Predicting Flow Reversals in a Computational Fluid Dynamics Simulated Thermosyphon using Data Assimilation - Submission to PLOS Journals

Andrew J. Reagan[1,*], Yves Dubief [2], Peter Sheridan Dodds[1], Christopher M. Danforth[1]

1 Department of Mathematics & Statistics, Vermont Complex Systems Center, Computational Story Lab, & the Vermont Advanced Computing Core, The University of Vermont, Burlington, VT 05405
2 School of Engineering, Vermont Complex Systems Center & the Vermont Advanced Computing Core, The University of Vermont, Burlington, VT 05405

* andrew.reagan@uvm.edu

Abstract

Abstract

A thermal convection loop is a annular chamber filled with water, heated on the bottom half and cooled on the top half. With sufficiently large forcing of heat, the direction of fluid flow in the loop oscillates chaotically, dynamics analogous to the Earth's weather. As is the case for state-of-the-art weather models, we only observe the statistics over a small region of state space, making prediction difficult. To overcome this challenge, data assimilation (DA) methods, and specifically ensemble methods, use the computational model itself to estimate the uncertainty of the model to optimally combine these observations into an initial condition for predicting the future state. Here, we build and verify four distinct DA methods, and then, we perform a twin model experiment with the computational fluid dynamics simulation of the loop using the Ensemble Transform Kalman Filter (ETKF) to assimilate observations and predict flow reversals. We show that using adaptively shaped localized covariance outperforms static localized covariance with the ETKF, and allows for the use of less observations in predicting flow reversals. We also show that a Dynamic Mode Decomposition (DMD) of the temperature and velocity fields recovers the low dimensional system underlying reversals, finding specific modes which together are predictive of reversal direction.

Introduction

Prediction of the future state of complex systems is a fundamental challenge of science and engineering, and ultimately integral to the functioning of society. Some of these systems include weather [1], health [2], the economy [3], marketing [4] and transportation [5]. For weather in particular, predictions are made using supercomputers integrating numerical weather models, projecting our current best guess of the atmospheric state into the future. The accuracy of these predictions depends on the accuracy of the models themselves, and the quality of our knowledge of the current state of the atmosphere.

Model accuracy has improved with better meteorological understanding of weather processes and advances in computing technology [6]. To solve the initial value problem,

techniques developed over the past 50 years are now broadly known as *data assimilation* (DA). Formally, data assimilation is the process of using all available information, including short-range model forecasts and physical observations, to estimate the current state of a system as accurately as possible [7]. The best-guess of the current state is often referred to as the *analysis* state.

Here, we employ a fluid dynamics experiment as a test bed for improving numerical weather prediction algorithms, focusing specifically on data assimilation methods. Our approach is inspired by the historical development of current methodologies, and provides a tractable system for rigorous analysis. The experiment is a thermal convection loop, which by design simplifies our problem into the prediction of natural convection. The thermosyphon, a type of natural convection loop or non-mechanical heat pump, can be likened to a toy model of climate [8]. The dynamics of thermal convection loops have been explored under both periodic [9] and chaotic [7, 10–19] regimes. A full characterization of the computational behavior of a loop under flux boundary conditions by Louisos et. al. describes four regimes: chaotic convection with reversals, high Rayleigh number (Ra) aperiodic stable convection, steady stable convection, and conduction/quasi-conduction [20]. For the remainder of this work, we focus on the chaotic flow regime.

Physical Experiment and Computational Model

The reduced order system describing a thermal convection loop was originally derived by Gorman [13] and Ehrhard and Müller [14]. Here we present this three dimensional system in non-dimensionalized form. In S2 Appendix, we present a more complete derivation of these equations, following the derivation of Harris [8]. For the mean fluid velocity $\frac{\mathrm{d}x_1}{\mathrm{d}t}$, temperature difference between the 3 o'clock and 9 o'clock positions $\frac{\mathrm{d}x_2}{\mathrm{d}t}$ (also referred to presently as ΔT_{3-9}), and deviation from conductive temperature profile $\frac{\mathrm{d}x_3}{\mathrm{d}t}$, these equations are:

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = \alpha(x_2 - x_1), \tag{1}$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = \beta x_1 - x_2(1 + Kh(|x_1|)) - x_1 x_3, \tag{2}$$

$$\frac{\mathrm{d}x_3}{\mathrm{d}t} = x_1 x_2 - x_3(1 + Kh(|x_1|)). \tag{3}$$

The function $h(x)$ is a defined piece-wise analytic polynomial, and is provided in the full derivation as Equation S14. The parameters α, β, and K, along with scaling factors for time and each model variable can be fit to data using standard parameter estimation techniques.

Operated by Dave Hammond, UVM's Scientific Electronics Technician, the experimental thermosyphons access the chaotic regime of state space found in the principled governing equations. We quote the detailed setup from Darcy Glenn's undergraduate thesis [21] and provide Fig. 2 for details of the experiment:

> The [thermosyphon] is a bent semi-flexible plastic tube with a 10-foot heating rope wrapped around the bottom half of the upright circle. The tubing used is light-transmitting clear THV from McMaster-Carr, with an inner diameter of 7/8 inch, a wall thickness of 1/16 inch, and a maximum operating temperature of 200F. The outer diameter of the circular thermosyphon is 32.25 inches. Together, the tubing inner diameter and outer diameter of the thermosyphon produce a ratio of approximately 1:36. There are 1 inch 'windows' when the heating cable is coiled in a helix

pattern around the outside of the tube, so the heating is not exactly
uniform. The bottom half is then insulated using aluminum foil, which
allowed fluid in the bottom half to reach 176F. A forcing of 57 V, or 105
Watts, is required for the heating cable so that chaotic motion is observed.
Temperature is measured at the 3 o'clock and 9 o'clock positions using
unsheathed copper thermocouples from Omega.

We confirm that the experiment accesses the chaotic regime of state space using a time
series of the temperature difference as measured at the 3 o'clock and 9 o'clock positions
in Fig. 1. We first test our ability to predict this experimental thermosyphon using
synthetic data.

**Figure 1. A time series of the physical thermosyphon, from the
Undergraduate Honor's Thesis of Darcy Glenn [21].** The temperature
difference (plotted) is taken as the difference between temperature sensors in the 3 and
9 o'clock positions. The sign of the temperature difference indicates the flow direction,
where positive values are clockwise flow. We note that the experimental thermosyphon
is not perfectly balanced, resulting in non-symmetric residence in each flow direction.

**Figure 2. Schematic of the experimental, and computational, setup from
Harris *et al.*(2012).** The loop radius is given by R and inner radius by r. The top
temperature is labeled T_c and bottom temperature T_h, gravity g is defined downward,
the angle ϕ is prescribed from the 6 o'clock position, and temperature difference
between 3 o'clock and 9 o'clock positions ΔT_{3-9} is labeled.

We perform all computational simulations of the thermal convection loop with the
open-source finite volume C++ library OpenFOAM [22]. The open-source nature of
this software enables its integration with the data assimilation framework that our
present work provides.

We consider the incompressible Navier-Stokes equations with the Boussinesq
approximation to model the flow of water inside a thermal convection loop. For brevity,
we omit the equations themselves, and include them in S1 Appendix. The solver in
OpenFOAM that we use, with some modification, is `buoyantBoussinesqPimpleFoam`.
Solving is accomplished by the Pressure-Implicit Split Operator (PISO) algorithm [23].
We find that modification of the code is necessary for laminar operation.

We create both 2-dimensional and 3-dimensional meshes using OpenFOAM's native
meshing utility `blockMesh` shown in Figures 3 and 4. After creating a mesh, we refine
the mesh near the walls to capture boundary layer phenomena and renumber the mesh
for solving speed. We use the `refineWallMesh` utility to refine the mesh near walls,
and the `renumberMesh` utility to renumber the mesh. The resulting 2D mesh contains
80,000 points (80 across the diameter and 1000 around).

Figure 3. A snapshot of the mesh used for CFD simulations. Shown is an
initial stage of heating for a fixed value boundary condition, 2D, laminar simulation
with a mesh of 40000 cells without wall refinement with walls heated at 340K on the
bottom half and cooled to 290K on the top half. The cells have been colored with a
truncated temperature range (299–301K) to highlight the flow structures.

Available boundary conditions (BCs) we find to be stable in OpenFOAM's solver are
constant gradient, fixed value conditions, and turbulent heat flux. Constant gradient
simulations are stable, but the behavior is empirically different from our physical
system. While it is possible that a fixed value BC is acceptable due to the thermal

Figure 4. The 3D mesh viewed as a wire-frame from within. Here there are
900 cells in each slice (not shown), for a total mesh size of 81,000 cells. Simulations
using this computational mesh are prohibitively expensive for use in a real time
ensemble forecasting system, but are possible offline.

diffusivity and thickness of the walls of the experimental setup, we find that this is also
inadequate. Simulations with a turbulent heat flux BC implemented through the
`externalWallHeatFluxTemperature` library are unstable with the laminar turbulence
model we use and resulted in physically unrealistic results. We employ the third-party
library `groovyBC` to use a gradient condition that computes the flux using a fixed
external temperature T_{inf} and fixed wall heat transfer coefficient h as

$$-\frac{\partial T}{\partial x_j} = h\left(T - T_{\text{inf}}\right)$$

where we choose h to be the reference value for aluminum (the material used in the   73
experimental setup).   74
   With the mesh, BCs, and solver chosen, we now simulate the flow. From the data of   75
$T, \phi, \vec{u}$ and p that are saved at each timestep (temperature, cell face flux, velocity, and   76
pressure, respectively), we extract the mass flow rate and average temperature at the   77
$12, 3, 6$ and 9 o'clock positions on the loop. Since ϕ is saved as a face-value flux, we   78
compute the mass flow rate over the cells i of top (12 o'clock) slice as   79

$$\sum_i \phi_{f(i)} \cdot v_i \cdot \rho_i \tag{4}$$

where $f(i)$ corresponds the face perpendicular to the loop angle at cell i and ρ is   80
reconstructed from the Boussinesq approximation $\rho = \rho_{\text{ref}}(1 - \beta(T - T_{\text{ref}}))$.   81

Methods   82

Data Assimilation   83

We perform initial tests of the data assimilation algorithms described here with the   84
Lorenz '63 system, which is analogous to the above equations with Lorenz's $\beta = 1$, and   85
$K = 0$. The canonical choices of $\sigma = 10, \beta = 8/3$ and $\rho = 28$ produce the well known   86
butterfly attractor, and we use these values for all examples here. From these tests, we   87
will find the optimal data assimilation parameters (inflation factors) for predicting time 88
series with this system. Having done so, we then focus our efforts on making prediction   89
using computational fluid dynamics models.   90
   We first implement the 3D-Var filter. Simply put, 3D-Var is the variational   91
(cost-function) approach to finding the analysis. It has been shown that 3D-var solves   92
the same statistical problem as optimal interpolation (OI) [24]. The usefulness of the   93
variational approach comes from the computational efficiency, when solved with an   94
iterative method. Specifically, the multivariate 3D-Var amounts to finding the $\mathbf{x}_a$ that   95
minimizes the cost function   96

$$J(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + (\mathbf{y}_o + H(\mathbf{x}))^T \mathbf{R}(\mathbf{y}_o - H(\mathbf{x})). \tag{5}$$

   Next, we implement the "gold-standard" Extended Kalman Filter (EKF). The   97
tangent linear model (TLM) is precisely the model (written as a matrix) that   98
transforms a perturbation at time t to a perturbation at time $t + \Delta t$, analytically   99
equivalent to the Jacobian of the model. Using the notation of Kalnay [25], this   100

amounts to making a forecast with the nonlinear model M, and updating the error covariance matrix $\mathbf{P}$ with the TLM L, and adjoint model L^T:

$$\mathbf{x}^f(t_i) = M_{i-1}[\mathbf{x}^a(t_{i-1})],$$
$$\mathbf{P}^f(t_i) = L_{i-1}\mathbf{P}^a(t_{i-1})L_{i-1}^T + \mathbf{Q}(t_{i-1})$$

where $\mathbf{Q}$ is the noise covariance matrix (model error). In the experiments with Lorenz '63 presented in this section, $\mathbf{Q} = 0$ since our model is perfect. In numerical weather prediction, $\mathbf{Q}$ must be approximated, e.g., using statistical moments on the analysis increments [26–28].

The analysis step is then written as (for H the observation operator):

$$\mathbf{x}^a(t_i) = \mathbf{x}^f(t_i) + \mathbf{K}_i\mathbf{d}_i, \tag{6}$$
$$\mathbf{P}^a(t_i) = (\mathbf{I} - \mathbf{K}_i\mathbf{H}_i)\mathbf{P}^f(t_i) \tag{7}$$

where

$$\mathbf{d}_i = \mathbf{y}_i^o - \mathbf{H}[x^f(t_i)]$$

is the innovation. We compute the Kalman gain matrix to minimize the analysis error covariance P_i^a as

$$\mathbf{K}_i = \mathbf{P}^f(t_i)\mathbf{H}_i^T[\mathbf{R}_i + \mathbf{H}_i\mathbf{P}^f(t_i)\mathbf{H}^T]^{-1}$$

where $\mathbf{R}_i$ is the observation error covariance. Since we are making observations of the truth with random normal errors of standard deviation ϵ, the observational error covariance matrix $\mathbf{R}$ is a diagonal matrix with ϵ along the diagonal. The most difficult (and most computationally expensive) part of the EKF is deriving and integrating the TLM. For this reason, the EKF is not used operationally, and later we will turn to statistical approximations of the EKF using ensembles of model forecasts. With our CFD model we have no such TLM, and we provide more detail on the TLM approaches applicable to the Lorenz '63 system in S3 Appendix.

The computational cost of the EKF is mitigated through the approximation of the error covariance matrix $\mathbf{P}_f$ from the model itself, without the use of a TLM. One such approach is the use of a forecast ensemble, where a collection of models (ensemble members) are used to statistically sample model error propagation. With ensemble members spanning the model analysis error space, the forecasts of these ensemble members are then used to estimate the model forecast error covariance.

The only difference between this approach and the EKF, in general, is that the forecast error covariance $\mathbf{P}^f$ is computed from the ensemble members, without the need for a tangent linear model:

$$\mathbf{P}^f \approx \frac{1}{K-2}\sum_{k\neq l}\left(\mathbf{x}_k^f - \overline{\mathbf{x}}_l^f\right)\left(\mathbf{x}_k^f - \overline{\mathbf{x}}_l^f\right)^T.$$

The ETKF introduced by Bishop is one type of square root filter, and we present it here to provide background for the formulation of the LETKF [29]. For a square root filter in general, we begin by writing the covariance matrices as the product of their matrix square roots. Because $\mathbf{P}_a$ and $\mathbf{P}_f$ are symmetric positive-definite (by definition), we can write

$$\mathbf{P}_a = \mathbf{Z}_a\mathbf{Z}_a^T \quad , \quad \mathbf{P}_f = \mathbf{Z}_f\mathbf{Z}_f^T \tag{8}$$

where $\mathbf{Z}_a$ and $\mathbf{Z}_f$ are the matrix square roots of $\mathbf{P}_a$ and $\mathbf{P}_f$. We are not concerned that this decomposition is not unique, and note that $\mathbf{Z}$ must have the same rank as $\mathbf{P}$ which will prove computationally advantageous. The power of the SRF is now seen as

we represent the columns of the matrix $\mathbf{Z}_f$ as the difference from the ensemble members from the ensemble mean, to avoid forming the full forecast covariance matrix $\mathbf{P}_f$. The ensemble members are updated by applying the model M to the states $\mathbf{Z}_f$ such that an update is performed by

$$\mathbf{Z}_f = M\mathbf{Z}_a. \qquad (9)$$

To summarize, the steps for the ETKF are to (1) form $\mathbf{Z}_f^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Z}_f$, assuming that computing $\mathbf{R}^{-1}$ is easy, and (2) compute its eigenvalue decomposition, and apply it to $\mathbf{Z}_f$.

The LEKF implements a strategy that becomes important for large simulations: localization. Namely, the analysis is computed for each grid-point using only local observations, without the need to build matrices that represent the entire analysis space. Localization removes long-distance correlations from $\mathbf{B}$ and allows greater flexibility in the global analysis by allowing different linear combinations of ensemble members at different spatial locations [30]. The general formulation of the LEKF by Ott goes as follows, quoting directly from [31]:

1. Globally advance each ensemble member to the next analysis timestep. Steps 2–5 are performed for each grid point.

2. Create local vectors from each ensemble member.

3. Project that point's local vectors from each ensemble member into a low dimensional subspace as represented by perturbations from the mean.

4. Perform the data assimilation step to obtain a local analysis mean and covariance.

5. Generate local analysis ensemble of states.

6. Form a new global analysis ensemble from all of the local analyses.

7. Wash, rinse, and repeat.

Proposed by Hunt *et al.*(2007) with the stated objective of computational efficiency, the LETKF is named from its most similar algorithms from which it draws [32]. With the formulation of the LEKF and the ETKF given, the LETKF can be described as a synthesis of the advantages of both of these approaches. The LETKF is the method sufficiently efficient for implementation on the full OpenFOAM CFD model of 240,000 model variables, and so we present it in more detail and follow the notation of Hunt *et al.*(2007). As in the LEKF, we explicitly perform the analysis for each grid point of the model. The choice of observations to use for each grid point can be selected a priori, and tuned adaptively. Starting with a collection of background forecast vectors $\{\mathbf{x}_{b(i)} : i = 1, \ldots, k\}$, we perform steps 1 and 2 in a global variable space, then steps 3–8 for each grid point:

1. Apply H to $\mathbf{x}_{b(i)}$ to form $\mathbf{y}_{b(i)}$, average the $\mathbf{y}_b$ for $\overline{\mathbf{y}_b}$, and form $\mathbf{Y}b$.

2. Similarly form $\mathbf{X}_b$. Now for each grid point:

3. Form the local vectors.

4. Compute $\mathbf{C} = (\mathbf{Y}_b)^T\mathbf{R}^{-1}$ (perhaps by solving $\mathbf{R}\mathbf{C}^T = \mathbf{Y}_b$.

5. Compute $\tilde{\mathbf{P}}_a = ((k-1)\mathbf{I}/\rho + \mathbf{C}\mathbf{Y}_b)^{-1}$ where $\rho > 1$ is a tun-able covariance inflation factor.

6. Compute $\mathbf{W}_a = \left((k-1)\tilde{\mathbf{P}}_a\right)^{1/2}$.

7. Compute $\overline{\mathbf{w}}_a = \tilde{\mathbf{P}}_a \mathbf{C} \left(\mathbf{y}_o - \overline{\mathbf{y}}_b \right)$ and add it to the column of $\mathbf{W}_a$.

176

8. Multiply $\mathbf{X}_b$ by each $\mathbf{w}_{a(i)}$ and add $\tilde{\mathbf{x}}_b$ to get $\left\{ \mathbf{x}_{a(i)} : i = 1, \ldots, k \right\}$ to complete each grid point.

177
178

9. Combine all of the local analysis into the global analysis.

179

We implement the LETKF on our mesh using the full 80 cells across with zone sizes of center 10, and sides 15, resulting in 3200 local variables for 100 zones. In parallel, these 100 local computations can all be carried out simultaneously over an arbitrary number of processors.

180
181
182
183

Adaptive covariance localization

184

Using the "square" sections of the loop to localize, we shift the zone to the left or right to follow the dominate flow direction at the center of that local window. In Fig. 5 a schematic of localization using square, circular, and adaptive location shows a situation in which adaptive localization will potentially capture more relevant information for finding the analysis state of any given cell. As we note in the caption of Fig. 5, while we are motivated by localization around flow structures like Panel C, we simply shift the covariance in Panel A so that our method is most general and computationally efficient.

185
186
187
188
189
190
191

Figure 5. Schematic of the adaptive covariance localization. In Panel A we see a zonal (square) covariance that is most similar to the covariance used for both control experiments and sliding covariance experiments. Panel B shows a localized covariance using a "local radius", and Panel C shows an idealized, fully adaptive covariance. While we are motivated by localization around flow structures like Panel C, we simply shift the covariance in Panel A so that our method is most general and computationally efficient.

Denote the velocity vector of cells on a perpendicular slice of the loop at $\vec{U}$, the tangent vector to the slice $\vec{U}$ by $\vec{T}$, the zone width as $z_{\max}$ and then the localization shift α_{local} for that slice of the loop is taken to be

192
193
194

$$\alpha_{\text{local}} = \text{floor}\left((\vec{U} \cdot \vec{T}) / \max(U) \times z_{\max} \right). \tag{10}$$

Dynamic mode decomposition

195

We employ the "standard" algorithm of Tu to compute the Dynamic Mode Decomposition [33]. Tu's "standard" algorithm is as follows with X and Y taken as the first and last $N - 1$ columns of the snapshot matrix D:

$$X = U\Sigma V \qquad\qquad\qquad \text{(Take SVD of } X.)$$
$$\tilde{A} = U^T Y V \Sigma^{-1} \qquad\qquad\qquad \text{(Build the } A \text{ matrix.)}$$
$$\tilde{A}w = \lambda w \qquad\qquad\qquad \text{(Compute eigenvectors and values.)}$$
$$\hat{\theta}w = Uw \qquad\qquad\qquad \text{(Compute corresponding modes.)}$$

Given a system state U^* we project this state onto the DMD basis by taking the real part of $\Phi = \text{re}\,(U^* \cdot w)$ and use the psuedoinverse to compute the projection as

$$(\Phi^T \cdot \Phi)^{-1} \cdot \Phi^T \cdot U^*.$$

This projection is a vector which contains the linear coefficients on the basis of DMD modes for the given state.

196
197

Results

Data assimilation

We confirm the performance of the DA methods described above by testing each (on the Lorenz '63 system) for increasingly long times between observations, by increasing the DA window length in Fig. 6. As the time between observations increases, the nonlinearity of the Lorenz '63 system results in the failure of the EKF and difficultly for the EnKF with small ensemble size. The ETKF and EnSRF perform the best of the methods tested and we chose the ETKF for future use with the CFD model.

Figure 6. The RMS error (not scaled by climatology) for our EKF and EnKF filters. Error is measured as the difference between forecast and truth at the end of an assimiliation window for the latter 2500 assimiliation windows in a 3000 assimilation window Lorenz '63 run. Error is measured in the only observed variable, x_1. Increasing the assimilation window led to an decrease in predictive skill, as expected.

The results in Fig. 6 rely on tuned covariance inflation, both additive and multiplicative, pre-computed for each window and DA technique. We choose optimal additive inflation μ and multiplicative inflation Δ by selecting for the lowest error in an exhaustive search through a maximum factors of 1.5 in each, an example is shown in Fig. 7. We use these optimal data assimilation parameters (inflation factors) for the remainder of this work.

Figure 7. The RMS error averaged over 100 model runs of length 1000 windows is reported for the ETKF for varying additive and multiplicative inflation factors Δ and μ. Each of the 100 model runs starts with a random IC, and the analysis forecast starts randomly. The window length here is 390 seconds. The filter performance RMS is computed as the RMS value of the difference between forecast and truth at the assimiliation window for the latter 500 windows, allowing a spin-up of 500 windows.

Limited observations & adaptive covariance

An initial test of prediction skill with limited observations in a twin model experiment showed that we needed 1000 spatial measurements of the temperature to predict flow reversals within 1 assimilation window. In an attempt to decrease the required observations to a experimentally realizable number, we implement a simple, adaptively localized covariance for data assimilation. Since we first saw a modest improvement in the prediction skill with full temperature observations, we hope that this improvement increases and is sufficient to get down to needing as few as 32 observations to predict reversals 1 assimilation window (of length 10 seconds) into the future.

In Fig. 8 we see that over an assimilation of 200 seconds, the ensemble converges on the hidden, true state. To test the performance of flow reversal prediction, we take the average of the ensemble flow direction (the average of each value of ϕ) as the predicted flow direction, and count how often we predict reversals both when they do and do not occur. Varying both the number of model variables and the strength of covariance shifting in Fig. 9, we find that covariance shifting improves flow reversal prediction skill even when spatial observation density is decreased. With full observations (spacing of 1), we obtain a the best predictions with a covariance shift of 2. For 1/2 and 1/5 observations [a spacing of 2 (5) to observe every other (fifth) variable], we again have

Figure 8. Convergence of 20 ensembles using sliding windows, starting from initially random states. Here, as in most of the experiments, only temperature is observed and assimilated. Flux is computed as in Equation 4, on the left hand side of the thermosyphon, and scaled by a factor of 10^8. Assimilation takes place every 10 model seconds.

Figure 9. Prediction skill as fraction of reversals that we correctly predicted across different numbers of observations and sliding windows of localized covariance. Decreasing observation density makes the prediction problem more difficult while at the same time make the data assimilation stable numerically, and we see a decrease in prediction skill with no covariance shifting. With covariance shifting, skill improves for each observational density and most dramatically with less observation density.

the best predictions with a shift of 2. And for a spacing of 10, observing every 10th   230
variable, we achieve greater prediction skill with a covariance shift of 10.   231

Computing the average flow direction inside a localized covariance zone is   232
straightforward, and computationally easy since the velocity is immediately available,   233
making incorporation of this scheme into any data assimilation method easy. Since   234
observations are also sparse in large weather models, we expect that using an adaptive   235
local covariance scheme could lead to improved prediction skill with sparse   236
observations [34].   237

Dynamic mode decomposition   238

To incorporate limited observations into a high-dimensional CFD simulation, we   239
combine ideas from both CFD literature and data assimilation to make predictions. We   240
proceed with Tu's algorithm using snapshots every 10 seconds for the first 900 seconds of   241
model time. A full picture of this time series can be found in the Figure in S4 Appendix.   242

In this reduced space, we extract the modes that correspond to the instability leading   243
to flow reversals. With a known low-dimensional model of the thermosyphon dynamics,   244
we take this opportunity to test whether DMD can discover the underlying system. The   245
time series of the model state projection onto a specific DMD mode will represent the   246
time dynamics of a mode that is representative of a single low dimensional variable.   247

To look at all of the modes at once, we examine the average magnitude of the   248
projection from all model states onto each mode in comparison to the projection of all   249
states that 1,3,5, and 7 time steps before a reversal. The magnitude of the mode   250
projection of a predictive mode before a reversal should stand out against the projection   251
average across all states, and decay back towards the average further from the reversal   252
in time. For modes 21 and 79, we directly observe in Fig. 10 that the average projection   253
from states just 1 second before reversal is the most different from the average state   254
projection, and the further away from the reversal the more similar the states become to   255
the average.   256

Figure 10. The $\log_{10}$ average projection onto each DMD mode for different sets of model states. DMD constructed as snapshots every 10 seconds for the first 900 seconds of model time, and model states from the first 2000 seconds are all projected onto the DMD modes. All states average shown in black, and the average of the subset of states that occur 1 second, 3 seconds, 5 seconds, and 7 seconds before a reversal are shown in other colors. The symmetry of the loop generates modes that often come in pairs.

We are particularly interested in whether the mode projection time series is predictive of flow reversals, as is true with the hidden system. The insets of Panel A and Panel B in Fig. 11 show two such timeseries, with stars indicating the time of flow reversals. Individually, these modes increase in amplitude when flow reversals happen. As a dominant mode, the time series of Mode 2 tracks closely to the timeseries from which the modes were generated, while the dynamics of the projection of Mode 79 are less obvious.

Figure 11. Panel A: The temperature profile of the thermosyphon of Mode 2, with inset of the projection of time series states onto Mode 2 (the projection coefficient). The color scale on the thermosyphon spans the values 1 to 0 in the DMD mode. The inset figure is the projection coefficient from time 100 to time 5000, with the projection range being shown from -300 to 300 (as in Panel C) and the starred reversals labeled as in Panel C. Panel B: Likewise, the temperature profile of the thermosyphon of Mode 79, with inset of the projection of time series states onto Mode 79 (the projection coefficient). The color scale and inset figure axes are the same as Panel A. Panel C: A butterfly-shaped phase plane shows the value of the projection onto modes 2 and 79 for each time in the first 2000 time steps of our ground truth model run. In blue and green stars the states that occur directly before a flow are highlighted, and are isolated into separate quadrants of phase space.

By combining the state projection onto specific mode time series into a phase plane, the combined signal from two modes is used for discovering states that separate reversals in direction and from other states in the phase plane. In Fig. 11 we see that the dominant dynamics from mode 2 plotted with those of mode 79 are able to strongly separate reversals into quadrants of the low-dimensional space. This result indicates that DMD could be used to improve predictability of reversals.

Concluding Remarks

The first output of our work is a general data assimilation framework for MATLAB and Julia. By utilizing an object-oriented (OO) design, the model and data assimilation algorithm code are separate and can be changed independently. The principal advantage of this approach is the ease of incorporation of new models and DA techniques (code available at `https://github.com/andyreagan/julia-openfoam`).

We next present the results pertaining to the accuracy of forecasts for synthetic data (twin model experiments). There are many possible experiments given the choice of assimilation window, data assimilation algorithm, localization scheme, model resolution, observational density, observed variables, and observation quality. We focused on considering the effect of observations and observational locations on the resulting forecast skill, and we find that there is a threshold for the required number of observations to make useful predictions. In general, and unsurprisingly, we see that increasing observational density leads to improved forecast accuracy. With too few observations, the data assimilation is unable to recover the underlying dynamics. Using adaptively localized covariance holds promise for data assimilation with data-scarce models, to overcome the lack of data.

The ability of DMD to recover the lower dimensional dynamics is expected but with 240,000 variables is nonetheless an accomplishment. When modeling systems for which there are unknown but useful dimension reductions, as demonstrated here, DMD can be a useful tool to find such dimension reductions. When computational model runs are exceedingly costly or time consuming, the best-guess state projection onto DMD modes provides insights into the system dynamics that could not otherwise be obtained.

The numerical coupling of CFD to experiment by DA should be generally useful to improve the skill of CFD predictions of experiments. In addition, the CFD model can provide better knowledge of unobservable quantities of interest in fluid flow that use the experimental data to find the analysis state provided by DA. Adaptive covariance localization further enhances the benefit provided by DA in this context.

References

1. Hsiang SM, Burke M, Miguel E. Quantifying the Influence of Climate on Human Conflict. Science. 2013;341(6151). Available from: http://www.sciencemag.org/content/341/6151/1235367.abstract.

2. Ginsberg J, Mohebbi MH, Patel RS, Brammer L, Smolinski MS, Brilliant L. Detecting influenza epidemics using search engine query data. Nature. 2008;457(7232):1012–1014.

3. Sornette D, Zhou WX. Predictability of large future changes in major financial indices. International Journal of Forecasting. 2006;22(1):153–168.

4. Asur S, Huberman BA. Predicting the future with social media. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. vol. 1. IEEE; 2010. p. 492–499.

5. Savely R, Cockrell B, , Pines S. Apollo Experience Report - Onboard Navigational and Alignment Software. Technical Report. 1972;.

6. Bauer P, Thorpe A, Brunet G. The quiet revolution of numerical weather prediction. Nature. 2015;525(7567):47–55.

7. Yang SC, Baker D, Li H, Cordes K, Huff M, Nagpal G, et al. Data Assimilation as Synchronization of Truth and Model: Experiments with the Three-Variable Lorenz System*. Journal of the atmospheric sciences. 2006;63(9):2340–2354.

8. Harris KD, Ridouane EH, Hitt DL, Danforth CM. Predicting flow reversals in chaotic natural convection using data assimilation. arXiv preprint arXiv:11085685. 2011;.

9. Keller JB. Periodic oscillations in a model of thermal convection. J Fluid Mech. 1966;26(3):599–606.

10. Welander P. On the oscillatory instability of a differentially heated fluid loop. International Geophysics Series. 1995;59.

11. Creveling H, PAZ D, Baladi J, Schoenhals R. Stability characteristics of a single-phase free convection loop. Journal of Fluid Mechanics. 1975;67(part 1):65–84.

12. Gorman M, Widmann P. Chaotic flow regimes in a convection loop. Phys Rev Lett. 1984;(52):2241–2244.

13. Gorman M, Widmann P, Robbins K. Nonlinear dynamics of a convection loop: a quantitative comparison of experiment with theory. Physica D. 1986;(19):255–267.

14. Ehrhard P, Müller U. Dynamical behaviour of natural convection in a single-phase loop. Journal of Fluid mechanics. 1990;217:487–518.

15. Yuen P, Bau H. Optimal and adaptive control of chaotic convection. Phys Fluids. 1999;(11):1435–1448.

16. Jiang Y, Shoji M. Spatial and temporal stabilities of flow in a natural circulation loop: influences of thermal boundary condition. J Heat Trans. 2003;(125):612–623.

17. Burroughs E, Coutsias E, Romero L. A reduced-order partial differential equation model for the flow in a thermosyphon. Journal of Fluid Mechanics. 2005;543:203–238.

18. Desrayaud G, Fichera A, Marcoux M. Numerical investigation of natural circulation in a 2D-annular closed-loop thermosyphon. International journal of heat and fluid flow. 2006;27(1):154–166.

19. Ridouane EH, Danforth CM, Hitt DL. A 2-D numerical study of chaotic flow in a natural convection loop. International Journal of Heat and Mass Transfer. 2010;53(1):76–84.

20. Louisos WF, Hitt DL, Danforth CM. Chaotic flow in a 2D natural convection loop with heat flux boundaries. International Journal of Heat and Mass Transfer. 2013;61:565–576.

21. Glenn D. Characterizing weather in a thermosyphon: an atmosphere that hangs on a wall; 2013. Undergraduate Honors Thesis, University of Vermont.

22. Jasak H, Jemcov A, Tukovic Z. OpenFOAM: A C++ library for complex physics simulations. In: International Workshop on Coupled Methods in Numerical Dynamics, IUC, Dubrovnik, Croatia; 2007. p. 1–20.

23. Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. Journal of Computational physics. 1986;62(1):40–65.

24. Lorenc AC. Analysis methods for numerical weather prediction. Quarterly Journal of the Royal Meteorological Society. 1986;112(474):1177–1194.

25. Kalnay E. Atmospheric modeling, data assimilation, and predictability. Cambridge university press; 2003.

26. Danforth CM, Kalnay E, Miyoshi T. Estimating and correcting global weather model error. Monthly weather review. 2007;135(2):281–299.

27. Li H, Kalnay E, Miyoshi T, Danforth CM. Accounting for model errors in ensemble data assimilation. Monthly Weather Review. 2009;137(10):3407–3419.

28. Danforth CM, Kalnay E. Using singular value decomposition to parameterize state-dependent model errors. Journal of the Atmospheric Sciences. 2008;65(4):1467–1478.

29. Bishop CH, Etherton BJ, Majumdar SJ. Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. Monthly weather review. 2001;129(3):420–436.

30. Kalnay E, Li H, Miyoshi T, Yang SC, Ballabrera-Poy J. 4-D-Var or ensemble Kalman filter? Tellus A. 2007;59(5):758–773.

31. Ott E, Hunt BR, Szunyogh I, Zimin AV, Kostelich EJ, Corazza M, et al. A local ensemble Kalman filter for atmospheric data assimilation. Tellus A. 2004;56(5):415–428.

32. Hunt BR, Kostelich EJ, Szunyogh I. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. Physica D: Nonlinear Phenomena. 2007;230(1):112–126.

33. Tu JH, Rowley CW, Luchtenburg DM, Brunton SL, Kutz JN. On dynamic mode decomposition: theory and applications. arXiv preprint arXiv:13120041. 2013;.

34. Bishop CH, Hodyss D. Adaptive Ensemble Covariance Localization in Ensemble 4D-VAR State Estimation. Monthly Weather Review. 2011 April;139(4):7.

35. Reagan A. Predicting Flow Reversals in a Computational Fluid Dynamics Simulated Thermosyphon using Data Assimilation. arXiv:13122142. 2013;.

36. Patankar SV, Spalding DB. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. International Journal of Heat and Mass Transfer. 1972;15(10):1787–1806.

37. Ferziger JH, Perić M. Computational methods for fluid dynamics. vol. 3. Springer Berlin; 1996.

38. Rall LB. Automatic Differentiation: Techniques and Applications (Lecture Notes in Computer Science). Springer; 1981.

39. Burgers G, Jan van Leeuwen P, Evensen G. Analysis scheme in the ensemble Kalman filter. Monthly weather review. 1998;126(6):1719–1724.

40. Anderson MJ. A new method for non-parametric multivariate analysis of variance. Austral Ecology. 2001;26(1):32–46.

41. Tippett MK, Anderson JL, Bishop CH, Hamill TM, Whitaker JS. Ensemble square root filters*. Monthly Weather Review. 2003;131(7):1485–1490.

42. Evensen G. The ensemble Kalman filter: Theoretical formulation and practical implementation. Ocean dynamics. 2003;53(4):343–367.

Supporting Information captions

S1 Appendix. Computational Details and Explicit Equations Used.

S2 Appendix. The Ehrhard and Müller Equations.

S3 Appendix. Data Assimiliation.

S4 Appendix. Additional DMD Details.