

Uncertainty Quantification and Sensitivity Analysis in Dynamical Systems

Matthew Rocklin

Abstract

Dynamical systems describe the time evolution of a point in a space of potential states. If knowledge about the initial point is uncertain then the dynamical system evolves a probability distribution forward rather than a single point. Tracking this distribution is challenging and approximate methods are often necessary.

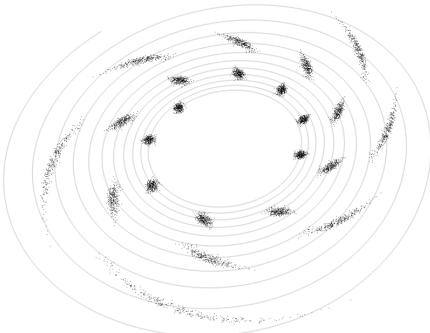
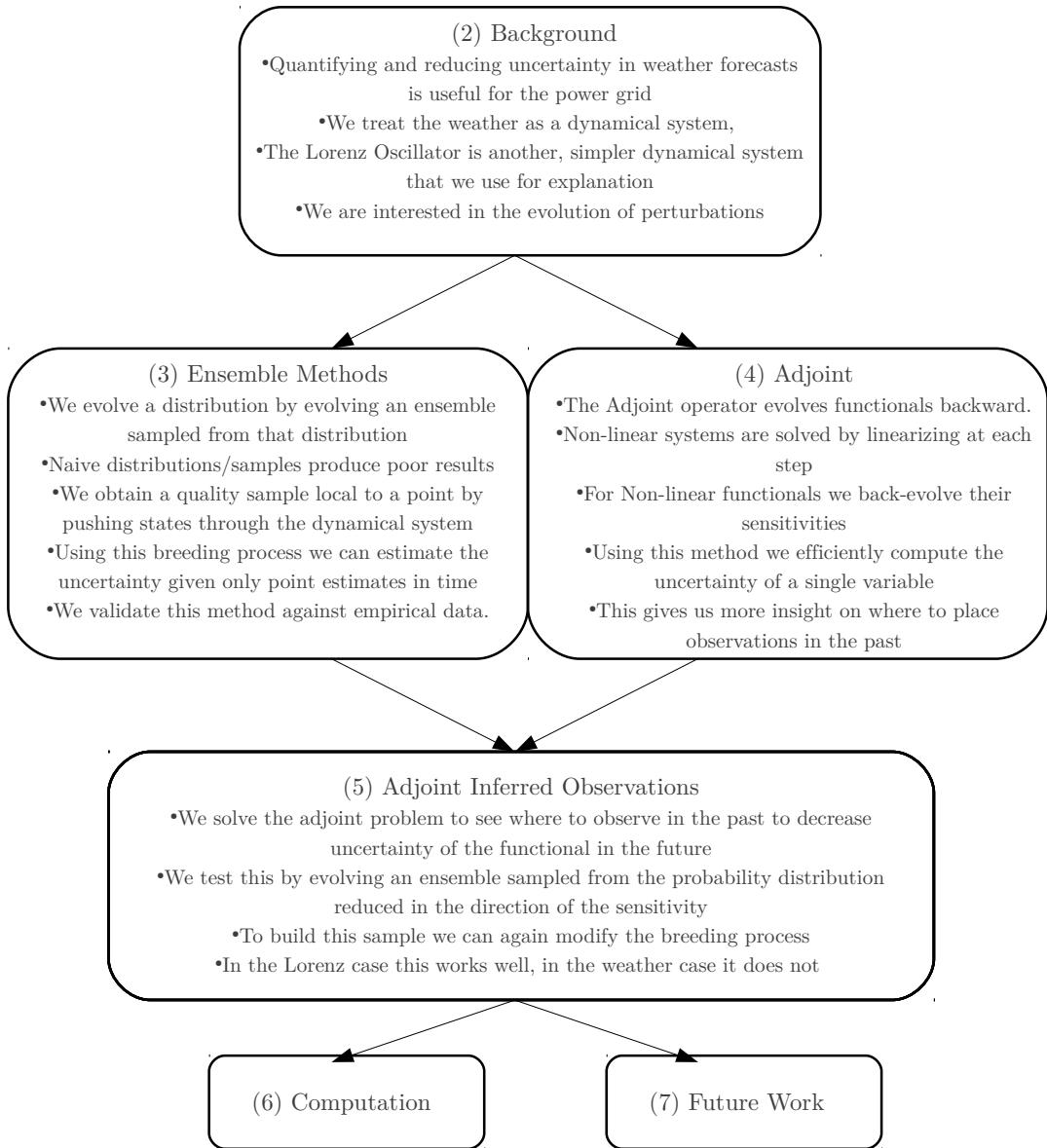
We study uncertainty quantification of short-term forecasts in dynamical systems. We compare sampling (Monte Carlo) against adjoint methods on low and high dimensional systems. For sampling we particularly consider the formation of sensible initial distributions, so-called "errors of the day", in a low-rank attractor given model dynamics and expectation values distributed in time. Adjoint methods are shown to be a cheaper alternative when close to the true solution which can also be used to pinpoint optimal observations to reduce targeted uncertainties in the final distribution.

For a motivating example we consider uncertainty in windspeed of short-term forecasts to aid with advance management of resources on the power grid.

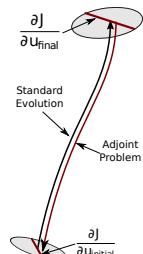
Contents

1	Introduction	5
2	General Background	6
2.1	Motivation: Unit Commitment in Power Systems	6
2.2	Numerical Weather Prediction Models - WRF and WRFVAR	6
2.3	Dynamical Systems	7
2.4	Modeling Uncertainty	8
2.5	The Lorenz Oscillator - A Visually Motivating Example	8
2.5.1	Definition	9
2.5.2	Local Behavior - Perturbations	9
3	Ensemble Methods for Uncertainty Quantification	12
3.1	A Naive Solution	12
3.2	Singular, Lyapunov, and Bred Vectors	14
3.2.1	Singular Vectors	14
3.2.2	Lyapunov Vectors	14
3.2.3	Bred Vectors	15
3.3	Uncertainty Quantification - Obtaining Estimates	17
3.4	Validation	18
4	Adjoint Methods for Sensitivity Analysis	20
4.1	Definition	20
4.2	Non-Linear Dynamics: Local Linear and Adjoint operators	21
4.3	Non-Linear Functionals: Sensitivities as Adjoint Variables	21
4.4	Uncertainty Quantification of a Single Functional	22
4.5	Application	22
4.6	Repeatability	24
5	Quantifying the Effects of Adjoint-Informed Observations	26
5.1	Introducing Local Conditions to Bred Vector Ensembles	26
5.2	A Test	26
5.3	Executing this test on the Lorenz Oscillator	27
5.4	Transitioning to the Full Weather System	27
6	Distributed Computation	31
7	Conclusion and Future Work	32
8	Personal Results	33

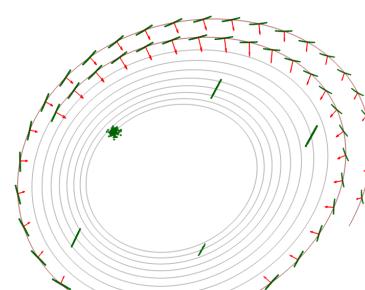
The Paper at a Glance



(a) Ensemble Uncertainty Quantification



(b) Adjoint Uncertainty Quantification



(c) Ensemble-with-Observation co-evolving with Adjoint Sensitivity

1 Introduction

How can we evolve a probability distribution through a dynamical system instead of a single state. I.e. given uncertainty in initial conditions how does this uncertainty propagate through time?

This question is of particular interest in short term weather forecasting. An accurate forecast of the weather is both valuable and challenging. Lorenz[1] showed in 1963 that the forecast of the weather was sensitive to perturbations in initial conditions. That is if there is uncertainty about the current weather the uncertainty of forecasts grows exponentially with forecast time.

The problem of weather forecasting is currently the domain of numerical weather prediction and is not the subject of this paper. Instead of improving forecasts we interest ourselves with the quantifying how uncertain these forecasts are due to the uncertainty in initial conditions. We approach this topic of uncertainty quantification through the lens of dynamical systems and ensembles.

We often care about uncertainty in some focused aspect of the forecast (i.e. how likely is it to rain in Chicago?) rather than the entire probability distribution at the forecast time. Because of this restriction our problem can be made much simpler through adjoint methods.

In section 2 we discuss the motivation and background on this topic. In section 3 we apply ensemble techniques to this problem and demonstrate their efficacy. In section 4 we present adjoint methods and show how they can be used to compute specific uncertainties with greater efficiency. In section 5 we test adjoint methods against ensemble methods in a specific problem to discover if they agree.

2 General Background

2.1 Motivation: Unit Commitment in Power Systems

To motivate the relevance of our work we consider weather-dependent elements on the power grid (wind, solar, air conditioning).

A careful balance between energy generation and energy consumption must be maintained for the day to day stability of the power grid. Too much or too little power can result in destruction of infrastructure and costly power loss to the general public.

As we introduce weather variable sources and sinks such as renewables (wind, solar) and air conditioning units we find ourselves at the mercy of the weather, an infamously unpredictable system. While cheap renewable energy is appreciated it must be carefully supplemented with a precise amount of traditionally controlled power to exactly match the expected consumption. For engineering reasons this amount must be decided well ahead of time and so we are tasked with predicting how much power weather-dependent units are likely to produce or consume so that we can adjust accordingly. This must be computed roughly twelve to twenty-four hours in advance.[2]

In addition to estimating wind or sun at a specific location we also need to know how certain we are of this estimate in order to develop optimal control strategies across the grid.

- 1: Motivating Problem: How can we quantify 12-hour Uncertainty of the weather at a specific location?

If we can estimate this uncertainty and find that it is too large how can we use targeted observations to reduce uncertainty?

- 2: Motivating Problem: Where should we observe to improve 12 hour forecasts of a specific location?

In general we interest ourselves in uncertainty and sensitivity in *targeted regions*. These regions are those that will hold wind farms, solar panels, etc... in a future power grid.

2.2 Numerical Weather Prediction Models - WRF and WRFVAR

The Weather Research and Forecasting Model (WRF) is a state of the art numerical weather prediction code developed by the National Center for Atmospheric Research (NCAR), NOAA, and the academy in general. This paper uses Advanced Research WRF branch which represents the best available model for weather evolution [3].

For our purposes the credentials and details of the internals of WRF are unimportant. This code is a complex work written by domain scientists in thousands of pages of Fortran. It is our desire to imbue the computations of this model with uncertainty information with as little contact with the original code as possible. For sampling this will simply involve running the code several times. For adjoint methods we will use automatic differentiation.

Automatic differentiation methods take computer codes and generate programs to compute the derivatives of these codes. While this process is ideally automated very complex codes such as WRF can prove difficult. WRFVAR is such a derivative code that is the result of this process on a simplified Adiabatic WRF model. We will use the derivative of this simplified model as proxy to the derivative of the full model, a decision supported by [4].

2.3 Dynamical Systems

We define a *dynamical system* as a manifold of states, $\mathbf{x} \in M$, and a fixed time-evolution rule on those states, f . This rule can be either continuous, in which case we have an ordinary differential equation or discrete, in which case we have a map from the state-space to itself.

$$\begin{aligned}\dot{\mathbf{x}} &= f(\mathbf{x}) && \text{Continuous} \\ \mathbf{x}_{i+1} &= \mathbf{x}_i + f(\mathbf{x}_i) && \text{Discrete}\end{aligned}$$

We obtain a *trajectory* of a start-state by evolving that state forward in time and recording its path. In the continuous and discrete cases we define

$$\begin{aligned}\mathbf{x}(t) &= \mathbf{x}(0) + \int_0^t f(\mathbf{x}(t')) dt' && \text{Continuous} \\ \mathbf{x}_t &= \mathbf{x}_0 + \sum_{i=0}^t f(\mathbf{x}_i) && \text{Discrete}\end{aligned}$$

In this work we are primarily concerned with the evolution of *perturbations* of states $\delta\mathbf{x}$. Because our manifold is everywhere described by the same \mathbb{R}^n we can think of either perturbed states ($\mathbf{x} + \delta\mathbf{x}$) living in the same state space as \mathbf{x} or of perturbations ($\delta\mathbf{x}$) living in spaces (T_x) tangent to points along the trajectory. These two options are pictorially represented in Fig. 1 on the left and right respectively.

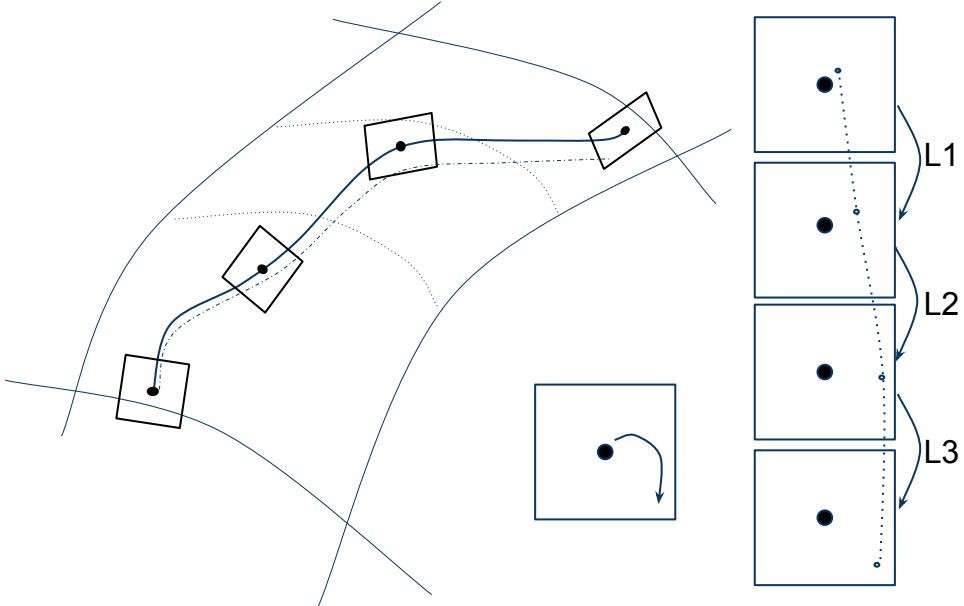


Figure 1: Left: The Evolution of a state (solid) and a perturbed state (dashed) through the dynamical system Right: The evolution of the same perturbation through a sequence of tangent spaces Bottom: The evolution of a perturbation through the tangent spaces identified as the same space

We will also use linear functionals on the linear spaces $\lambda : T_x \rightarrow \mathbb{R}$. We depict these functionals as red lines in Fig. 2, representing a projection to a 1-D subspace. In section 4 we show how the adjoint to the linear model propagates these backwards along the trajectory.

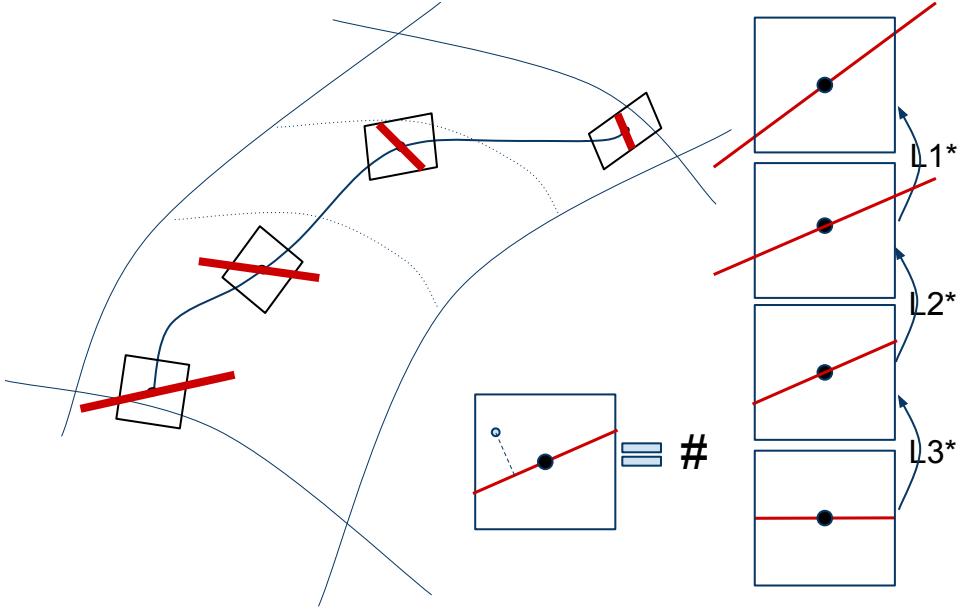


Figure 2: The evolution of a functional backwards through the tangent spaces of a dynamical system. It starts by selecting the x-coordinate of the system at forecast time but evolves into a more complex projection (left and right). Center-bottom: A depiction of how a perturbation is matched with a functional to produce a number.

2.4 Modeling Uncertainty

We define a space of possible weather states. For simplicity we discretize the physical region of interest both along the land and into several vertical levels. At each latitude, longitude, and height level there are several variables (temperature, pressure, wind, etc....) The space of discretized weather states can then be represented by \mathbb{R}^n where

$$n = n_{\text{east-west}} \times n_{\text{north-south}} \times n_{\text{vertical}} \times n_{\text{variables}} \quad (1)$$

The state of the weather at any given time can then be represented by a point in \mathbb{R}^n . Uncertain knowledge of the state can be represented as a probability distribution, $P : \mathbb{R}^n \rightarrow \mathbb{R}$ with non-negative values that sum to one, rather than an exact state. Intuitively we think of very certain distributions as tightly bundled and uncertain distributions as being very diffuse or multi-modal in physical space.

The distribution P can in general be quite complex. When we want to say simple but meaningful statements about P we often appeal to its average (expectation value) or spread (variance) and erroneously use these as proxy to the distribution itself.

The next simplest yet convenient approximation is to represent P as a Normal distribution with mean μ covariance matrix Σ . For small systems (such as the Lorenz Oscillator) it is possible to approximately evolve this distribution through time in a local sense.

For sufficiently well behaved P it may be intuitive to visualize the level sets of a small subspace as contours centered (hopefully) on an average state with spread correlated with the variance.

2.5 The Lorenz Oscillator - A Visually Motivating Example

The Lorenz Oscillator is a three dimensional dynamical system which exhibits chaotic behavior [1]. It is simple, easy to analyze/visualize and is convenient for demonstration. It resembles our target system, meteorology, in its chaotic behavior, attractor of valid states and distinct and separable zones. It fails

to resemble meteorology in that it is very low dimensional and lacks a concept of locality among its variables.

We rely on the Lorenz Oscillator to provide visual demonstration and intuition for many of the presented ideas. We will point out when dissimilarities prevent accurate demonstration.

2.5.1 Definition

We define a state in the Lorenz Oscillator with the three variables x, y, z . An initial state $\langle x_i, y_i, z_i \rangle \in \mathbb{R}^3$ is evolved through time to a final state $\langle x_f, y_f, z_f \rangle = \mathcal{M}_{t_i \rightarrow t_f}(\langle x_i, y_i, z_i \rangle)$ according to the following ordinary differential equations

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z\end{aligned}\tag{2}$$

With parameters σ, ρ, β constant for a given system. Common values which yield chaotic behavior are $\sigma, \beta, \rho = 10, 8/3, 28$.

2.5.2 Local Behavior - Perturbations

We consider a perturbed state $\mathbf{x} + \delta\mathbf{x} = \langle x + \delta x, y + \delta y, z + \delta z \rangle$, and its evolution $(\mathbf{x} + \delta\mathbf{x})(t)$ in relation to the trajectory $\mathbf{x}(t)$ over very short timescales and with $|\delta\mathbf{x}|$ small. The evolution of $(\mathbf{x} + \delta\mathbf{x})(t)$ is described by Eqns. 2

$$\begin{aligned}\dot{(x + \delta x)} &= \sigma((y + \delta y) - (x + \delta x)) \\ \dot{(y + \delta y)} &= (x + \delta x)(\rho - (z - \delta z)) - (y - \delta y) \\ \dot{(z + \delta z)} &= (x + \delta x)(y + \delta y) - \beta(z + \delta z)\end{aligned}$$

Terms can be rearranged to produce

$$\begin{aligned}\dot{x} + \dot{\delta x} &= \sigma(y - x) + \sigma(\delta y - \delta x) \\ \dot{y} + \dot{\delta y} &= x(\rho - z) - y + \delta x(\rho - z) - x\delta z - \delta y - \delta x\delta z \\ \dot{z} + \dot{\delta z} &= xy - \beta z + y\delta x + x\delta y - \beta\delta z + \delta x\delta y\end{aligned}$$

We can subtract off the terms familiar from Eqn. 2 and drop terms quadratic in δ to obtain a linear evolution rule for the perturbation $\langle \delta x, \delta y, \delta z \rangle$ valid for small perturbations:

$$\begin{aligned}\dot{\delta x} &= \sigma(\delta y - \delta x) \\ \dot{\delta y} &= \delta x(\rho - z) - x\delta z - \delta y \\ \dot{\delta z} &= y\delta x + x\delta y - \beta\delta z\end{aligned}\tag{3}$$

Because we have removed terms non-linear in δ this can be rewritten in matrix form

$$\begin{bmatrix} \dot{\delta x} \\ \dot{\delta y} \\ \dot{\delta z} \end{bmatrix} = \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho - z & -1 & -x \\ y & x & -\beta \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix}$$

The 3×3 matrix is the Jacobian J of the Lorenz system and from it we can compute how perturbations $\delta\mathbf{x} = \langle \delta x, \delta y, \delta z \rangle$ evolve local to a specific state $\mathbf{x} = \langle x, y, z \rangle$. $\delta\mathbf{x}(t)$ evolves from $\delta\mathbf{x}(a)$ through time as follows

$$\delta\mathbf{x}(b) = e^{\int_a^b J(\mathbf{x}(t))dt} \delta\mathbf{x}(a) \quad (4)$$

A simple method to approximately compute the evolution of $\delta\mathbf{x}$ presented in equation 4 is forward Euler. We step forward uniformly to compute the next state from the current one, ignoring non-linear effects.

$$\begin{aligned} \delta\mathbf{x}(b) &= e^{\int_a^b J_{\mathbf{x}(t)} dt} \delta\mathbf{x}(a) \\ &\approx e^{\sum_{i=0}^{n-1} J_{\mathbf{x}_i} \Delta t} \cdot \delta\mathbf{x}(a) : \Delta t = \frac{b-a}{n}, \mathbf{x}_i = \mathbf{x}(i\Delta t) \\ &\approx \prod_{i=0}^{n-1} e^{J_{\mathbf{x}_i} \Delta t} \cdot \delta\mathbf{x}(a) \\ &\approx \prod_{i=0}^{n-1} [I + J_{\mathbf{x}_i} \Delta t] \cdot \delta\mathbf{x}(a) : \text{for } |J_{\mathbf{x}_i} \Delta t| \ll 1 \\ &\approx \prod_{i=0}^{n-1} L_i \cdot \delta\mathbf{x}(a) : \text{for } L_i = I + J_{\mathbf{x}_i} \Delta t \\ &\approx L_{n-1} L_{n-2} \dots L_2 L_1 L_0 \cdot \delta\mathbf{x}(a) \end{aligned}$$

The linear operators L_i approximately evolve perturbations $\delta\mathbf{x}_t$ at time t to perturbations $\delta\mathbf{x}_{t+\Delta t}$ at time $t + \Delta t$

$$\delta\mathbf{x}_{t+\Delta t} = (I + J_{\mathbf{x}(t)} dt) \delta\mathbf{x}_t = L_{\mathbf{x}(t)} \delta\mathbf{x} \quad (5)$$

$$\begin{pmatrix} \delta x_{t+\Delta t} \\ \delta y_{t+\Delta t} \\ \delta z_{t+\Delta t} \end{pmatrix} = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + dt \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho - z_i & -1 & -x_i \\ y_i & x_i & \beta \end{bmatrix} \right) \begin{pmatrix} \delta x_t \\ \delta y_t \\ \delta z_t \end{pmatrix}$$

Perturbations which grow $|\delta\mathbf{x}_{t+\delta t}| > |\delta\mathbf{x}_t|$ are unstable while perturbations which shrink $|\delta\mathbf{x}_{t+\delta t}| < |\delta\mathbf{x}_t|$ are stable. In the Lorenz and other chaotic systems we often find that there exists a submanifold of the state space away from which perturbations shrink. That is states which are perturbed off of the manifold are *attracted* back to it. This submanifold is called an *attractor*. The attractor for the Lorenz system is depicted in Fig. 3. States which are on the attractor tend to stay on the attractor even though they exhibit chaotic behavior within it.

If the nonlinear equation is sufficiently linear around \mathbf{x} over the timescale $t \rightarrow t + \delta t$ then the linear operator $L_{\mathbf{x}} = I + J_{\mathbf{x}} dt$ should approximate the difference between two non-linear evolutions of the model.

$$M_{t \rightarrow t+\delta t}(\mathbf{x} + \delta\mathbf{x}) - M_{t \rightarrow t+\delta t}(\mathbf{x}) \approx L_{\mathbf{x}} \delta\mathbf{x} \quad (6)$$

Note that $L_{\mathbf{x}}$ is defined separately for each state $\mathbf{x}(t)$ in the trajectory. If we sample $\mathbf{x}(t)$ sufficiently densely in time we can construct linear operators $L_{\mathbf{x}_i}$ along the trajectory and compose these operators to approximate the evolution of perturbations over a longer time period

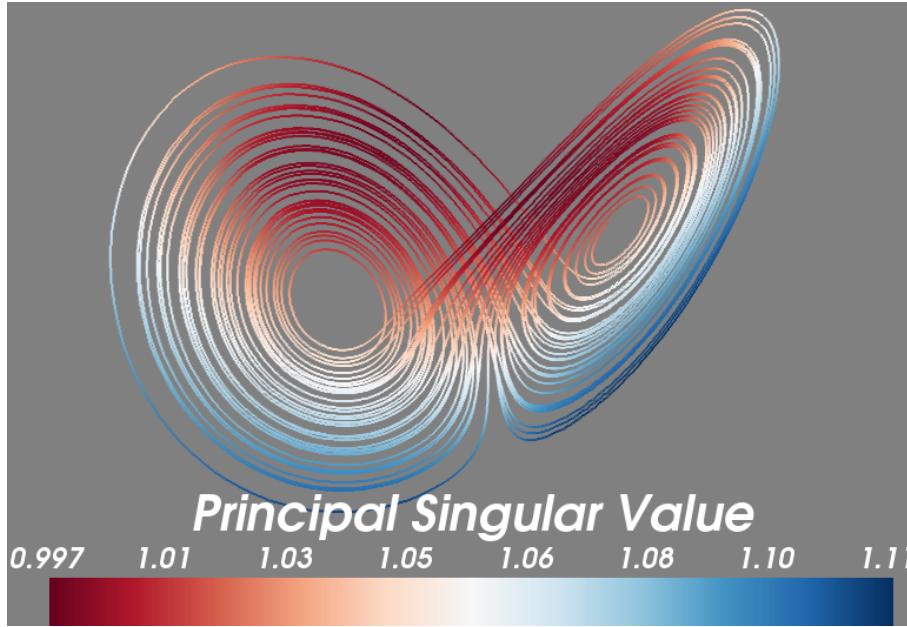


Figure 3: The principal singular value of the linearized evolution operator L with a time-step $dt = .01$. Notice how these values are greater than 1 almost everywhere. This leads to growth of errors at short scales and chaotic behavior at long scales. The Singular Value Decomposition of the Lorenz Oscillator will be covered in more depth in section 3.2.1

$$\delta\mathbf{x}(b) = M_{a \rightarrow b}(\mathbf{x} + \delta\mathbf{x}) - M_{a \rightarrow b}(\mathbf{x}) \approx L_{\mathbf{x}_n} L_{\mathbf{x}_{n-1}} \dots L_{\mathbf{x}_2} L_{\mathbf{x}_1} \delta\mathbf{x}(a) \quad (7)$$

From a trajectory of states we can therefore define a new system for the perturbations, $\delta\mathbf{x}(t)$, evolved along this trajectory, $\mathbf{x}(t)$. While all of these systems exhibit exponential structured growth they are quantitatively quite different.

3 Ensemble Methods for Uncertainty Quantification

Our knowledge of the current state of the weather is inexact. Instead of knowing the true state of weather at present time we have only a probability distribution. Probability distributions can be complex and evolving them through dynamical systems is often challenging.

A common approach to this problem is to sample from the probability distribution at the initial time, evolve this sample forward to the final time, and then use sample statistics (mean, standard deviation, correlations) to approximate the probability distribution at final time. This method may lose valuable information but is simple to implement and commonly done in practice.

In the meteorological problem we find that the probability distribution at the initial time is not readily available. Instead the National Center for Atmospheric Research (NCAR) releases best-guess estimates (not probability distributions) of North America every six hours. Through these point estimates distributed in time and the assumption that the computational model reliably represents the true weather over short periods we will construct an ensemble which predicts uncertainty with reasonable accuracy.

In this section we discuss two problems:

1. How can we estimate the current uncertainty given only point estimates distributed through time?
2. Can we verify that ensemble methods effectively evolve a probability distribution to the forecast time?

We begin in section 3.1 by examining the failings of a naive ensemble method on the Lorenz Oscillator. We find that it is important to be able identify locally feasible states. We approach this issue in section 3.2 through a sequence of methods culminating with a discussion of Bred Vectors in section 3.2.3. In section 3.3 we modify the Breeding approach to formulate a solution to problem 1, estimating uncertainty in space given estimates distributed through time. Finally we pursue problem 2 in section 3.4 by testing the method proposed in 3.3 on wind-speed data collected at two sites in Illinois.

3.1 A Naive Solution

We pose a simplistic solution so that its failings may motivate our future efforts.

Given a best-guess start state μ we pose that we are certain of this state within some common-sense bounds ($\pm 3C$, $\pm 10\%$ wind speed, etc....). We represent this uncertainty with a normal distribution $\mathcal{N}(\mu, \Sigma)$ with Σ a covariance matrix denoting the standard deviations in the variables just mentioned.

We then sample from this distribution and evolve an ensemble forward. As the ensemble evolves forward in time we compute sample statistics as a proxy to the actual distribution.

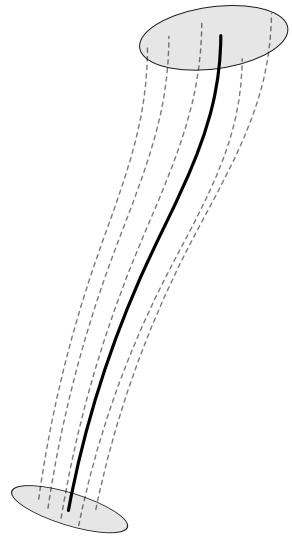


Figure 4: The evolution of a probability distribution (bottom) is represented by the evolution of an ensemble (lines) drawn from the distribution. At the forecast time (top) sample statistics are taken from the ensemble to form a probability distribution.

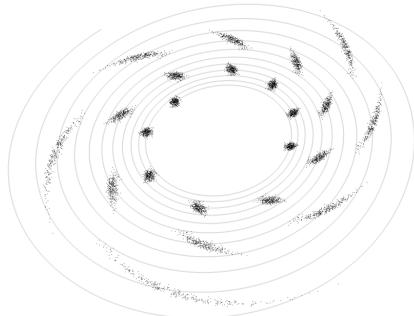


Figure 5: Evolution of an ensemble approximates the evolution of a probability distribution.

To investigate the effects of this solution, we apply this strategy to the Lorenz Oscillator. Given a best-guess start state $\mu = (x_0, y_0, z_0)$ we pose that we are certain of this location within .1 units. We represent this uncertainty with a normal distribution $\mathcal{N}(\mu, \Sigma)$ with $\Sigma = .1 \cdot I$ a covariance matrix denoting .1 standard deviation isotropically.

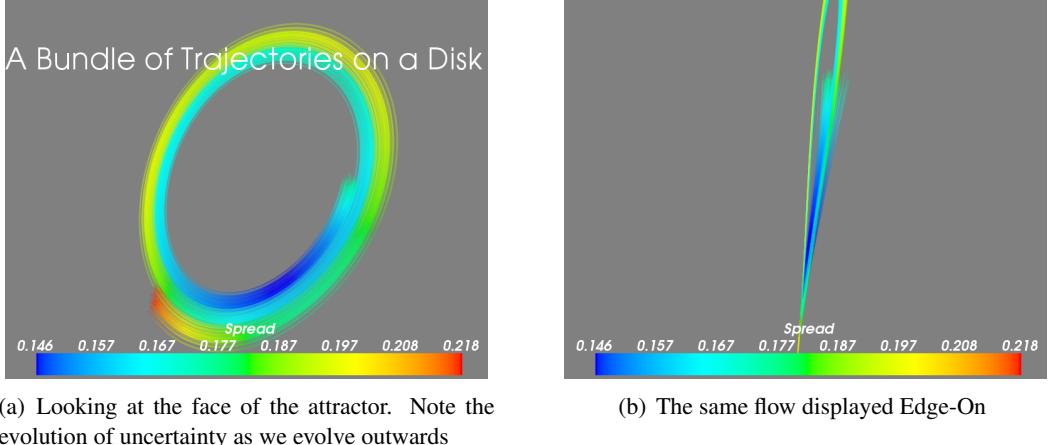


Figure 6: A Normal bundle of states evolving on the Lorenz Attractor. Note how the original start states collapse onto the 2-D attractor resulting in a notable loss of spread.

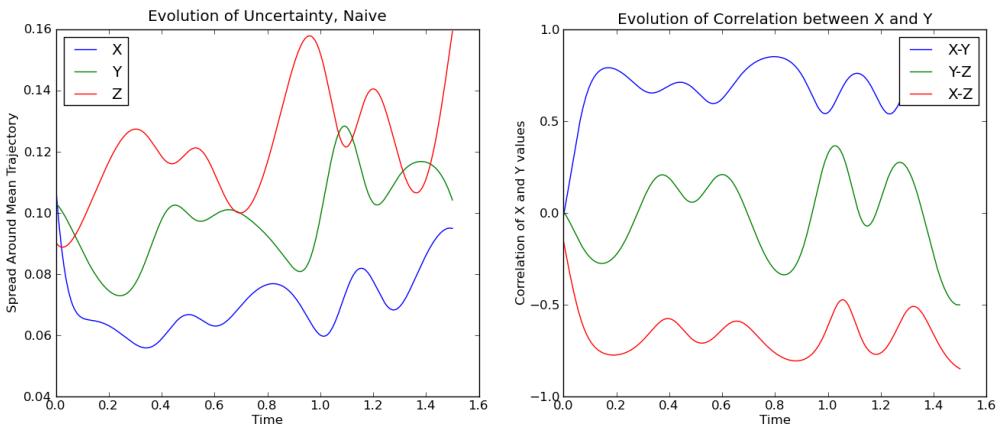


Figure 7: Left: The spread of test particles drops suddenly and then grows in a structured way. The drop can be seen as the test particles rapidly fall onto the attractor unexpectedly reducing uncertainty. How can we start a bundle so that all points are on the attractor? Right: While originally the perturbations were created with no interdependence we see that they quickly correlate with each other. How can we ensure that this trait is present in the creation of the ensemble?

We sample one hundred realizations of this distribution and evolve them independently through the system. Tracks are plotted in Fig. 6 and two statistics plotted in Fig. 7. The statistics are chosen to highlight some failings of this approach.

First, in Fig. 6 we see that the original perturbed states do not originally lie on the attractor. As they evolve they begin to fall towards it, aggregating as they do so. This causes an immediate and unexpected decrease in the spread or uncertainty of the states, visible in the blueing of color or more explicitly in Fig. 7a, (time between (0, .2)).

Second, in Fig. 7b we plot the evolution of the correlation coefficient of the x, y, z variables of

the Lorenz Attractor across the 100 states over time. While the original states were formed with no correlation we see that after a brief time strong dependencies develop ($X - Y, X - Z$).

The first note is concerning because, dictating an uncertainty at the start time we expect that this uncertainty should increase, not decrease as we see in the plots. The second note is concerning because it shows that the uncertainty in the distribution of states is highly structured and correlated. Projecting these ideas back into the weather domain we learn that

1. Correlation structure is likely to be present in the probability distribution. This correlation is intuitive as we expect nearby points will likely fall or rise together and that different variables (pressure and wind velocity for example) will have strong effects on each other in nearby times and locations.
2. If we do not create perturbations which lie on the attractor of feasible weather states we will underestimate the growth of uncertainty.

There is however an important silver lining to this experiment. A poor-quality ensemble evolved through the model to become a high-quality ensemble with appropriate correlations and growing uncertainty. Thus:

- 3 The mechanics to form an ensemble of reasonable perturbations are present within the model itself.

3.2 Singular, Lyapunov, and Bred Vectors

In the last section we found that it was important to initialize our ensemble on the attractor. In this section we will outline three methods to identify the feasible states of the system local to a given point. In section 3.3 we will extend the last of these three, the breeding process, to incorporate best-guess estimates distributed through time.

3.2.1 Singular Vectors

For the Lorenz Oscillator the attractor is a two-dimensional manifold embedded in \mathbb{R}^3 . Given a state we can find the local attractor by examining the linear forward evolution operator L_x discussed in section 2.5.2. We compute its Singular Value Decomposition.

$$L_x = U\Sigma V^T \quad (8)$$

With U, V orthogonal matrices and Σ diagonal with entries $\sigma_{1,1} > \sigma_{2,2} > \sigma_{3,3}$.

Consider the space of perturbations around any given state x_0 . Because the system is chaotic, perturbations will likely grow in at least one direction ($\sigma > 1$). Perturbations moving away from the attractor however will be pulled back, diminishing in magnitude ($\sigma < 1$). This difference will be reflected in the values of the singular value decomposition. The vector of lowest singular value ($\sigma_{2,2} < 1$) will be orthogonal to the attractor while the other singular vectors define the local tangent plane of the attractor. This is depicted in Fig. 8

This decomposition yields a local representation of the attractor. By creating perturbations only in the subspace spanned by the first two singular vectors we can locally ensure that our ensemble is initialized on the attractor.

3.2.2 Lyapunov Vectors

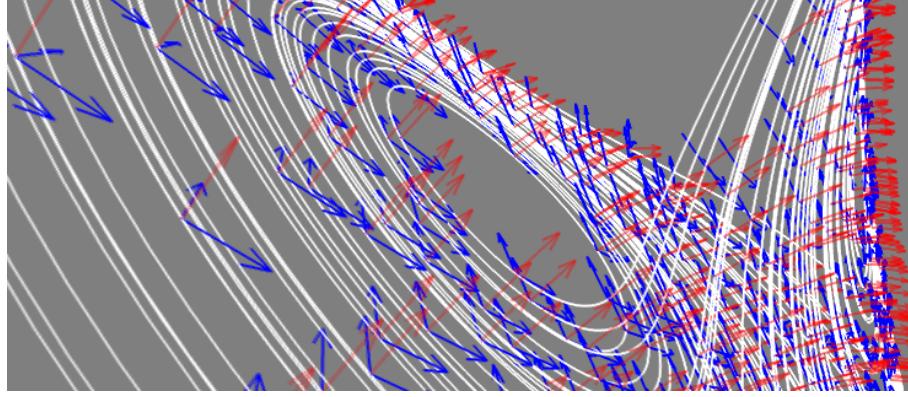


Figure 8: The singular vectors with large singular values ($\sigma > 1$) (blue) help to locally define the plane of the attractor while vectors with small singular values ($\sigma < 1$) (red) are orthogonal to the attractor.

For larger systems computing the complete singular value decomposition of the linear evolution operator may be infeasible. In several cases we have only the ability to apply an operator, and can not represent it numerically or compute its decomposition. We can approximately compute the leading singular vectors through Krylov methods. The resultant vectors in this field are called *Lyapunov vectors* in this field.

For a given linear evolution operator, L , Lyapunov vectors are the result of the iterative application of $L^T L$ on a set of random seed vectors followed by orthonormalization. By looking at the singular value decomposition of $L = U\Sigma V^T$ we find that

$$L^T L = V \Sigma U^T U \Sigma V^T = V \Sigma^2 V^T \quad (9)$$

And so, by repeated application of $L^T L$, we will converge on the k principal singular vectors (v_1, v_2, \dots) of L with a rate given by the ratio of the square of the singular values $\sigma_k^2/\sigma_{k+1}^2$.

A variant found in Kalnay et al [5] is that Lyapunov vectors may be computed as in section 2.5.2 by sequential application of L on a seed ensemble of initial perturbations. These are evolved forward along the trajectory but are periodically orthonormalized to maintain separation and prevent computational blowup.

In either case, Lyapunov vectors represent those global perturbations which yield the fastest growing errors.

3.2.3 Bred Vectors

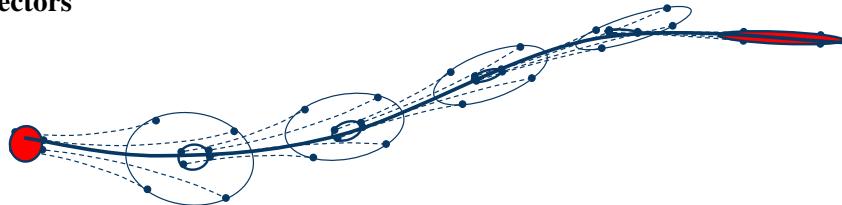


Figure 10: Evolution of an ensemble through the breeding process

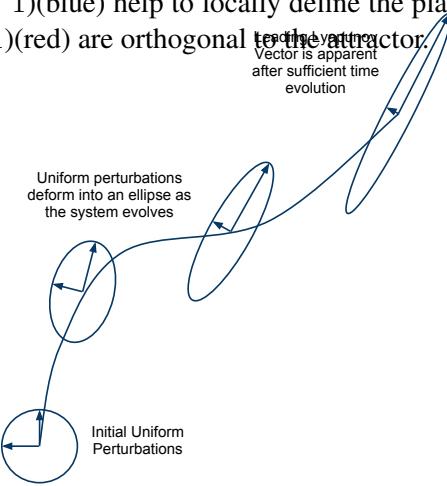


Figure 9: Depiction of the evolution of perturbations into Leading Lyapunov Vectors

Bred Vectors have been more recently accepted as a method to create initial perturbations on the attractor [5, 6]. Rather than evolve perturbations δx using the linear operator L , states in the ensemble (here called Bred Vectors) are computed using perturbed states $x + \delta x$ and the full non-linear model M .

As these states evolve they both fall onto the attractor and are kept local to the mean state. In the limit of infinitesimal timesteps and infinite time evolution, bred vectors converge to the leading Lyapunov vector[7]. For non-limiting cases however the two produce qualitatively different ensembles.

Because the Singular/Lyapunov vectors are defined for the whole linear operator, they maximize growth of *global* perturbations. Conversely in the full model different regions do not communicate with each other over short time periods. As a result bred vectors are composed of many separate fast-growing local errors, with several occurring independently in the same vector[6]. Bred vectors emphasize fast-growing regional perturbations while Lyapunov vectors emphasize fast-growing large scale perturbations[5].

To compute bred vectors we perform the following process. For more details see Kalnay-2002 [5]:

Input: Mean state \bar{x} , Model M , norm on states $|\cdot|$, intended spread σ

1. Create an ensemble of uniformly perturbed states centered around a mean state \bar{x} with spread σ

$$\begin{aligned} \{x_i\} &= \bar{x} + \{\delta x_i\} \\ \delta x_i &\sim \mathcal{N}(0, \sigma I) \end{aligned}$$

2. Evolve the ensemble forward a fixed time $t \leftarrow t + \Delta t$

$$\begin{aligned} \{x_i\} &\leftarrow \{M_{t \rightarrow t + \Delta t}(x_i)\} \\ \bar{x} &\leftarrow M_{t \rightarrow t + \Delta t}(\bar{x}) \end{aligned}$$

3. Rescale the ensemble around the mean so that the spread of the perturbations is again about σ .

$$\begin{aligned} \{\delta x_i\} &\leftarrow \{x_i\} - \bar{x} \\ \{\delta x_i\} &\leftarrow \left\{ \frac{\sigma}{|\delta x_i|} \delta x_i \right\} \\ \{x_i\} &\leftarrow \bar{x} + \{\delta x_i\} \end{aligned}$$

4. Repeat items 2,3 until satisfied.

Output: Ensemble of states $\{x_i\}$ on the attractor with spread $|\delta x_i| \approx \sigma$

In bred vector literature the stopping condition is left ambiguous. It is desirous to breed for long enough to collapse states to the attractor but not so long that all bred vectors converge to the leading Lyapunov vector [5]. In practice we continue the breeding process until there is no longer not a sharp decrease in the spread of the ensemble after a single iteration. We expect the component of the ensemble that is not on the attractor to contract significantly and the component of the ensemble that is on the attractor to expand slowly. When the rescaling factor $\frac{\sigma}{|\{\delta x_i\}|}$ reaches one we suspect that this condition is likely met.

Patil et al. [8] showed that for local regions the attractor was relatively low-dimensional (on the order of 3-10 dimensions), meaning that there were only a few things that the weather could do in a specific time and place. This means that if we restrict our views to specific regions then we need only compute on the order of 10 bred vectors to encompass local variability. Global variability is much larger in scope, consisting of the product of the variability of many small regions.

3.3 Uncertainty Quantification - Obtaining Estimates

We now have the ability to initialize a local ensemble of states on the attractor around a *best-guess* state. While the best-guess field is available for any given time at the North American Regional Reanalysis (NARR) archive, uncertainty information is not. How can we obtain uncertainty estimates and how can we evolve them forward?

To solve this problem we leverage our weather model and the NARR estimates at nearby times to estimate uncertainty using a breeding process similar to the breeding process described in section 3.2.3.

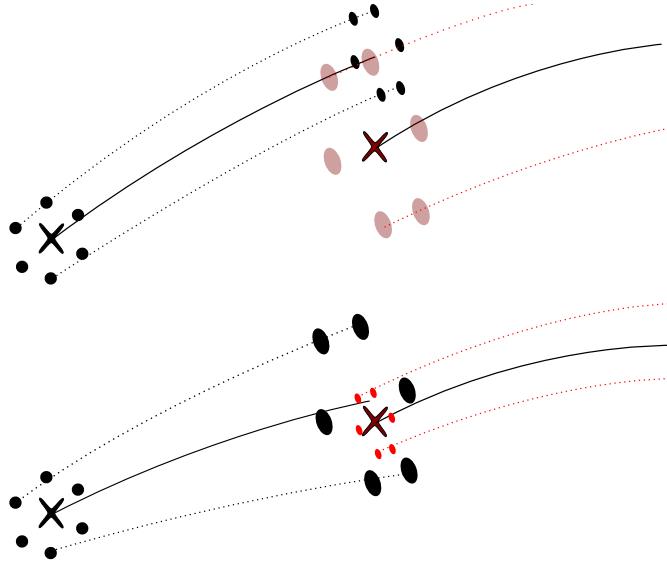


Figure 11: Evolution of original states (black) with recentering, rescaling and continued evolution around future NARR state (red). The spread of the second ensemble is determined by the error of the first forecast

An Algorithm:

1. Start with an ensemble $\{x_{i,t_0}\}$ on the attractor centered around the best guess state $x_{bg,t=0}$ with arbitrary spread, η_0 .

$$\begin{aligned} \{x_{i,t_0}\} &= x_{bg,t_0} + \{\delta_{i,t_0}\} \\ |\delta_{i,t_0}| &\approx \eta_0 \end{aligned}$$

2. Evolve this ensemble forward to the next NARR analysis time to obtain

$$\begin{aligned} \{x_{i,t_1}\} &= \{M_{t_0 \rightarrow t_1}(x_{i,t_0})\} \\ x_{bg,t_1} &= M_{t_0 \rightarrow t_1}(x_{bg,t_0}) \end{aligned}$$

3. Obtain the actual best-guess state at t_1 from the NARR repository

$$x_{bg,t_1}^*$$

4. Compute the spread of this ensemble η_{t_1} and the error between the evolved best-guess estimate and the known estimate

$$\begin{aligned} \eta_{t_1} &= |\{x_{i,t_1} - x_{bg,t_1}\}| \\ \sigma &= |x_{bg,t_1}^* - x_{bg,t_1}| \end{aligned}$$

- Center the ensemble around the known x_{bg,t_1}^* with respread the states based on the error σ between the known and the evolved mean states

$$\begin{aligned}\{\delta x_{i,t_1}\} &= \{x_{i,t_1} - x_{bg,t_1}\} \\ \{x_{i,t_1}\} &\leftarrow \left\{ \frac{\sigma}{\eta_{t_i}} \delta x_{i,t_1} + x_{bg,t_1}^* \right\}\end{aligned}$$

3.4 Validation

To validate this process we compare the computed mean and standard deviation of windspeed and temperature with observed values in two locations in Illinois. We find that with occasional exceptions, computations do tend to represent the average and uncertainty well.

To test if this process can be used in practice we wanted a greater level of accuracy than our normal, qualitative tests. To balance this desire with computational constraints we used nested domains and distributed individual jobs among several compute nodes. For every time step the outer domain (also the coarsest) is solved once. Then the next inner domain uses the result of the previous computation as a boundary condition and proceeds to solve over the same time period in the smaller domain at finer spatial and time resolutions. This continues once again for the innermost domain.

We breed an ensemble of thirty states forward for five days. At regular 12 or 24 hour intervals we recenter the ensemble so that the sample average coincides with the observed best-guess weather state at that time . Also at each interval we rescale the uncertainty as depicted in Fig. 11. We evolve this ensemble forward for five days. For the same five days we take wind and temperature data from two sites in Peru, IL and Chicago, IL and compare these measurements with our expectation and uncertainty measurements.

Deviations between this observed data and our averaged forecast measure the quality of the WRF solver. The extent to which these deviations match our uncertainty measurements measure the quality of our uncertainty quantification technique. In Fig. 13 we measure WRF by comparing the data points (circles) with the solid line (mean forecast). We measure the UQ method by comparing the standard deviation uncertainty (colored bands) with the distribution of the circles away from the solid line.

We find little difference in forecast or uncertainty accuracy between 12 and 24 hour reanalysis times.

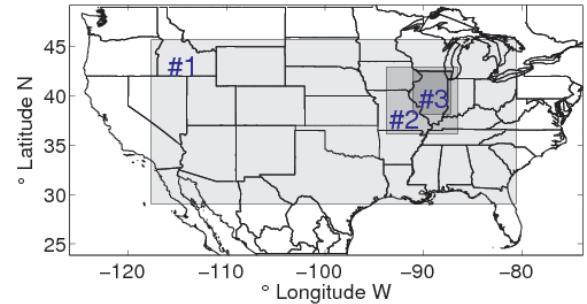


Figure 12: Nested domains focused on Illinois used for the validation computation.

#1 – 32 km^2 – (130×60)

#2 – 6 km^2 – (126×121)

#3 – 2 km^2 – (202×232)

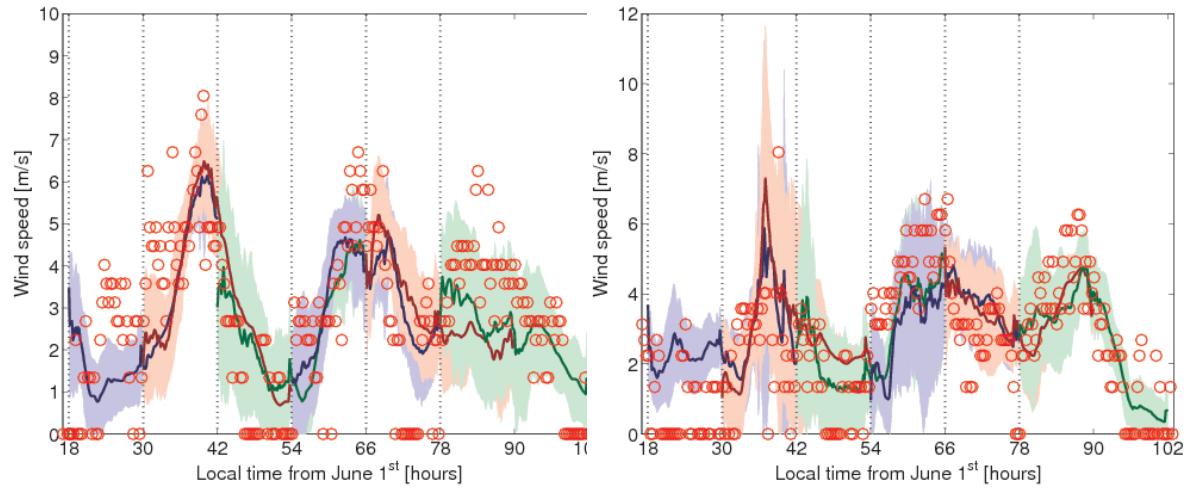


Figure 13: Wind-speed ($\pm 2\sigma$) predictions and measurements (o) for Peru (left) and Chicago, IL. The vertical dashed lines denote the beginning of a new 24-hour prediction window; different colors are used to indicate ensembles started at different times. We have at least two ensembles valid at most times.

4 Adjoint Methods for Sensitivity Analysis

In section 3 we proposed a method to estimate uncertainty in initial conditions and evolve this uncertainty forward in time. In Fig. 13 we demonstrated how this method quantified the uncertainty through time of the wind at a particular site. In estimating this single uncertainty at a single site our method also estimated the uncertainty everywhere at great expense.

In this section we will show how solving the adjoint problem backwards from the forecast to initial time can compute targeted uncertainties more efficiently. Additionally, because this allows forecast uncertainties to be computed at the initial time optimal sites for observations can be more directly discovered, solving one of our motivating problems.

In section 4.1 we define and discuss the adjoint to a linear operator. In section 4.2 we show how adjoint variables can be back-evolved along non-linear trajectories. In sections 4.3, 4.4 we discuss the sensitivities of functionals and how they can be back-evolved as adjoint variables and applied to probability distributions to estimate uncertainty. Finally in section 4.5 we give explicit examples both on the Lorenz and Weather systems and discuss the challenges on generalizing these computational results. In section 5 we will attempt to validate the ideas and processes described in this section.

4.1 Definition

We define the adjoint of a linear operator and discuss its applicability to our problem. This is covered in more detail by D. Estep [9].

In general consider two spaces X , and Y and a linear operator between them.

$$L : X \rightarrow Y$$

For each space X , Y there is a space of linear functionals, X^* , Y^* , $x^* \in X^*$, $x^* : X \rightarrow \mathbb{R}$. We define an operator *adjoint* to L as follows.

$$\begin{aligned} L^* &: Y^* \rightarrow X^* \\ x^* &= L^*(y^*) = y^* \circ L \end{aligned}$$

It is clear that x^* is a linear function from X to \mathbb{R} and thus an element in X^* .

If we further qualify the spaces X and Y as finite dimensional inner product spaces then this process becomes familiar matrix algebra. Choose a basis in X and Y and represent vectors x , y as elements in \mathbb{R}^n , \mathbb{R}^m representing the components in that basis. The operator L becomes a matrix of size $m \times n$. Matching bases for X^* , Y^* can be found so that every linear functional can be written as an n/m -dimensional row vector.

Consider the inner product

$$\langle y, Lx \rangle \text{ or } y(Lx)$$

The parentheses are added on the right to imply how we usually think of this computation. We may choose to apply the matrix L to the left instead.

$$\begin{aligned} \langle y, Lx \rangle &= \langle Ly, x \rangle \\ \text{or} \\ y^T(Lx) &= (L^T y)^T x \end{aligned} \tag{10}$$

The numerical equivalence of the left and right side is clear. Thinking about the states and spaces involved in the two sides is illustrative however. On the left we evolve a state $x \in X$ to a state $Lx \in Y$ and then compute a known functional y^* on that state, $y^*(L(x))$. On the right we transform the known

functional $y^* \in Y^*$ to a functional $L^*(y^*) \in X^*$ and then apply this functional directly onto the original state x . We see from the numerical equality that these two processes yield the same value.

If X and Y represent past and future spaces of some forward operator L then the adjoint L^* allows us to back-evolve functionals on Y in the future to functionals on X in the past.

4.2 Non-Linear Dynamics: Local Linear and Adjoint operators

If we have multiple linear forward operators L_1, L_2, L_3, \dots chained between several spaces X, Y, Z, \dots as in section 2.5.2 then we can back evolve functionals from the latter spaces to the former ones ($\dots \rightarrow Z \rightarrow Y \rightarrow X$) by chaining together the adjoint operators $L_1^*, L_2^*, L_3^*, \dots$ in reverse order.

To accurately back-evolve adjoint variables across large time scales we may need to consider the non-linear effects of the system.

In section 2.5.2 we showed how small perturbations could be evolved along a non-linear trajectory using local linear operators.

$$\begin{aligned}\delta \mathbf{x}_{i+1} &= \delta \mathbf{x}_i + dt J_{\mathbf{x}_i} \delta \mathbf{x}_i \\ \delta \mathbf{x}_{i+1} &= L_i \delta \mathbf{x}_i\end{aligned}$$

where $J_{\mathbf{x}_i}$ is the Jacobian of the system valid at state \mathbf{x}_i , the point on the trajectory at time i . We defined the linear operator $L_i = I + dt J_{\mathbf{x}_i}$ so that $\delta \mathbf{x}_i$ was propagated by the composition of a sequence L_i .

$$\delta \mathbf{x}_t = L_{t-1} L_{t-2} \dots L_2 L_1 L_0 \delta \mathbf{x}_0 \quad (11)$$

To back-propagate an adjoint variable λ_t along a trajectory \mathbf{x}_i to time zero we compute the adjoint problem.

$$\lambda_0 = L_0^* L_1^* \dots L_{t-2}^* L_{t-1}^* \lambda_t \quad (12)$$

That this yields the desired results can be quickly seen by taking the inner product of Eq. 11 with λ_t .

$$\begin{aligned}\langle \lambda_t, \delta \mathbf{x}_t \rangle &= \langle \lambda_t, L_{t-1} L_{t-2} \dots L_2 L_1 L_0 \delta \mathbf{x}_0 \rangle \\ \langle \lambda_t, \delta \mathbf{x}_t \rangle &= \langle L_0^* L_1^* \dots L_{t-2}^* L_{t-1}^* \lambda_t, \delta \mathbf{x}_0 \rangle \\ \langle \lambda_t, \delta \mathbf{x}_t \rangle &= \langle \lambda_0, \delta \mathbf{x}_0 \rangle\end{aligned}$$

Demonstrating that λ_0 is the linear functional at time 0 which accurately computes the value at time t , $\langle \lambda_t, \delta \mathbf{x}_t \rangle$ in terms of the initial perturbation $\delta \mathbf{x}_0$. To back-evolve an adjoint variable in a non-linear system we sample the trajectory sufficiently densely in time, linearize at each step, and apply (or solve) the composition of the adjoint operators.

4.3 Non-Linear Functionals: Sensitivities as Adjoint Variables

Define a Functional J over the space of states at time t . Consider the change in J due to small perturbations in the state at time t , $\delta \mathbf{x}_t$.

$$\delta J_t \approx \frac{\partial J}{\partial \mathbf{x}_t} \delta \mathbf{x}_t \quad (13)$$

$\frac{\partial J}{\partial \mathbf{x}_t}$ is called the *sensitivity* of J with respect to changes in \mathbf{x}_t . $\frac{\partial J}{\partial \mathbf{x}_t}$ is a linear functional which maps perturbations $\delta \mathbf{x}_t \in T_{\mathbf{x}_t}$ to real numbers $\delta J \in \mathbb{R}$. We can use the adjoint operators to back-evolve this sensitivity to compute the sensitivity of J at the final time due to perturbations $\delta \mathbf{x}_0$ at the initial time.

$$\begin{aligned}
\delta J_t &\approx \left\langle \frac{\partial J}{\partial \mathbf{x}_t}, \delta \mathbf{x}_t \right\rangle \\
&\approx \left\langle \frac{\partial J}{\partial \mathbf{x}_t}, L_{t-1} L_{t-2} \dots L_1 L_0 \delta \mathbf{x}_0 \right\rangle \\
&\approx \left\langle L_0^* L_1^* \dots L_{t-2}^* L_{t-1}^* \frac{\partial J}{\partial \mathbf{x}_t}, \delta \mathbf{x}_0 \right\rangle \\
&\approx \left\langle \frac{\partial J}{\partial \mathbf{x}_0}, \delta \mathbf{x}_0 \right\rangle \text{ where we have identified} \\
\frac{\partial J}{\partial \mathbf{x}_0} &\equiv L_0^* L_1^* \dots L_{t-2}^* L_{t-1}^* \frac{\partial J}{\partial \mathbf{x}_t}
\end{aligned}$$

By computing $\frac{\partial J}{\partial \mathbf{x}_0}$ we can now efficiently compute the effect of perturbations in initial conditions on the functional J at the final time.

4.4 Uncertainty Quantification of a Single Functional

In section 2.4 we described the uncertainty of the weather as a probability distribution ($P : M \rightarrow \mathbb{R}$) over the state of the weather $\mathbf{x} = \mu + \delta \mathbf{x}$, a continuous random variable.

Now we wish to consider the more focused problem of estimating the uncertainty of $J(\mathbf{x}_t)$. Because \mathbf{x}_t is a random variable so too is $J(\mathbf{x}_t)$ and in principle with a known $P(\mathbf{x})$ we could compute a $P_J(J(\mathbf{x}))$, $P_J : \mathbb{R} \rightarrow \mathbb{R}$, through integration over the entire domain of \mathbf{x} .

This is effectively the approach we took to estimate the uncertainty of wind speed in Peru, IL in Fig. 13, performing the integration with Monte Carlo techniques. We found that integrating over entire domain of \mathbf{x} is costly.

If the uncertainty in \mathbf{x} is small compared to the relevant scale in J then we may consider the linearized problem presented in section 4.3

$$\delta J_t \approx \frac{\partial J}{\partial \mathbf{x}_t} \delta \mathbf{x}_t \quad (14)$$

Where $\delta \mathbf{x}_t$ is the random variable $\delta \mathbf{x} = \mathbf{x} - \mu$, μ constant, and $\frac{\partial J}{\partial \mathbf{x}_t}$ the gradient of J with respect to \mathbf{x}_t .

The subspace of the domain of \mathbf{x} orthogonal to $\frac{\partial J}{\partial \mathbf{x}_t}$ is (in the linearized case) irrelevant to the computation of the uncertainty of $J(\mathbf{x})$. To estimate the uncertainty of J we need only consider the probability distribution along the vector $\frac{\partial J}{\partial \mathbf{x}_t}$. Furthermore, as we saw in section 4.3, we may compute the value of $\frac{\partial J}{\partial \mathbf{x}_t}$ using the adjoint evolved $\frac{\partial J}{\partial \mathbf{x}_T}$ for any previous time $T < t$. We may estimate the uncertainty in J_t by evaluating the probability distribution $P_{\mathbf{x}_T}$ along the gradient to J_t , $\frac{\partial J_t}{\partial \mathbf{x}_T}$ at any time T .

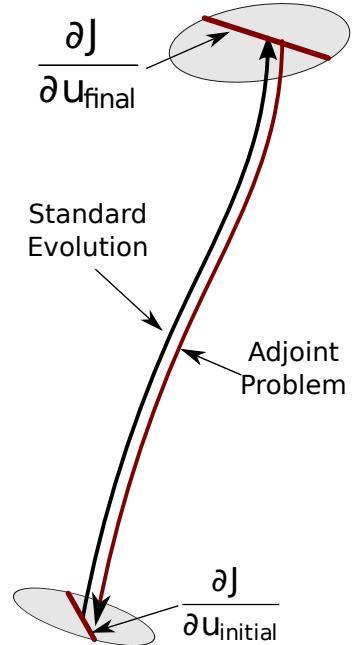


Figure 14: In analogy to Fig. 4 we can estimate uncertainty by back-evolving a sensitivity rather than forward-evolving an ensemble/probability distribution.

4.5 Application

We now consider our second problem.

Given a target region/variable of interest specified by J , where should we observe to reduce uncertainty of forecasts in this target region/variable?

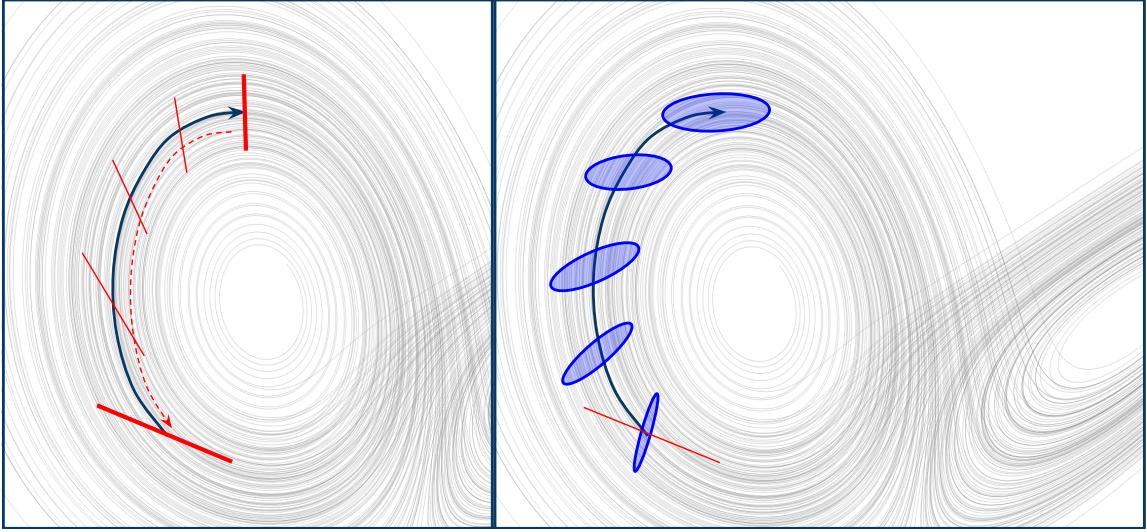


Figure 15: Left: An adjoint variable $\frac{\partial J}{\partial \mathbf{x}_t}$ evolved backwards (red dashed arrow) along a particle's trajectory (blue arrow). We use this variable to highlight that we want low uncertainty along the vertical variable. Right: Evolution of a probability distribution forward. We have reduced the uncertainty along the back-evolved gradient of J to produce the desired reduced uncertainty in the final state.

For example in Fig. 15 we want to reduce uncertainty in the vertical variable, $J(x, y, z) = y$. We set up an adjoint variable $\lambda_{\text{future}} = (0, 1, 0)$ to reflect this sensitivity and evolve it backwards to find that, in the past, we should work to reduce uncertainty in a diagonal direction, $\lambda_{\text{past}} = (.3, .6, -.8)$. We may test the effectiveness of this strategy by forming a distribution at past time with low spread in the λ_{past} direction and evolving it forwards, investigating the uncertainty at the forecast time.

For a concrete example in the weather domain, consider a functional J as "The sum of the temperature on the ground in the panhandle of Texas at time t_0 ".

$$J = \int_R T(t_0, x, y, z) dx dy dz : R \text{ is Panhandle at ground level} \quad (15)$$

Any functional, of which an example is the sensitivity $\frac{\partial J}{\partial \mathbf{x}_i(t_0)}$, can be expressed by an inner product with the appropriate adjoint variable λ

$$\frac{\partial J}{\partial \mathbf{x}(t_0)} = \sum_{i \in \{T, P, U, V, W, \dots\}} \int \delta \mathbf{x}_i(t_0, x, y, z) \lambda_i(t_0, x, y, z) dx dy dz \quad (16)$$

Where T, P, U, V, W, \dots are the variables temperature, pressure, wind velocity components, etc.... In this case $\lambda_T = 1$ when the corresponding component in $\delta \mathbf{x}$ corresponds to a perturbation in ground-level temperature variable in the panhandle region and $\lambda_i = 0$ elsewhere. As we consider functionals $\lambda(t)$ and perturbations $\delta \mathbf{x}(t)$ at other points in time the λ 's will not in general be this simple.

As in the last example we back-evolve the sensitivity $\frac{\partial J}{\partial \mathbf{x}_t}$ through time. Because $\frac{\partial J}{\partial \mathbf{x}_t}$ depends on every latitude, longitude, vertical level, and physical variable in our weather model it is challenging to display. In Fig 16 we plot just $\frac{\partial J}{\partial T_t}$ at ground level for $t \in \{0, -5, -8, -11, -14, -17\}$ hours; we see that the simple indicator function from equation 16 at time 0 is advected backwards in time.

Notice the bright spots in these images. Small perturbations near these spots at these times strongly affect the functional J evaluated at time zero. We are motivated to observe in these specific locations in the past to reduce uncertainty of J at time $t = t_0$. In this particular case most of the sensitivity evolution is caused by advection due winds blowing from East to West.

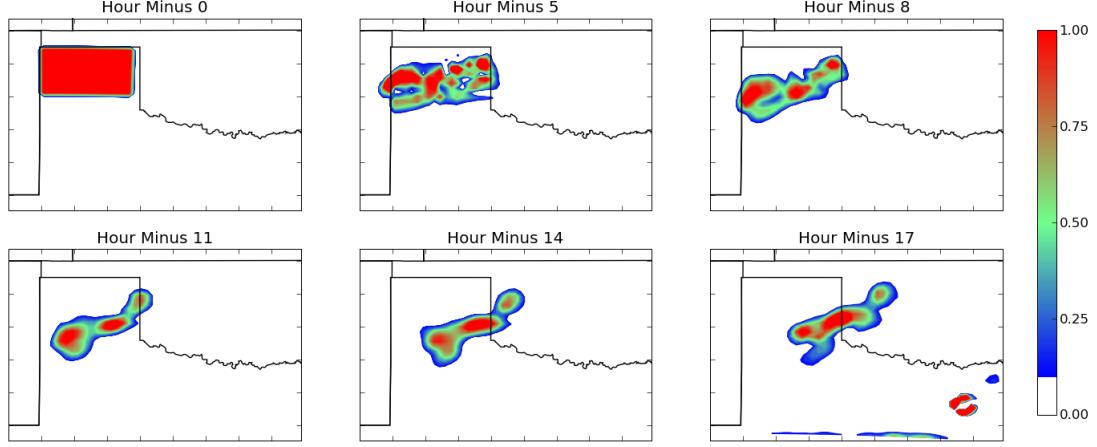


Figure 16: Adjoint Variable summing the temperature in the north of Texas evolved backwards through time. These images are analogous to the red vectors, λ in Fig. 15, but show only the absolute value of the temperature at ground level $|\frac{\partial J}{\partial T}|$ rather than the entire state. Note the predictable advection due to wind and the growth of numerical instabilities in the Southern and South-East regions after 17 hours of back-evolution.

4.6 Repeatability

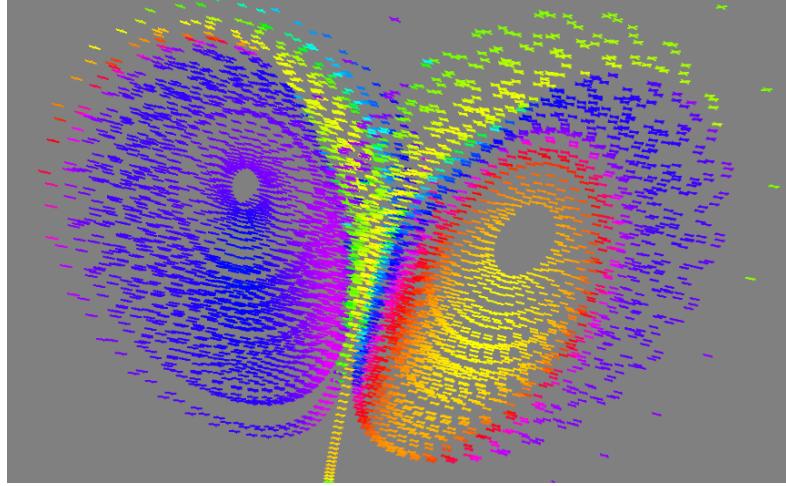


Figure 17: At each point we evolve the adjoint variable $(0,1,0)$ back $.5$ time units and plot the backward propagated adjoint variable as an RGB value. Note how sensitivity to different variables strongly depends on the region in state-space.

This sensitivity result is highly dependent on the state of the weather at time t_0 . Had the winds been blowing West to East instead of East to West then the images in Fig. 16 would look very different. This dependence of the adjoint variable on the state of the dynamical system is demonstrated in Fig. 17. This figure shows how the backwards evolution of the same adjoint variable $\lambda_{\text{future}} = (0, 1, 0)$ can vary substantially based on the region of the attractor. For example in red regions the adjoint variable is back evolved nearer to $\lambda_{\text{past}} = (1, 0, 0)$ while in blue regions it is back-evolved to $\lambda_{\text{past}} = (0, 0, 1)$.

The attractor for the weather system has this same difficulty. The evolution of the adjoint variables depends highly on the transient state of the system. Unlike the Lorenz case however we are unable to

exhaustively sample and visualize the trends. Are there trends or clearly separated regions in the weather attractor as there are in the Lorenz system?

This poses an issue for solving our second motivating problem. We are now required to heavily sample the distribution of weather states from historical data to make an informed decision about where to place observation posts to lower uncertainty in expectation.

This analysis has been used effectively for rapid deployment of observations for extreme weather events such as forecasting the amount of rainfall to be expected by a severe storm. Xiao et. al. [4] investigated a single strong storm event occurring in Antarctica that was particularly misforecasted. Baker et. al [10] discuss the feasibility of computing sensitivities for storm events as they are detected to determine aerial observation flight-paths.

5 Quantifying the Effects of Adjoint-Informed Observations

In section 4 we showed how the uncertainty of a functional could be efficiently estimated by solving the adjoint problem to a given trajectory. We also found that this process clearly highlights where additional observations would reduce targeted uncertainties. In this section we perform numerical experiments to simulate this decision. Our goals are

1. To check the assumptions made in the previous section on both the Lorenz and Weather systems
2. To investigate the feasibility of this method in selecting sites for meteorological observation for short-term forecasts.

We will find that the uncertainty in the weather system does not track the adjoint method beyond a few hours. This could be due to differences between the Advective WRF and full WRF model or non-linearities of the evolution transformation. This could be due to differences between the Advective WRF and full WRF model or non-linearities of the evolution transformation.

In section 5.1 we introduce a method to add local conditions to bred ensembles. In sections 5.2, 5.3, 5.4 we outline and present results for a simulation of adding adjoint-informed observations in the general, Lorenz, and Weather systems respectively.

5.1 Introducing Local Conditions to Bred Vector Ensembles

In section 3.3 we modified the breeding process to compute global uncertainty information using a sequence of known best-guess weather estimates from the NARR database. We will again modify the breeding process, this time on a local level to introduce novel certainty information to our ensemble, reflecting our new observations.

We could create a bred vector ensemble in the normal way and then, in the final states, replace the a subset of the components in each element of the ensemble with the components of the average state. If we then look at the standard deviation across the ensemble we would see a strong decrease in uncertainty around this subset, encoding our observations.

These states have been moved off of the attractor however and, upon evolving even a short time from this certainty-enhanced state, we'll find a rapid shift back to full uncertainty. To solve this we impose the local reduced uncertainty condition repeatedly *as we breed the vectors*. This way the condition is present as the ensemble evolves. The result is that the ensemble evolves on a lower-dimensional attractor with axis of uncertainty now unavailable.

5.2 A Test

If we have the ability to create ensembles imbued with local regions of certainty we can start to quantitatively examine the effects of observation. This allows us to check the results of the adjoint sensitivity procedure for regions targeted for wind-farm development. We describe a test for this procedure and apply it on the Lorenz system below.

1. Take a start state at time t_0 . Evolve forward through t_i to t_f so that $t_0 < t_i < t_f$
2. At time t_f create an adjoint variable which highlights a target region,
3. Back-Evolve the adjoint variable to time t_i .
4. Select sites for potential observation
5. Evolve three ensembles of bred vectors from time t_0 to time t_i under three different conditions
 - (a) Evolve ensemble under constraint that the observation sites have zero uncertainty.
 - (b) Evolve ensemble under constraint that the randomly chosen sites have zero uncertainty.
 - (c) Evolve a control ensemble without constraint

6. Evolve each of these ensembles from t_i to t_f without constraint.
7. Compare uncertainty in target region between these three ensembles.

5.3 Executing this test on the Lorenz Oscilator

We perform the above test on the Lorenz Oscilator. For this test we will try to reduce the uncertainty of the Z-variable, setting the sensitivity $\lambda_f = [0, 0, 1]$.

Trajectories on the Lorenz Oscilator were solved using a simple constant timestep Euler method. Adjoint and Tangent models were solved using the same timestep which results in near perfect connection between the backsolved adjoint and the forward-solved ensemble. The only deviation from the ideal comes from the perturbations not being infinitessimally close to the true trajectory. You can observe in Fig. 5.3d a slight bowing to the ensemble. This bowing is caused due to the non-linear terms present in the full solution not being entirely negligible at the scale of the spread of the ensemble. We could also test the errors that would arise from additional deviations from the ideal such as coarser time steps and larger ensemble sizes.

5.4 Transitioning to the Full Weather System

How would this test differ on WRF? What difficulties have we run into and should we expect?

This is the test we propose in the larger weather system with a few changes.

- Our state space is larger. Rather than $\lambda_f = (0, 0, 1)$ we'll set λ_f to be active in the region of interest and zero elsewhere.
- Observing the actual back-evolved λ_i is infeasible. Due to practical constraints on observation we will have to restrict our λ_i to have a small spatial support. Learning how and where to observe in λ_i in-silico could provide useful insight for physical measurements.
- Due to higher dimensionality we will no longer need to worry about dimensional collapse but do need to worry about sufficiently spanning the space.
- It is challenging to find a uniform uncertainty ensemble that satisfies all of our conditions.
- Our Adjoint solver is the Adjoint of a different, simpler non-linear model that approximates the full model.

We attempted to perform this test on the full weather model but ran into issues which prevented us from quantifying the benefits of adjoint-directed observations. Results from these tests are below.

We start by selecting a target region (the Panhandle of Texas), specifying a sensitivity, and evolving it backwards from the future to the present time as shown in Fig. 18.

We then select one particularly sensitive location within the sensitivity valid at time_present. We choose this location to be our observation point. Our goal then is to create an ensemble valid at time_present which has uniform uncertainty everywhere except this one point, where the ensemble should have uncertainty zero. This uncertainty field is depicted in Fig. 19 in the upper right.

To obtain such an ensemble at time_present we breed an ensemble from time_past to time_present with the following constraints.

- Uncertainty everywhere should be constant (except as follows)
- Uncertainty near observation point should decay to zero as a gaussian with given spread
- Uncertainty near the boundaries should decay to zero as a gaussian with given spread (for computational stability)
- Spread of the ensemble should fixed under some norm

The result of this process is an ensemble which is (hopefully) on the attractor with the uncertainty distribution that we desire.

We then release this ensemble and evolve it from time_present to time_future and plot the standard deviation field in Fig. 19a. To compare this experiment against a control we create a second ensemble without the observation present and plot the difference of the two uncertainty fields in Fig. 19b.

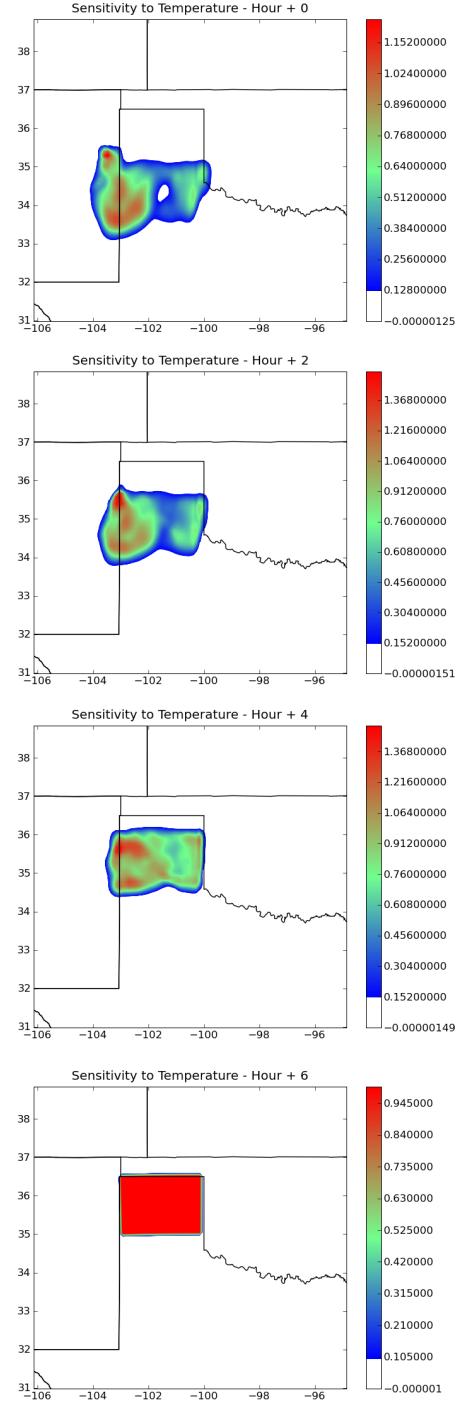
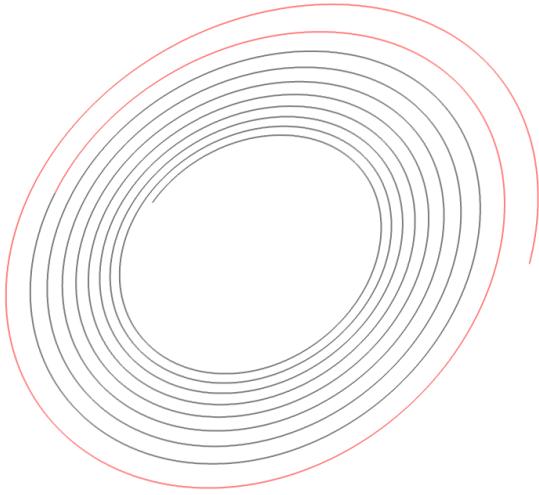
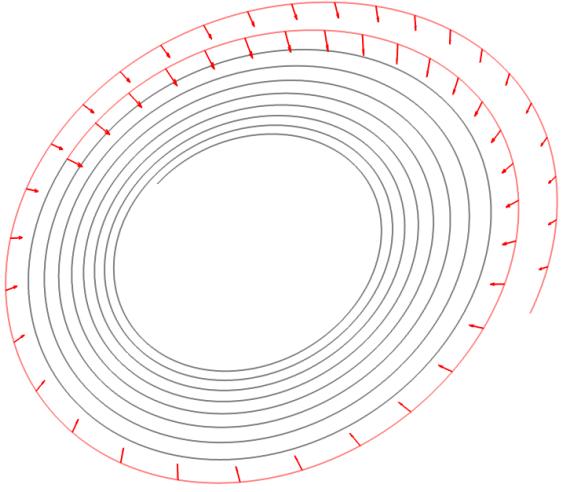


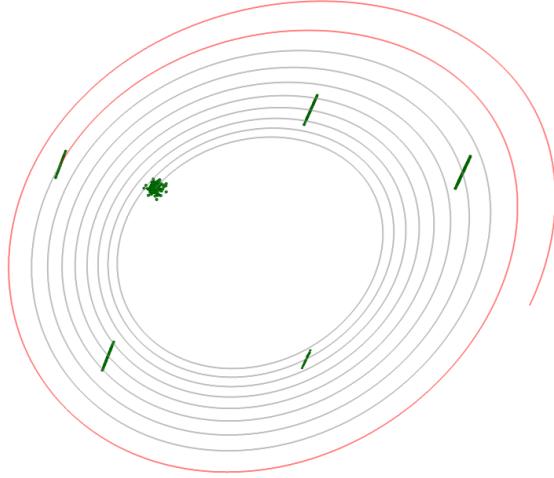
Figure 18: Evolution of the sensitivity backwards.



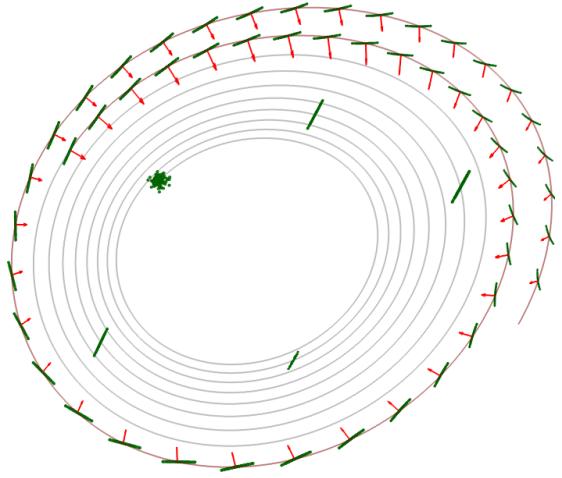
(a) We are interested in a prediction from time t_i . We start a mean-path evolution from $t_0 \rightarrow t_i \rightarrow t_f$. This is shown with $t_0 \rightarrow t_i$ in gray and $t_i \rightarrow t_f$ in red.



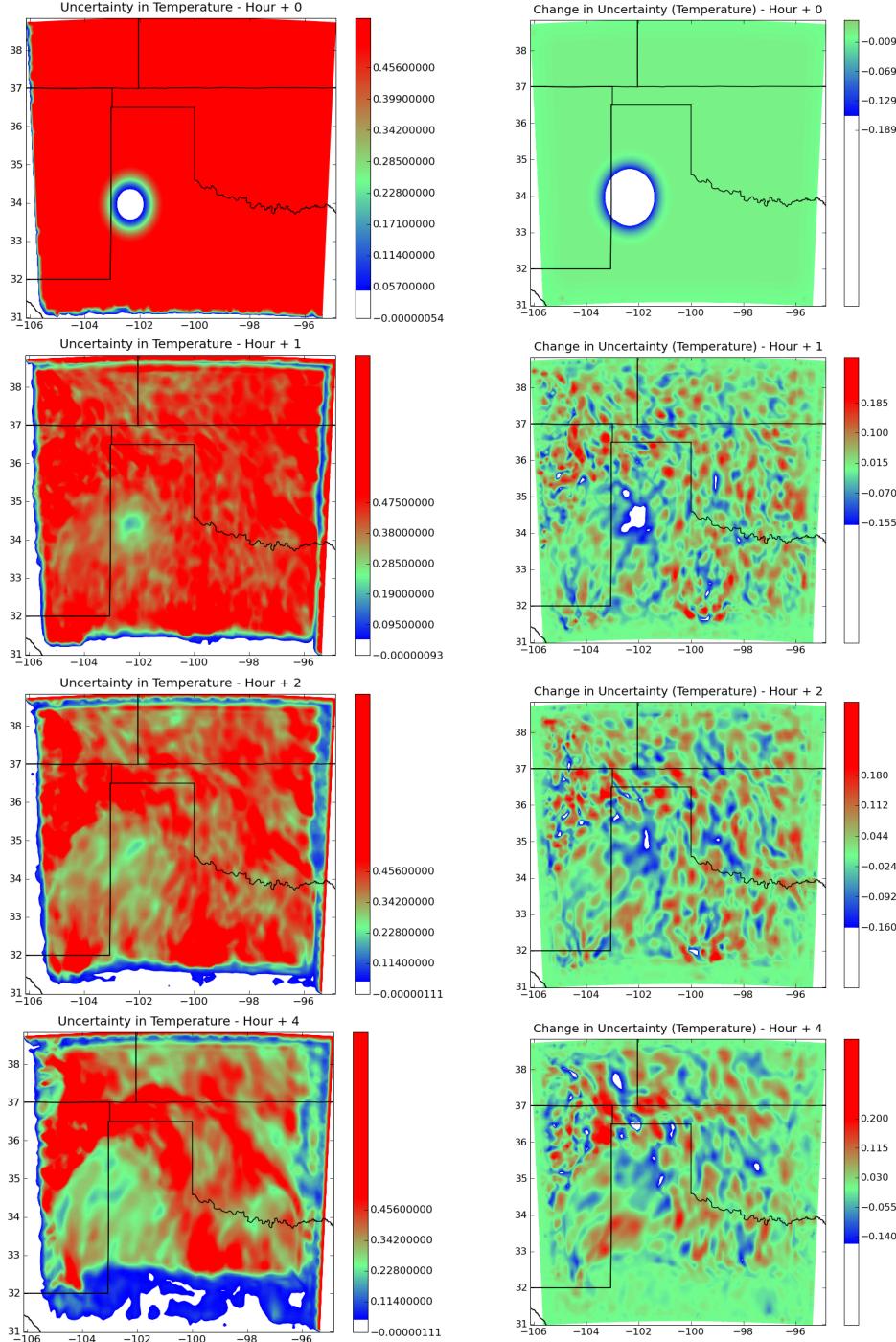
(b) We set the adjoint variable at time t_f equal to $\lambda_f = (0, 0, 1)$ and back evolve it to obtain $\lambda_i = (0.31, -1.41 - 1.86)$



(c) We want to create an ensemble valid at t_i with reduced uncertainty in the λ_i direction. To do this we breed an ensemble along the $t_0 \rightarrow t_i$ trajectory with some local conditions. 1) Uncertainty is constant (renormalizing step of bred vector process) 2) Two dimensional. Because the Lorenz Oscillator is such a small system we quickly lose dimensionality (converge to the leading Lyapunov vector). To fix this we equate the top two singular values of the singular value decomposition. This spreads the states out to a disk on the attractor. 3) Reduced uncertainty in the λ_i direction. After spreading out we constrict this particular direction.



(d) We then end the breeding process at time t_i and release the ensemble to evolve naturally to time t_f . We compare to the adjoint variable over the timespan $t_i \rightarrow t_f$. Observe on the far right that the ensemble (green dots) have very little uncertainty in the Z-direction (red arrow). Non-linearity (bowing in the ensemble) is present but minimal.



(a) Standard deviation of the Temperature of the thirty member ensemble for each point on the ground level as a function of time. At the top we see flat uncertainty with a single observation. As time progresses the certainty evolve towards the panhandle (the target region). Resolution of the certainty is lost after only a few hours of evolution.

(b) The experiment on the left was run twice, once with an observation and once without. The images above are the result of subtracting off the standard deviation of the second control experiment. The certainty resulting from the observation is clearer but still becomes lost to noise after a few hours evolution.

Figure 19: Evolution of certainty over five hours.

6 Distributed Computation

Experiments done on the full weather model (WRF) would have taken a prohibitively long time on a single workstation machine. Because of the ensemble approach much of the work is highly parallelizable. For simplicity, each single application of the weather Model on a state $x_{t_b} = M_{t_a \rightarrow t_b}(x_{t_a})$ is viewed as an atomic calculation. Communication between such states (such as to compute the spread of an ensemble under some norm) was accomplished using a shared network file system.

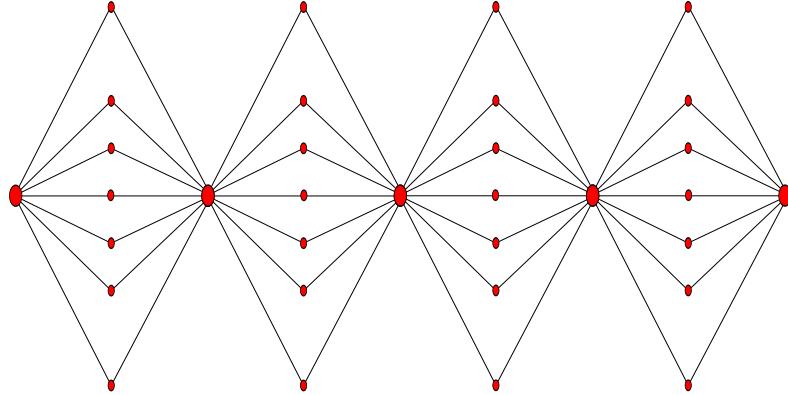


Figure 20: Depiction of the workflow of the breeding processes. Most of the work is embarrassing with occasional barriers for the recentering/respreading of the ensemble

Scheduling and allocating machines was done with automatic generation of Condor scripts through Python. All computation was done on an informal cluster of workstation machines. Code is available to provide a layer of abstraction from mathematical weather models (M), boundary conditions, states (x_t), and ensembles of states ($\{x_{t,i}\}$) to executables, NetCDF files, structured ensembles of files, and their evolution and interaction on a compute cluster.

High resolution computations performed in comparison with validation data involved more intense nested domains. For this individual evolutions, $x_{t_b} = M_{t_a \rightarrow t_b}(x_{t_a})$, were not viewed as atomic and multiple compute nodes were used in a tightly coupled compute system. For this we relied on WRF's distributed memory parallelization implementation. Scaling information is available in [2].

Adjoint computations were not a major computational burden and were done on single workstation machines. The adjoint codes themselves were developed at NCAR by the WRF development team. They are the result of running Automatic Differentiation on a simpler adiabatic weather model. Validation of these models was done in [4].

The experiments on the Lorenz Oscillator were lightweight enough to be accomplished a single workstation machine. Visualization was done through VTK hooks into Python provided by MayaVi. All Lorenz images are immediate results of Python scripts and are easily reproducible. (will have a weblink here to image scripts)

7 Conclusion and Future Work

We presented the following ideas

- A background on dynamical systems, ensemble methods for uncertainty quantification, and the evolution of adjoint variables
- The breeding vector process can be *modified* to reflect a wide range of requirements on your ensemble.
- In particular we can modify the scale of perturbations using the NARR database to accurately quantify uncertainty
- We can also modify the perturbations locally to bake in local knowledge of certainty
- Finally we discussed the effectiveness to which adjoint methods can be used to mitigate uncertainty in target regions.

Our end goal is to produce a process to determine which sites will significantly reduce uncertainty in a specified target region. We have proposed such a process and shown its efficacy on the Lorenz System for single start states. The transition from the Lorenz System to the full WRF model is one major step, a survey of behavior at a broad sampling of weather states is another.

In section 5.4 we discussed some of the challenges in moving to the full WRF system. This transition is burdened by an increase in dimensionality, decrease in control over the computational processes (we're treating WRF as a black box), and a vast increase in computation time and needed organization.

If we find that the adjoint solver can reliably find observations sites which significantly reduce uncertainty in the target region we then need to conduct a broad survey of the weather to examine the extent to which these sites are good predictors throughout the year and under different weather conditions. To this end we need to conduct a data-intensive survey of North America's meteorological history. This data can be found on the North American Regional Reanalysis (NARR) database.

Furthermore the data that we are working with is a simplification of what is actually available. This data has already gone through the data assimilation process from sparse observations to a best-guess field. If we wish to propose observation sites for future investment we should go down to the more detailed level. It may be that there are already several observation posts in this area or that observation is difficult due to the surrounding geography. These sorts of questions are unanswerable with our current model for uncertainty. Integrating the data assimilation process into our system may be a future step. At the very least we should compare our estimates for uncertainty at current time against what data assimilation processes produce.

8 Personal Results

While this project does not have a conclusive research result it did educate me in the following respects:

Uncertainty: I learned to think of measurements as distributions rather than numbers. This change in thinking has lead me to pursue statistics as a general way to approach problems. I gained some anecdotal experience with some motivating examples behind this shift in thinking.

Linearized analysis: This work solidified my understanding of forward perturbation analysis; looking at how perturbations of a solution evolve around that solution. It also introduced me to the adjoint problem to this process. Computationally I'm now familiar with automatic differentiation, how it's computed and it can be used it to carry abundant local information in a computation.

Sampling in High Dimensional Systems: This work pushed me to understand the more theoretical side of sampling; what problems does it try to solve, when is it valid, when can we know that it's valid. I've gained respect for the problems present in high dimensional systems, particularly in representing uncertain information through sampling and mindfullness of the relevant low-dimensional subspaces.

Distributed Computation: On the more practical side I dealt with a problem that could not fit on a single computer. I approached this problem from several directions: implementing a simple job scheduler, finding and using a well developed scheduler (Condor), and finally running MPI jobs on a managed cluster. This roundabout approach both forced me to learn several technologies and also gave me a great appreciation of creating an abstraction layer between mathematical statements and computational implementation.

Framework for Future Research: Most importantly this work has turned me on to other work in the field. I have enjoyed reading in this topic and hope to pursue it in the future. The intuition built here has served me well when reading about related research.

References

- [1] EN Lorenz. Deterministic Nonperiodic Flow. *Atmos. Sci.*, 20:130–141, 1963.
- [2] Emil M. Constantinescu, Victor M. Zavala, Matthew Rocklin, Sangmin Lee, and Mihai Anitescu. A Computational Framework for Uncertainty Quantification and Stochastic Optimization in Unit Commitment With Wind Power Generation. *IEEE Transactions on Power Systems*, 26(1):431–441, February 2011.
- [3] William C. Skamarock, Joseph B. Klemp, Jimy Dudhia, David O. Gill, Dale M. Barker, Michael G. Duda, Xiang-Yu Huang, Wei Wang, and Jordan G. Powers. A description of the Advanced Research WRF Version 3. *NCAR Technical Note*, NCAR/TN47(June), 2008.
- [4] Qingnong Xiao, Ying-Hwa Kuo, Zaizhong Ma, Wei Huang, Xiang-Yu Huang, Xiaoyan Zhang, Dale M. Barker, John Michalakes, and Jimy Dudhia. Application of an Adiabatic WRF Adjoint to the Investigation of the May 2004 McMurdo, Antarctica, Severe Wind Event. *Monthly Weather Review*, 136(10):3696–3713, October 2008.
- [5] Eugenia Kalnay, Matteo Corazza, and Ming Cai. Are Bred Vectors the same as Lyapunov Vectors? *EGS XXVII General Assembly*, 2002.
- [6] M Corazza, E Kalnay, DJ Patil, SC Yang, R Morss, Ming Cai, I Szunyogh, Brian Hunt, and James Yorke. Use of the breeding technique to estimate the structure of the analysis "errors of the day". *Nonlinear Processes in Geophysics*, pages 233–243, 2003.
- [7] E Kalnay. *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, 2002.
- [8] D. Patil, Brian Hunt, Eugenia Kalnay, James Yorke, and Edward Ott. Local Low Dimensionality of Atmospheric Dynamics. *Physical Review Letters*, 86(26):5878–5881, June 2001.
- [9] DJ Estep. A short course on duality, adjoint operators, Green's functions, and a posteriori error analysis, 2004.
- [10] Nancy L. Baker and Roger Daley. Observation and background adjoint sensitivity in the adaptive observation-targeting problem. *Quarterly Journal of the Royal Meteorological Society*, 126(565):1431–1454, July 2007.