Nesterov acceleration for ensemble Kalman inversion and variants

Article

It is advisable to refer to the publisher's version if you intend to cite from the work. See Guidance on citing.

To link to this article DOI: http://dx.doi.org/10.1016/j.jcp.2025.114063
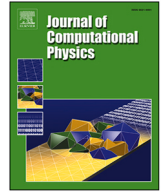
www.reading.ac.uk/centaur

CentAUR

Short note

Nesterov acceleration for ensemble Kalman inversion and variants

Sydney Vernon [a], Eviatar Bach [ID] [a,b,c,*], Oliver R. A. Dunbar [ID] [a]

[a] *California Institute of Technology, Pasadena, California, United States*
[b] *University of Reading, Reading, United Kingdom*
[c] *National Centre for Earth Observation, Reading, United Kingdom*

A B S T R A C T

Ensemble Kalman inversion (EKI) is a derivative-free, particle-based optimization method for solving inverse problems. It can be shown that EKI approximates a gradient flow, which allows the application of methods for accelerating gradient descent. Here, we show that Nesterov acceleration is effective in speeding up the reduction of the EKI cost function on a variety of inverse problems. We also implement Nesterov acceleration for two EKI variants, unscented Kalman inversion and ensemble transform Kalman inversion. Our specific implementation takes the form of a particle-level nudge that is demonstrably simple to couple in a black-box fashion with any existing EKI variant algorithms, comes with no additional computational expense, and with no additional tuning hyperparameters. This work shows a pathway for future research to translate advances in gradient-based optimization into advances in gradient-free Kalman optimization.

1. Introduction

1.1. Inverse problem

We assume that we have a model $\mathcal{G}$ with unknown parameters $u^* \in \mathbb{R}^d$, and observations $y \in \mathbb{R}^k$ generated as

$$y = \mathcal{G}(u^*) + \eta, \tag{1}$$

where η is measurement noise, assumed to be Gaussian $\eta \sim \mathcal{N}(0, \Gamma)$. The inverse problem is then to estimate u^* given y. We assume access to the model $\mathcal{G}$. Note that y can include multiple independent observations of the same quantities.

We define the cost function

$$\mathcal{J}(u) = -\log \mathbb{P}(y|u) = \frac{1}{2}(y - \mathcal{G}(u))^\top \Gamma^{-1}(y - \mathcal{G}(u)), \tag{2}$$

and note that the maximum likelihood estimate (MLE) of u^* is given by

$$u_{\text{MLE}} = \arg\min_u \mathcal{J}(u).$$

1.2. Ensemble Kalman inversion

We now turn to the estimation of u_{MLE} using ensemble Kalman inversion [EKI,1]. While we adopt the notation and terminology of [1], this methodology has a long lineage from randomized likelihood optimizers and ensemble smoothers [2–6] as used for state

---

* Corresponding author.
  E-mail addresses: eviatarbach@protonmail.com (E. Bach), odunbar@caltech.edu (O.R.A. Dunbar).

and parameter estimation in weather forecasting, and for oil reservoir history matching. Moreover, the idea of using an ensemble to approximate a gradient dates back to the introduction of the ensemble Kalman filter [7]. We note that while variants or extensions [8–10] of EKI can also be used to approximate the Bayesian inverse problem of estimating a posterior distribution $\mathbb{P}(u|y)$, here we restrict ourselves to finding a point estimate.

We begin with the continuous-time version of EKI. We first note that

$$-\nabla \mathcal{J}(u) = \frac{d\mathcal{G}}{du}^{\top} \Gamma^{-1}(y - \mathcal{G}(u)). \tag{3}$$

Then, we consider having a probability distribution over values of u and define the covariance matrices

$$C^{uu} = \mathbb{E}[(u - \mathbb{E}[u]) \otimes (u - \mathbb{E}[u])],$$
$$C^{u\mathcal{G}} = \mathbb{E}[(u - \mathbb{E}[u]) \otimes (\mathcal{G}(u) - \mathbb{E}[\mathcal{G}(u)])].$$

We also note that under assumptions that the second derivative of $\mathcal{G}$ is small [11],

$$C^{u\mathcal{G}} \approx C^{uu} \frac{d\mathcal{G}}{du}\bigg|_{\mathbb{E}[u]}^{\top}. \tag{4}$$

We now consider the covariance-preconditioned gradient flow

$$\dot{u} + C^{uu}\nabla \mathcal{J}(u) = 0. \tag{5}$$

This differs from a typical gradient flow in the use of the C^{uu} as a time-dependent preconditioner. This is, however, a natural choice, since this type of gradient flow is affine invariant [11]. Notice that Eq. (5) is a mean-field or McKean–Vlasov equation, in the sense that the time evolution of u depends on its probability distribution. Given some initial distribution $\mathbb{P}(u(0))$, Eq. (5) defines a Liouville equation which propagates this distribution forward in time.

Substituting Eqs. (3) and (4) into Eq. (5), we obtain

$$\dot{u} = C^{u\mathcal{G}}\Gamma^{-1}(y - \mathcal{G}(u)). \tag{6}$$

This is the mean-field EKI [11]. Importantly, the derivative of the model $\frac{d\mathcal{G}}{du}$ is replaced with the derivative-free covariance approximation. We now approximate this further, and replace $C^{u\mathcal{G}}$ by its Monte Carlo approximation from a finite ensemble of particles $\{u^{(n)}\}_{n=1}^{N}$ and discretize in time with an explicit method using a time-step of Δt, obtaining the usual form of EKI:

$$C_j^{u\mathcal{G}} = \frac{1}{N}\sum_{n=1}^{N}\left[(u_j^{(n)} - \overline{u_j}) \otimes (\mathcal{G}(u_j^{(n)}) - \overline{\mathcal{G}(u)}_j)\right], \tag{7a}$$

$$C_j^{\mathcal{G}\mathcal{G}} = \frac{1}{N}\sum_{n=1}^{N}\left[(\mathcal{G}(u_j^{(n)}) - \overline{\mathcal{G}(u)}_j) \otimes (\mathcal{G}(u_j^{(n)}) - \overline{\mathcal{G}(u)}_j)\right], \tag{7b}$$

$$\overline{u}_j = \frac{1}{N}\sum_{n=1}^{N}u_j^{(n)}, \qquad \overline{\mathcal{G}(u)}_j = \frac{1}{N}\sum_{n=1}^{N}\mathcal{G}(u_j^{(n)}), \tag{7c}$$

$$u_{j+1}^{(n)} = u_j^{(n)} + \Delta t C_j^{u\mathcal{G}}(\Gamma + \Delta t C_j^{\mathcal{G}\mathcal{G}})^{-1}(y - \mathcal{G}(u_j^{(n)})). \tag{7d}$$

Note that this discretization is not quite a forward Euler method, as in Eq. (7d) we have replaced Γ^{-1} by $(\Gamma + \Delta t C_j^{\mathcal{G}\mathcal{G}})^{-1}$. This discretization, however, is still consistent with the limit of Eq. (6) as $\Delta t \to 0$, and also maintains consistency with a derivation of EKI based on the discrete-time ensemble Kalman filter as well as having favorable convergence properties; see [12].

Sometimes, the initial ensemble $\{u_0^{(n)}\}_{n=1}^{N}$ is drawn from a prior distribution over u. Since we are here interested in the MLE, and due to the invariant subspace property of EKI [1,13], the choice of initial ensemble can be considered a regularization for the maximum likelihood problem; that is, EKI approximates the MLE but with regularization coming from the fact that the solution must be in the span of the initial ensemble. The recursion Eq. (7d) is repeated until some $j = J$, and the ensemble mean $\overline{u}_J$ taken as an approximation of u_{MLE}. The convergence of the particle approximation to the mean-field limit is proved in [14].

We also note that EKI is often derived from the ensemble Kalman filter or from connection with the Bayesian solution of a linear inverse problem, wherein observations are perturbed for each ensemble member [1]. However, in this paper, since we use the formulation of EKI based on MLE, perturbed observations do not arise.

Several variants of EKI exist. Unscented Kalman inversion [UKI,15] chooses the ensemble using quadrature points at every iteration and has favorable performance for some problems. Ensemble transform Kalman inversion (ETKI), based on the ensemble transform Kalman filter [16], avoids building covariance matrices in the parameter and observation spaces, and has favorable computational complexity if $N \ll d, k$.

1.3. Nesterov acceleration

Nesterov acceleration, sometimes referred to as momentum, is a method to accelerate gradient descent [17,18]. This was extended to continuous-time gradient flows in [19]. Namely, continuous-time Nesterov acceleration transforms a gradient flow

$$\dot{u} + \nabla \mathcal{J}(u) = 0$$

to

$$\ddot{u} + \lambda(t)\dot{u} + \nabla\mathcal{J}(u) = 0. \tag{8}$$

Here, $\lambda(t)$ is the momentum coefficient, which is taken in [19] to be $\lambda(t) = \frac{3}{t}$. In this paper, we will apply this Nesterov acceleration approach to EKI, as well as its variants UKI and ETKI.

1.4. Motivation and related work

Each iteration of Eq. (7d) requires evaluating the forward model $\mathcal{G}(\cdot)$ N times. This is often computationally expensive, as in the case of climate models [9]. Thus, we would like J to be as small as possible, which motivates the need for acceleration.

Nesterov acceleration for EKI was first suggested by [20]; however, detailed numerical tests were not conducted in that work, nor was Nesterov applied to EKI variants. Another method for accelerating convergence of EKI was considered in [21]. A related implementation was also considered for the ensemble Kalman updates with a different objective [22]. There momentum was applied to the ensemble mean and covariance, and with constant momentum parameter. By contrast, we will showcase a simpler particle-wise update, and one with a momentum parameter that changes with iteration in a manner consistent with the gradient-based theory. Momentum has also been considered for particle-based sampling methods in [23].

2. Nesterov acceleration for EKI

We now add Nesterov acceleration to EKI. We begin by considering a covariance-preconditioned Nesterov gradient flow obtained by formally replacing the gradient in Eq. (8) with the covariance-preconditioned gradient as in Eq. (5),

$$\ddot{u} + \lambda(t)\dot{u} + C^{uu}\nabla\mathcal{J}(u) = 0.$$

We note that although improved rates of convergence have previously been proved for continuous-time Nesterov, it remains to be proved that the covariance-preconditioned Nesterov will also have this improvement; however, our numerical results suggest this to be the case.

Now, employing Eqs. (3) and (4), we obtain the mean-field Nesterov-accelerated EKI

$$\ddot{u} + \lambda(t)\dot{u} = C^{u\mathcal{G}}\Gamma^{-1}(y - \mathcal{G}(u)). \tag{9}$$

Due to the non-uniqueness of discretization of Eq. (9), it is difficult to proceed directly with the derivation from these mean-field equations. Instead, we next write our proposed discrete algorithm, and then motivate its consistency with Eq. (9) by taking small-timestep and large-particle limits.

Algorithm 1 Nesterov-accelerated ensemble Kalman inversion.

Require: $\{u_0^{(n)}\}_{n=1}^N$, $J \in \mathbb{N}$, y, Γ, $\mathcal{G}$
1: Compute $C_1^{u\mathcal{G}}, C_1^{\mathcal{G}\mathcal{G}}$
2: $u_1^{(n)} \leftarrow u_0^{(n)} + \Delta t C_1^{u\mathcal{G}}(\Gamma + \Delta t C_1^{\mathcal{G}\mathcal{G}})^{-1}(y - \mathcal{G}(u_1^{(n)})), \forall n = 1, \dots, N$
3: **for** $j = 1, \dots, J$ **do**
4: $\quad \lambda_j \leftarrow \frac{j-1}{j+2},$
5: $\quad v_j^{(n)} \leftarrow u_j^{(n)} + \lambda_j(u_j^{(n)} - u_{j-1}^{(n)}), \forall n = 1, \dots, N$
6: $\quad$ Compute $C_j^{v\mathcal{G}}, C_j^{\mathcal{G}\mathcal{G}}$
7: $\quad u_{j+1}^{(n)} \leftarrow v_j^{(n)} + \Delta t C_j^{v\mathcal{G}}(\Gamma + \Delta t C_j^{\mathcal{G}\mathcal{G}})^{-1}(y - \mathcal{G}(v_j^{(n)})), \forall n = 1, \dots, N$
8: **end for**
9: **return** $\{u_{J+1}^{(n)}\}_{n=1}^N$

We write out the Nesterov-accelerated EKI as Algorithm 1. One can view this implementation of Nesterov acceleration as performing a particle-wise "nudge" based on each particle's history, and then carrying out the regular EKI update Eq. (7d) with the nudged ensemble.

In Algorithm 1 we define λ_j in line with the Nesterov acceleration literature; however, this choice is not unique. We explore other choices and their effect on the convergence in Appendix B.

With any choice of λ_j, the accelerated algorithm retains the subspace property of EKI, that particles will always remain in the span of the initial ensemble [1]. This can be seen inductively, from the steps of Algorithm 1. Firstly, $\{u_0^{(\cdot)}\}$ and $\{u_1^{(\cdot)}\}$ are by definition in this span; then, if $\{u_{k-1}^{(\cdot)}\}$, $\{u_k^{(\cdot)}\}$ are in the span, $\{v_k^{(\cdot)}\}$ is too, as each member update is a linear combination of elements in the span. Then by the original argument [1], $\{u_{k+1}^{(\cdot)}\}$ is also in the span, as a classical EKI update of $\{v_k^{(\cdot)}\}$.

In Appendix A we use a formal argument to find the continuous-time limit of Algorithm 1, but highlight a need for stronger rigorous arguments to show convergence to Eq. (9) in the limit of infinite ensemble size.

Applying the same idea to the UKI and ETKI algorithms, we obtain Nesterov-accelerated versions of these two algorithms in Appendix C.

3. Numerical experiments

We conduct numerical experiments on the impact of Nesterov acceleration on three different inverse problems. Data is generated from equation Eq. (1), so we must define the forward map $\mathcal{G}$, noise covariance Γ, and select a ground-truth parameter u^*. We must additionally pick a probability distribution on u_0, from which the initial ensemble is drawn. We present results of experiments over fifty trials, each with different random draws of the initial ensemble and different noise realizations η in Eq. (1). When reporting the cost function values in the experiments that follow, we use the ensemble mean over the forward map evaluations $\overline{\mathcal{G}(u)}_j$ in expression Eq. (2).

3.1. Example inverse problems

In the exponential sine problem (Exp Sin), we consider:

$$u = [u_1, u_2]^\top,$$
$$f(t, u) = \exp\left(u_1 \sin(t) + u_2\right),$$
$$\mathcal{G}(u) = \begin{bmatrix} \mathrm{mean}_{t \in [0, 2\pi]}(f(t, u)) \\ \max_{t \in [0, 2\pi]}(f(t, u)) - \min_{t \in [0, 2\pi]}(f(t, u)) \end{bmatrix}.$$

That is, we look to estimate the amplitude and vertical shift parameters $u = \begin{bmatrix} u_1, u_2 \end{bmatrix}^\top$ of function f, using the observations given by $\mathcal{G}$. We define the true parameters to be $u^* = [1, 0.8]^\top$. We take a Lognormal$(-1.38, 0.06)$ initial distribution for u_1 and $\mathcal{N}(0, 0.5)$ for u_2, and assume these are independent.

In the Lorenz '96 problem (Lorenz96), we solve for the initial conditions of the D-dimensional Lorenz '96 dynamical system [24] given a state some time after that is integrated from those initial conditions, following a problem used by [13]. Here, we have

$$u = x(t_0) \in \mathbb{R}^D, \qquad \mathcal{G}(u) = x(t_K) \in \mathbb{R}^D,$$
$$\frac{dx_k}{dt} = -x_k - x_{k-1}(x_{k-2} - x_{k+1}) + F,$$

where the indices are cyclical and $x(t) = [x_1(t), \ldots, x_D(t)]^\top$. We use the fourth-order Runge–Kutta method with timestep 0.05, and $t_K = 0.4$. The dimension is set to $D = 20$ and the forcing parameter to $F = 8$, for which the Lorenz '96 model is chaotic. We draw the truth u^* from the D-dimensional standard normal distribution. Taking u^* to be the true state at time t_0, we run through the dynamics until time $t_k = t_0 + 1000$. The initial ensemble members for u are each drawn from the D-dimensional standard normal distribution as well.

Finally, the Darcy problem (Darcy) describes subsurface flow over the domain $\mathcal{D} = [0, 1]^2$. We observe pointwise measurements from a pressure field $p(x)$. We seek to optimize parameter κ to define a discretized permeability field $a(x, \kappa)$. Due to the high dimensionality of the discretized field when spatial resolution is high, we choose to represent it efficiently with a Gaussian random field (GRF) of a given covariance structure (e.g., see the literature review of [25]). We choose a Matérn covariance of smoothness 1 and correlation length 0.25. This covariance is expanded as a Karhunen–Loève eigenfunction expansion, and truncated to d terms (degrees of freedom); the learned u are then coefficients of the truncated expansion. Our forward map then becomes:

$$u \in \mathbb{R}^d, \quad \mathcal{G}(u) = \mathrm{vec}\left(\begin{bmatrix} p(x_{1,1}) & \cdots & p(x_{1,N}) \\ \vdots & \ddots & \vdots \\ p(x_{N,1}) & \cdots & p(x_{N,N}) \end{bmatrix}\right),$$

where $p(x)$ is obtained by solving Darcy's law with a finite-difference scheme. Darcy's law is given by:

$$-\nabla(a(x, \kappa(u))p(x)) = f(x),$$

where $f(x)$ is a given velocity field (flow) and $p(x)$ is set to 0 at the boundary. We set $d = 50$ and $N = 80$, and $u^* = [-1.5, \ldots, -1.5]^\top$. The distribution of the initial ensemble for u is taken to be independent $\mathcal{N}(0, 1)$ for each degree of freedom.

3.2. Implementation

We conduct experiments using the open-source Julia package `EnsembleKalmanProcesses.jl` [26]. This package has implementations of EKI, UKI, and ETKI algorithms, and we have added Nesterov acceleration implementations for these variants as well as the different choices of λ_j.

3.3. Results

Fig. 1 shows the results of applying Nesterov acceleration to the three algorithms EKI, ETKI, and UKI on the three inverse problems. For the first and second row, EKI and ETKI, the ensemble size is taken to be 10, 20, and 52, respectively, for the three inverse problems, weakly related to the increasing problem dimension (Appendix D shows that acceleration occurs regardless of ensemble size). The results are summarized over random trials where the data realization y and the initial ensemble $\{u_0^n\}_{n=1}^N$ are both redrawn. In all three cases, and for all three algorithms, Nesterov acceleration results in faster reduction of the cost function, as well as appearing to
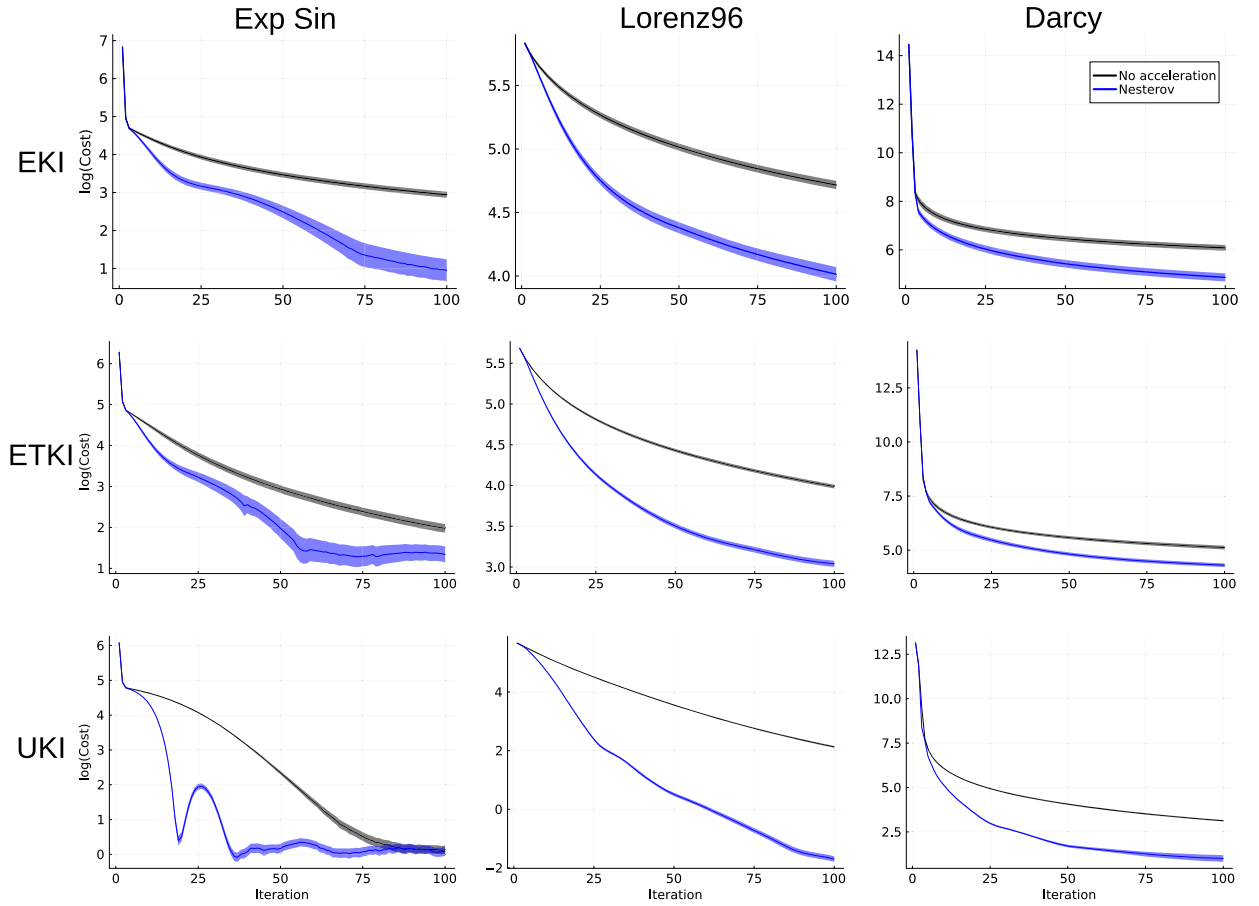
Fig. 1. Comparison of EKI, ETKI, and UKI convergence with and without acceleration, over the three inverse problems of Section 3.1, as measured by the logarithm of the cost function Eq. (2). The results were averaged over 50 trials for exponential sine and Lorenz '96, and 10 trials for Darcy. Ribbons denote one standard error from the mean.

converge to better optima. This suggests that Nesterov acceleration can successfully be applied to EKI variants, such as those described in [15].

For additional verification of the robustness of the method, we conducted three other experiments; the results are shown in Appendix D. Namely, we test the performance of Nesterov-accelerated EKI with a variety of ensemble sizes, with different timesteps Δt, and with different expressions for the momentum coefficient. In all cases, Nesterov-accelerated EKI exhibits improvement compared to non-accelerated EKI.

In all the experiments, Nesterov-accelerated EKI performed at least as well as non-accelerated EKI, and often significantly better, across problems with varying dimensionality of the input space and nonlinearity of the forward model. Despite some variance over different random trials, the improvements are robust also when looking at individual trials. In all the cases tested, the improvement in the decrease of the cost function was observed over the first few iterations in addition to performing better in the long run, indicating that Nesterov acceleration can be useful even in settings where only a small number of iterations J can be afforded.

In some of the cases, such as the UKI experiment (bottom row of Fig. 1), temporary "bumps" where the cost function increases with Nesterov-accelerated EKI can be observed. This may be because Nesterov-accelerated gradient descent, unlike gradient descent, is not strictly a descent method, in the sense that the objective function Eq. (2) is not guaranteed to be monotonically non-increasing with iterations. This behavior is also observed when accelerating gradient descent [19].

4. Conclusions

We have demonstrated that Nesterov acceleration improves performance of EKI on a variety of inverse problems. The form of update we used here is written as a parameter-wise nudging that is simple to implement, non-intrusive to the EKI update, and instantly adaptable to other algorithm variants. There is no additional computational cost, nor are there additional algorithm hyperparameters to tune. We demonstrated this with algorithms for accelerated UKI and ETKI, and also observed improved convergence.

5. Future work

On a theoretical level, it remains to be proven that the Nesterov acceleration in fact improves the convergence rate of covariance-preconditioned gradient flows, as has previously been done for regular gradient flows. Adaptive timestepping through modification of Δt has been explored for EKI [10], and it remains for future work to apply acceleration in such settings. The motivation of Appendix A also still requires rigorous justification to show whether Eq. (9) is the mean-field limit of Eq. (A.6).

Another avenue of future work is the extension of the current work to the ensemble Kalman sampler [EKS,8]. Although there has been previous work on acceleration of continuous-time EKS [23,27], the discrete-time versions of the algorithms are complex and depend in detail on the form of the EKS update. Our approach may offer simple implementations of such updates, and more broadly, application to other empirically approximated non-Bayesian objective functions.

This work demonstrates that improvements of gradient descent algorithms can be transferred to EKI and its variants, by using the EKI's continuous-time gradient flow limit as a bridge. This paves the way for other gradient-based optimizer advancements popular with the machine learning community to be transferred into the EKI framework. Examples of recently developed methods that could be implemented for EKI acceleration are the Information Theoretic Exact Method (ITEM) [28] or its cousin, triple momentum [29,30] (corresponding to ITEM with constant momentum parameters); these methods have provable optimality. Additionally, there is the popular family of adaptive methods (namely, Adam, Nadam, Adan, Adamax, and Adagrad [31–33]) that lack in theoretical guarantees and continuum limits, but are used commonly in the machine learning community.

Data availability

The package EnsembleKalmanProcesses.jl contains the code used for the paper, as stated in the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Sydney Vernon: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Investigation; **Eviatar Bach:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Oliver R. A. Dunbar:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Conceptualization.

Acknowledgment

Appendix A. Continuous-time limit for Nesterov-accelerated EKI

This analysis closely follows the limit of [19]. First, for brevity define

$$\mathcal{F}_n(\{w_j^{(\cdot)}\}) = C_j^{u\mathcal{G}}(\{w_j^{(\cdot)}\})(\Gamma + \Delta t C^{\mathcal{G}\mathcal{G}}(\{w_j^{(\cdot)}\}))^{-1}(y - \mathcal{G}(w_j^{(n)})),$$

$$\overline{\mathcal{F}}_n(\{W^{(\cdot)}(t)\}) = C^{u\mathcal{G}}(\{W^{(\cdot)}(t)\})\Gamma^{-1}(y - \mathcal{G}(W^{(n)}(t))).$$

These are the discrete- and continuous-time expressions for the ensemble update, respectively. We use the notation $\{x^{(\cdot)}\}$ to denote the full ensemble $\{x^{(n)}\}_{n=1}^N$. We also assume that $\mathcal{G}$ is a continuous function of its argument. $C_j^{u\mathcal{G}}(\{w_j^{(\cdot)}\})$ and $C^{\mathcal{G}\mathcal{G}}(\{w_j^{(\cdot)}\})$ indicate the respective empirical covariances computed using the ensemble. We can express the discrete Nesterov update of Algorithm 1 as follows:

$$v_j^{(n)} = u_j^{(n)} + \lambda_j(u_j^{(n)} - u_{j-1}^{(n)}),$$
$$u_{j+1}^{(n)} = v_j^{(n)} - \Delta t \mathcal{F}_n(\{v_j^{(\cdot)}\}). \tag{A.1}$$

Rewriting and using $\lambda_j = \frac{j-1}{j+2}$, this becomes

$$\frac{u_{j+1}^{(n)} - u_j^{(n)}}{\sqrt{\Delta t}} = \frac{j-1}{j+2}\left(\frac{u_j^{(n)} - u_{j-1}^{(n)}}{\sqrt{\Delta t}}\right) - \sqrt{\Delta t}\,\mathcal{F}_n(\{v_j^{(\cdot)}\}). \tag{A.2}$$

We associate a continuous-time trajectory $U^{(n)}(t)$ with the discrete trajectories $u_j^{(n)}$ via the ansatz $u_j^{(n)} = U^{(n)}(t) = U^{(n)}((j-2)\sqrt{\Delta t})$, that is, set $j = t/\sqrt{\Delta t} - 2$. We assume that $U^{(n)}(t)$ is twice-differentiable with respect to t. Then, one can Taylor expand $U(t + \sqrt{\Delta t})$ and $U(t - \sqrt{\Delta t})$ about t to derive the expressions:

$$\frac{u_{j+1}^{(n)} - u_j^{(n)}}{\sqrt{\Delta t}} = \dot{U}^{(n)}(t) + \frac{1}{2}\sqrt{\Delta t}\ddot{U}^{(n)}(t) + o(\sqrt{\Delta t}),$$

$$\frac{u_j^{(n)} - u_{j-1}^{(n)}}{\sqrt{\Delta t}} = \dot{U}^{(n)}(t) - \frac{1}{2}\sqrt{\Delta t}\ddot{U}^{(n)}(t) + o(\sqrt{\Delta t}).$$

Taylor expansion of $U(t - \sqrt{\Delta t})$ in $v_j^{(n)}$ about t gives:

$$v_j^{(n)} = U^{(n)}(t) + \frac{j-1}{j+2}\Big(U^{(n)}(t) - U^{(n)}(t - \sqrt{\Delta t})\Big),$$

$$= U^{(n)}(t) + \sqrt{\Delta t}\frac{j-1}{j+2}\dot{U}(t) + o(\sqrt{\Delta t}). \tag{A.3}$$

We would now like to show that we can write

$$\mathcal{F}_n(\{v_j^{(\cdot)}\}) = \mathcal{F}_n(\{U^{(\cdot)}(t)\}) + o(1). \tag{A.4}$$

To do so, we first note that for matrices $A = \Gamma$ and $\Delta t B = -\Delta t C^{\mathcal{GG}}(U^{(\cdot)}(t))$, and taking Δt small enough so that $\Delta t\|A^{-1}B\| < 1$, where $\|\cdot\|$ is the operator norm, it follows from Neumann series expansion that

$$(A - \Delta t B)^{-1} = A^{-1} + \sum_{k=0}^{\infty} (\Delta t)^k (A^{-1}B)^k A^{-1} = o(\Delta t).$$

We then substitute Eq. (A.3) into $\mathcal{F}_n(\{v_j^{(\cdot)}\})$ and Taylor expand to first order:

$$\Big(C_j^{u\mathcal{G}}(\{U^{(\cdot)}(t)\}) + \sqrt{\Delta t}c_1 + o(\sqrt{\Delta t})\Big) \tag{A.5}$$
$$\times \Big(\Gamma + \Delta t C^{\mathcal{GG}}(\{U^{(\cdot)}(t)\}) + (\Delta t)^{3/2}c_2 + o((\Delta t)^{3/2})\Big)^{-1}$$
$$\times \Big(y - \mathcal{G}(U^{(n)}(t)) + \sqrt{\Delta t}c_3 + o(\sqrt{\Delta t})\Big),$$

where c_1, c_2, and c_3 are not functions of Δt. Then, defining $D = \Gamma + \Delta t C^{\mathcal{GG}}(\{U^{(\cdot)}(t)\})$, we have for sufficiently small Δt the Neumann series expansion

$$(D + (\Delta t)^{3/2}c_2 + o((\Delta t)^{3/2}))^{-1}$$
$$= \Big(I + D^{-1}(\Delta t)^{3/2}c_2 + o((\Delta t)^{3/2})\Big)^{-1}D^{-1},$$
$$= (I - D^{-1}(\Delta t)^{3/2}c_2 + o((\Delta t)^{3/2}))D^{-1},$$

where we have used the fact that D^{-1} is of order Δt for sufficiently small Δt. Substituting back into Eq. (A.5) and expanding, we find the desired result Eq. (A.4).

Using again the fact that D^{-1} is $o(\Delta t)$ for Δt sufficiently small, we deduce

$$\sqrt{\Delta t}\mathcal{F}_n(\{U^{(\cdot)}(t)\}) = \sqrt{\Delta t}\,\overline{\mathcal{F}}_n(\{U^{(\cdot)}(t)\}) + o(\sqrt{\Delta t}).$$

Substitution into Eq. (A.2) results in

$$\dot{U}(t) + \frac{1}{2}\sqrt{\Delta t}\ddot{U}(t) + o(\sqrt{\Delta t}) = \left(1 - \frac{3\sqrt{\Delta t}}{t}\right)\left(\dot{U}(t) - \frac{1}{2}\sqrt{\Delta t}\ddot{U}(t)\right)$$
$$+ \sqrt{\Delta t}\,\overline{\mathcal{F}}_n(\{U^{(\cdot)}(t)\}) + o(\sqrt{\Delta t}).$$

By asymptotic matching on the order $\sqrt{\Delta t}$, one derives an informal relationship:

$$\ddot{U}(t) + \frac{3}{t}\dot{U}(t) = -\overline{\mathcal{F}}_n(\{U^{(\cdot)}(t)\}). \tag{A.6}$$

From here, one would like to take the large-particle limit $N \to \infty$ and show that the mean-field equation is of the form Eq. (9). As seen in [14], where this is carried out for the standard EKI algorithm, this ends up being a technical argument for even linear or mildly nonlinear $\mathcal{G}$, and theory involves jointly taking limits $\Delta t \to 0$ and while fixing $N\Delta t = 1$. Here we have taken $\Delta t \to 0$ before $N \to \infty$ as a motivation, but these limits may not commute.

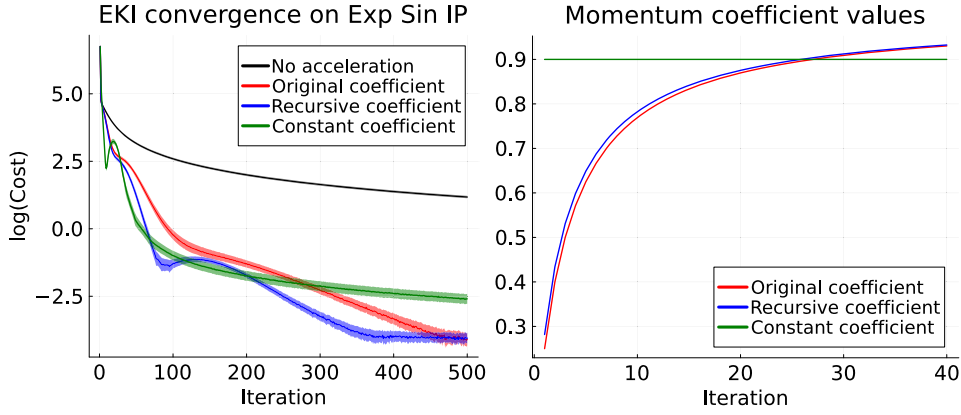**Fig. B.2.** The effect of different choices of the λ_j coefficient for EKI acceleration on the Exp Sin inverse problem of Section 3.1. Left: convergence as measured with the logarithm of the cost function Eq. (2) Ribbons denote one standard error from the mean. Right: values of the λ_j over the first 40 iterations.

Appendix B. Choice of λ_j

Different choices of λ_j (respectively $\lambda(t)$) are made in the literature. In this work we considered three options:

- **Original** [18]: $\lambda_j = \frac{j-1}{j+2}$.
- **Recursive** [19]: $\lambda_j = \theta_j(\theta_{j-1}^{-1} - 1)$, where $\theta_0 = 1$,

$\theta_{j+1} = \frac{1}{2}\left(\sqrt{\theta_j^4 + 4\theta_j^2} - \theta_j^2\right)$.

- **Constant**: $\lambda_j \equiv c$, with c a tuning parameter.

Both the Original and Recursive definitions asymptotically behave as $1 - \frac{3}{j} + O\left(\frac{1}{j^2}\right)$, a property needed for acceleration of Eq. (8); see [19]. The Constant definition, on the other hand, has been shown not to improve asymptotic convergence rates for Eq. (8); see [34]. It is, however, a common practice in the literature. In the numerical experiments that follow in Section 3, we used the Recursive definition unless otherwise stated.

The difference in coefficients, and their effect on performance, is illustrated in a numerical example below.

Fig. B.2 shows the results of applying the two different expressions for the momentum coefficient on the exponential sine problem, as well as a fixed momentum coefficient $\lambda_j = 0.9$ that was tuned for good performance on this problem. We observe both the Original and Recursive give consistently better performance than without acceleration, and moreover appear to attain a higher rate of convergence. In this experiment we see that the Recursive slightly outperforms the Original, though this is likely experiment dependent. The Constant coefficient actually performs worse than without acceleration for the first 10 iterations, then achieves fast convergence for 100 or so iterations, before its convergence stagnates and eventually the dynamic coefficients overtake it.

Appendix C. Ensemble transform Kalman inversion and unscented Kalman inversion

We present the algorithms for accelerated ensemble transform Kalman inversion (ETKI) in Algorithm 2 and accelerated unscented Kalman inversion (UKI) in Algorithm 3. In both cases, the only difference in the accelerated versions of the algorithms is the introduction of the "nudged" ensembles $\{v_j^{(n)}\}_{n,j}$ over which we evaluate the model $\mathcal{G}$. We still write out the full algorithms in easily implementable forms, as they often look significantly different from other forms written in the existing literature (e.g., [15]). These forms also highlights the simplicity of the proposed accelerations.

We do not provide derivations of the new methods here but offer some motivation and interpretation. To motivate ETKI, we recall that one can derive the EKI from the stochastic ensemble Kalman filter (EnKF) [1]. ETKI is found by applying a similar derivation to the class of square-root filters known as ensemble transform Kalman filter (ETKF); the power of this update differs from EKI, in that one computes all matrices in the ensemble space, allowing for preferential scaling to EKI in the output dimensions, assuming Γ is easily invertible [16]. It also does not accumulate errors due to stochastic perturbations.

UKI is obtained, in its basic form, by replacing EKI's Monte Carlo approximation of a Gaussian with a known quadrature rule. It also contains additional regularization, typically by the prior. In lower dimensions this can lead to significant efficiency increases, and very fast convergence on some problems. There are a number of free parameters and matrices in this algorithm, and for our experiments we select these based on the work of [15]: we take the prior to be $N(m, C)$ and take $r = m$, $\Sigma_\nu = 2\Gamma$, and $\Sigma_\omega = (2 - \alpha^2)C$. We therefore need only to choose $\alpha \in (0, 1]$ in experiments. UKI picks the initial ensemble deterministically; therefore, the different trials in the numerical experiments vary only due to using different data realizations.

Algorithm 2 Nesterov-accelerated ensemble transform Kalman inversion.

    function compute_increment$(u, \bar{u}, \{\mathcal{G}(u^{(n)})\}_n, \Gamma)$

2:      Compute $\overline{\mathcal{G}}(u) \leftarrow \frac{1}{N}\sum_n \mathcal{G}(u^{(n)})$

        $\mathrm{col}_n(\Delta u) \leftarrow \frac{1}{\sqrt{N-1}}(u^{(n)} - \bar{u})$

4:      $\mathrm{col}_n(\Delta \mathcal{G}(u)) \leftarrow \frac{1}{\sqrt{N-1}}(\mathcal{G}(u^{(n)}) - \overline{\mathcal{G}}(u))$

        $\Omega(u) \leftarrow (I + \Delta\mathcal{G}(u)^\top \Gamma^{-1} \Delta\mathcal{G}(u))^{-1}$

6:      $w(u) \leftarrow \Omega \Delta \mathcal{G}(u)^\top \Gamma^{-1}(y - \overline{\mathcal{G}}(u))$

        $\delta u \leftarrow \bar{u} + \Delta u(w(u) + \sqrt{N-1}\sqrt{\Omega(u)})$

8:      **return** δu

    end function

10:

Require: $u_0 = \{u_0^{(n)}\}_{n=1}^N$, $J \in \mathbb{N}$, y, Γ, $\mathcal{G}$

    Compute $\bar{u}_0 \leftarrow \frac{1}{N}\sum_n u_0^{(n)}$

12: $\delta u_0 \leftarrow$ compute_increment$(u_0, \bar{u}_0, \{\mathcal{G}(u^{(n)})\}_n, \Gamma)$

    $u_1 \leftarrow \bar{u}_0 + \delta u_0$

14: **for** $j = 1, \ldots, J$ **do**

      $\lambda_j \leftarrow \frac{j-1}{j+2}$,

16:     $v_j^{(n)} \leftarrow u_j^{(n)} + \lambda_j(u_j^{(n)} - u_{j-1}^{(n)}), \forall n = 1, \ldots, N$

      Compute $\bar{v}_j \leftarrow \frac{1}{N}\sum_n v_j^{(n)}$

18:     $\delta v_j \leftarrow$ compute_increment$(v_j, \bar{v}_j, \{\mathcal{G}(v^{(n)})\}_n, \Gamma)$

      $u_{j+1} \leftarrow \bar{v}_j + \delta v_j$

20: **end for**

    return $u_{J+1} = \{u_{J+1}^{(n)}\}_{n=1}^N$

Appendix D. Additional experiments

We briefly present three additional experiments in support of the above results.

We test the performance of Nesterov-accelerated EKI on the Darcy problem with two different ensemble sizes, $N = 10$ and $N = 200$, shown in Fig. D.3. In the case of $N = 10$, both algorithms can explore only the nine-dimensional subspace spanned by the initial ensemble, while the true parameters are in a fifty-dimensional space; this leads to higher costs and large spread across trials due to sampling error. For $N = 200$ case there is no restriction as the ensemble subspace contains the true parameters, and the cost is greatly reduced. In both cases, we again observe clearly that the Nesterov acceleration is beneficial for convergence, though the spread due to sampling error in the small-ensemble regime causes some overlapping performance in the first few iterations.

Fig. D.4 shows the results of applying EKI and Nesterov-accelerated EKI on the exponential sine problem with varying Δt. Although the timestep has an influence on the performance, the Nesterov-accelerated EKI with the worst choice of timestep still outperforms
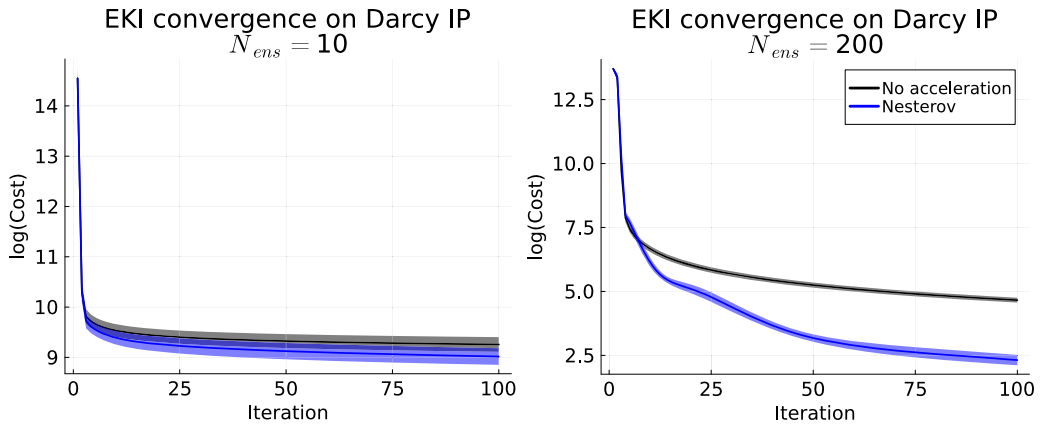

Fig. D.3. The effect of ensemble size on EKI convergence with and without acceleration, on the Darcy inverse problem of Section 3.1, as measured by the logarithm of the cost function Eq. (2). Nesterov-accelerated EKI applied to the Darcy problem, with ensemble sizes 10 and 200. An experiment with ensemble size 52 is given in Fig. 1.

Algorithm 3 Nesterov-accelerated unscented Kalman inversion.

    function generate_ensemble($m, C, r, \alpha, \Sigma_\omega$)

        $\hat{m} \leftarrow r + \alpha(m - r)$, $\hat{C} \leftarrow \alpha^2 C + \Sigma_\omega$

3:       $L \leftarrow \text{Cholesky}(\hat{C})$

        $N \leftarrow \dim(\hat{m})$, $\gamma \leftarrow \sqrt{N}\min\{\sqrt{\frac{4}{N}}, 1\}$

$$\begin{cases} u^{(0)} \leftarrow \hat{m} \\ u^{(n)} \leftarrow \hat{m} + \gamma\,\text{col}_n(L) & n = 1, \ldots, N \\ u^{(N+n)} \leftarrow \hat{m} - \gamma\,\text{col}_n(L) & n = 1, \ldots, N \end{cases}$$

6:       **return** u

    end function

9: **function** update_mean_cov($u, \{\mathcal{G}(u^{(n)})\}_n, y, \Sigma_\nu$)

        Compute $\hat{m}$ and $\hat{C}$ from u,

        $N \leftarrow \dim(\hat{m})$ $\gamma \leftarrow \sqrt{N}\min\left\{\sqrt{\frac{4}{N}}, 1\right\}$

12:      $\hat{C}^{u\mathcal{G}} \leftarrow \sum_n \frac{1}{2\gamma^2}(u^{(n)} - \hat{m})(\mathcal{G}(u^{(n)}) - \mathcal{G}(\hat{m}))^\top$

        $\hat{C}^{\mathcal{G}\mathcal{G}} \leftarrow \sum_n \frac{1}{2\gamma^2}(\mathcal{G}(u^{(n)}) - \mathcal{G}(\hat{m}))(\mathcal{G}(u^{(n)}) - \mathcal{G}(\hat{m}))^\top + \Sigma_\nu$

        $m \leftarrow \hat{m} + C^{u\mathcal{G}}(C^{\mathcal{G}\mathcal{G}})^{-1}(y - \mathcal{G}(\hat{m}))$

15:      $C \leftarrow \hat{C} - C^{u\mathcal{G}}(C^{\mathcal{G}\mathcal{G}})^{-1}(C^{u\mathcal{G}})^\top$

        return m, C

    end function

18:

Require: $m_0, C_0, y, \mathcal{G}, r, \alpha, \Sigma_\omega, \Sigma_\nu$

    $u_0 \leftarrow \text{generate\_ensemble}(m_0, C_0, r, \alpha, \Sigma_\omega)$

    $m_1, C_1 \leftarrow \text{update\_mean\_cov}(u_0, \{\mathcal{G}(u_0^{(n)})\}_n, y, \Sigma_\nu)$

21: **for** $j = 1, \ldots, J$ **do**

    $u_j \leftarrow \text{generate\_ensemble}(m_j, C_j, r, \alpha, \Sigma_\omega)$

    $\lambda_j \leftarrow \frac{j-1}{j+2}$

24:     $v_j^{(n)} \leftarrow u_j^{(n)} + \lambda_j(u_j^{(n)} - u_{j-1}^{(n)})$

    $m_{j+1}, C_{j+1} \leftarrow \text{update\_mean\_cov}(v_j, \{\mathcal{G}(v_j^{(n)})\}_n, y, \Sigma_\nu)$

    end for
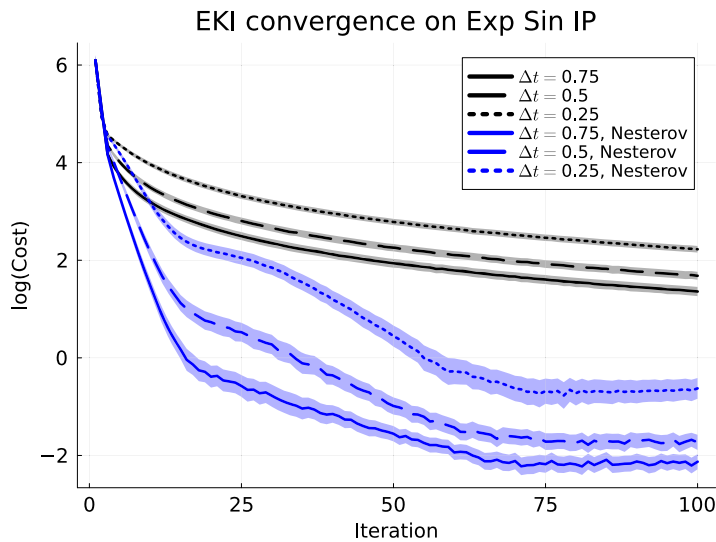
27: **return** m_{J+1}, C_{J+1}

Fig. D.4. The effect of taking different Δt on EKI convergence with and without acceleration on the Exp Sin inverse problem of Section 3.1, as measured by the logarithm of the cost function Eq. (2).

(after 10 iterations) the non-accelerated EKI with the best choice of timestep. This suggests that the Nesterov acceleration can compensate for a poorly chosen timestep.

References

[1] M.A. Iglesias, K.J.H. Law, A.M. Stuart, Ensemble Kalman methods for inverse problems, Inverse Probl. 29 (4) (2013) 045001. https://doi.org/10.1088/0266-5611/29/4/045001

[2] P.J. van Leeuwen, G. Evensen, Data assimilation and inverse methods in terms of a probabilistic formulation, Mon. Weather Rev. 124 (12) (1996) 2898–2913. https://doi.org/10.1175/1520-0493(1996)124<2898:DAAIMI>2.0.CO;2

[3] G. Evensen, P.J. Van Leeuwen, An ensemble Kalman smoother for nonlinear dynamics, Mon. Weather Rev. 128 (6) (2000) 1852–1867. https://doi.org/10.1175/1520-0493(2000)128<1852:AEKSFN>2.0.CO;2

[4] Y. Chen, D.S. Oliver, D. Zhang, Efficient ensemble-based closed-loop production optimization, SPE J. 14 (04) (2009) 634–645. https://doi.org/10.2118/112873-PA

[5] Y. Chen, D.S. Oliver, Ensemble randomized maximum likelihood method as an iterative ensemble smoother, Math. Geosci. 44 (2012) 1–26. https://doi.org/10.1007/s11004-011-9376-z

[6] M. Bocquet, P. Sakov, An iterative ensemble Kalman smoother, Q. J. R. Meteorol. Soc. 140 (682) (2014) 1521–1535. https://doi.org/10.1002/qj.2236

[7] G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res. Oceans 99 (C5) (1994) 10143–10162. https://doi.org/10.1029/94JC00572

[8] A. Garbuno-Inigo, F. Hoffmann, W. Li, A.M. Stuart, Interacting Langevin diffusions: gradient structure and ensemble Kalman sampler, SIAM J. Appl. Dyn. Syst. 19 (1) (2020) 412–441. https://doi.org/10.1137/19M1251655

[9] E. Cleary, A. Garbuno-Inigo, S. Lan, T. Schneider, A.M. Stuart, Calibrate, emulate, sample, J. Comput. Phys. 424 (2021) 109716. https://doi.org/10.1016/j.jcp.2020.109716

[10] M. Iglesias, Y. Yang, Adaptive regularisation for ensemble Kalman inversion, Inverse Probl. 37 (2) (2021) 025008. https://doi.org/10.1088/1361-6420/abd29b

[11] E. Calvello, S. Reich, A.M. Stuart, Ensemble Kalman methods: a mean field perspective, (2024). arXiv:2209.11371

[12] D. Blömker, C. Schillings, P. Wacker, A strongly convergent numerical scheme from ensemble Kalman inversion, SIAM J. Numer. Anal. 56 (4) (2018) 2537–2562. https://doi.org/10.1137/17M1132367

[13] X.T. Tong, M. Morzfeld, Localized ensemble Kalman inversion, Inverse Prob. 39 (6) (2023) 064002. https://doi.org/10.1088/1361-6420/accb08

[14] Z. Ding, Q. Li, Ensemble Kalman inversion: mean-field limit and convergence analysis, Stat. Comput. 31 (1) (2021) 9. https://doi.org/10.1007/s11222-020-09976-0

[15] D.Z. Huang, T. Schneider, A.M. Stuart, Iterated Kalman methodology for inverse problems, J. Comput. Phys. 463 (2022) 111262. https://doi.org/10.1016/j.jcp.2022.111262

[16] M.K. Tippett, J.L. Anderson, C.H. Bishop, T.M. Hamill, J.S. Whitaker, Ensemble square root filters, Mon. Weather Rev. 131 (7) (2003) 1485–1490. https://doi.org/10.1175/1520-0493(2003)131<1485:ESRF>2.0.CO;2

[17] Y.E. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$, Sov. Mathematics—Doklady 27 (2) (1983) 372–376.

[18] Y. Nesterov, Introductory Lectures on Convex Optimization, Vol. 87 of Applied Optimization, Springer US, Boston, MA, Boston, MA, 2004. https://doi.org/10.1007/978-1-4419-8853-9

[19] W. Su, S. Boyd, E.J. Candès, A Differential equation for modeling Nesterov's accelerated gradient method: theory and insights, J. Mach. Learn. Res. 17 (153) (2016) 1–43. https://jmlr.org/papers/v17/15-084.html.

[20] N.B. Kovachki, A.M. Stuart, Ensemble Kalman inversion: a derivative-free technique for machine learning tasks, Inverse Probl. 35 (9) (2019) 095005. https://doi.org/10.1088/1361-6420/ab1c3a

[21] N. Chada, X. Tong, Convergence acceleration of ensemble Kalman inversion in nonlinear settings, Math. Comput. 91 (335) (2022) 1247–1280. https://doi.org/10.1090/mcom/3709

[22] M.M. Nilsen, A.S. Stordal, R.J. Lorentzen, P.N. Raanes, K.S. Eikrem, Accelerated ensemble optimization using momentum methods, Sci. Rep. 14 (1) (2024) 25336. https://doi.org/10.1038/s41598-024-76916-7

[23] Z. Liu, A.M. Stuart, Y. Wang, Second Order Ensemble Langevin Method for Sampling and Inverse Problems, (2022). arXiv:2208.04506

[24] E.N. Lorenz, Predictability: a problem partly solved, in: Proceedings of a Seminar Held at ECMWF on Predictability, 4-8 September 1995, 1, ECMWF, Shinfield Park, Reading, 1996, pp. 1–18.

[25] J. Latz, M. Eisenberger, E. Ullmann, Fast sampling of parameterised Gaussian random fields, Comput. Methods Appl. Mech. Eng. 348 (2019) 978–1012. https://doi.org/10.1016/j.cma.2019.02.003

[26] O.R.A. Dunbar, I. Lopez-Gomez, A. Garbuno-Iñigo, D.Z. Huang, E. Bach, J.-l. Wu, EnsembleKalmanProcesses.Jl: Derivative-free Ensemble-Based Model Calibration, J. Open Source Softw. 7 (80) (2022) 4869. https://doi.org/10.21105/joss.04869

[27] Y. Wang, W. Li, Accelerated information gradient flow, J. Sci. Comput. 90 (1) (2021) 11. https://doi.org/10.1007/s10915-021-01709-3

[28] A. Taylor, Y. Drori, An optimal gradient method for smooth strongly convex minimization, Math. Program. 199 (1) (2023) 557–594. https://doi.org/10.1007/s10107-022-01839-y

[29] B. Van Scoy, R.A. Freeman, K.M. Lynch, The fastest known globally convergent first-order method for minimizing strongly convex functions, IEEE Control Syst. Lett. 2 (1) (2018) 49–54. https://doi.org/10.1109/LCSYS.2017.2722406

[30] A. d'Aspremont, D. Scieur, A. Taylor, Acceleration methods, Found. Trends Optim. 5 (1-2) (2021) 1–245. https://doi.org/10.1561/2400000036

[31] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, (2017). arXiv:1412.6980

[32] T. Dozat, Incorporating Nesterov momentum into Adam, in: Proceedings of the 4th International Conference on Learning Representations, Workshop Track, San Juan, Puerto Rico, 2016. https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ.

[33] X. Xie, P. Zhou, H. Li, Z. Lin, S. Yan, Adan: adaptive Nesterov momentum algorithm for faster optimizing deep models, IEEE Trans. Pattern Anal. Mach. Intell. 46 (12) (2024) 9508–9520. https://doi.org/10.1109/TPAMI.2024.3423382

[34] N.B. Kovachki, A.M. Stuart, Continuous time analysis of momentum methods, J. Mach. Learn. Res. 22 (17) (2021) 1–40. http://jmlr.org/papers/v22/19-466.html.