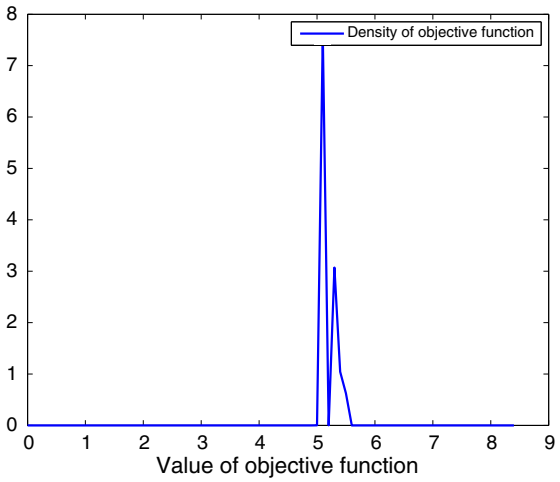
(b) AGM, h=0.1, iteration 1



(c) AGM, h=0.1, iteration 2

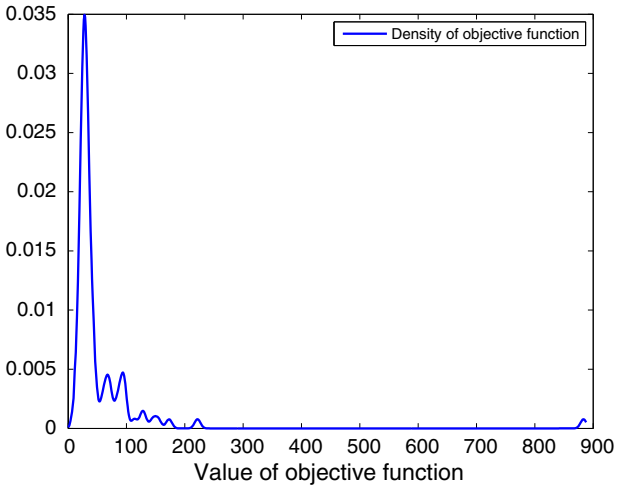


(d) AGM, h=0.1, iteration 3

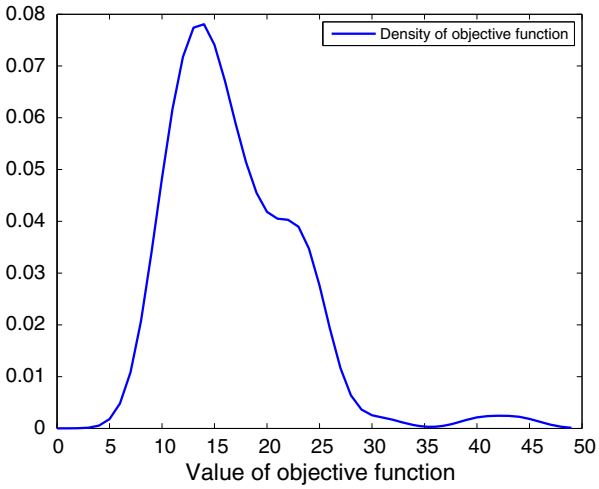


(e) AGM, h=0.1, iteration 4

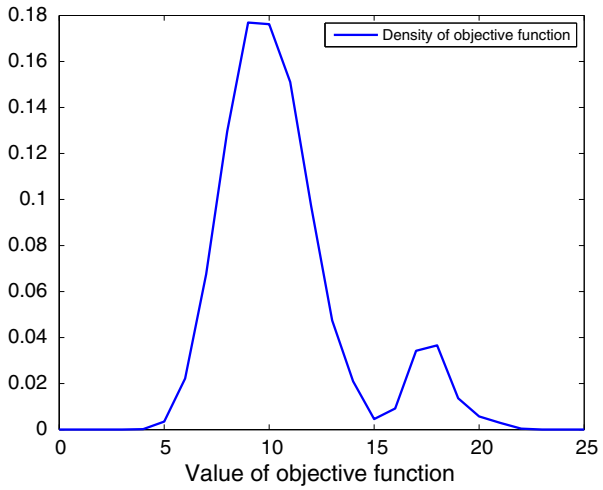
Fig. 5 Density of objective function



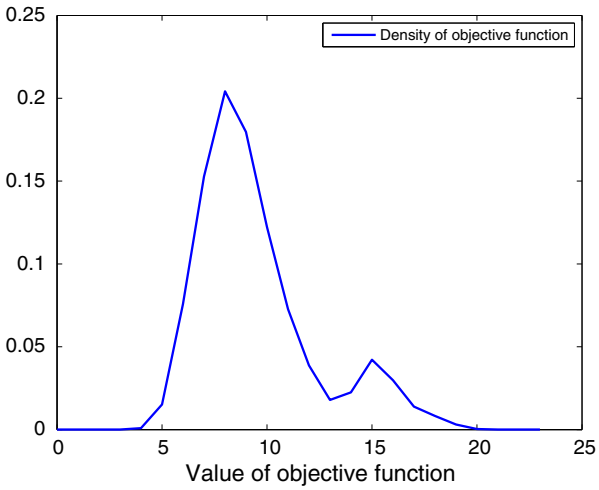
(a) AGM, h=0.05, iteration 1



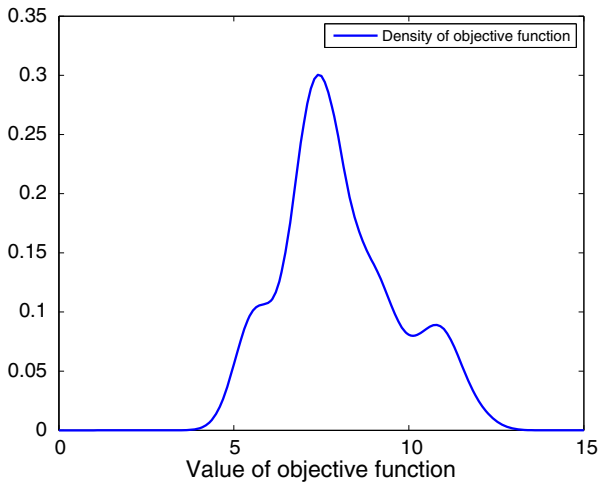
(b) AGM, h=0.05, iteration 2



(c) AGM, h=0.05, iteration 3



(d) AGM, h=0.05, iteration 4



(e) AGM, h=0.05, iteration 5

Fig. 6 Density of objective function

in Table 2. We are not going to focus too much on the results of the EnKF run since the IAGM is more time-consuming and since we know that the iteration would improve the results of the EnKF. It is mainly included as a reference to show that AGM and IAGM does not perform worse than the ordinary EnKF.

Clearly, we see that the IAGM with $h = 0.1$ has iterated too many times already after three iterations. There is almost no variability left in the objective function as seen in Fig. 5 and Table 2. We do, however, see that the minimum, the mean, and the median of the objective function is greatly reduced after a few iterations with both $h = 0.1$ and $h = 0.05$. We do not know what the true posterior density of the objective function look like. However, we argue that it is reasonable to believe that this density should be skewed and possibly similar to a chi-square density. Most of the objective function values should be small; however, if a large sample from the posterior density is provided, some values should also be large. Remember that for a large sample from the posterior, some values should be in the tail where the posterior is very small. This, of course, corresponds to a large objective function value. From Figs. 5 and 6, we argue that the EnKF has too many large values and that iterations 3 and 4 for AGM with $h = 0.1$ has to small values (not capturing the whole distribution). We should, however, be careful with the conclusion here since we do not know the true density as mentioned above.

As a curiosity, we compute the correlation between the final estimated log permeability field and porosity field for the three methods. For EnKF, it was 0.7162; for IAGM with $h = 0.1$, it was 0.8671; and for IAGM with $h = 0.05$, it was 0.8755. However, since we do not know the true posterior, it is impossible to draw any conclusion; however, the empirical correlation between log permeability and porosity in the reference field is 0.8479.

Table 2 Statistics from the objective function for each filter

Iteration	1	2	3	4	5
EnKF-Mean	20.74	NA	NA	NA	NA
Median	18.43	NA	NA	NA	NA
Max	58.32	NA	NA	NA	NA
Min	9.97	NA	NA	NA	NA
IAGM h01-Mean	36.88	8.62	5.46	5.19	NA
Median	20.64	8.05	5.38	5.12	NA
Max	366.08	19.48	7.00	5.49	NA
Min	7.55	5.55	5.05	5.06	NA
IAGM h005-Mean	54.68	16.87	10.65	9.44	8.01
Median	31.30	15.37	9.90	8.74	7.73
Max	883.38	44.10	20.20	18.15	12.01
Min	12.53	8.73	6.25	6.07	5.24

Table 3 Mean of the posterior standard deviations of the log permeability fields

Iteration	1	2	3	4	5
Initial	1.41	NA	NA	NA	NA
EnKF	0.52	NA	NA	NA	NA
IAGM h01	1.06	0.62	0.13	0.058	NA
IAGM h005	1.15	1.03	0.93	0.82	0.73

In addition, we compute the mean of the posterior standard deviations for the log permeability fields and the mean of the root mean square error (RMSE) for the porosity fields. The results are shown in Tables 3 and 4.

The standard deviations of IAGM with $h = 0.1$ confirm our previous suspicion about the ensemble collapse as there is very little variability within the ensemble (Table 3). Although it collapses to a model that match the data and have a low value of the objective function, this is unwanted since we can not quantify the uncertainty. This is a classical bias variance trade-off problem, and we rather select a method that maintains variability at the expense of precision. However, both after two iterations with $h = 0.1$ and five iterations with $h = 0.05$, we maintain a higher posterior uncertainty than one run with EnKF and at the same time reduce the average data mismatch and the minimum value of the objective function, which means that these methods have produced at least one sample with higher posterior likelihood than EnKF. It is difficult to get very low RMSE for the porosity (Table 4); however, we do observe that the RMSE is reduced for each iteration which means that iterating leads to improved results in terms of RMSE. We also plot the final mean of the log permeability fields in Fig. 7, and oil and water production history for well P2 with ensemble rerun from time zero with IAGM, $h = 0.05$, and EnKF after the final iteration in Fig. 8. Using visualization for comparison is not advised, but it is meaningful to at least show some of the data and the estimated fields to help understand the numbers better. Note that the ensembles from IAGM are weighted, so the statistical spread of the forecasts is not equivalent to the spread we see in the plot. The final mean of the porosity fields has the same trend as the porosity fields but are omitted here.

Table 4 Mean of the RMSE for the porosity fields

Iteration	1	2	3	4	5
Initial	0.067	NA	NA	NA	NA
EnKF	0.066	NA	NA	NA	NA
IAGM h01	0.060	0.056	0.050	0.049	NA
IAGM h005	0.061	0.058	0.056	0.053	0.052

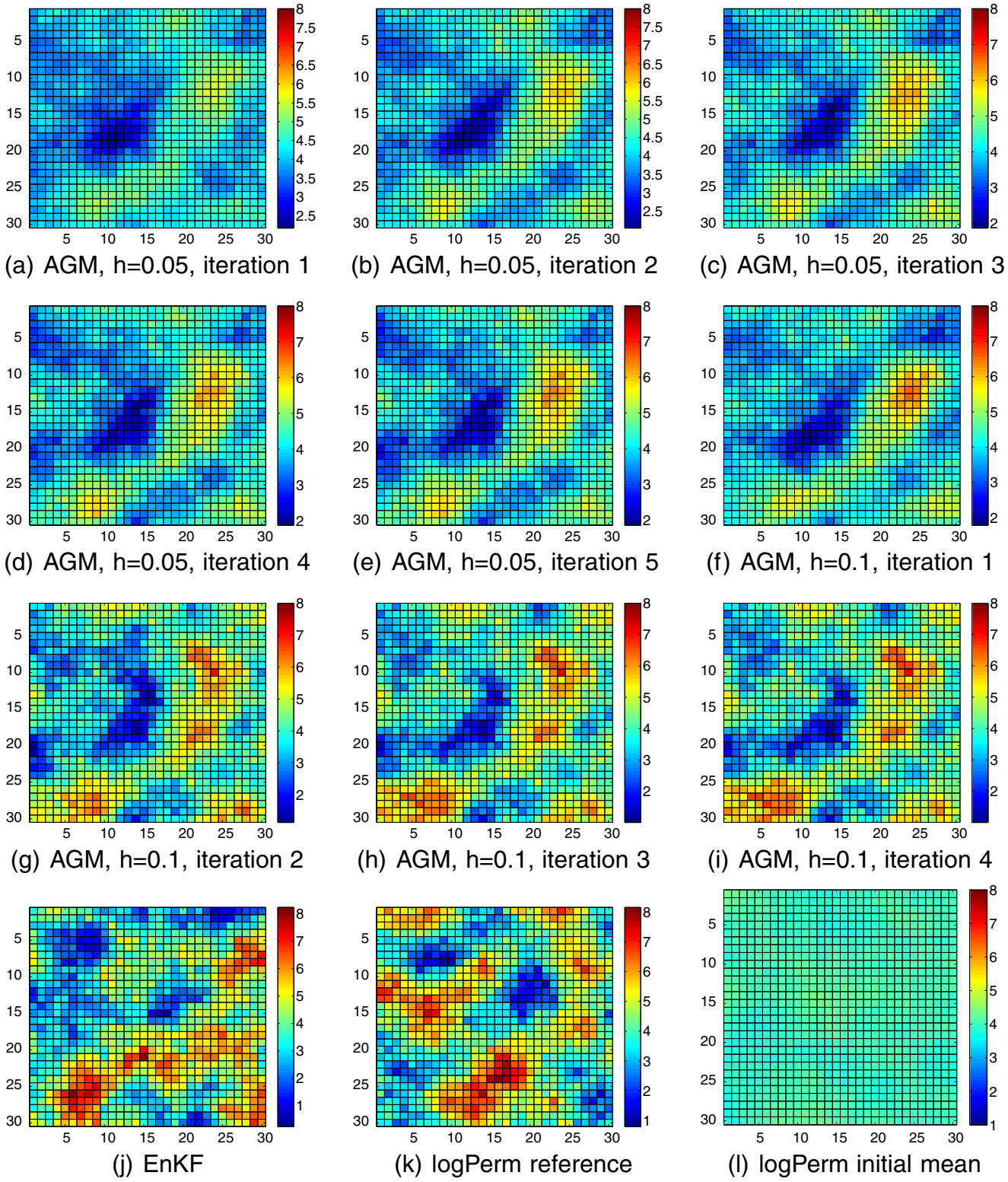


Fig. 7 Final mean of log permeability fields

7 Summary and conclusions

In this paper, we have extended the AGM for improved data assimilation in large-scale models. By studying on how importance sampling can be improved by iterations, we have taken this idea and defined iterative versions of the AGM. The method is promising, both theoretically and from simulation studies; however, this is more time-consuming

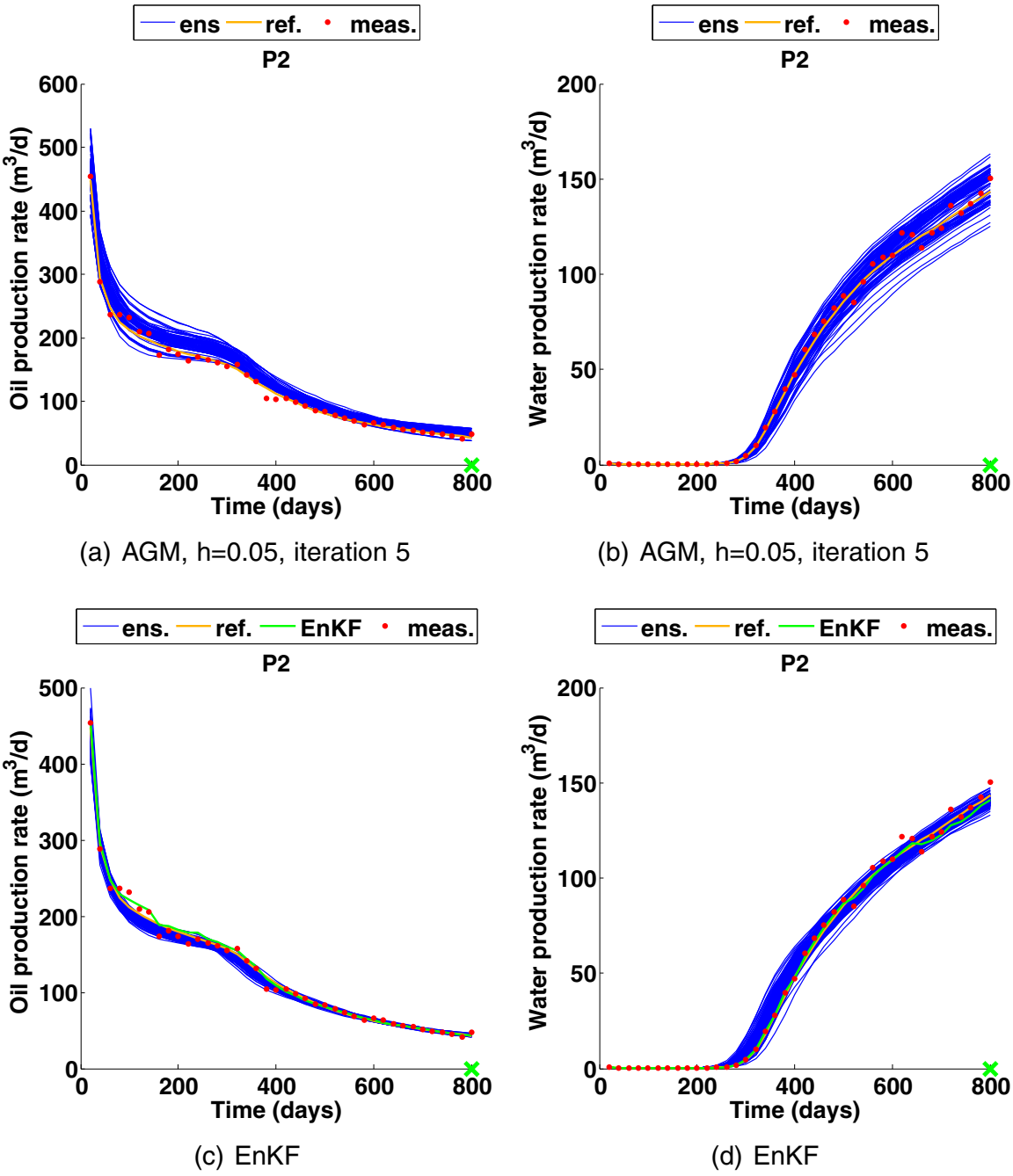


Fig. 8 Reruns of final ensemble for IAGM, h=0.05, iteration 5 and EnKF for water and oil production rates in well P2

than, e.g., EnKF since it requires J runs of the AGM algorithm were J is the number of iterations. It is most useful when we estimate parameters or initial conditions and not very suitable for general filtering in stochastic models where the system forgets its initial state although an extension to such models is easy to create using, e.g., proxy models. In this scenario, it is possible to define an iterative version where the iterations are performed at each time step.

The iterative AGM showed good results in the simulation studies, especially for $h = 0.1$ where two iterations were enough to get a good history match in the 2D model. This is promising from a theoretical point of view since the bias of the AGM decrease with h. Also, we see from the results after one iteration that for the original AGM, both 0.05 and 0.1 were too low values for h to get a good history match for the 2D model.

The method needs more investigation, and especially, comparison with other iterative methods is needed before any conclusion can be drawn. Also, there is no reason why we should fix h at each iteration, perhaps starting with

a larger h increase the convergence rate. To improve the computation time, it is also possible to define the iterative adaptive Gaussian smoother, but then some care has to be taken about the amount of measurements in order to avoid the curse of dimensionality. The authors have already started investigation on these areas.

Acknowledgments The authors acknowledge financial support from the Research Council of Norway (PETROMAKS) and industrial sponsors through the project "Realistic Geology & Integrated Workflow".

References

1. Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. J. Geophys. Res. **99**(C5), 10143–10162 (1994)
2. Burgers, G., van Leeuwen, P., Evensen, G.: Analysis scheme in the ensemble Kalman filter. Mon. Weather Rev. **126**(6), 1719–1724 (1998)
3. Bengtsson, T., Snyder, C., Nychka, D.: Toward a nonlinear ensemble filter for high-dimensional systems. J. Geophys. Res. **108**, 35–45 (2003)
4. Hoteit, I., Pham, D.-T., Triantafyllou, G., Korres, G.: A new approximative solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. Mon. Weather Rev. **136**, 317–334 (2008)
5. Stordal, A.S., Valestrand, R., Karlsen, H.A., Skaug, H.J., Nævdal, G.: Comparing the adaptive Gaussian mixture filter with the ensemble Kalman filter on synthetic reservoir models. Comput. Geosci. **16**(2), 467–482 (2011)
6. Stordal, A.S., Karlsen, H.A., Nævdal, G., Skaug, H.J., Vallès, B.: Bridging the ensemble Kalman filter and particle filters. Comput. Geosci. **15**(2), 293–305 (2011)
7. Valestrand, R., Nævdal, G., Shafieirad, A., Stordal, A.S., Dovera, L.: Refined adaptive Gaussian mixture filter—application on a real field case. In: EAGE Annual Conference & Exhibition Incorporating SPE Europec, Copenhagen, Denmark (2012)
8. Valestrand, R., Nævdal, G., Stordal, A.S.: Evaluation of EnKF and variants on a field model. Oil & Gas Science and Technology-Revue d'IFP Energies nouvelles (2012)
9. Doucet, A., de Freitas, N., Gordon, N. (eds.): Sequential Monte-Carlo methods in practice. Springer, New York (2001)
10. Stordal, A.S.: Sequential Data Assimilation in High Dimensional Nonlinear Systems. PhD thesis, University of Bergen (2011)
11. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer, Heidelberg (2004)
12. Bengtsson, T., Bickel, P., Li, B.: Curse-of-dimensionality revisited: collapse of particle filter in very large scale systems. Probab. Stat. **2**, 316–334 (2008)
13. Stordal, A.S., Karlsen, H.A., Nævdal, G., Oliver, D.S., Skaug, H.J.: Filtering with state space localized Kalman gain. Phys. D **241**(13), 1123–1135 (2012)
14. Gordon, N.: Bayesian Methods for Tracking. PhD thesis, University of London (1993)
15. Givens, G.H., Raftery, A.E.: Local adaptive importance sampling for multivariate densities with strong nonlinear relationships. J. Am. Stat. Assoc. **91**(433), 132–141 (1996)
16. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)
17. Geweke, J.: Bayesian inference in econometric models using monte carlo integration. Econometrica **24**, 1317–1399 (1989)
18. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proc.-F **140**(2), 107–113 (1993)
19. Emerick, A.A., Reynolds, A.C.: Combining the ensemble Kalman filter with Markov chain Monte Carlo for improved history matching and uncertainty characterization. SPE J (2012). doi:10.2118/141336-PA