

Computational Science
Laboratory Technical Report
CSL-TR-3-2013
July 22, 2013

Alexandru Cioaca,
Adrian Sandu, and Eric de Sturler

*Efficient methods for computing
observation impact
in 4D-Var data assimilation*

Computational Science Laboratory
Computer Science Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060
Phone: (540)-231-2193
Fax: (540)-231-6075
Email: sandu@cs.vt.edu
Web: <http://csl.cs.vt.edu>



Innovative Computational Solutions



Low-rank Approximations for Computing Observation Impact in 4D-Var Data Assimilation

Alexandru Cioaca, Adrian Sandu

Abstract

We present an efficient computational framework to quantify the impact of individual observations in four dimensional variational data assimilation. The proposed methodology uses first and second order adjoint sensitivity analysis, together with matrix-free algorithms to obtain low-rank approximations of observation impact matrix. We illustrate the application of this methodology to important applications such as data pruning and the identification of faulty sensors for a two dimensional shallow water test system.

Contents

1	Introduction	1
2	4D-Var Data Assimilation	2
2.1	Formulation	2
2.2	Computational aspects	3
3	Observation Impact	3
3.1	4D-Var sensitivity to observations	3
3.2	Observation impact matrix	5
3.2.1	Main features	5
3.2.2	Computational ingredients	6
4	Efficient Computation of Observation Sensitivity and Impact	7
4.1	An iterative (serial) approach for matrix-free low-rank approximations with SVD	7
4.2	An ensemble-based (parallel) approach for matrix-free low-rank approximations	9
5	Applications	10
5.1	Test problem: shallow water equations	10
5.2	Data assimilation setup	11
5.3	Experimental results	12
5.3.1	Validating the low-rank approximations of the observation impact matrix	12
5.3.2	Observation impact	13
5.3.3	Pruning observations based on sensitivity values	17
5.3.4	Identifying faulty data using observation sensitivity	18
6	Conclusions and future work	19

1. Introduction

Data assimilation is a dynamic-data driven application that integrates information from physical observations with numerical model predictions. This paper describes a systematic approach to quantify the contribution of each observation data point in improving the model state estimates. The focus is on the four-dimensional variational (4D-Var) data assimilation approach, which fits a model trajectory against time-distributed observations, to obtain maximum likelihood estimates of the initial state and model parameters.

Computing the “observation impact” (OI) means quantifying the individual (or group) contributions of “assimilated” data points to reduce the uncertainty in state or model parameters. OI provides a numerical measure to distinguish between crucial and redundant information, and benefits applications such as big data pruning, error detection/correction, intelligent sensor placement, and other decision-making processes.

There are several approaches available in the scientific literature to compute OI. In early studies OI was associated with the energy of perturbations along the dominant directions of maximum error growth [1, 2]. Other approaches study observation impact using metrics from information theory [3, 4], statistical design [5], and robust control [6]. More recently, observation impact has been assessed through adjoint-sensitivity analysis, as a counterpart measure to observation sensitivity [7–9]. The current state-of-the-art uses the second-order sensitivity equations of 4D-Var system [10–12], a powerful approach followed by the present research study. The efficient implementation of this methodology is hindered by several computational limitations related to the calculation of second-order derivative approximations.

This work develops a systematic and efficient approach to compute OI in a sensitivity analysis framework. Observation impact is formalized as a sensitivity matrix which maps changes in observation space to associated changes in solution space. This matrix is usually not available in explicit (full) form. Our computational approach makes smart use of tangent linear and adjoint models [13–15] to propagate first and second order sensitivities through the data assimilation system, and obtain products of the OI matrix with user defined vectors. Matrix-free linear algebra tools are necessary for this. Eigenvalue and singular value decompositions are used to obtain low-rank approximations that capture the main features of observation impact, while significantly reducing the computational burden. As of recently, low-rank approximations of matrices have become very popular in image processing [16], information retrieval [17], and machine learning [18] to extract correlations and remove noise from data. Two alternative ways to compute a low-rank approximation for observation impact are given, one of serial nature and one highly parallel. Several applications of OI are exemplified using the two dimensional shallow water equations test problem.

The remaining part of the paper is structured as follows. Section 2 reviews the formulation of 4D-Var data assimilation. Section 3 reviews the 4D-Var sensitivity equations necessary for the observation impact matrix (following the original derivation [11]). We discuss the structural and functional properties of

the impact matrix and the required numerical tools for evaluating its action. Section 4 introduces low-rank approximations of the observation impact matrix. Numerical experiments to exemplify and validate the observation impact matrix are presented in Section 5. Conclusions and future directions of research are outlined in Section 6.

2. 4D-Var Data Assimilation

2.1. Formulation

Data assimilation is the process by which model predictions are constrained with real measurements [19, 20]. It combines three sources of information: an a priori estimate of the initial state of the system (“background”), knowledge of the physical laws governing the behavior of the system (captured by a numerical model), and measurements of the real system (“observations”). Data assimilation solves the inverse problem for improving estimates of model states, initial and boundary conditions, or various system parameters.

Four-dimensional variational (4D-Var) assimilation is formulated as a PDE-constrained nonlinear optimization. The improved initial state \mathbf{x}_0^a (“analysis”) is obtained by minimizing the following cost function:

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2} (\mathbf{x}_0 - \mathbf{x}_0^b)^T \cdot \mathbf{B}_0^{-1} \cdot (\mathbf{x}_0 - \mathbf{x}_0^b) \quad (1a)$$

$$+ \frac{1}{2} \sum_{k=1}^N (\mathcal{H}_k(\mathbf{x}_k) - \mathbf{y}_k)^T \cdot \mathbf{R}_k^{-1} \cdot (\mathcal{H}_k(\mathbf{x}_k) - \mathbf{y}_k),$$

$$\mathbf{x}_0^a = \arg \min_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0) \quad \text{subject to } \mathbf{x}_k = \mathcal{M}_{t_0 \rightarrow t_k}(\mathbf{x}_0). \quad (1b)$$

Here \mathcal{M} denotes the numerical model used to evolve the initial state vector \mathbf{x}_0 in time. \mathcal{H}_k denotes the observation operator at assimilation time t_k , and maps the model state $\mathbf{x}_k \approx \mathbf{x}(t_k)$ to the observation space. \mathbf{R}_k is the observation error covariance matrix. A practical assumption is that observations are independent, hence uncorrelated, which reduces \mathbf{R}_k to a diagonal matrix containing observation variances. The covariance matrices \mathbf{B}_0 and \mathbf{R}_k are predefined, and the quality of their approximation influences the resulting analysis.

The first term of the 4D-Var cost functional (1a) quantifies the mismatch between the initial solution (\mathbf{x}_0) and the background state (\mathbf{x}_0^b) at the initial time (t_0). This mismatch is computed in a least-squares sense, scaled by the inverse background error covariance matrix \mathbf{B}_0 . The second term measures the mismatch between the model trajectory (initialized from \mathbf{x}_0) and observations \mathbf{y}_k taken at times t_k , $k = 0, \dots, N$ scaled by the inverse observation error covariances \mathbf{R}_k . Through \mathbf{B}_0 and \mathbf{R}_k the inverse problem takes into account the uncertainty in the data and model predictions. 4D-Var can be viewed as a method of Bayesian inference, which computes the maximum likelihood initial solution conditioned by the observations.

When assimilating observations only at the initial time t_0 the method is known as three-dimensional variational (3D-Var), as the additional time dimension is not present.

2.2. Computational aspects

The minimization problem (1b) is computed numerically using gradient-based iterative nonlinear solvers such as quasi-Newton, nonlinear conjugate gradients, and truncated Newton methods.

The iterative solver starts from an initial guess and advances to a minimizer, along directions of descent computed using the gradients of \mathcal{J} . The descent directions depend on all three available sources of information; intermediate iterations can be viewed as the result of partially assimilating information given by observations.

The iterative nature of the solution approach has several important consequences. The resulting analysis is not the global minimizer, but rather is a local one. When the computational cost per iteration is high, the available computational resources constrain the number of iterations that can be performed. Thus in practice analyses are almost always suboptimal.

We will call the numerical model \mathcal{M} (1b) the *forward model* (FWD) or forecast model. Applying gradient-based methods to minimize (1b) requires the derivatives of \mathcal{J} with respect to the initial model states. This can be achieved using adjoint modeling [13, 14], a methodology that has been successfully implemented in optimization, sensitivity analysis and uncertainty quantification [15, 21, 22]. The adjoint models (ADJ) can be based either on the linearization of the original differential equations (“continuous” adjoint approach), or the linearization of the numerical method (“discrete” adjoint approach). A convenient approach to generate discrete adjoint models is automatic differentiation [23], which takes as input the source code of the FWD model, performs line by line differentiation, and returns the source code of the adjoint models.

3. Observation Impact

3.1. 4D-Var sensitivity to observations

We now establish a relation between the analysis \mathbf{x}_0^a and the observations \mathbf{y}_k . This is done using sensitivity analysis, and expresses how slight changes in the observational data translate into changes in the resulting analysis. This section follows the 4D-Var sensitivity approach of Daescu [11].

Consider the problem of finding a vector $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ that minimizes the twice continuously differentiable cost function:

$$\mathbf{x}^a = \arg \min_{\mathbf{x}} \mathcal{J}(\mathbf{x}, \mathbf{u}). \quad (2)$$

The function depends on the state vector \mathbf{x} and parameter vector $\mathbf{u} \in \mathbb{R}^m$. For any value $\bar{\mathbf{u}}$ of the parameter, the solution obeys the first order optimality condition

$$\nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}^a, \bar{\mathbf{u}}) = 0. \quad (3)$$

Assume that the Hessian of the function is positive definite at the optimum, $\nabla_{\mathbf{x}, \mathbf{x}}^2 \mathcal{J}(\mathbf{x}^a, \bar{\mathbf{u}}) > 0$. The implicit function theorem applied to (3) guarantees

there exists a vicinity of $\bar{\mathbf{u}}$ where the optimal solution is a smooth function of the parameters, $\mathbf{x}^a = \mathbf{x}^a(\mathbf{u})$, and has the sensitivity

$$\nabla_{\mathbf{u}} \mathbf{x}^a(\mathbf{u}) = -\nabla_{\mathbf{u}, \mathbf{x}}^2 \mathcal{J}(\mathbf{u}, \mathbf{x}^a) \cdot (\nabla_{\mathbf{x}, \mathbf{x}}^2 \mathcal{J}(\mathbf{u}, \mathbf{x}^a))^{-1} .$$

We apply this procedure to the 4D-Var cost function (1a) with the parameters \mathbf{u} being the observational data \mathbf{y}_k . The first-order necessary condition reads:

$$\nabla_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0^a) = \mathbf{B}_0^{-1} (\mathbf{x}^a - \mathbf{x}^b) + \sum_{k=1}^N \mathbf{M}_{0,k}^T \mathbf{H}_k^T \mathbf{R}_k^{-1} (\mathcal{H}_k(\mathbf{x}_k) - \mathbf{y}_k) = 0 , \quad (4)$$

where $\mathbf{M}_{0,k} = \mathcal{M}'_{t_0 \rightarrow t_k}(\mathbf{x}_0)$ is the tangent linear propagator associated with the numerical model \mathcal{M} , and $\mathbf{H}_k = \mathcal{H}'_k(\mathbf{x}_k)$ is the linearized observation operator at time t_k . Differentiating (4) with respect to the observations \mathbf{y}_k

$$\nabla_{\mathbf{y}_k, \mathbf{x}_0}^2 \mathcal{J}(\mathbf{x}_0^a) = -\mathbf{R}_k \mathbf{H}_k \mathbf{M}_{0,k}$$

leads to the following expression for the sensitivity of the analysis to observations:

$$\nabla_{\mathbf{y}_k} \mathbf{x}_0^a = \left(\frac{\partial \mathbf{x}_0^a}{\partial \mathbf{y}_k} \right)^T = \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{M}_{0,k} \mathbf{A}_0 , \quad (5)$$

$$\mathbf{A}_0 = (\nabla_{\mathbf{x}_0, \mathbf{x}_0}^2 \mathcal{J}(\mathbf{x}_0^a))^{-1} . \quad (6)$$

Consider a verification cost functional $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ that measures the discrepancy between the analysis and a verification solution \mathbf{x}_0^v :

$$\Psi(\mathbf{x}_0^a) = \frac{1}{2} (\mathbf{x}_0^a - \mathbf{x}_0^v)^T \mathbf{C} (\mathbf{x}_0^a - \mathbf{x}_0^v) . \quad (7)$$

Here the verification solution \mathbf{x}_0^v is also defined at t_0 . The matrix \mathbf{C} is a weighting matrix corresponding to a particular norm or restricting the verification to a subdomain of the solution space. Using chain-rule differentiation and (5), the *sensitivity to observations* (the gradient of Ψ with respect to observations \mathbf{y}_k) is:

$$\begin{aligned} \nabla_{\mathbf{y}_k} \Psi(\mathbf{x}_0^a) &= \nabla_{\mathbf{y}_k} \mathbf{x}_0^a \cdot \nabla_{\mathbf{x}_0^a} \Psi(\mathbf{x}_0^a) \\ &= \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{M}_{0,k} \mathbf{A}_0 \mathbf{C} (\mathbf{x}_0^a - \mathbf{x}_0^v) . \end{aligned} \quad (8)$$

The sensitivity (8) can be computed via the following steps:

- The sensitivity of the verification function with respect to the analysis is

$$\nabla_{\mathbf{x}_0^a} \Psi(\mathbf{x}_0^a) = \mathbf{C} (\mathbf{x}_0^a - \mathbf{x}_0^v) .$$

- A ‘‘supersensitivity’’ is obtained through solving a linear system with the matrix the 4D-Var Hessian eqn:inverse-hessian

$$\mu = \mathbf{A}_0 \cdot \nabla_{\mathbf{x}_0^a} \Psi(\mathbf{x}_0^a) . \quad (9)$$

This step dominates the computational cost of the sensitivity calculation.

- Finally, the verification sensitivity to observations \mathbf{y}_k is the supersensitivity vector propagated to time t_k through the tangent linear model, mapped through the observation selection operator, and scaled with the inverse error covariance:

$$\nabla_{\mathbf{y}_k} \Psi(\mathbf{x}_0^a) = \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{M}_{0,k} \cdot \boldsymbol{\mu}. \quad (10)$$

The sensitivities of Ψ to the background \mathbf{x}_0^b and to the error covariance matrices \mathbf{B} and \mathbf{R} can be derived in a similar fashion [11].

3.2. Observation impact matrix

3.2.1. Main features

Equation (5) defines a matrix whose elements are the sensitivities of each component of the analysis vector \mathbf{x}_0^a to each component of the observation vector \mathbf{y}_k assimilated at time t_k . The *observation impact matrix* collects the sensitivities to all observations as follows:

$$\mathbf{T} = \left(\frac{\partial \mathbf{x}^a}{\partial \mathbf{y}} \right)^T = \begin{pmatrix} \nabla_{\mathbf{y}_1} \mathbf{x}_0^a \\ \nabla_{\mathbf{y}_2} \mathbf{x}_0^a \\ \vdots \\ \nabla_{\mathbf{y}_N} \mathbf{x}_0^a \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1^{-1} \mathbf{H}_1 \mathbf{M}_{0,1} \\ \mathbf{R}_2^{-1} \mathbf{H}_2 \mathbf{M}_{0,2} \\ \vdots \\ \mathbf{R}_N^{-1} \mathbf{H}_N \mathbf{M}_{0,N} \end{pmatrix} \mathbf{A}_0. \quad (11)$$

Each of the n columns of \mathbf{T} represents the sensitivities of one particular model state to all observations. For real applications $n \sim 10^7 - 10^{10}$. Each row of \mathbf{T} contains the sensitivities of each state to one particular observation. In typical data assimilation applications the number of observations (rows) is two-three orders of magnitude smaller than the number of model states (columns).

We now seek to understand the structure of the observation impact matrix (11), which is the transpose of the sensitivity matrix $\partial \mathbf{x}^a / \partial \mathbf{y}$. The impact matrix is the product of submatrices of type $\mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{M}_{0,k}$, with the inverse of the 4D-Var Hessian (6). The Hessian is symmetric and positive-definite when evaluated at the minimum, but can lose positive-definiteness when evaluated at an inexact analysis, such as when the minimization of 4D-Var was incomplete. The inverse of the 4D-Var Hessian at the minimum approximates the analysis error covariance [24]. This a posteriori error covariance characterizes the 4D-Var process and quantifies the uncertainty reduction due to assimilating the observations. Each column of this matrix represents the error covariance corresponding to a certain model state. The tangent linear model $\mathbf{M}_{0,k}$ propagates this error covariance to the observational time t_k . The propagated perturbations are mapped to observation space through \mathcal{H}_k , and then scaled with \mathbf{R}_k^{-1} . The inverse observation error covariance can be interpreted as a measure of trust assigned to each data point. Large values of the result are associated to points in space and time where better measurements can benefit the assimilation process.

This metric can be assessed not only at times when we have observations available, but also at intermediate times, and can prove to be useful for deploying adaptive observations.

Small changes in observations $\Delta \mathbf{y}$ lead to a change in the analysis $\Delta \mathbf{x}_0^a$ which, to first order, can be calculated using the observation impact matrix (11) as

$$\Delta \mathbf{x}_0^a = \mathbf{T}^T \cdot \Delta \mathbf{y}. \quad (12)$$

We call the change in analysis $\Delta \mathbf{x}_0^a$ the *observation impact*. It is computed via the following steps:

- Each observation change $\Delta \mathbf{y}_k$ is scaled and pulled back to the initial time via the adjoint model:

$$\overline{\Delta \mathbf{y}}_k = \mathbf{M}_{0,k}^T \mathbf{H}_k^T \mathbf{R}_k^{-1} \cdot \Delta \mathbf{y}_k.$$

- The aggregated contribution of all observation changes is

$$\overline{\Delta \mathbf{y}} = \sum_{k=0}^N \overline{\Delta \mathbf{y}}_k,$$

and in practice can be computed via *a single* adjoint run.

- The observation impact is obtained by solving a linear system whose matrix is the 4D-Var cost function Hessian

$$\Delta \mathbf{x}_0^a = \mathbf{A}_0 \cdot \overline{\Delta \mathbf{y}}.$$

3.2.2. Computational ingredients

For real models the matrices that appear in (11) are very large. The calculations of observation impact rely on matrix-vector products. The linear operator $\mathbf{M}_{0,k}$ is applied by running the tangent linear model from t_0 to t_k . Application of the linearized observation mapping operator \mathcal{H}_k and scaling by \mathbf{R}_k^{-1} can be performed directly as they are relatively inexpensive.

The inverse Hessian has been successfully used in practical applications to compute the supersensitivity. This was done by solving iteratively a linear system for the supersensitivity, the system matrix being the 4D-Var Hessian. Iterative solvers of Krylov-type require only Hessian-vector products. These products can be computed by running the second-order adjoint model. Furthermore, the linear solvers can be preconditioned for faster convergence and several matrix-free methods are readily available [25]. When the second-order adjoint is not available, several approximations can be used as follows.

1. The finite difference of gradients

$$\nabla_{\mathbf{x}_0, \mathbf{x}_0}^2 \mathcal{J}(\mathbf{x}_0^a) \cdot \mathbf{u} \approx \frac{\nabla_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0^a + \epsilon \cdot \mathbf{u})^T - \nabla_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0^a)^T}{\epsilon}.$$

requires two first-order adjoint runs, which can be performed in parallel. The accuracy of this approximation is typically low.

2. The Gauss-Newton approximation of the Hessian

$$\nabla_{\mathbf{x}_0, \mathbf{x}_0}^2 \mathcal{J}(\mathbf{x}_0^a) \cdot \mathbf{u} \approx \mathbf{B}_0^{-1} \cdot \mathbf{u} + \sum_{k=1}^N \mathbf{M}_{0,k}^T \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{M}_{0,k} \cdot \mathbf{u}$$

discards second order terms which contain the observation residual, so this approximation is independent of the observation data. It requires one tangent-linear model run followed by one first-order adjoint run.

3. Limited memory quasi-Newton approximations are based on the sequence of solutions and gradients generated during the numerical optimization procedure. An example of such an approximation is L-BFGS [26].

4. Efficient Computation of Observation Sensitivity and Impact

The computation of observation sensitivity or impact is a non-trivial task since for practical problems of interest it is not feasible to build the full matrices. We seek to develop computationally inexpensive approximations of the impact matrix that capture the most important features, and whose accuracy can be improved with an increased computational effort. Our approach is based on the Singular Value Decomposition (SVD), a powerful tool to generate low-rank approximations of large matrices. We present two algorithms, one iterative (inherently serial), and one ensembled-based (inherently parallel).

Consider the SVD of (11) and the corresponding low-rank approximations

$$\mathbf{T} = \mathbf{U} \mathbf{S} \mathbf{V}^T, \quad \mathbf{T}_{(p)} = \mathbf{U}_{(p)} \mathbf{S}_{(p)} \mathbf{V}_{(p)},$$

where \mathbf{U} and \mathbf{V} are orthogonal, \mathbf{S} is diagonal, $\mathbf{U}_{(p)}$ and $\mathbf{V}_{(p)}$ are the right and the left singular vectors associated with the largest p singular values, and $\mathbf{S}_{(p)}$ has these dominant singular values on the diagonal formed.

$\mathbf{T}_{(p)}$ has the smallest "reconstruction error" in both Frobenius $\|\mathbf{T} - \mathbf{T}_{(p)}\|_F$ and L^2 norms $\|\mathbf{T} - \mathbf{T}_{(p)}\|_2$ among all the rank p approximations of \mathbf{T} . The accuracy increases as more dominant singular modes are added. The cut-off threshold is particular to the problem under study and can be determined from the singular value spectrum decay.

4.1. An iterative (serial) approach for matrix-free low-rank approximations with SVD

An ideal iterative SVD algorithm for our problem uses one matrix-vector product per iteration, and reveals one new singular pair with each iteration. The singular vectors are discovered in decreasing order of the magnitude of their singular values. Thus, running the algorithm for p iterations would generate the leading p singular values and their associated singular vectors.

There are no classic algorithms to compute iteratively the SVD of matrix available only in operator form. We change our problem to computing the

leading eigenvectors of the product between the observation impact matrix and its transpose, $\mathbf{T}^* \mathbf{T} = \mathbf{U} \mathbf{S}^2 \mathbf{U}^*$, where

$$\begin{aligned} \mathbf{T}^* \mathbf{T} &= \mathbf{A}_0^* \sum_{k=1}^N \overline{\mathbf{M}}_{0,k}^* \overline{\mathbf{M}}_{0,k} \mathbf{A}_0 = \sum_{k=1}^N \mathbf{T}_k^* \mathbf{T}_k, \\ \overline{\mathbf{M}}_{0,k} &= \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{M}_{0,k}, \quad \mathbf{T}_k = \overline{\mathbf{M}}_{0,k} \mathbf{A}_0. \end{aligned} \quad (13)$$

This problem is solved using Krylov-based approaches (Lanczos [27], Arnoldi [28]), e.g., by the Jacobi-Davidson algorithm [29], available in the JDQZ software library [30].

Our algorithm computes a low-rank approximation of \mathbf{A}_0 , the inverse 4D-Var Hessian. An iterative procedure can be used to compute the smallest eigenvalues and the corresponding eigenvectors of the Hessian matrix, which are the dominant eigenpairs of \mathbf{A}_0 . The low-rank approximation of the inverse Hessian reads:

$$\mathbf{A}_0 = (\mathbf{V} \mathbf{D} \mathbf{V}^*)^{-1} = \mathbf{V} \mathbf{D}^{-1} \mathbf{V}^* \approx \left(\mathbf{V}_{(p)} \mathbf{D}_{(p)}^{-1} \mathbf{V}_{(p)}^* \right). \quad (14)$$

We replace the inverse Hessian in (13) with its low-rank approximation:

$$\begin{aligned} \mathbf{T}_k^* \mathbf{T}_k &\approx \left(\mathbf{V}_{(p)} \mathbf{D}_{(p)}^{-1} \mathbf{V}_{(p)}^* \right) \overline{\mathbf{M}}_{0,k}^* \overline{\mathbf{M}}_{0,k} \left(\mathbf{V}_{(p)} \mathbf{D}_{(p)}^{-1} \mathbf{V}_{(p)}^* \right) \\ &= \mathbf{V}_{(p)} \mathbf{D}_{(p)}^{-1} \mathbf{W}_k^* \mathbf{W}_k \mathbf{D}_{(p)}^{-1} \mathbf{V}_{(p)}^*, \\ \mathbf{W}_k &= \overline{\mathbf{M}}_{0,k} \mathbf{V}_{(p)}. \end{aligned} \quad (15)$$

The columns of \mathbf{W}_k are the Hessian eigenvectors $\mathbf{V}_{(p)}$ propagated forward and scaled by the tangent linear model $\overline{\mathbf{M}}_{0,k}$. The p tangent linear models can be performed in parallel. This allows us to approximate (13) as:

$$\begin{aligned} \mathbf{T}^* \mathbf{T} &\approx \mathbf{V}_{(p)} \mathbf{D}_{(p)}^{-1} \mathbf{W} \mathbf{D}_{(p)}^{-1} \mathbf{V}_{(p)}^*, \\ \mathbf{W} &= \sum_{k=1}^N \mathbf{W}_k^* \mathbf{W}_k. \end{aligned}$$

With the eigendecomposition:

$$\mathbf{D}_{(p)}^{-1} \mathbf{W} \mathbf{D}_{(p)}^{-1} = \mathbf{V}_{\text{red}} \mathbf{D}_{\text{red}} \mathbf{V}_{\text{red}}^*, \quad (16)$$

the approximation of (13) becomes

$$\mathbf{T}^* \mathbf{T} \approx \left(\mathbf{V}_{(p)} \mathbf{V}_{\text{red}} \right) \mathbf{D}_{\text{red}} \left(\mathbf{V}_{(p)} \mathbf{V}_{\text{red}} \right)^*. \quad (17)$$

This represents a truncated singular vector decomposition of \mathbf{T} , with \mathbf{D}_{red} the matrix of dominant singular values, and $\mathbf{V}_{(p)} \mathbf{V}_{\text{red}}$ the matrix of left singular vectors.

Algorithm 1 summarizes the main computational steps. The computational cost is dominated by the first step, where the expensive second-order adjoint model is run repeatedly to generate the Hessian eigenpairs. The iterative approach is suited for applications that benefit from an iterative improvement of the low-rank approximation. The methodology can be applied to any data assimilation system for which first and second order adjoint models are available.

Algorithm 1 Iterative algorithm for low-rank approximations

- 1: Solve iteratively the eigenvalue problem for the 4D-Var Hessian (14)
 - 2: Map newly generated eigenvectors through the tangent linear model (15)
 - 3: Compute the truncated SVD of the resulting matrix (16)
 - 4: Project the left singular vectors onto the eigenvector base of the 4D-Var Hessian and build the low-rank approximation of \mathbf{T} (17)
-

4.2. An ensemble-based (parallel) approach for matrix-free low-rank approximations

This approach uses a “randomized SVD” algorithm [31] to compute the Moore-Penrose pseudoinverse [32] of the Hessian. The 4D-Var Hessian matrix \mathbf{A}_0^{-1} is available only in operator form, i.e., only matrix vector products can be evaluated by running the second order adjoint. The randomized algorithm is as follows:

- 1: Draw p random vectors and form a matrix Ω .
- 2: Compute the product $\mathcal{Y} = \mathbf{A}_0^{-1}\Omega$ using Hessian-vector multiplications, i.e., running the second order adjoint model for each column.
- 3: Construct the QR decomposition $\mathcal{Y} = \mathbf{Q}\mathbf{R}$.

Each of the above steps can be performed in parallel.

The columns of \mathbf{Q} form an orthonormal basis for the range of \mathcal{Y} . Randomized SVD uses a series of algebraic manipulations, starting from the observation that \mathbf{Q} is also the orthonormal factor in the QR decomposition of \mathbf{A}_0^{-1} :

$$\mathbf{A}_0^{-1} = \mathbf{Q}\mathbf{B}, \quad \mathbf{B} = \mathbf{Q}^* \mathbf{A}_0^{-1}, \quad \mathbf{B}^* = \mathbf{A}_0^{-1} \mathbf{Q}. \quad (18)$$

Next, compute an SVD of \mathbf{B} :

$$\mathbf{B} = \mathbf{U}_B \Sigma_B \mathbf{V}_B^* \quad (19)$$

and replace (19) in (18) to obtain the SVD of \mathbf{A}_0^{-1} :

$$\mathbf{A}_0^{-1} = \mathbf{Q} \mathbf{U}_B \Sigma_B \mathbf{V}_B^* = \mathbf{U}_A \Sigma_B \mathbf{V}_B^*. \quad (20)$$

The left singular vectors of \mathbf{A} represent the projections of the left singular vectors of \mathbf{B} onto the columns of \mathbf{Q} . The singular values and right singular vectors of \mathbf{A} are the same as those of \mathbf{B} . The pseudoinverse of the 4D-Var Hessian \mathbf{A}_0^{-1} reads:

$$\mathbf{A}_0^+ \approx \mathbf{V}_B \Sigma_B^+ \mathbf{U}_A^*. \quad (21)$$

The observation impact matrix is approximated using the tangent linear model $\overline{\mathbf{M}}_{0,k}$ and the pseudoinverse \mathbf{A}_0^+ :

$$\mathbf{T} \approx \sum_{k=1}^N \overline{\mathbf{M}}_{0,k} \mathbf{A}_0^+. \quad (22)$$

The computational flow is summarized in Algorithm 2.

Algorithm 2 Sampling algorithm for low-rank approximations

- 1: Build the matrix \mathbf{B} , through parallel second-adjoint runs (18)
 - 2: Compute a full SVD of \mathbf{B} (19)
 - 3: Project the left singular vectors of \mathbf{B} in \mathbf{Q} and form the SVD of \mathbf{A}_0^{-1} (20)
 - 4: Compute the Hessian pseudoinverse \mathbf{A}_0^+ (21)
 - 5: Build the impact matrix \mathbf{T} through parallel tangent linear runs (22)
-

Computing the rows of \mathbf{B} (18) is done as matrix-vector products through second-order adjoint runs that can be performed in parallel. The last step, which propagates the components of the pseudoinverse through the linearization of the model, is achieved by multiple tangent linear model runs in parallel. The tangent model results are checkpointed at each of the observation times, so that only one run across the entire time horizon is necessary for each input vector.

5. Applications

We illustrate several applications of the observation impact matrix in 4D-Var using the two-dimensional shallow water equations. We first describe the system and its numerical discretization, then present the 4D-Var implementation and the experimental setting for data assimilation. Observation sensitivity is computed both in full and using low-rank approximation to assess how well the latter captures the essential features. We apply this analysis to three applications, namely, detecting the change in impact from perfect data to noisy data, pruning the least important observations, and detecting faulty sensors.

5.1. Test problem: shallow water equations

The two-dimensional shallow-water equations (2D SWE) [33] approximate the movement of a thin layer of fluid inside a basin:

$$\begin{aligned} \frac{\partial}{\partial t} h + \frac{\partial}{\partial x}(uh) + \frac{\partial}{\partial y}(vh) &= 0 \\ \frac{\partial}{\partial t}(uh) + \frac{\partial}{\partial x}\left(u^2h + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}(uvh) &= 0 \\ \frac{\partial}{\partial t}(vh) + \frac{\partial}{\partial x}(uvh) + \frac{\partial}{\partial y}\left(v^2h + \frac{1}{2}gh^2\right) &= 0. \end{aligned} \tag{23}$$

Here $h(t, x, y)$ is the fluid layer thickness, and $u(t, x, y)$ and $v(t, x, y)$ are the components of the velocity field of the fluid. The gravitational acceleration is denoted by g .

We consider a spatial domain $\Omega = [-3, 3]^2$ (spatial units), and an integration window is $t_0 = 0 \leq t \leq t_f = 0.1$ (time units). Boundary conditions are considered periodic. The space discretization is realized using a finite volume scheme, and the time integration uses a fourth Runge-Kutta scheme, following

Table 1: Normalized CPU times of different sensitivity models. The forward model takes one time unit to run.

FWD	1		
TLM	2.5	FWD + TLM	3.5
FOA	3.7	FWD + FOA	4.7
SOA	12.8	FWD + TLM + FOA + SOA	20

the method Lax-Wendroff [34]. The model uses a square $-q \times q$ uniform spatial discretization grid, which brings the number of model (state) variables to $n = 3q^2$.

We use the automatic differentiation tool TAMC [35, 36] to build various sensitivity models, as follows. The tangent-linear model (TLM) propagates perturbations forward in time. The first-order adjoint model (FOA) propagates perturbations backwards in time, and efficiently computes the gradient of a scalar cost functional defined on the model states. The second-order adjoint model (SOA) computes the product between the Hessian of the cost function and a user-defined vector [22].

The overhead introduced by the sensitivity models is considerable. Table 1 presents the CPU times of TLM, FOA, and SOA models (normalized with respect to that of one FWD run). One SOA integration is about 3.5 times more expensive than a single first-order adjoint run, while the FOA takes 3.7 times longer than the forward run. These relative costs depend on the discretization methodology and the implementation of sensitivities. Our previous research showed how to build efficient adjoint models by reusing computations performed in the forward run [22]. For example, for the shallow water model, the alternative continuous adjoints we built required a fraction of the forward model CPU time to run.

5.2. Data assimilation setup

The 4D-Var system is set up for a simple version of the “circular dam” problem [37]. The reference initial height field h is a Gaussian bell of a width equal to 1 length units centered on the grid, and the reference initial velocity vector components are constant $u = v = 0$. The physical interpretation is that the front of water falls to the ground (h decreases) under the effect of gravity and creates concentric ripples which propagate towards the boundaries. Figures 1 represent snapshots of the reference trajectory at initial and final time.

The computational grid is square and regular with $q = 40$ grid points in each direction, for a total of 4800 model variables (states). The simulation time interval is set to 0.01 seconds, using $N = 100$ timesteps of size 0.0001 (time units).

The h component of the a priori estimate (background) \mathbf{x}^b is generated by adding a correlated perturbation to the h reference solution at initial time. The background error covariance \mathbf{B}_0 corresponds to a standard deviation of 5% of the reference field values. For the u and v components we use white noise to

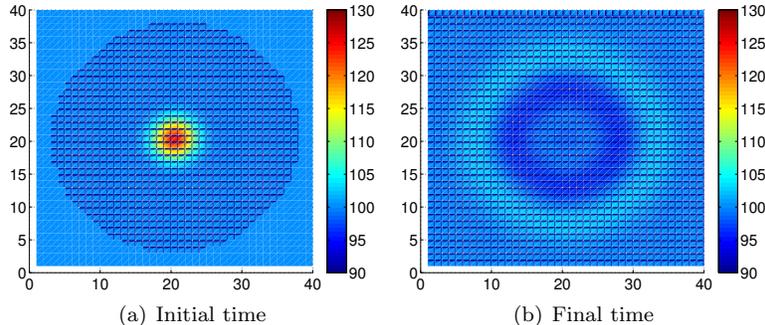


Figure 1: The height field h at the beginning and at the end of the reference trajectory.

prescribe perturbations. The spatial error correlation uses a Gaussian decay model, with a correlation distance of five grid points.

Synthetic observations are generated at the final time, by adding random noise to the reference trajectory. Since the observation errors are assumed uncorrelated, the observation error covariance matrix \mathbf{R} is diagonal. The standard deviation for observation noise is 1% of the largest absolute value of the observations for each variable. The observation operator \mathcal{H} is linear and selects observed variables at specified grid points. For the following experiments, we consider observations of all variables at each grid point.

The minimization of the 4D-Var cost function is performed with the L-BFGS-B solver [26] using a fixed number of 100 iterations.

5.3. Experimental results

5.3.1. Validating the low-rank approximations of the observation impact matrix

In the first experiment, we study the effect of two possible sources of errors: data noise and SVD truncation errors. These issues are inherent to performing the data assimilation and the observation impact analysis.

We apply our computational methodology to the data assimilation scenario introduced in Section 5.2. To assess how the reanalysis is affected by the presence of noise, we first assimilate perfect observations (i.e., reference model values), then assimilate the same observations with added small noise. Figure 2 reveals that the convergence of the numerical optimization procedure is similar in the two scenarios, and leads to a similar decrease in the root mean square (RMS) error for each variable.

We now apply the systematic approach presented in Section 4 to each one of the two assimilation scenarios, in order to compute the sensitivity to observations (8) when the verification solution \mathbf{x}_0^v is the background \mathbf{x}_0^b and \mathbf{C} is the identity matrix:

$$\nabla_{\mathbf{y}_k} \Psi(\mathbf{x}_0^a) = \mathbf{T}(\mathbf{x}_0^a - \mathbf{x}_0^b). \quad (24)$$

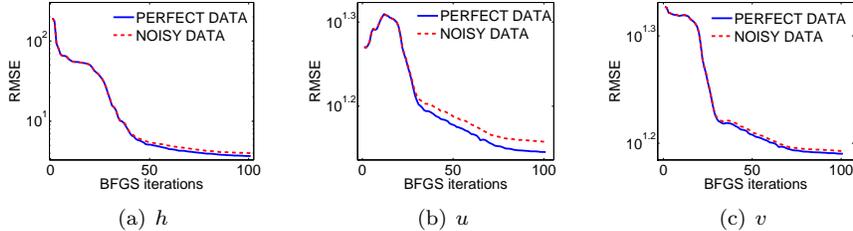


Figure 2: Root mean square error decrease for each variable versus the number of L-BFGS iterations.

This procedure provides the sensitivity of the 4D-Var increment norm $\|\mathbf{x}_0^a - \mathbf{x}_0^b\|$ to data. Large sensitivity values will be associated to observations that had a more important contribution to the 4D-Var correction.

Figure 3 plots the observation sensitivity (24) for each one of h , u and v , for both perfect and noisy scenarios. The standard deviation for observation noise is equal to 1% of the largest absolute value of the observations for each variable. We notice the observation sensitivity is locally correlated. For the h observations $\nabla_{\mathbf{y}_k} \Psi(\mathbf{x}_0^a)$ is almost symmetric, while for u and v is aligned along the East-West and South-North directions. Observation sensitivities for perfect and for noisy data exhibit similar spatial features and numerical values. Reasonably small noise in data does not significantly affect the sensitivity fields.

The results in Figure 3 are obtained with the full sensitivity matrix, built offline. We compare these results with a low-rank approximation of the sensitivity matrix computed using Algorithm 1. The low rank approximation of the sensitivity is

$$(\nabla_{\mathbf{y}_k} \Psi(\mathbf{x}_0^a))_{(p)} = \mathbf{T}_{(p)} \cdot (\mathbf{x}_0^a - \mathbf{x}_0^b). \quad (25)$$

Figure 4 displays the singular value spectrum of the observation impact matrix and the reconstruction error of the low-rank truncated values for observation sensitivity. It can be noticed that the decay in singular value spectrum resembles that of the truncation error. Based on this, a good choice for the rank of the approximation is $p \approx 1600$, which corresponds to one third of the full workload.

In Figure 5 we plot the low-rank approximation (25) of sensitivity to h observations and the corresponding truncation error. A visual comparison with the full-rank observation sensitivity plotted in Figure 3(a) reveals that the low-rank approximation captures well the main features of the sensitivity field.

5.3.2. Observation impact

The computational methodology developed in Section 4 is now applied to compute the forward observation impact. Specifically, we map changes in the observations (innovation vector $\Delta \mathbf{y} = \mathbf{y} - \mathcal{H}(\mathbf{x})$) to changes in the analysis ($\Delta \mathbf{x}_0^a$) using the relation (12).

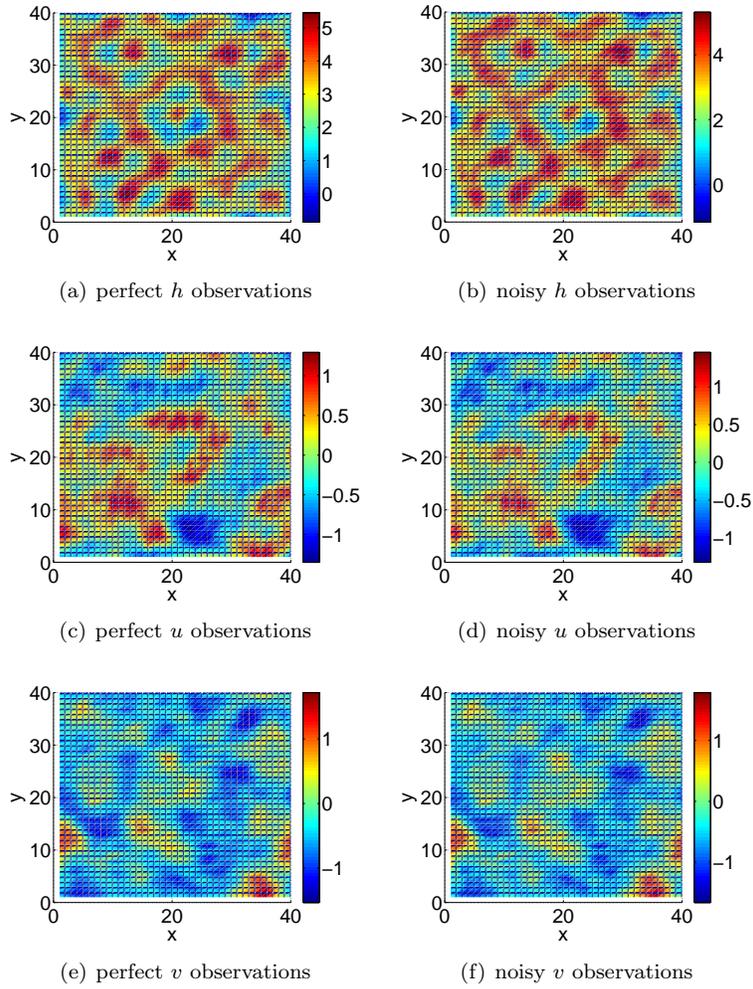


Figure 3: Sensitivity field (24) for perfect and for noisy observations of h , u , and v .

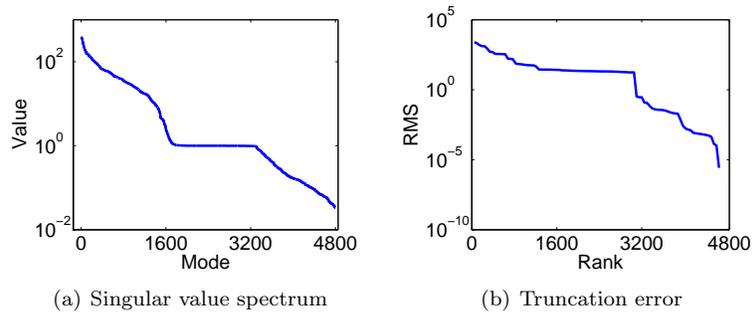


Figure 4: Singular value decay for the observation impact matrix \mathbf{T} (11) and the corresponding truncation error norm.

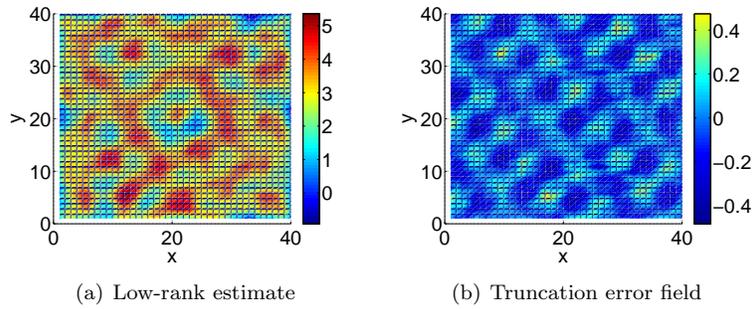


Figure 5: Low-rank approximation of observation sensitivity (25) for h data and the associated truncation error field for 1600 modes.

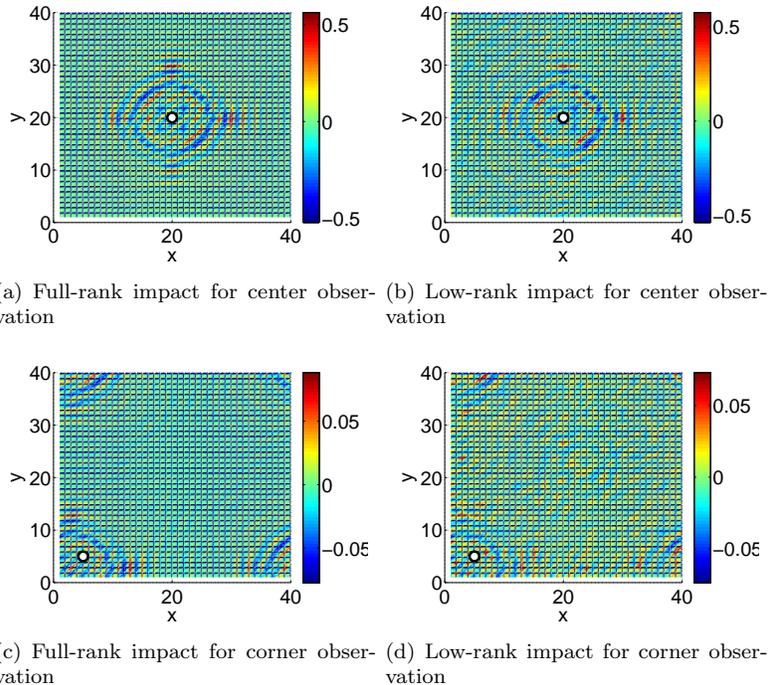


Figure 6: The impact of two h observations from assimilation time t_100 , placed in the center and in the corner of the grid. Both full rank and reduced rank solutions are shown. 500 modes are used for the reduced rank approximation.

We compute the observation impact in full-rank and low-rank corresponding to two observations of h , one in the center (grid coordinates $(20, 20)$) and one in the corner (at $(5, 5)$) at the locations represented with white markers. This is achieved by multiplying \mathbf{T}^T with a vector containing just the innovation brought by the observation whose impact we want to evaluate (all other vector entries have value zero). The resulting impact fields are plot in Figure 6. The spatial features of the observation impact have a local radial correlation in both cases. This means the information carried by the observation is spread in its proximity by the 4D-Var process; this is also true across the periodic boundaries of the shallow water system. Moreover, the low-rank approximations are able to pick up the important features of the full-rank calculations, and provide impacts of a similar magnitude.

Having already computed the SVD of the observation impact matrix, we also look at the directions in the data space $\Delta\mathbf{y}$ that have the largest impact on the analysis, and the directions in the 4D-Var correction space $\Delta\mathbf{x}$ that benefit most from the assimilation. These directions are given by the dominant left and right singular vectors of \mathbf{T}^T , respectively. Figure 7 plots the first dominant left (and

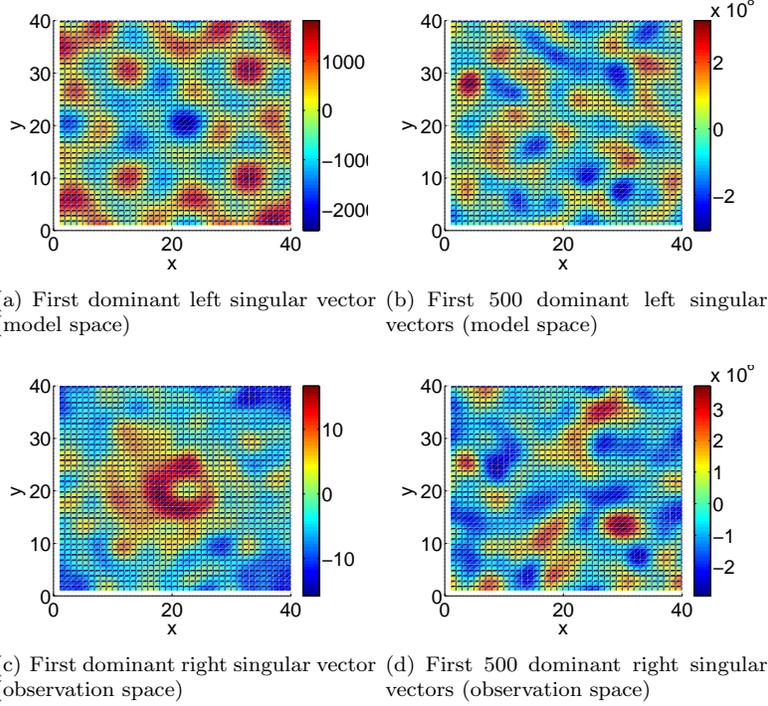


Figure 7: Principal directions of growth in observation and solution space, as defined by the dominant left and right singular vectors of the observation impact matrix.

right) singular vectors corresponding to the h variable, and also the composition of the most important 500 directions for the observation and solution space using the formula

$$v_{\text{domin}} = \sum_{i=1}^{500} s_i^2 * v_i, \quad (26)$$

where s_i and v_i are the singular pair corresponding to the i -th mode.

5.3.3. Pruning observations based on sensitivity values

In this experiment we illustrate how sensitivity analysis can be used to select the most useful data points.

For this, we compute the 4D-Var reanalysis and then apply the observation impact methodology to compute the sensitivity of the cost function Ψ to each one of the 4800 observations (24). We then split our observation set in three subsets of 1600 observations, corresponding to each one of h , u and v . Within each subset, we order the observations by their sensitivity; the top 800 observations form the HIGH sensitivity set, and the ones in the bottom 800 form the

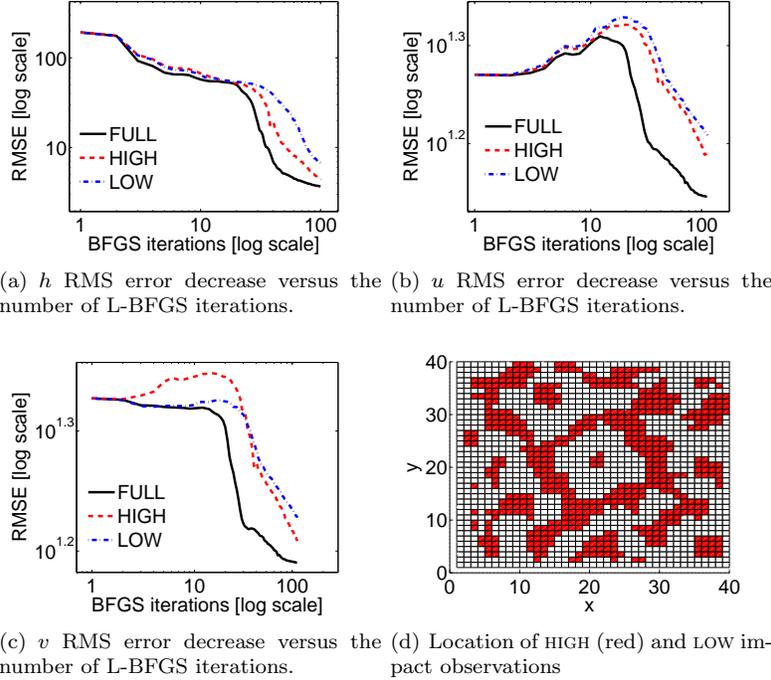


Figure 8: Location of high and low sensitivity observations, and their use in data assimilation.

LOW sensitivity set. This procedure partitions our initial 4800 observations in two halves with respect to observation sensitivity, taken variable-wise. The 4D-Var data assimilation process is repeated using either one of the HIGH or LOW data sets.

Figure 8(a) shows the distribution of HIGH versus LOW observations on our computational grid. We run 4D-Var to assimilate only the HIGH, followed by assimilating only the LOW data points, and compare the decrease in the RMS true error in h in Figure 8(b). The convergence results show that assimilating observations of larger sensitivity yields slightly better results. When the number of optimizer iterations increases, the two scenarios exhibit similar performance, comparable to using all the observations.

5.3.4. Identifying faulty data using observation sensitivity

Finally, we consider the problem of detecting isolated faulty sensors and try to solve it using our new computational capabilities. We prescribe certain sensors to exhibit the wrong behavior and expect the observation sensitivity analysis to reflect this.

We perform the 4D-Var scenario above again, using perfect observations at each grid point, with the exception of two points where the observation values are increased by a factor of 10 (thus simulating two faulty instruments). The two points are located at coordinates (20, 20), the grid center where fluid height field has large values initially, and at coordinates (10, 10), which is initially outside the water height profile.

The 4D-Var data assimilation uses the faulty data together with the correct values. Figure 9(a) plots the correction in the variable h , i.e., the quantity $(\mathbf{x}_0^a - \mathbf{x}_0^b)$. The correction field is smooth and there is no indication that there are very large errors in some data points. Figure 9(b) plots the supersensitivity vector μ obtained by mapping the increment through the inverse 4D-Var Hessian (9). The supersensitivity field exhibits a clear structure consisting of large values in concentric circles around the two locations with faulty observations. Figure 9(c) shows the observation sensitivity \mathbf{y} , obtained by propagating the supersensitivity through the tangent linear model (10). The observation impact methodology clearly identifies two points of abnormally large sensitivity. These are exactly the locations of the faulty sensors, and because their values are very different than the surrounding field, they have an abnormally large impact on the analysis. This is precisely identified by the observation impact metrics.

6. Conclusions and future work

This research is motivated by dynamic data driven applications which use measurements of the real system state to constrain computer model predictions. We develop efficient approaches to compute the observation impact in 4D-Var data assimilation, i.e., the contribution of each data point to the final analysis. Quantification of observation impact is important for applications such as data pruning, tuning the parameters of data assimilation, identification of faulty sensors, and optimal configuration of sensor networks. Immediate applications include numerical weather prediction, climate and air-quality forecast, hydrology, renewable energy systems and biohazard proliferation.

While 4D-Var is the state of the art data assimilation methodology, and is widely used in many operational centers, it is one of the most computationally challenging approaches to data assimilation. The computation of observation impact adds to the cost of performing analyses. We review the derivation of the sensitivity equations that lead to the observation impact matrix, and make in-depth comments about its structure, functionality and computation requirements. One contribution of this work is to develop highly efficient implementation strategies for the 4D-Var sensitivity equations via a smart use of adjoint models. Two matrix free algorithms are proposed, one serial and one parallel, to compute approximations of observation impact matrix. These algorithms compute SVD-based low-rank approximations, which capture the most important features of observation impact matrix. The accuracy of the generated approximations scales with the computational cost.

A second contribution of this work is to illustrate the use of observation impact in solving two practical applications: pruning observations of small impact,

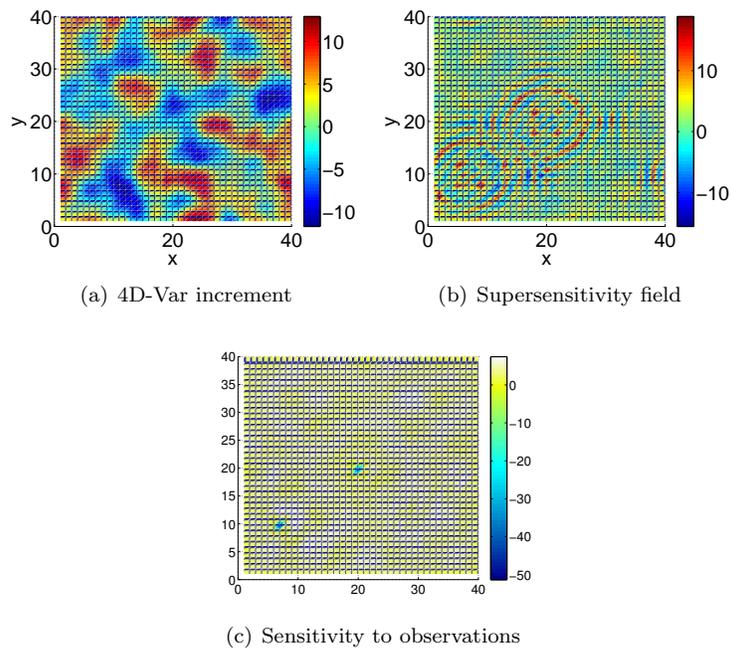


Figure 9: Observation sensitivity field when the assimilated data is corrupted at two locations with coordinates $(10,10)$ and $(20,20)$. The location of the faulty sensors is unknown to the data assimilation system, but is retrieved via the observation impact methodology.

and detecting faulty sensors. Numerical experiments that validate the proposed computational framework are carried out with a two dimensional shallow water equations system.

Several future research directions emerge from this study. On the computational side, the impact approximation algorithms can be further developed to achieve superior performance. A rigorous analysis of approximation errors will be developed to guide the choice of the number of iterations and the truncation level. On the application side, our will be integrated in real large scale applications to provide a measure of importance of different measurements in real time assimilation. In hindsight, the impact can be used to prune the data subsets and to detect erroneous data points. In foresight, it can be used to design efficient strategies of sensor placement for targeted and adaptive observations.

Acknowledgements

This work was supported by the National Science Foundation through the awards NSF DMS-0915047, NSF CCF-0635194, NSF CCF-0916493 and NSF OCI-0904397, and by AFOSR DDDAS program through the awards FA9550-12-1-0293-DEF and AFOSR 12-2640-06.

References

- [1] T. Palmer, T. Gelaro, J. Barkmeijer, and R. Buizza. Singular vectors, metrics, and adaptive observations. *Journal of the Atmospheric Sciences*, 55(4):633–653, 1998.
- [2] W. Liao, A. Sandu, and G. Carmichael. Total energy singular vector analysis for atmospheric chemical transport models. *Monthly Weather Review*, 134(9):2443–2465, 2006.
- [3] D. Zupanski, A.Y. Hou, S.Q. Zhang, M. Zupanski, C.D. Kummerow, and S.H. Cheung. Applications of information theory in ensemble data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 133(627):1533–1545, 2007.
- [4] K. Singh, A. Sandu, and M. Jardak. Information theoretic metrics to characterize observations in variational data assimilation. *Proceedings of the International Conference on Computational Science*, 9(1):1047–1055, 2012.
- [5] L.M. Berliner, Z.Q. Lu, and C. Snyder. Statistical design for adaptive weather observations. *Journal of the Atmospheric Sciences*, 56(15):2536–2552, 1999.
- [6] W. Kang and L. Xu. Optimal placement of mobile sensors for data assimilations. *Tellus A*, 64(0), 2012.

- [7] R. Gelaro, Y. Zhu, and R.M. Errico. Examination of various-order adjoint-based approximations of observation impact. *Meteorologische Zeitschrift*, 16(6):685–692, 2007.
- [8] R. Gelaro and Y. Zhu. Examination of observation impacts derived from observing system experiments (OSEs) and adjoint models. *Tellus A*, 61(2):179–193, 2009.
- [9] Y. Tremolet. Computation of observation sensitivity and observation impact in incremental variational data assimilation. *Tellus A*, 60(5):964–978, 2008.
- [10] F.X. Le Dimet and H.E. Ngodock. Sensitivity analysis in variational data assimilation. *Journal of the Meteorological Society of Japan*, 75(1):245–255, 1997.
- [11] D.N. Daescu. On the sensitivity equations of four-dimensional variational (4D-Var) data assimilation. *Monthly Weather Review*, 136(8):3050–3065, 2008.
- [12] D.N. Daescu and R. Todling. Adjoint sensitivity of the model forecast to data assimilation system error covariance parameters. *Quarterly Journal of the Royal Meteorological Society*, 136(653):2000–2012, 2010.
- [13] D.G. Cacuci. Sensitivity theory for nonlinear systems. I: Nonlinear functional analysis approach. *Journal of Mathematical Physics*, 22:2794, 1981.
- [14] Z. Wang, I.M. Navon, F.X. Le Dimet, and X. Zou. The second order adjoint analysis: theory and applications. *Meteorology and Atmospheric Physics*, 50(3):3–20, 1992.
- [15] A. Sandu and L. Zhang. Discrete second order adjoints in atmospheric chemical transport modeling. *Journal of Computational Physics*, 227(12):5949–5983, 2008.
- [16] J. Ye. Generalized low rank approximations of matrices. *Machine Learning*, 61(1-3):167–191, 2005.
- [17] M.W. Berry and R.D. Fierro. Low-rank orthogonal decompositions for information retrieval applications. *Numerical Linear Algebra with Applications*, 3(4):301–327, 1996.
- [18] I.S. Dhillon and D.S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2):143–175, 2001.
- [19] R. Daley. *Atmospheric data analysis*. Cambridge University Press, Cambridge, 1991.
- [20] E. Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, Cambridge, 2002.

- [21] A. Sandu, D.N. Daescu, G.R. Carmichael, and T. Chai. Adjoint sensitivity analysis of regional air quality models. *Journal of Computational Physics*, 204(1):222–252, 2005.
- [22] A. Cioaca, A. Sandu, and M. Alexe. Second-order adjoints for solving PDE-constrained optimization problems. *Optimization Methods and Software*, 27(4-5):625–653, 2011.
- [23] A. Griewank. On automatic differentiation. *Mathematical Programming: recent developments and applications*, 6:83–107, 1989-.
- [24] I.Y. Gejadze, F.X. Le Dimet, and V. Shutyaev. On analysis error covariances in variational data assimilation. *SIAM Journal on Scientific Computing*, 30(4):1847–1874, 2008.
- [25] A. Cioaca, A. Sandu, and E. de Sturler. Efficient methods for computing observation impact in 4d-var data assimilation. Technical report, 2012.
- [26] C. Zhu, R.H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.
- [27] C. Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Government Press Office, 1950.
- [28] W.E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9(1):17–29, 1951.
- [29] G.L.G. Sleijpen and H.A. Van der Vorst. A Jacobi–Davidson iteration method for linear eigenvalue problems. *SIAM Review*, 42(2):267–293, 2000.
- [30] D.R. Fokkema and M.B. van Gijzen. Short manual for the JDQZ-package, 1999.
- [31] E. Liberty, F. Woolfe, P.G. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51):20167–20172, 2007.
- [32] T.O. Lewis and T.G. Newman. Pseudoinverses of positive semidefinite matrices. *SIAM Journal on Applied Mathematics*, 16(4):701–703, 1968.
- [33] D.D. Houghton and A. Kasahara. Nonlinear shallow fluid flow over an isolated ridge. *Communications on Pure and Applied Mathematics*, 21(1):1–23, 1968.
- [34] R. Liska and B. Wendroff. Composite schemes for conservation laws. *SIAM Journal of Numerical Analysis*, 35(6):2250–2271, 1998.

- [35] R. Giering. Tangent linear and adjoint model compiler. *User Manual, TAMC Version*, 4, 1997.
- [36] R. Giering and T. Kaminski. Recipes for adjoint code construction. *ACM Transactions on Mathematical Software*, 24(4):437–474, 1998.
- [37] K. Anastasiou and C.T. Chan. Solution of the 2d shallow water equations using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids*, 24(11):1225–1245, 1997.