LEARNING LANDMARK GEODESICS USING 
THE ENSEMBLE KALMAN FILTER

Andreas Bock* and Colin J. Cotter 
Department of Mathematics 
Imperial College London 
South Kensington Campus 
London SW7 2AZ, UK

(Communicated by Kody Law)

ABSTRACT. We study the problem of diffeomorphometric geodesic landmark matching where the objective is to find a diffeomorphism that, via its group action, maps between two sets of landmarks. It is well-known that the motion of the landmarks, and thereby the diffeomorphism, can be encoded by an initial momentum leading to a formulation where the landmark matching problem can be solved as an optimisation problem over such momenta. The novelty of our work lies in the application of a derivative-free Bayesian inverse method for learning the optimal momentum encoding the diffeomorphic mapping between the template and the target. The method we apply is the ensemble Kalman filter, an extension of the Kalman filter to nonlinear operators. We describe an efficient implementation of the algorithm and show several numerical results for various target shapes.

1. Introduction. A central matching problem in shape analysis is to find a diffeomorphism that, via its group action, brings into alignment two sets of so-called landmarks (point clouds in our domain \( \Omega := \mathbb{R}^d \)) called the template and the target - a problem in line with the metric pattern theory \([22]\) framework of Grenander \([9]\). For applications in computational anatomy \([8]\) it is often convenient to choose diffeomorphometric methods for matching since this class provides transformations that can represent smooth biological growth. A popular method is the large deformation diffeomorphic metric mapping (LDDMM) \([1, 14, 35]\). In LDDMM we study a curve \( t \mapsto \varphi_t \in \text{Diff}_V(\Omega), \ t \in [0,1] \) on the group of diffeomorphisms \( \text{Diff}_V(\Omega) \) induced by a curve \( t \mapsto u_t \) on the smooth vector space \( V \rightarrow C_0^1(\Omega, \mathbb{R}^d) \) (the space of continuously differentiable functions on \( \Omega \) vanishing at infinity and taking values in \( \mathbb{R}^d \)) via the following equation, see e.g. \([6]\):

\[
\dot{\varphi}_t = u_t \circ \varphi_t, \quad \varphi_0 = \text{id}.
\]

Letting \( \varphi_1 \cdot q \) denote the group action of \( \varphi_1 \) on \( q \), the LDDMM matching problem can be solved by finding the one-parameter family of velocity fields \( t \mapsto u_t \in V \) such that the distance between a target \( q_1 \) and \( \varphi_1 \cdot q_0 \) is minimized. The candidate curve of velocities in \( V \) is the one whose kinetic energy \( \int_0^1 \| u_t \|_V^2 \, dt \) is minimized subject to...
to (1). When the ambient space is the plane one may think of this problem as one of finding the velocity which takes the least amount of energy to deform the underlying sheet so that $q_0$ is aligned with $q_1$ without breaking or folding it onto itself. Figure 1 illustrates this, where we have computed (using shooting, which is explained later) geodesics between two landmark configurations. It is clear from this figure that the landmarks follow in some sense the “short path” to each other without intersecting - note that the speed of these geodesics is not visualized. As such, no matter is “destroyed” or “created” by the deformation. See [31, 30] for

![Figure 1. A matching between landmarks where the geodesics are shown.](image)

the technical development of the family of methods using this approach and [32] for an extension of LDDMM called metamorphosis. A particular strength of this framework is that the geodesic motion of the shape can be encoded by an initial momentum conjugate to the template. This means that the full curve $\varphi$ can be described by a single initial momentum rather than a curve of velocities in $V$. The technical details are described in section 2. LDDMM can therefore be viewed as an inverse problem in the sense that we want to find the momentum parameter leading to the observation given by the evolution of the template shape at time $t = 0$ to fit the target shape at time $t = 1$. A popular method for solving such inverse problems is called shooting and we highlight here some central references explaining this approach. Shooting, which we used to produce Figure 1 typically employs a Newton method to explore the space of momentum provided some initial guess; the gradient-based optimisation scheme of Beg [1] popularised this for LDDMM. See [21] for a general introduction to geodesic shooting for different data. See [17] for an introduction to numerical implementation aspects of landmark geodesic equations.

In this paper we build a surrogate Bayesian model on the tangent space of the landmark manifold and treat the landmark matching problem as a Bayesian inverse
problem to arrive at a matching algorithm that does not use adjoint-based differentiation methods such as e.g. [17] (and in this sense is derivative-free). An inverse problem in its most abstract form seeks to recover the input parameter \( p \in P \) that is mapped to a known state \( q \in Q \) by some typically known uncertainty-to-state operator \( G := O \circ f : P \to Q \), where \( O \) is the observation operator and \( f \) a forward operator e.g. via the solution of a differential equation such as the geodesic equations for landmarks. When \( f \) is the operator mapping from initial momentum \( p \) to a candidate target \( G(p) \), and \( O = \text{id} \) (i.e. we observe all the points \( f(p) \)) we can also view \( G(p) = q_1 \) as a Bayesian inverse problem [29, 15] where the aim is to find a distribution of candidate functions \( p \) that minimize the kinetic energy of a transformation, see [4, 19] for examples of Bayesian approaches in shape registration. This means inverting, in some sense, \( G \) so that \( p = G^{-1}(q) \). In practice, we are often simply looking for an approximation: \( p \approx G^{-1}(q) \), and the inverse may not be unique or classically defined. We can view shape matching as the inverse problem of finding the velocity \( p \) such that an initial shape \( q_0 \) is mapped to \( q_1 \approx G(p) = (O \circ f)(p) \) where \( f \) is the forward geodesic motion of the template landmarks. In a crude sense we wish to bound \( \| q_1 - G(p) \|_Q \) by the following triangle inequality:

\[
\| q_1 - G(p) \|_Q \leq \| q_1 - G_h(p) \|_Q + \| G_h(p) - G(p) \|_Q \\
\leq \| q_1 - G_h(p) \|_Q + \| G_h - G \|_{P^*} \| p \|_P.
\]

where \( G_h \) is a numerical approximation of \( G \) and \( \| \cdot \|_{P^*} \) is a formal dual norm. In the present context we control \( \| G_h - G \|_{P^*} \) by numerical discretization of the forward operator. The aim is to learn the optimal momentum \( p \) that allows us to take \( \| q_1 - G_h(p) \|_Q \) to zero, and we do so by applying a nonlinear filtering method called the ensemble Kalman filter (enKf). As for geodesic shooting we exploit the linearity of the space of momentum to define an iterative Bayesian method. In this setting, a collection - or ensemble - of initial momentum is drawn from a proposed prior distribution and is iteratively updated by measuring its likelihood as a function of how close the average template landmarks are from the target under the flow of the diffeomorphism generated by the initial momentum. At the enKf level the algorithm is embarrassingly parallel in the ensemble dimension and we present several numerical results using a parallel implementation. The enKf algorithm does not rely on the derivative of \( G \), paving the way for researchers to use other forward operators instead of those for LDDMM that are used in this paper.

1.1. Outline. A mathematical treatment of LDDMM using a reproducing kernel Hilbert space framework is provided in section 2 along with a presentation of Hamilton’s equations for landmarks. Section 3 describes the enKf in detail and its application to landmark matching. Next, section 4 shows numerical examples for various landmark configurations and demonstrates the robustness of our approach. Section 5 summarises this paper.

2. Large deformation diffeomorphic metric mapping. To set up the notation we first present some preliminaries in section 2.1 before describing the classic LDDMM framework in section 2.2.

2.1. Mathematical preliminaries. In this paper, \( \| \cdot \|_{\mathbb{R}^d} \) denotes the standard Euclidean norm and we let \( \langle h, g \rangle_Q := \langle h, g \rangle_{\mathbb{R}^{d \times M}} := \sum_{i=1}^{M} (h^i, g^i)_{\mathbb{R}^d} \) for \( h, g \in \mathbb{R}^{d \times M} \). Further, let \( \langle \cdot, \cdot \rangle_{0^d, \Omega} \) be the \( L^2(\Omega) \) norm of vector \( d \)-valued functions defined over \( \Omega \).
We use $q_t \in Q := \mathbb{R}^{d \times M}$ to denote a vector of $M$ landmarks $q_t = \{q^i_t\}_{i=1}^M$ at time $t \in [0, 1]$, with $q^i_t \in \mathbb{R}^d$ for $d = 2$. When the time index is omitted $q$ refers to the one-parameter family of landmark positions and we say $q \in Q := \{q \mid t \mapsto q_t \in Q, t \in [0, 1]\}$. We further restrict the space of velocities and let $V \hookrightarrow C^2_{0}(\Omega)$ denote a vector space to be specified later on, and $V = L^2([0, 1], \mathbb{R}^d)$ the space of square-integrable curves taking values in the sufficiently smooth space $V$ which we define later. Further, when $\varphi_t \in \text{Diff}_V(\Omega)$ is a diffeomorphism we understand the action of $\varphi_t$ on $q_t \in Q$ as: $\varphi_t.q_t = \{\varphi_t.q^i_t\}_{i=1}^M$, where $\varphi_t.q^i_t = q^i_t \circ \varphi^{-1}_t$. For brevity we shall use the notation: $\dot{q}_t = u_t \circ q_t$, $t \in [0, 1]$, to describe the evolution of the collection of landmarks whereby each landmark $i = 1, \ldots M$ is governed by: $\dot{q}^i_t = u_t \circ q^i_t$, $t \in [0, 1]$.

In other words, when the velocities $u_t$ are sufficiently smooth it can be shown that the evolution of the landmarks is diffeomorphic. Next we discuss a practical way that in which we have modelled this space. In this paper the space $V$ is a reproducing kernel Hilbert space [35, Chapter 9] with kernel $K_V : \Omega \times \Omega \rightarrow \mathbb{R}$. We denote by $L_V : V \rightarrow L^2(\Omega, \mathbb{R}^d)$ the symmetric operator generating $V$ which for all $u \in V$ satisfies the following:

$$\|u\|^2_V = \langle L_V u, u \rangle_{\Omega^d, \Omega},$$

$$\langle u, u \rangle_{\Omega^d, \Omega} \leq C \langle L_V u, u \rangle_{\Omega^d, \Omega},$$

for some constant $C > 0$, with an associated inverse described by $K_V$. We assume that the kernel $K_V$ is Gaussian:

$$K_V(x, x') = e^{-\frac{\|x-x'\|^2_{\mathbb{R}^d}}{2\tau^2}},$$

(2)

where $\tau > 0$ is a kernel parameter that determines the interaction of the landmarks which we refer to as the size of the landmarks. This is easily seen: as $\tau \rightarrow 0$, $x$ and $y$ may be closer in the plane before $K_V(x, y)$ takes values away from zero. To summarise, we see that a smooth evolution of the landmarks can be represented via different parameterizations of a smooth kernel.

2.2. Geodesics. The LDDMM matching problem between two configurations of landmarks $q_0$ and $q_1$ seeks to minimize the following functional as a function of $u \in V$:

$$\mathcal{L} = \frac{1}{2} \int_0^1 \|u_t\|^2_V \, dt,$$

subject to the evolution equation $\dot{q}_t = u_t \circ q_t$ and the boundary conditions $q_0$. Note that (3) can be viewed as a regularization term for the geodesics. We address the end-point condition momentarily. Letting $p$ denote the conjugate momentum to $q$ we can write the Lagrangian associated to (3):

$$\mathcal{L} = \int_0^1 (p_t - q_t \circ u_t)_{Q, Q^*} - \frac{1}{2} ||u_t||^2_V \, dt.$$
Differentiating $\mathcal{L}$ with respect to arbitrary variations $\delta u$, $\delta p$ and $\delta q$ in $u$, $p$ and $q$, respectively, gives us the equations:

\begin{align*}
\langle L_V u_t, \delta u \rangle_{Q,Q^*} &= (p_t | \delta u \circ q_t)_{Q,Q^*}, \quad \forall \delta u \in V, \quad (4) \\
(p | \delta q - \nabla u_t \circ q \delta q)_{Q,Q^*} &= 0, \quad \forall \delta q \in Q, \quad (5) \\
(\delta p | q_t - u_t \circ q_t)_{Q,Q^*} &= 0, \quad \forall \delta p \in Q^*. \quad (6)
\end{align*}

Using the properties of the RKHS this variational system has an explicit solution in terms of $p_t$ and $q_t$. Landmarks can be viewed as measures (see e.g. [11, Section 6.1]) since landmarks can be lifted to $x_1$ when $t = 1$ in (7)-(9), defining the misfit function by:

$$y \mapsto \int_{\Omega} \delta u(x) \delta q_t(x),$$

Integrating by parts in (5) and using $L_V^{-1} = K_V$ we can write Hamilton’s equations as follows where we seek $u \in \mathcal{V}$, $p \in \mathcal{Q}$ (since $\mathcal{Q} \simeq \mathcal{Q}^*$) and $q \in \mathcal{Q}$:

\begin{align*}
    u_t(q_j^t) &= \sum_{i=1}^M K_V(q_i^t, q_j^t)p_i^t, \quad j = 1, \ldots, M, \quad (7) \\
    \dot{p}_t &= -\nabla u_t^\top \circ q_t p_t, \quad (8) \\
    \dot{q}_t &= u_t \circ q_t, \quad (9)
\end{align*}

subject to the boundary condition $q_0$. When we do not enforce an end-point condition, (8) and (9) are both simple ODEs. Note that $u$ is fully described by $K_V$, $p$ and $q$. Since $K_V$ is provided as a parameter and $\mathfrak{q}_0$ is known the system (7)-(9) is fully described by $p_0$ and the integrand of (3) can be written as:

$$\|u_t\|_V^2 = \sum_{i,j=1}^M p_i^t K_V(q_i^t, q_j^t)p_j^t. \quad (10)$$

The initial momentum encodes the forward geodesic motion to provide destination shapes $q_1$ at $t = 1$ by integration in time. Using similar notation as in section 1 we define, for a template $\mathfrak{q}_0$, the forward map $f_{\mathfrak{q}_0} : Q \to Q$ by:

$$p_0 = \{p_j^0\}_{j=1}^M \mapsto f_{\mathfrak{q}_0}[p_0]. \quad (11)$$

To avoid confusion we let $\mathfrak{q}_1$ denote the desired target and $f_{\mathfrak{q}_0}[p_0]$ the landmarks at time $t = 1$ in (7)-(9), defining the misfit function by:

$$p_0 \mapsto \mathfrak{q}_1 - f_{\mathfrak{q}_0}[p_0],$$

which we aim to minimize in the coming sections.

For $T$ timesteps $0, \ldots, T - 1$ of size $\Delta t = T^{-1}$ we choose a forward Euler scheme to discretize the time derivative in (8) and (9). This leads to a discrete forward operator depending on $T$ which, with a slight abuse of notation, shall also be denoted $f$. All of our simulations are implemented in Python using Pytorch [24] and KeOps (https://kernel-operations.io/keops/index.html, see also [3]). Appendix A contains details on how to obtain and run our code.

In summary, the evolution equation of the landmarks, along with a principle of least energy leads us to a system of equations for $u$, $q$ and $p$. Along with this and our choice of the space $V$ for the velocities we naturally obtain an operator that maps initial momenta $p_0$ to different positions $f_{\mathfrak{q}_0}[p_0]$. Next we demonstrate that
the enKf can be used to provide us with the correct initial “kick” that sets the template landmarks in motion towards the desired target.

3. Bayesian inverse problem.

3.1. The ensemble Kalman filter. The enKf is a Monte Carlo data assimilation [25] algorithm dating back to 1994 [7] where the objective is to estimate the posterior \( \rho(\mathbf{X}|\mathbf{Y}) \) of a system i.e. the probability of \( \mathbf{X} \in \mathcal{X} \) given \( \mathbf{y} \in \mathcal{Y} \), via Bayes’ rule:

\[
\rho(\mathbf{X}|\mathbf{Y}) \propto \rho(\mathbf{X})\rho(\mathbf{Y}|\mathbf{X}),
\]

for some prior information \( \rho(\mathbf{X}) \sim \mathcal{N}(\mu, \mathbf{C}) \) about a state \( \mathbf{X} \) and likelihood \( \rho(\mathbf{Y}|\mathbf{X}) \) of a prediction \( \mathbf{Y}|\mathbf{X} \in \mathcal{Y} \). Whereas the covariance \( \mathbf{C} \) of the prior in the standard Kalman filter is prescribed [16], the enKf instead computes sample statistics from a collection, or ensemble, \( \mathbf{X} = \{X_i\}_{i=1}^{N_E} \) of \( N_E \) state vectors taking values in \( \mathcal{X} \) equipped with an inner product \( \langle \cdot, \cdot \rangle_{\mathcal{X}} \). For such an \( \mathbf{X} \) we compute the sample mean \( \bar{\mathbf{X}} \) and write the action of the covariance \( \mathbf{C}_E \):

\[
\bar{\mathbf{X}} := \frac{1}{N_E} \sum_{i=1}^{N_E} X_i, \quad (13)
\]

\[
\mathbf{C}_E[\cdot] := \frac{1}{N_E - 1} \sum_{i=1}^{N_E} (X_i - \bar{\mathbf{X}})(X_i - \bar{\mathbf{X}})^T, \quad (14)
\]

We consider the following evolving system, for iterations \( k \geq 1 \) and ensemble members \( i = 1, \ldots, N_E \), where \( H : \mathcal{X} \to \mathcal{Y} \) maps from state to prediction:

\[
\mathbf{Y}^k_i = H\mathbf{X}^k_i + \eta^k_i, \quad \eta^k_i \sim \mathcal{N}(0, \mathbf{R}),
\]

where \( \mathbf{R} \) is the covariance of the noise. Here, \( \mathbf{X}^0 \) is the initial ensemble. Using the enKf approximations (13) and (14) we write the ensemble Kalman gain as follows:

\[
K_E := \mathbf{C}_E H^T (H\mathbf{C}_E H^T + \mathbf{R})^{-1},
\]

so when a new observation \( \mathbf{y}^{k+1} \) is available we can form samples of the posterior distribution as follows:

\[
\mathbf{X}^{k+1}_i := \mathbf{X}^k_i + K_E(\mathbf{y}^{k+1} - H\mathbf{X}^k_i), \quad i = 1, \ldots, N_E. \quad (17)
\]

Note that (17) parallelizes across the ensemble members. The trade-off here is that we must evolve a system of equations. The enKf update is in fact also a nonlinear system owing to the dependence of the Kalman gain on the ensemble itself via the sample statistics, breaking with the Gaussian assumption of the ensemble. In the limit of large ensembles the enKf can be shown to converge to the Kalman filter [18, 20].

While the standard enKf is typically used as an inference tool it has also been proposed as a way to solve general class inverse problems [23, 13, 12, 2, 27], supported by a wide array of numerical evidence in particular for data assimilation in atmospheric science (see e.g. [28] and its bibliography). We briefly outline the steps of an iterative enKf method for a Bayesian inverse problem roughly on the form given \( \mathbf{y} \), find the \( \mathbf{x} \) such that \( \mathbf{y} \approx H[\mathbf{x}] \), where \( H \) now is a specified uncertainty-to-state operator. The key in this approach is that the term \( \mathbf{y} - H\mathbf{X}_i \) in (17) now represents the misfit that we want to minimize (modulo possible added noise).

- Denote by \( \mathbf{X}^0 = \{X^0_i\}_{i=1}^{N_E} \) the initial ensemble.
- For \( k = 0, \ldots, n - 1 \):
1. Propagate the ensemble through the uncertainty-to-state operator:

\[ Y^k := \{ y_i^k \}_{i=1}^{N_E} = \{ H[X^k_i] + \eta_i \}_{i=1}^{N_E}, \quad \eta_i \in \mathcal{N}(0, R), \quad i = 1, \ldots, N_E. \] (18)

2. Verify convergence of \( y - \bar{Y}^k \), otherwise proceed.
3. Compute \( K_E \) using \( X^k \) and \( Y^k \) in (13) and (14).
4. Update the ensemble:

\[ X^{k+1}_i = X^k_i + K_E(y - Y^k), \]

where these operations are understood element-wise across the ensemble.

Under certain assumptions on the linearity of \( H \) and the covariance operators \( C \) and \( R \) it can be shown [13, Section 2.6] that the iterative enKf approximates, without the use of derivatives, a solution to the Tikhonov-Philips [34] regularised functional:

\[ x \rightarrow \| y - Hx \|_Y^2 + \| x - \bar{x} \|_X^2, \]

where in this context \( \bar{x} \) is the average of \( x \).

Next we show that the enKf provides a massively parallel and derivative-free method for applications in shape analysis. We demonstrate the utility of this algorithm in the next section where we show numerical evidence of convergence and accuracy for landmark matching problems.

3.2. Application to landmark matching. Ideally we want to solve the inverse problem of finding the initial momentum \( p_0 \) such that \( f_{q_0}(p_0) \) is close to a target \( q_1 \): \n
\[ p^*_0 := \arg \inf_{p_0} \| q_1 - f_{q_0}(p_0) \|_Q^2. \] (19)

While this can be done by traditional shooting approaches, the aim here is to present a derivative-free and parallelizable method. We emphasize that while \( f_{q_0} \) does involve the derivatives by the nature of (8) and (9) stemming from the Lagrangian associated with (3), the optimization of (19) is carried out without having to compute derivatives of \( f_{q_0} \) itself. Indeed, we will in fact not be solving (19), but rather a version of it where we optimize over an ensemble of initial momenta which, formally speaking, can be considered as a collection of different samples \( p_0 \) of the prior (defined over \( Q \)) that each map the template to different target locations. The likelihood is provided by the forward operator as it gives us information about how suitable a candidate \( p_0 \) is in the sense of minimising the misfit between the proposed and the desired target. Note that the energy term in the LDDMM functional is conserved along geodesics and can be rewritten as a quadratic form on the initial momentum \( p_0 \) cf. (10). This can be interpreted as the logarithm of the probability density for a prior Gaussian distribution on the momentum and provides the link between the LDDMM and the Bayesian formulation.

We will be looking at a deterministic problem and therefore assume that the “measurements” provided by \( f_{q_0} \) are not noisy (this corresponds to eliminating \( \eta_i \) in (18)), since our aim is to determine if the enKf is applicable to the highly nonlinear problem at hand. In other words, we wish to demonstrate that the enKf can be used to solve the inverse problem (19) without relying on adjoint differentiation software. This is an important point as the method here only uses evaluations of \( f_{q_0} \) which is useful when it is difficult or impossible to evaluate its derivatives (e.g., a black-box implementation). We shall answer this in the affirmative to motivate further research into diffeomorphic matching of other mathematical objects such as images, densities, etc. where a parallel paradigm could be useful. Our aim is
then to use the enKf to generate a sequence of ensembles such that the average of
the forward map applied to each of its elements converges to the desired target \( q_1 \).
Before we make these statements precise we introduce some notation:

- The space \( Q_E \) of ensemble momenta is defined as the space whose elements
  are collections of \( N_E \) elements of \( Q \). Since the enKf is an iterative method
  we let \( P^k \) denote the ensemble of momenta at iteration \( k \) of the filter. The
  superscript \( j \) in e.g. \( P^k \) refers to the \( j \)th member of the ensemble \( P^k \), whereas the
  subscript \( i \) in e.g. \( P^k \) refers to the \( i \)th \( d \)-dimensional entry of \( P^k \). The objects
  \( P^k \) represent the initial momentum previously referred to as \( p_0 \) where the subscript
  denoted time but in the ensemble setting we prefer this notation. \( \bar{P}^k \) denotes
  the average across the ensemble members of \( P^k \) defined for each landmark component
  \( i \) by:
  \[
  \bar{P}^k_i = \frac{1}{N_E} \sum_{j=1}^{N_E} P^k_{i,j}.
  \]

- We now define the ensemble forward map \( F : Q_E \rightarrow Q \) via (11) by:
  \[
  F_{q_0} [P^k] := \frac{1}{N_E} \sum_{j=1}^{N_E} f_{q_0} [P^k_{i,j}].
  \]
  This maps each ensemble member pair to a configuration of destination landmarks
  and then averaging each landmark across the ensemble. In other words, the forward
  operator is mapped over the elements of the ensemble and then we take an average in
  the space \( Q \).

- We define the Kalman update operator at iteration \( k \), \( K^k_P : Q \rightarrow Q \) be defined
  by:
  \[
  K^k_P = \text{Cov}^k_{PQ} (\text{Cov}^k_{QQ} + \xi \Lambda)^{-1},
  \]
  where \( \xi > 0 \) is determined later and \( \Lambda \in \mathbb{R}^{M \times M} \) is the identity matrix
  and the actions of the covariance matrices are given by:
  \[
  \text{Cov}^k_{QQ} [\cdot] = \frac{1}{N_E - 1} \sum_{j=1}^{N_E} \langle (f_{q_0} [P^k_{i,j}] - F_{q_0} [P^k]) \rangle \langle (f_{q_0} [P^k] - F(P^k), \cdot) \rangle_Q,
  \]
  \[
  \text{Cov}^k_{PQ} [\cdot] = \frac{1}{N_E - 1} \sum_{j=1}^{N_E} \langle (P^k_{i,j} - \bar{P}^k) \rangle \langle (f_{q_0} [P^k_{i,j}] - F_{q_0} [P^k], \cdot) \rangle_Q.
  \]

Algorithm 1 describes the enKf applied to shape matching and takes a similar
form to the algorithm described in [12, Section 2.2] for some number of landmarks
\( M \), \( P^0 \) denotes the initial ensemble, \( q_0 \) and \( q_1 \) the template and target we wish
to match, \( n \) is a maximum number of Kalman iterations and \( \epsilon \) is a predefined
error tolerance. In practice this error tolerance depends on the noise level of the
measurements of \( q_1 \) (if these came from some instruments, for instance), but in
this paper we work only with synthetic data so we leave the investigation of a more
sophisticated early termination criteria as future work. An interesting observation
we draw from Algorithm 1 is that although the Kalman gain is inherently nonlinear,
the momentum ensemble at iteration $k \geq 1$ is a linear combination of the ensemble at step $k$. The choice of initial ensemble is therefore important as the enKf seeks the best approximation to the target in the space of shapes spanned by applying the forward operator to the ensemble momenta.

**Algorithm 1** Ensemble Kalman Filter for diffeomorphic shape matching

1: procedure enKfDiffeo($P^0, q_0, q_1, n, \epsilon, N_E, T, \xi, \tau$)
2:  \hspace{1em} $k \leftarrow 0$
3:  while $k < n$ do
4:    \hspace{1em} if $\|q_1 - F_{q_0}[P^k]\|_Q \leq \epsilon$ then return $P^k$
5:      \hspace{1em} else
6:        \hspace{2em} for $j \leftarrow 1, \ldots, N_E$ do
7:          \hspace{3em} $P_{k+1,j} \leftarrow P_{k,j} + K_p^k (q_1 - F_{q_0}[P_{k,j}])$
8:        \hspace{2em} end for
9:      \hspace{1em} $k \leftarrow k + 1$
10:  \hspace{1em} end if
11: end while
12: return $P^k$
13: end procedure

While the outer loop (lines 3-11) of Algorithm 1 must be performed sequentially, the inner loop (lines 6-8) can to some extent be performed in parallel. We now elaborate on the intermediate steps that have been elided for brevity. It is clear that the operator $F_{q_0}[P^k]$ is constructed from $N_E$ solutions to the system (7)-(9) for each ensemble member of $P^k$, at each $k \geq 0$. The kernel operations in (7) are computationally intensive, so to obtain an efficient implementation our code (cf. Appendix (A)) uses multiprocessing to parallelize the forward map across the ensemble members. The operators (22) and (23) are then constructed sequentially after which we carry out the actual Kalman update in line 7 in parallel. This optimization is easy to implement, widely available in consumer hardware and essential to a practical method. A fully sequential implementation is prohibitively slow.

4. **Numerical examples.** In this section we show some numerical experiments using Algorithm 1. We generate, for $M \in \{10, 50, 150\}$, synthetic targets by sampling normally distributed initial momenta $p_0 \sim \mathcal{N}(0, 1)^{d \times M}$ and applying the forward map $f$ defined in (11) to generate template-target configurations such as those presented in Figure 2. For convenience we always sample the template shape from the unit circle. The objective in this section is two-fold; first we study the effect of the regularization parameter $\xi$ in section 4.1 and in section 4.2 the interplay between $N_E$ (the size of the ensemble) and $M$ (the number of landmarks).

4.1. **The effect of regularization.** We first investigate how the scalar $\xi$ in (21) affects the performance of Algorithm 1. First we define the data misfit $E^k$ at iteration $k$ as:

$$E^k := \|q_1 - F_{q_0}[P^k]\|_Q^2.$$  

(24)

The components of our initial ensembles $P^0$ are sampled from the uniform distribution:

$$P^{0,j} \sim U[-1, 1]^{d \times M}, \quad j = 1, \ldots, N_E.$$  

(25)
We keep the remaining parameters fixed in the experiments shown throughout this paper, see Table 1. As we are investigating the convergence of the filter we use a very low error tolerance. For simplicity, the landmark size \( \tau \) and the parameter \( T \) used in the Euler discretizations of (8) and (9) are fixed throughout our experiments. This choice of \( \tau \) implies some interaction between the landmarks making the matching harder as the geodesic paths of e.g. neighbouring landmarks must not get too close but also gives a more intuitive matching for \( t \in (0, 1) \). If set too low, the landmarks do not have enough interaction resulting in unnatural intermediate shapes. The discretization parameter \( T = 15 \) was chosen so as to keep the computational cost manageable on standard consumer hardware while allowing large deformations where the evolution of the shape can clearly be visualized such as in Figure 1. While not a focus here, this is often of interest in practice.

Table 1. Global parameters used for Algorithm 1.

| Variable | Value | Description |
|----------|-------|-------------|
| \( n \)  | 50    | Kalman iterations |
| \( T \)  | 15    | time steps |
| \( \tau \) | 1     | landmark size (cf. (2)) |
| \( \epsilon \) | 1e-05 | absolute error tolerance |
For $M = N_E = 50$ Figure 3 shows the log data misfits for three different targets (shown later on in Figure 5) using different values of $\xi$. There is a clear trade-off between the rate of convergence and overfitting. As these figures show, a high value of $\xi$ smooths the convergence of the filter at the cost of curtailing the rate. This is expected as the covariance $\text{Cov}_{QQ}^k$ plays a smaller role in the term $[\text{Cov}_{QQ}^k + \xi \Lambda]^{-1}$ in (21) as the value of $\xi$ increases. In other words, autocorrelations are valued higher in the filter than (possibly) spurious modes that may have a higher impact on the Kalman gain when $\xi$ is small. We find that some regularization $\xi$ is necessary in the later iterations of the filter, while setting $\xi$ too high impedes convergence in the initial iterations.

4.2. Landmark versus ensemble size. Now we look at how Algorithm 1 performs for each value of $M$ above for different ensemble sizes $N_E \in \{10, 50, 100\}$ when provided with random ensembles whose members are normally distributed cf. (25). We let $\xi = 1$ but otherwise use the parameter values described in Table 1.

We first present applications of Algorithm 1 to synthetic template-target pairs such as those presented in Figure 2 for the $(M, N_E) \in \{(10, 10), (50, 50), (150, 100)\}$ in Figures 4, 5 and 6, respectively, at various iterations. Note that again we linearly interpolate between landmarks when plotting these figures. We ran these on a 2014 MacBook Pro with a 2.5GHz Intel Core i7 processor and 16GB of 1600MHz DDR3 RAM. We use the 4 physical cores available on this platform to speed up the algorithm. While a process is spawned for each ensemble member, the actual
orchestration of the computations is handled by Python’s multiprocessing module. Good matches are obtained in each case. While these figures only show the geodesics for one particular realisation of (25), we also have evidence that Algorithm 1 shows some robustness with respect to the choice of initial momentum. Indeed, for \( M = 10 \), Figure 7 shows convergence of \( E^k \) as a function of the Kalman iteration \( k \) for three different target configurations (each row corresponds to the targets in Figure 4) and values of \( N_E = 10, 50, 100 \) (corresponding to each column) for 20 different draws of random initial ensembles distributed according to (25). Figures 8 and 9 show the same information where \( M = 50 \) and \( M = 150 \) for the landmark configurations in Figures 5 and 6, respectively. We observe some similarities and differences in convergence depending on the choice of \( M \). Overall we observe rapid convergence in the first 10 or so iterations, with stagnation in the residual \( E^k \) obtained around \( k \geq 30 \). We see an improvement in the smoothness of the convergence as the ensemble size increases (the ensemble can simply span a larger space), most notably seen in the \( M = 10 \) and \( M = 50 \) cases. As we increase the ensemble size for \( M = 150 \), we observe more consistent convergence across realisations of (25) which aligns with our expectations. In particular for \( M = 150 \) we observe some oscillations in the error in the early iterations of the filter. This can be attributed to spurious correlations between distant landmarks - we suggest ways in which this can be abated in section 5.

Letting \( P^f \) denote the truth that satisfies \( q_1 = f_{q_0}[P^f] \) we define the relative error \( R^k \) at iteration \( k \) of Algorithm 1 as:

\[
R^k := \| \bar{P}^k - P^f \|_Q / \| P^f \|_Q.
\]

Figures 10-12 show the relative error corresponding to the misfits in Figures 7-9. We observe a similar trend in each of these, namely that the relative error decreases as a function of \( k \) until a minimum is reached, after which it increases. This is a result of overfitting by attempting to drive (24) to zero and is aligned with the literature [12]. Future work includes exploring adaptive regularisation schemes where \( \alpha \) depends on \( k \) or early termination criteria such as that presented in the aforementioned reference. The relative error displays the same robustness with respect to the choice of initial ensemble as the misfits when the ensemble size is increased. As \( M \) increases so does the number of Kalman iterations needed to reach consensus, possibly owing to spurious correlations between landmarks or ensemble members, although this consensus appears reached in each of our experiments.

Tables 2-4 show the mean relative error at the final iteration across the experiments in Figure 10-12.

| Ensemble size \( N_E \) | 10          | 50          | 100         |
|--------------------------|-------------|-------------|-------------|
| Config                   |             |             |             |
| Config 1                 | 8.004e-01   | 7.899e-01   | 7.882e-01   |
| Config 2                 | 1.030e+00   | 9.968e-01   | 9.938e-01   |
| Config 3                 | 8.055e-01   | 7.821e-01   | 7.785e-01   |
LEARNING GEODESICS USING THE ENKF
Figure 4. Progression of Algorithm 1 for various targets using $M = 10$ and $N_E = 10$. Computation times for 50 iterations: 6s for each configuration.

Table 3. Relative error at the last iteration of algorithm 1 for different values of $N_E$ for fixed $M = 50$. The rows correspond to the configurations in Figure 5.

| Ensemble size $N_E$ | 10         | 50         | 100        |
|---------------------|------------|------------|------------|
| Config.             |            |            |            |
| 9.566e-01           | 9.358e-01  | 9.342e-01  |
| 9.618e-01           | 9.426e-01  | 9.408e-01  |
| 9.677e-01           | 9.416e-01  | 9.399e-01  |

Table 4. Relative error at the last iteration of algorithm 1 for different values of $N_E$ for fixed $M = 150$. The rows correspond to the configurations in Figure 6.

| Ensemble size $N_E$ | 10         | 50         | 100        |
|---------------------|------------|------------|------------|
| Config.             |            |            |            |
| 1.014e+00           | 9.885e-01  | 9.865e-01  |
| 1.022e+00           | 9.936e-01  | 9.914e-01  |
| 1.011e+00           | 9.923e-01  | 9.910e-01  |

5. Summary & outlook. In this paper we have presented a new robust approach to solving shape matching problems using a regularised derivative-free, massively parallel method. We have obtained high accuracy despite the global nature of Algorithm 1, paving the way for further investigation of Bayesian inversion techniques in the context of shape matching. The enKf we use here is agnostic to the forward model $f$ and can therefore be easily implemented alongside existing software packages or black box third-party implementations.

Several extensions are possible. First, since the enKf is based on covariance matrices we may want to introduce some control over the way landmarks should influence each other. For instance, landmarks that are distant in space should not necessarily have a significant impact on the Kalman gain. The notion of covariance localisation in enKf literature [10] provides a useful tool in this case. Localisation
LEARNING GEODESICS USING THE ENKF
means modifying the Kalman matrices by (Fröbenius) multiplication by a correlation matrix \( L \) whose entries take values in \([0,1]\) using e.g. a loosely defined rule: \( L_{ij} \approx 1 \) if the information represented at \( i \) is sufficiently close to affect the information at \( j \), and vice versa; otherwise \( L_{ij} \approx 0 \). Total localisation i.e. \( L_{ij} = \delta_{ij} \) where \( \delta \) is the Kronecker delta may be used in the first instance to tune the trade-off between convergence rate and stability. For our applications localisation makes intuitive sense; parts of the shape that are far from each other do not affect each other in the Kalman update.

Furthermore, a practical way of improving the accuracy of our matches is to implement a restart in analogy with e.g. the generalised minimal residual method [26] for linear systems. That is, if consensus has been reached in the ensemble for some tolerance i.e. at a certain iteration \( k \), the quantity:

\[
\| P_{k,j} - \bar{P}^k \|_Q,
\]

is below a certain threshold for all \( j = 1, \ldots, N_E \), we argue that no more information can be extracted from the ensemble. In such a case, a restart could be applied where we generate a whole new momentum ensemble generated from draws centered at the previous average momentum \( \bar{P}^k \). This will then give a new ensemble with new directions in which to search. While this provides a way of controlling the information coming from the momentum, an adaptive regularization strategy could also be investigated i.e. \( \xi \) becoming a function of \( k \).

Finally, since we are dealing with a discretized version of the forward problem future work includes quantifying the error in the Bayesian inverse problem via this discretization error drawing inspiration from the work from [5]. Future work also includes applying Algorithm 1 to real data and rigorously treating the Bayesian inversion problem. Filters such as the unscented Kalman filter [33] or machine learning approaches could be explored.
Figure 6. Progression of Algorithm 1 for various targets using $M = 150$ and $N_E = 100$. Computation times for 50 iterations (top to bottom): 5m22s, 5m23s, 5m23s.
Figure 7. Convergence of $E^k$ where $M = 10$. 
Figure 8. Convergence of $E^k$ where $M = 50$. 
Figure 9. Convergence of $E^k$ where $M = 150$. 
Figure 10. Evolution of the relative error $R^k$ corresponding to the misfits in Figure 7 where $M = 10$. 
Figure 11. Evolution of the relative error $R^k$ corresponding to the misfits in Figure 8 where $M = 50$. 
**Figure 12.** Evolution of the relative error $R^k$ corresponding to the misfits in Figure 9 where $M = 150$. 
Appendix A. Code. All the source code used to run the simulations presented in this paper is available from this repository [https://github.com/andreasbock/enkf_landmarks](https://github.com/andreasbock/enkf_landmarks). Consult the README.md in the repository for details on how to reproduce the experiments presented here.

REFERENCES

[1] M. F. Beg, M. I. Miller, A. Trouvé and L. Younes, Computing large deformation metric mappings via geodesic flows of diffeomorphisms, *Int. J. Comput. Vis.*, 61 (2005), 139–157.

[2] N. K. Chada, M. A. Iglesias, L. Roininen and A. M. Stuart Parameterizations for ensemble Kalman inversion, *Inverse Problems*, 34 (2018), 31pp.

[3] B. Charlier, J. Feydy, J. A. Glauçès, F.-D. Collin and G. Durif, Kernel operations on the GPU, with autodiff, without memory overflows, preprint, arXiv:2004.11127.

[4] C. J. Cotter, S. L. Cotter and F.-X. Vialard, Bayesian data assimilation in shape registration, *Inverse Problems*, 29 (2013), 21pp.

[5] S. L. Cotter, M. Dashti and A. M. Stuart, Approximation of Bayesian inverse problems for PDEs, *SIAM J. Numer. Anal.*, 48 (2010), 322–345.

[6] P. Dupuis, U. Grenander and M. I. Miller, Variational problems on flows of diffeomorphisms for image matching, *Quart. Appl. Math.*, 56 (1998), 587–600.

[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 (1994), 10143–10162.

[8] U. Grenander and M. I. Miller, Computational anatomy: An emerging discipline, *Quart. Appl. Math.*, 56 (1998), 617–694.

[9] U. Grenander and M. I. Miller, Representations of knowledge in complex systems, *J. Roy. Statist. Soc. Ser. B.*, 56 (1994), 549–603.

[10] S. J. Greybush, E. Kalnay, T. Miyoshi, K. Ide and B. R. Hunt, Balance and ensemble Kalman filter localization techniques, *Monthly Weather Review*, 139 (2011), 511–522.

[11] D. D. Holm, A. Trouvé and L. Younes, The Euler-Poincaré theory of metamorphosis, *Quart. Appl. Math.*, 67 (2009), 661–685.

[12] M. A. Iglesias, A regularizing iterative ensemble Kalman method for PDE-constrained inverse problems, *Inverse Problems*, 32 (2016), 35pp.

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

[14] S. C. Joshi and M. I. Miller, Landmark matching via large deformation diffeomorphisms, *IEEE Trans. Image Process.*, 9 (2000), 1357–1370.

[15] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Applied Mathematical Sciences, 160, Springer-Verlag, New York, 2005.

[16] R. E. Kalman, A new approach to linear filtering and prediction problems, *Trans. ASME Ser. D. J. Basic Engry.*, 82 (1960), 35–45.

[17] L. Kühnel, S. Sommer and A. Arnaudon, Differential geometry and stochastic dynamics with deep learning numerics, *Appl. Math. Comput.*, 356 (2019), 411–437.

[18] F. Le Gland, V. Monbet and V.-D. Tran, Large sample asymptotics for the ensemble Kalman filter, in *The Oxford Handbook of Nonlinear Filtering*, Oxford Univ. Press, Oxford, 2011, 598–631.

[19] J. Ma, M. I. Miller, A. Trouvé and L. Younes, Bayesian template estimation in computational anatomy, *NeuroImage*, 42 (2008), 252–261.

[20] J. Mandel, L. Cobb and J. D. Beezley, On the convergence of the ensemble Kalman filter, *Appl. Math.*, 56 (2011), 533–541.

[21] M. I. Miller, A. Trouvé and L. Younes, Geodesic shooting for computational anatomy, *J. Math. Imaging Vision*, 24 (2006), 209–228.

[22] D. Mumford, Pattern theory: The mathematics of perception, in *Proceedings of the International Congress of Mathematicians, Vol. I*, Higher Ed. Press, Beijing, 2002, 401–422.

[23] D. S. Oliver, A. C. Reynolds, N. Liu, *Inverse Theory for Petroleum Reservoir Characterization and History Matching*, Cambridge University Press, 2008.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer and J. Bradbury, et al., PyTorch: An imperative style, high-performance deep learning library, preprint, arXiv:1912.01703.

[25] S. Reich and C. Cotter, *Probabilistic Forecasting and Bayesian Data Assimilation*, Cambridge University Press, New York, 2015.
[26] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, *SIAM J. Sci. Statist. Comput.*, 7 (1986), 856–869.

[27] C. Schillings and A. M. Stuart, Analysis of the ensemble Kalman filter for inverse problems, *SIAM J. Numer. Anal.*, 55 (2017), 1264–1290.

[28] T. Schneider, S. Lan, A. Stuart and J. Teixeira, Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations, *Geophysical Research Letters*, 44 (2017), 12396–12417.

[29] A. M. Stuart, Inverse problems: A Bayesian perspective, *Acta Numer.*, 19 (2010), 451–559.

[30] A. Trouvé, Diffeomorphisms groups and pattern matching in image analysis, *Int. J. Comput. Vis.*, 28 (1998), 213–221.

[31] A. Trouvé, An infinite dimensional group approach for physics based models in pattern recognition, (1995).

[32] A. Trouvé and L. Younes, Metamorphoses through Lie group action, *Found. Comput. Math.*, 5 (2005), 173–198.

[33] R. Van Der Merwe, A. Doucet, N. De Freitas and E. A. Wan, The unscented particle filter, in *Advances in Neural Information Processing Systems*, 2000, 584–590.

[34] C. R. Vogel, *Computational Methods for Inverse Problems*, Frontiers in Applied Mathematics, 23, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.

[35] L. Younes, *Shapes and Diffeomorphisms*, Applied Mathematical Sciences, 171, Springer-Verlag, Berlin, 2010

Received March 2021; revised July 2021; early access August 2021.

E-mail address: me@andreasbock.dk
E-mail address: colin.cotter@imperial.ac.uk