Lagrangian Data Assimilation and Manifold Detection for a Point-Vortex Model

Project Proposal

Author:
David Darmon
AMSC
ddarmon (at) math.umd.edu

Advisor:
Dr. Kayo Ide
AOSC, IPST, CSCAMM, ESSIC
ide (at) umd.edu

Abstract

The process of assimilating data into geophysical models is of great practical importance. Classical approaches to this problem have considered the data from an *Eulerian* perspective, where the measurements of interest are flow velocities through fixed instruments. An alternative approach considers the data from a *Lagrangian* perspective, where the position of particles are tracked instead of the underlying flow field. The Lagrangian perspective also permits the application of tools from dynamical systems theory to the study of flows. However, very simple flow fields may lead to highly nonlinear particle trajectories. Thus, special care must be paid to the data assimilation methods applied. This project will apply Lagrangian data assimilation to a model point-vortex system using three assimilation schemes: the extended Kalman filter, the ensemble Kalman filter, and the particle filter. The effectiveness of these schemes at tracking the hidden state of the flow will be quantified. The project will also consider opportunities for observing system design (the optimization of observing systems through knowledge of the underlying dynamics of the observed system) by applying a methodology for detecting manifolds within the structure of the flow.

October 9, 2011

1 Background

1.1 Geophysical Flows and Lagrangian Transport

The motion of fluids across the Earth's surface impacts weather, climate, trade, and many other systems of practical interest and concern. Typically, the study of such flows has considered the velocity fields of the flows themselves (e.g. consider the famous Navier-Stokes equation, the solution of which is a flow velocity). The study of the velocity field is called the *Eulerian approach*. Recently, a complementary approach which explicitly considers the motion of particles within the flow has been developed, the *Lagrangian approach* [13]. This approach considers the motion of particles within the framework of dynamical systems theory. The analogy becomes obvious when we consider that dynamical systems theory studies the trajectories of state vectors in a velocity field 'induced' by differential equations. Tools from dynamical systems theory may be directly applied to the study of Lagrangian transport. The overarching goal of this project is to use a dynamical systems-informed approach to data assimilation to develop methods for observing Lagrangian flows.

1.2 Data Assimilation

Data assimilation is an iterative process for optimally determining the true state of an indirectly observed dynamical system. The process occurs in two stages. First, given some initial state, or a probability distribution on the initial state, a model for the system is integrated forward in time. This step is called *forecasting*. Second, at discrete times, observations (frequently indirect) of the system's state are made available. The goal is to combine these observations with the current prediction of the system's state from the forecast in some optimal way. This step is called *analysis* or *filtering*.

More formally, denote the state of our system by the n-vector $\mathbf{x}^t$. Then the evolution of this state forward in time is governed by a stochastic differential equation

$$d\mathbf{x}^t = M(\mathbf{x}^t, t)dt + d\boldsymbol{\eta}^t. \tag{1}$$

Here, $M(\cdot)$ is a possibly nonlinear operator on the state, and $\boldsymbol{\eta}^t$ is a Brownian motion with covariance matrix $\mathbf{Q}$. As the system evolves forward in time, we obtain noisy observations $\mathbf{y}_k^o \triangleq \mathbf{y}^o(t_k)$ of the system via

$$\mathbf{y}_k^o = h(\mathbf{x}^t(t_k)) + \boldsymbol{\epsilon}_k^t \tag{2}$$

where $h_k(\cdot)$ is a time-dependent, possibly nonlinear observation function of the systems true state and we take

$$\boldsymbol{\epsilon}^{\boldsymbol{t}} \sim \mathrm{N}(\mathbf{0}, \mathbf{R}^t). \tag{3}$$

That is, the observational noise is assumed to be multivariate normal with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{R}^t$.

Now that we have probabilistic models for the system dynamics and observation process, we may perform the forecasting and analysis stages of data assimilation. Let $\mathbf{x}^f(t_0)$ denote our prediction of the state at some initial time t_0. We wish to evolve this state forward in time. Typically this is done by either deterministic or stochastic numerical solutions of the (assumed) true dynamics (1) with an initial condition based on assumptions about the initial data (to start) or the analysis state (after observations). Once an observation becomes available, we use our observational model to update the state estimate. Through this iterative process of forecasting and analysis, we hope to devise a scheme to combine $\mathbf{x}^f(t_k)$ and $\mathbf{y}_k^o$ in an optimal way to obtain an analyzed state $\mathbf{x}^a(t_k)$ that, ideally, more closely matches the true state $\mathbf{x}^t(t_k)$ than either $\mathbf{x}^f(t_k)$ or $\mathbf{y}_k^o$ alone. This probabilistic approach also allows us to maintain information about the uncertainty in both $\mathbf{x}^f$ and $\mathbf{x}^a$. We will outline several such approaches in Section 2.1.

In the context of Lagrangian data assimilation, we consider $\mathbf{x}^t$ to be the concatenation of the state of the flow, which we will denote by $\mathbf{x}_F^t$, and the drifters, which we will denote by $\mathbf{x}_D^t$ [5]. Thus, the overall state of the system is given by

$$\mathbf{x}^t = \begin{pmatrix} \mathbf{x}_F^t \\ \mathbf{x}_D^f \end{pmatrix}. \tag{4}$$

We consider the drifters to be passive, and thus they do not affect the motion of the flow, and we may consider the flow dynamics to evolve according to (1), with $\mathbf{x}^t$ replaced by $\mathbf{x}_F^t$. Drifter are advected by the induced velocity field of the flow, and thus the evolution of their states is also governed by (1). A key feature of Lagrangian data assimilation is that the covariance matrix for $\mathbf{x}^t$ has the form

$$\mathbf{P}^t = E[(\mathbf{x}-\mathbf{x}^t)(\mathbf{x}-\mathbf{x}^t)^T] = E\left[\begin{array}{cc} (\mathbf{x}_F-\mathbf{x}_F^t)(\mathbf{x}_F-\mathbf{x}_F^t)^T & (\mathbf{x}_F-\mathbf{x}_F^t)(\mathbf{x}_D-\mathbf{x}_D^t)^T \\ (\mathbf{x}_D-\mathbf{x}_D^t)(\mathbf{x}_F-\mathbf{x}_F^t)^T & (\mathbf{x}_D-\mathbf{x}_D^t)(\mathbf{x}_D-\mathbf{x}_D^t)^T \end{array}\right]. \tag{5}$$

Thus, the covariance matrix captures correlations between the flow state and the drifters' states. Without these correlations, the instantaneous position of the drifters would offer no information concerning the state of the flow, even though their states is a direct consequence of the flow state.

1.3 Observing System Design

Up to this point, we have assumed that the initial positioning of the drifter governed by (9) was made arbitrarily. That is, we have not considered how the convergence of the state estimation might be optimized by the placement of the drifter. This is an important problem in oceanographic data assimilation, since measuring instruments are costly to deploy and obtaining measurements from such instruments requires great effort [1]. Under such circumstances, one might wish to choose an optimal placement of the drifter, given some constraints. We wish to design an optimal observing system, and thus this field of study is called observing system design. Early results have demonstrated that in the case of flow assimilation, Lagrangian methods are superior to Eulerian methods [7]. We wish to apply the tools of dynamical systems, especially the concept of a manifold, to optimally decide on the placement of our observing system, a passive drifter.

2 Approach

2.1 Phase I - Lagrangian Data Assimilation

2.1.1 Model System

For the system we wish to assimilate, we consider the deterministic point-vortex model with N_v vortices and N_d drifter [5]. The positions of the j^{th} vortex $\mathbf{z}_j = (x_j, y_j)$ and k^{th} drifter $\boldsymbol{\zeta}_k = (\xi_k, \eta_k)$ in the plane are governed by the system of ordinary differential equations

$$\frac{dx_j}{dt} = -\frac{1}{2\pi}\sum_{j'=1,j'\neq j}^{N_v} \frac{\Gamma_{j'}(y_j - y_{j'})}{l_{jj'}^2}, \qquad j = 1,\dots,N_v \tag{6}$$

$$\frac{dy_j}{dt} = \frac{1}{2\pi}\sum_{j'=1,j'\neq j}^{N_v} \frac{\Gamma_{j'}(x_j - x_{j'})}{l_{jj'}^2}, \qquad j = 1,\dots,N_v \tag{7}$$

$$\frac{d\xi_k}{dt} = -\frac{1}{2\pi}\sum_{j=1}^{N_v} \frac{\Gamma_j(\eta_k - y_j)}{l_{kj}^2}, \qquad k = 1,\dots,N_d \tag{8}$$

$$\frac{d\eta_k}{dt} = \frac{1}{2\pi}\sum_{j=1}^{N_v} \frac{\Gamma_j(\xi_k - x_j)}{l_{kj}^2}, \qquad k = 1,\dots,N_d \tag{9}$$

where l_{ij}^2 is the square of the Euclidean distance from a point at i to a point at j. We will focus on the case where the vorticities, Γ_j, are taken to be fixed and equal to 2π. We will also fix the initial conditions of the vortices at $z_j(0) = z_j$.

In the context of the data assimilation problem formulated above, we have that the evolution of our system is governed by the SDE

$$d\mathbf{x}^t = M(\mathbf{x}^t, t)dt + d\boldsymbol{\eta}^t \tag{10}$$

where

$$\mathbf{x}^t = \begin{pmatrix} \mathbf{z} \\ \boldsymbol{\zeta} \end{pmatrix} \tag{11}$$

and the deterministic evolution of the system, M, is given by the right-hand sides of (6)-(9).

2.1.2 Extended Kalman Filter

The data assimilation problem outlined in Section 1.2 with linear dynamics, linear observations, and Gaussian noise has an exact solution [6]. The stochastic process which solves (1) in this case is Gaussian, and can be fully characterized by its mean $\mathbf{x}^t$ and covariance matrix $\mathbf{P}^t$. By directly evolving both the mean and covariance matrix forward in time and performing a minimum mean-square estimate of $\mathbf{x}^a$, the optimal filter can be derived. This filter was first proposed by Rudolf Kalman, and as such is named the Kalman filter. An extension of the Kalman filter to deal with nonlinearities in both the dynamical equations and the observation equation is appropriately called the extended Kalman filter (EKF). This modification to the Kalman filter uses a tangent-linear model for the system dynamics and measurement operator, centered at the forecasted value $\mathbf{x}^f$. Thus, we denote the Jacobians of the evolution operator in (1) and the observation operator in (2) by

$$\mathbf{M}(t) = J[M(\mathbf{x}, t)]\big|_{\mathbf{x}=\mathbf{x}^f} \tag{12}$$

$$\mathbf{H}_k = J[h(\mathbf{x}, t_k)]\big|_{\mathbf{x}=\mathbf{x}^f}. \tag{13}$$

During the forecasting stage, we evolve forecasts for $\mathbf{x}^t$ and $\mathbf{P}^t$ forward in time by

$$\frac{d}{dt}\mathbf{x}^f = M(\mathbf{x}^f, t) \tag{14}$$

$$\frac{d}{dt}\mathbf{P}^f = \mathbf{M}(t)\mathbf{P}^f + \mathbf{P}^f\mathbf{M}^T(t) + \mathbf{Q}, \tag{15}$$

where the superscript T denotes a transpose. As an observation becomes available at time t_k, we assimilate this observation to obtain the analyzed estimates $\mathbf{x}^a$ and $\mathbf{P}^a$ using

$$\mathbf{x}^a_k = \mathbf{x}^f(t_k) + \mathbf{K}_k(\mathbf{y}^o_k - h_k(\mathbf{x}^f(t_k))) \tag{16}$$

$$\mathbf{P}^a_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^f(t_k) \tag{17}$$

where $\mathbf{K}_k$ is the Kalman gain and is given by

$$\mathbf{K}_k = \mathbf{P}^f(t_k)\mathbf{H}^T_k(\mathbf{H}_k\mathbf{P}^f(t_k)\mathbf{H}^T_k + \mathbf{R}^o_k)^{-1}. \tag{18}$$

This analysis estimate is then used as the initial condition for the forecast equations, and the next iterate of the data assimilation process begins.

The EKF has many deficiencies. As just described, it only weakly assumes nonlinearity in the governing dynamical equation (1) by using the tangent-linear model. It also assumes that the posterior estimate of the state is Gaussian: this is no longer guaranteed in the nonlinear case. Both of these violations of the assumptions of the Kalman filter result in a suboptimal filter that may diverge depending on the dynamics of the system under consideration.

2.1.3 Ensemble Kalman Filter

Another approach to data assimilation that addresses some weaknesses inherent in the EKF is the ensemble Kalman filter (EnKF) [3]. The EnKF represents the probability distribution function describing the state $\mathbf{x}^t$ by an *ensemble* of states $\{\mathbf{x}_i^f\}_{i=1}^N$. That is, we may use this ensemble to construct an empirical distribution function which may be used to compute (approximate) moments of interest.

Let $\bar{\mathbf{x}}^f$ be the ensemble average given by

$$\bar{\mathbf{x}}^f = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i^f. \tag{19}$$

Then the ensemble covariance matrix $\mathbf{P}_e^f$ may be computed by

$$\mathbf{P}_e^f = \frac{1}{N-1}\sum_{i=1}^N (\mathbf{x}_i^f - \bar{\mathbf{x}}^f)(\mathbf{x}_i^f - \bar{\mathbf{x}}^f)^T. \tag{20}$$

Thus, we may evolve the ensemble states forward in time using (1) and then compute the the ensemble average and covariance matrices. These now approximate the *true* mean value and covariance matrix, without any linearity assumptions, and the only approximation is in the ensemble size.

The analysis step requires care [15]. There are two main approaches. The *stochastic* approach generates the analysis ensemble by applying random perturbation to the observations (based on the presumed statistics of the observation process) and applying a Kalman-type update to the combined forecast and observation ensembles. The *square root filter* approach, which we will apply, takes advantage of properties shared by the forecast covariance matrix and the analysis covariance matrix to generate the new analysis ensemble deterministically.

While the EnKF has several advantages over the EKF, it still has several weaknesses. The forecasting stage preserves the full non-linearity and non-gaussianity (up to the ensemble approximation) of (1), but the analysis step still assumes that the posterior distribution of the state is Gaussian. As stated above, this is not necessarily the case for a nonlinear system. The EnKF also requires the evolution of ensembles forward in time: thus, whereas the EKF only requires the evolution of a single state mean and covariance matrix, the EnKF requires that we integrate N ensembles forward in time, where N must be relatively large to obtain good approximations to the mean and covariance matrix of the process. Finally, the EnKF still assumes a Gaussian posterior in the analysis step. While a Gaussian random vector is completely characterized by its mean and covariance matrix, this is not necessarily the case for an arbitrary random vector.

2.1.4 Particle Filter

The particle filter closely resembles the EnKF with the key advantage that it performs a full Bayesian update at the analysis step [11, 14]. Like the EnKF, the particle filter evolves an ensemble of states forward in time by using (1). Unlike the EnKF, at the analysis step, instead of generating a new analysis ensemble, the particle filter updates weights associated with each of the ensemble members by applying Bayes's rule. Recall that from (2), the observations are modeled as Gaussian random vectors. Thus, the likelihood of an observation is given by

$$p(\mathbf{y}_k^o|\mathbf{x}(t_k)) = C_1 \mathrm{exp}\left(-\frac{1}{2}(\mathbf{y}_k^o - h(\mathbf{x}(t_k)))^T(\mathbf{R}^o)^{-1}(\mathbf{y}_k^o - h(\mathbf{x}(t_k)))\right) \tag{21}$$

where C_1 is a normalization constant. Let $w_{i,k}$ be the weight of particle $\mathbf{x}_i^f$ at time t_k. At the start of data assimilation, we have no information about the system and thus initialize all the $w_{0,k}$ to equal values, ensuring that they sum to unity. That is, we assume a discrete uniform distribution. As an observation is made at time t_k, we update the weights by applying Bayes's theorem, given by (30). In this case, Bayes's theorem takes the form of

$$w_{i,k} = C_2 w_{i,k-1} p(\mathbf{y}_k^o|\mathbf{x}_i^f(t_k))) \tag{22}$$

where C_2 is again a normalization constant. At any analysis stage, we may recover the moments of the state of the system by applying the proper weighting of the states. For example, we recover the mean state by

$$\bar{\mathbf{x}}^f(t_k) = \sum_{i=1}^{N} w_{i,k}\mathbf{x}_i^f(t_k). \tag{23}$$

While the formulation presented above seems straightforward to implement, special attention must be paid to the weight of the particles. It is well known that in high dimensions, most of the weights become concentrated on a small number of particles, thus reducing the effective number of particles [2]. When the effective number of particles drops below a certain threshold, we wish to resample the particles to obtain an ensemble with a larger effective number of particles. This resampling step is key to the proper functioning of the particle filter. There are various methods for doing this. We will use residual resampling [8].

2.2 Phase II - Manifold Detection

Invariant manifolds play a key role in the description of Lagrangian flows, just as they do in the description of dynamical systems [13]. For flows, the streamlines of a system are invariant manifolds since a drifter trajectory starting on that streamline will remain on that streamline. This is similar to eigenpair solutions to a linear differential equation. If a trajectory begins on an eigenpair solution of the system, it must remain on that solution for all time. Similarly, due to the uniqueness of solutions, no trajectory may cross an invariant manifold. Thus, the invariant manifolds of a Lagrangian flow partition the flow into different types of behavior. By identifying these manifolds (and thus these partitions), we may be able to take advantage of the flow behavior within a given region to better perform data assimilation for the system.

We focus on the deterministic point-vortex system (6)-(9). We may imagine we have a drifter at each point on a grid in a defined region. In this case, we may define the following Lagrangian descriptor [9]

$$M(\mathbf{x}_D^t, t^*) = \int_{t^*-\tau}^{t^*+\tau} \left(\sum_{i=1}^{n} \left(\frac{dx_D^i(t)}{dt} \right)^2 \right)^{1/2} dt \tag{24}$$

where x_D^i represents the i^{th} position component of the drifter. For our system, the drifter moves on the plane, and thus we have $n = 2$. In this case, M measures the Euclidean arc length of the curve passing through $\mathbf{x}_D^t$ at time t. Clearly, M depends on our choice of τ and t^*. For a time independent flow (such as the one we consider), M provides a time-independent partition of the phase space of our system. Thus, we may consider $M(\mathbf{x}_D^t) \equiv M(\mathbf{x}_D^t, 0)$. This function may also be applied to time-dependent flows, in which case we must maintain the t^* dependency.

3 Implementation

All algorithms will initially be prototyped in MATLAB on a MacBook Pro with a 2.4 GHz Intel Core 2 Duo processor with 4 GB of RAM. The algorithms will initially be developed to run serially. As the year progresses (see Section 7 for more details), the code for evolving the particles in the ensemble Kalman filter and particle filter will be ported to run in parallel using MATLAB's Parallel Computing Toolbox. These modified versions will then be run on a computing cluster. Similarly, the manifold detection algorithm will initially be developed serially and then parallelized.

4 Databases

During Phase I, the databases will consist of archived numerical solutions to (1) with varied initial conditions for the positions of the drifters. These solutions will be generated using a stochastic Runge-Kutta fourth-order SDE solver with prescribed levels of dynamical noise [16]. Observational noise will be added to the locations of the drifters according to (2), again following prescribed levels of noise (see Section 5 for more details). These common databases will be used for all assimilation methods, allowing comparisons between and within methods.

Phase II does not require databases. The computation of M values is completely model dependent and does not require model-independent inputs.

5 Validation

5.1 Phase I - Lagrangian Data Assimilation

For validation of the Lagrangian data assimilation methods, we will consider a hierarchical approach. We will begin with the (physically unrealizable) condition that the true positions of the point-vortices are known for all time and estimate the positions of the drifters using the three data assimilation methods. This represents a 'best case' scenario and we expect the schemes to easily track the positions of the drifters under these conditions. Next, we will consider the case where the the point-vortices locations are unknown, but where both the dynamical noise in (1) and the observational noise in (2) are small relative to the motion of the particles. Finally, we will consider the case where both the dynamical and observational noise take on physically plausible values. In all cases, we may compare the true state to the assimilated state using the distance between the two as a function of time, given by the Euclidean norm of their difference

$$\delta^{a,f}(t) = ||\mathbf{x}^t(t) - \mathbf{x}^{a,f}(t)||_2. \tag{25}$$

In the first case (point-vortices locations known), we expect this distance to remain small. In the second case (state unknown, low noise), we expect this distance to remain small, though we expect the EKF to perform the worst. In the final case (state unknown, realistic noise), we expect to see greater filter divergence, though again with the EKF performing the worst.

Finally, for the particle filter and the extended Kalman filter, we will compare our results to a published study on data assimilation for the point-vortex model [14]. We expect the ensemble Kalman filter to perform better than the EKF but worse than the particle filter, due to its ability to track nonlinearities in the flow but failure to incorporate non-Gaussianity.

5.2 Phase II - Manifold Detection

Using the M function, we expect to obtain a good approximation to the analytically known stream function of the point-vortex model. For example, with two vortices, the stream function is given by

$$\psi(x,y) = \frac{1}{2}\log[(x-1)^2 + y^2] - \frac{1}{2}\log[(x+1)^2 + y^2] + \frac{1}{4}(x^2 + y^2) \tag{26}$$

where x and y are the real and imaginary parts of a ξ from Section 2.1.1 [14]. We may also compare the results from M to the results from computing finite-time Lyapunov exponents for the system [4]. As shown in [9], these results should be similar.

Finally, we may consider the motion of drifters near the manifolds of the system. We know that these drifters should not cross the manifold, providing a sanity check for the computation of the M function.

6 Testing

6.1 Phase I - Lagrangian Data Assimilation

Consider the distance metric defined by (25) where we restrict ourselves to only the vortices' locations. Then as in [14], we may consider a *failure statistic*

$$\hat{f}_{\delta_{div},n}(t) = \text{ fraction of times } \delta(t) > \delta_{div} \text{ at time } t \text{ in } n \text{ trials} \tag{27}$$

where we define a failure distance δ_{div} beyond which we consider the filter to have diverged. Thus, by using a common sample of n realizations of the system described by (1), we may obtain an estimate for the relative performance of the different data assimilation methods over the time course of data assimilation.

6.2 Phase II - Manifold Detection

Once we have successfully detected manifolds using the M function, we will be able to investigate how the placement of drifters with respect to the manifolds affects the accuracy of data assimilation [12]. We will characterize the performance using our knowledge of the true trajectory by computing the distance between the true and assimilated states and $\hat{f}_{\delta_{div},n}$. Thus, we can compare the performance for 'arbitrary' placement of drifters to those placed using approximate knowledge of the flow's stream function.

7 Project Schedule and Milestones

The tentative project schedule is as follows:

- Phase I
  - Produce database: now through mid-October
  - Develop extended Kalman Filter: now through mid-October
  - Develop ensemble Kalman Filter: mid-October through mid-November
  - Develop particle filter: mid-November through end of January
  - Validation and testing of three filters (serial): Beginning in mid-October, complete by February
  - Parallelize ensemble methods: mid-January through March
- Phase II
  - Develop serial code for manifold detection: mid-January through mid-February
  - Validate and test manifold detection: mid-February through mid-March
  - Parallelize manifold detection algorithm: mid-March through mid-April

The corresponding milestones are the following:

- Phase I
  - Complete validation and testing of extended Kalman filter: beginning of November
  - Complete validation and testing of (serial) ensemble Kalman filter: beginning of December
  - Complete validation and testing of (serial) particle filter: end of January
- Phase II
  - Complete validation and testing of (serial) manifold detection: mid-March
  - Parallelize ensemble methods: beginning of April
  - Parallelize manifold detection algorithm: end of April

8 Deliverables

The deliverables for this project will include: the collection of databases used for the filter validation and testing, a suite of software for performing EKF, EnKF, and particle filtering on the stochastic point-vortex model, and a suite of software for performing manifold detection on the deterministic point-vortex model. The EnKF, particle filter, and ensemble Kalman filter, as well as the computation of M, will all be parallelized.

References

[1] J. Baehr, D. McInerney, K. Keller, and J. Marotzke. Optimization of an observing system design for the north atlantic meridional overturning circulation. *Journal of Atmospheric and Oceanic Technology*, 25(4):625, 2008.

[2] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[3] G. Evensen. *Data assimilation: the ensemble Kalman filter*. Springer Verlag, 2009.

[4] G. Haller. Finding finite-time invariant manifolds in two-dimensional velocity fields. *Chaos*, 10(1):99–108, 2000.

[5] K. Ide, L. Kuznetsov, and C.K.R.T. Jone. Lagrangian data assimilation for point vortex systems. *Journal of Turbulence*, (3), 2002.

[6] A.H. Jazwinski. *Stochastic processes and filtering theory*. Dover Publications, 2007.

[7] A.J. Krener and K. Ide. Measures of unobservability. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 6401–6406. IEEE, 2009.

[8] J.S. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, pages 1032–1044, 1998.

[9] C. Mendoza and A.M. Mancho. Hidden geometry of ocean flows. *Physical review letters*, 105(3):38501, 2010.

[10] H. Salman. A hybrid grid/particle filter for lagrangian data assimilation. i: Formulating the passive scalar approximation. *Quarterly Journal of the Royal Meteorological Society*, 134(635):1539–1550, 2008.

[11] H. Salman. A hybrid grid/particle filter for lagrangian data assimilation. ii: Application to a model vortex flow. *Quarterly Journal of the Royal Meteorological Society*, 134(635):1551–1565, 2008.

[12] H. Salman, K. Ide, and C. Jones. Using flow geometry for drifter deployment in lagrangian data assimilation. *Tellus A*, 60(2):321–335, 2008.

[13] R.M. Samelson and S. Wiggins. *Lagrangian transport in geophysical jets and waves: the dynamical systems approach*. Springer Verlag, 2006.

[14] E.T. Spiller, A. Budhiraja, K. Ide, and C.K.R.T. Jones. Modified particle filter methods for assimilating lagrangian data into a point-vortex model. *Physica D: Nonlinear Phenomena*, 237(10-12):1498–1506, 2008.

[15] M.K. Tippett, J.L. Anderson, C.H. Bishop, T.M. Hamill, and J.S. Whitaker. Ensemble square root filters. *Monthly Weather Review*, 131, 2003.

[16] J. Wilkie. Numerical methods for stochastic differential equations. *Physical Review E*, 70(1):017701, 2004.

A Overview of the Idealized Data Assimilation Problem

Let $p \equiv p(\mathbf{x}^t, t)$ be the probability density of the state $\mathbf{x}^t$ described by (1). Then the Fokker-Planck equation for p is

$$\frac{\partial p}{\partial t} + \sum_i \frac{\partial m_i p}{\partial x_i} = \frac{1}{2} \sum_{i,j} \frac{\partial^2 p(\mathbf{Q})_{ij}}{\partial x_i \partial x_j} \tag{28}$$

where m_i is the i^{th} component of the evolution operator $M(\cdot)$ from (1) and $\mathbf{Q}$ is the covariance matrix characterizing $\boldsymbol{\eta}^t$ in (1) [10]. Given p, we 'know everything' about $\mathbf{x}^t$ in the sense that we can compute all possible moments of the distribution, which we may then use in the analysis step of data assimilation. However, tractable solutions to (28) rarely exist. Thus, the following methods (extended Kalman filter, ensemble Kalman filter, and particle filter) represent attempts to approximate p by applying various assumptions on the form of p that make the solution to (28) tractable.

When an observation arrives at time t_k, we wish to perform the analysis step. This step allows us to determine the posterior estimate of the state given the current observation, $p(\mathbf{x}^f(t_k)|\mathbf{y}_k^o)$, by applying Bayes's rule to combine the likelihood of the data $p(\mathbf{y}_k^o|\mathbf{x}^f(t_k))$ and the prior $p(\mathbf{x}^f(t_k))$. We drop the t_k dependence for brevity of presentation. Thus, the analysis estimate is given by

$$p(\mathbf{x}^f|\mathbf{y}_k^o) = \frac{p(\mathbf{x}^f, \mathbf{y}_k^o)}{p(\mathbf{y}_k^o)} = \frac{p(\mathbf{y}_k^o|\mathbf{x}^f)p(\mathbf{x}^f)}{p(\mathbf{y}_k^o)} \tag{29}$$

$$= \frac{p(\mathbf{y}_k^o|\mathbf{x}^f)p(\mathbf{x}^f)}{\int p(\mathbf{y}_k^o|\mathbf{x}^f)p(\mathbf{x}^f)\,d\mathbf{x}^f}. \tag{30}$$

As with the forecasting step, the application of Bayes's rule is an idealization of the analysis stage. As we will see, several assumptions may be made about the distribution of the prior and likelihood that simplify (30).