

Data Assimilation in Large Time-Varying Multidimensional Fields

Amir Asif, *Member, IEEE*, and José M. F. Moura, *Fellow, IEEE*

Abstract— In the physical sciences, e.g., meteorology and oceanography, combining measurements with the dynamics of the underlying models is usually referred to as *data assimilation*. Data assimilation improves the reconstruction of the image fields of interest. Assimilating data with algorithms like the Kalman–Bucy filter (KBf) is challenging due to their computational cost which for two-dimensional (2-D) fields is of $O(I^6)$ where I is the linear dimension of the domain. In this paper, we combine the block structure of the underlying dynamical models and the sparseness of the measurements (e.g., satellite scans) to develop four efficient implementations of the KBf that reduce its computational cost to $O(I^5)$ in the case of the block KBf and the scalar KBf, and to $O(I^4)$ in the case of the local block KBf (lbKBf) and the local scalar KBf (lsKBf). We illustrate the application of the lbKBf to assimilate altimetry satellite data in a Pacific equatorial basin.

Index Terms— Computed imaging, data assimilation, Kalman–Bucy filter, Gauss–Markov fields, physical oceanography, satellite altimetry.

I. INTRODUCTION

IN THE physical sciences, the images of interest are often modeled by partial differential equations (PDE's) and the measurements available are highly sparse. We are motivated by the reconstruction of image fields like ocean velocity or sea surface height in physical oceanography, and develop algorithms to couple the dynamics of the fields with sparse measurements. In oceanography, these problems are referred to as *data assimilation*. In engineering and signal processing, data assimilation is known as *estimation* or *filtering*.

For space-time dependent physical systems, a number of smoothing and signal processing algorithms have been developed, each of which emphasizes varying degrees of statistical structure or computational efficiency. Broadly speaking, the main approaches may be divided into two major classes: optimal interpolation methods and least square methods. The essential concept behind schemes based on optimal interpolation

Manuscript received May 26, 1996; revised February 22, 1999. This work was supported in part by the Office of Naval Research under Grants N00014-91-J-1001 and N00014-97-1-0800, and by MOST, Pakistan. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. W. Clem Karl.

A. Asif was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA. He is now with the Technical University of British Columbia, Surrey, B.C., Canada V3R 7P8 (e-mail: asif@techbc.ca).

J. M. F. Moura is with the Department of Electrical and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139-7250 USA, on leave from the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA (e-mail: moura@mit.edu).

Publisher Item Identifier S 1057-7149(99)08750-3.

is unbiased linear regression with minimal error covariance. Such methods are useful when the physics modeling the system are not known. Reference [1] develops a multiresolution framework, which uses a scale-recursive statistical model proceeding from coarser representations to finer ones. It allows characterization of the error statistics and calculates efficiently the maximum likelihood parameter estimates. The scheme resembles a batch processing approach, no physical dynamics are involved in the estimation process, and the sea surface height, the field under study, is basically interpolated. Least square methods minimize in a statistical sense the data misfit under the constraint of the model dynamics. These include the adjoint method [2] and the Kalman–Bucy filter (KBf) [3]. For problems of large dimensions, a straightforward implementation of the KBf is infeasible in terms of the computations and the storage required. In optical flow and computer vision, a number of practical KBf implementations such as in [4]–[6] have been developed to reduce the computations associated with the KBf. Reference [4] presents an approximate *information form* variant of the KBf. It approximates the information matrix by a sparsely banded matrix and uses a Taylor series polynomial approximation to propagate the banded structure through successive KBf iterations. This sparse banded matrix approximation corresponds to assuming that the estimation error process is a Markov random field (MRF). Reference [5] extends this approach to a square root information variant of the KBf.

In this paper, we consider the *direct form* of the KBf for data assimilation applications. Our algorithm has similarities with that in [4] and [5] but is distinct in important ways. In [4] and [5], the MRF approximation is used in conjunction with a truncated Taylor series approximation to simplify the error covariance matrix computation. In our work, we apply the MRF approximation to derive recursive algorithms for this matrix, no further simplification being used. Our updating algorithms are different from those in [4] and [5]. Our algorithms relate the constituent blocks in the error covariance matrix to its diagonal and its upper diagonal block entries. Due to the special sparse nature of the measurements, most of the off diagonal blocks need not be updated. We update only the diagonal and the upper diagonal blocks. Any other block, if required, may be derived directly from the MRF relationships.

The algorithms we present take full advantage of the unique aspects associated with the applications we consider, as follows.

- *Local Fields*: The fields are obtained from the discretization of partial differential equations (dPDE). In other words, the field values at a particular spatial location

(site) are mostly related to the field values at neighboring sites. In this paper, we restrict attention to models with first order time derivatives and arbitrary order spatial derivatives.

- *Sparse Measurements*: Examples of sparse measurements include undersampled scanned data, like altimetry data (sea surface height) collected by an orbiting satellite, or data gathered by sparsely distributed point sensors like moored or drifting buoys.

For fields defined on a two-dimensional (2-D) lattice of linear dimension I , following the usual approach, we stack the field variables in a long I^2 -dimensional state vector. The operational count of the standard implementation of the KBf grows with the cube of the state dimension, requiring then $O(I^6)$ computations. Fields described by dPDE models are localized. The local nature of the dPDE models is naturally reflected in the block banded structure of their system matrices, with blocks themselves banded and sparse. By coupling this block structure with the sparse measurements, we obtain computationally efficient KBf's. With scanned measurements, we obtain a block KBf (bKBf). With point sensors, we get a scalar KBf (sKBf). The bKBf and the sKBf implementations are both of $O(I^5)$.

To reduce further the complexity of the implementation, we explore the MRF simplification alluded to above, which is suggested by the local structure of the fields. As mentioned, this is an approximation on the error field associated with the KBf rather than on the field itself. The approach assumes that the estimation error field of the KBf is local [4], [7], i.e., that the error field at a particular site and at each time instant is essentially determined by its values at neighboring sites. Using the results in [8] and [9], this simplifies the inverse of the error covariance matrix associated with the filter to a banded block parameterization. We call these approximate implementations the *localized block KBf* (lbKBf) or the *localized scalar KBf* (lsKBf). Using further results in [8] and [9], we simplify the KBf, reducing by an additional factor of I the computational complexity of the bKBf and of the sKBf. The total operational count for the lbKBf and the lsKBf becomes of $O(I^4)$, a reduction by I^2 . The storage requirements of the localized implementations are $O(I^3)$, reduced from $O(I^4)$ for the direct implementation of the KBf.

The paper is structured as follows. In Section II, we present the block structure of the dynamical equations when these arise from the discretization of PDE's. In Section III, we discuss the sparse nature of the measurements of interest: scanned and point measurements. In Sections IV and V, we couple the sparse measurements with the block structure of the dynamical models to derive efficient data assimilation algorithms. Section VI evaluates the computational effort of these algorithms. In Section VII, we illustrate our algorithms in a small data assimilation study in ocean circulation with synthetic altimetry satellite data. Finally, Section VIII concludes the paper.

II. DYNAMICAL MODEL: DISCRETIZED LINEAR PDE'S

The dynamical models we consider are discretized partial differential equations (dPDE). The models are linear and are

restricted in this paper to first-order time derivatives. The linear models may arise from linearization of more complex nonlinear PDE's. We are interested in transport type problems. We do not consider here wave equation models which involve second-order time derivatives. The spatial derivatives may be of arbitrary order. We discuss first the case of scalar fields, like for example the intensity field in image processing or the stream function field in fluid dynamics and physical oceanography [10]. We then consider the problem of multiple fields like velocity and displacement fields in physical oceanography.

There is a rich class of numerical procedures to discretize PDE's. These include finite differences [11], finite elements [12], or spectral methods [13]. To focus discussion on the essentials of the approach and to avoid unnecessarily heavy notation, we restrict the discussion in this section to explicit finite difference schemes, in particular, to the forward-Euler method [11]. In actual application, see Section VII, we use other numerical schemes namely the leap frog method [11]: the details change, but the fundamental aspects of the results remain the same.

A. Forward-Euler Method

To discretize PDE's over a continuous 2-D spatial domain M and a continuous time interval T , M is divided into a regularly spaced mesh of size $\Delta x \times \Delta y$, and T into intervals of duration Δt . We assume M is rectangular and that the uniform grid has finite dimensions $I \times J$. The indices i and j represent the point $(x_i, y_j) = (i\Delta x, j\Delta y)$ on the grid. The index k represents the time instant $t_k = k\Delta t$.

The Forward-Euler (FE) method uses a single step forward difference approximation for time and space. First order time and first order spatial derivatives of the scalar field ψ in the (x, y) plane at time t are computed as

$$\begin{aligned} \frac{\partial \psi}{\partial t} &\approx \frac{\psi_{ijk+1} - \psi_{ijk}}{\Delta t}, \\ \frac{\partial \psi}{\partial x} &\approx \frac{\psi_{i+1jk} - \psi_{ijk}}{\Delta x}, \quad \text{and} \\ \frac{\partial \psi}{\partial y} &\approx \frac{\psi_{ij+1k} - \psi_{ijk}}{\Delta y}. \end{aligned} \quad (1)$$

The notation ψ_{ijk} represents the value $\psi(x_i, y_j, t_k)$ of the field ψ at the location (x_i, y_j) at time t_k . For the second-order derivatives, we use the following definitions

$$\begin{aligned} \frac{\partial^2 \psi}{\partial x^2} &\approx \frac{\psi_{i+1jk} - 2\psi_{ijk} + \psi_{i-1jk}}{\Delta x^2}, \\ \frac{\partial^2 \psi}{\partial y^2} &\approx \frac{\psi_{ij+1k} - 2\psi_{ijk} + \psi_{ij-1k}}{\Delta y^2}, \\ \frac{\partial^2 \psi}{\partial x \partial y} &\approx \frac{\psi_{i+1j+1k} - \psi_{i+1j-1k} - \psi_{i-1j+1k} + \psi_{i-1j-1k}}{4\Delta x \Delta y}. \end{aligned} \quad (2)$$

To compute the first order spatial derivatives at (x_i, y_j) at time t_k , FE uses only the values at the neighboring points (x_{i+1}, y_j) and (x_i, y_{j+1}) at the same time instant t_k . The time derivative at (x_i, y_j) at time t_k uses (x_i, y_j) at time t_{k+1} . The second order spatial derivatives involve, in addition to the sites mentioned above, the value of the field at time

t_k at sites (x_{i-1}, y_j) , (x_i, y_{j-1}) , (x_{i+1}, y_{j+1}) , (x_{i+1}, y_{j-1}) , (x_{i-1}, y_{j+1}) , and (x_{i-1}, y_{j-1}) . Equations (1)–(3) reflect the local coupling introduced by the discretization of the partial differential operators. We illustrate the type of dPDE models that result.

B. dPDE's

We discretize linear models

$$\frac{\partial \psi}{\partial t} = \mathcal{L}\psi \quad (4)$$

where \mathcal{L} is a PDE linear operator. For example, in two dimensions,

$$\mathcal{L}\psi = \sum_{m,n} \frac{\partial^m}{\partial x^m} \frac{\partial^n}{\partial y^n} a_{mn}(x, y)\psi. \quad (5)$$

The PDE model is completed with boundary and initial conditions. Boundary conditions (BC) of interest include Dirichlet, Neuman, mixed, or periodic [14].

For simplicity consider the 2-D diffusion equation

$$\frac{\partial \psi}{\partial t} = \alpha_1 \frac{\partial^2 \psi}{\partial x^2} + \alpha_2 \frac{\partial^2 \psi}{\partial y^2} + w \quad (6)$$

where the forcing term w may include random effects. Discretizing (6) by the FE method, (1)–(3), and using vector and matrix notation, we get

$$\Psi^{(k+1)} = \mathcal{A}\Psi^{(k)} + \mathcal{C}W^{(k)} \quad (7)$$

with $\Psi^{(k)} = \text{vec}[\Psi_i^{(k)}]$, i.e., $\Psi^{(k)} = [\Psi_1^{(k)T} \ \Psi_2^{(k)T} \ \dots \ \Psi_I^{(k)T}]^T$, and $\Psi_i^{(k)} = [\psi_{i1k} \ \psi_{i2k} \ \dots \ \psi_{iJk}]^T$. Similarly for $W^{(k)}$.

The matrix \mathcal{A} is an $IJ \times IJ$ matrix with the following structure:

$$\mathcal{A} = \begin{bmatrix} B_2 & B_3 & \mathbf{0} & \mathbf{0} & \cdot & \mathbf{0} \\ B_1 & B_2 & B_3 & \mathbf{0} & \cdot & \mathbf{0} \\ \mathbf{0} & B_1 & B_2 & B_3 & \cdot & \mathbf{0} \\ \cdot & \cdot & \ddots & \ddots & \ddots & \cdot \\ \mathbf{0} & \cdot & \mathbf{0} & B_1 & B_2 & B_3 \\ \mathbf{0} & \cdot & \mathbf{0} & \mathbf{0} & B_1 & B_2 \end{bmatrix}. \quad (8)$$

$$= (K_I^1)^T \otimes B_1 + \mathcal{I}_I \otimes B_2 + K_I^1 \otimes B_3 \quad (9)$$

where \otimes denotes the Kronecker product, \mathcal{I}_I is the identity matrix of dimension I , and K_I^1 is an all zero square matrix of order I except for the first upper diagonal of ones. For the FE, the blocks B_1 , B_2 , and B_3 are given by

$$B_1 = B_3 = \mathcal{I}_J \otimes b \quad \text{and} \\ B_2 = (K_J^1)^T \otimes \beta_1 + \mathcal{I}_J \otimes \beta_2 + K_J^1 \otimes \beta_3. \quad (10)$$

The parameters β_1 , β_2 , β_3 , and b are given by

$$\beta_1 = \beta_3 = \lambda_y, \quad \beta_2 = -2(\lambda_x + \lambda_y - \frac{1}{2}), \quad b = \lambda_x. \quad (11)$$

For other more general numerical procedures, the structure of the matrix \mathcal{A} and of the blocks B_i 's is similar. The structure of the matrix \mathcal{C} in (7) depends on the forcing term w . For simplicity here, we assume that w contains no derivatives. In this case, \mathcal{C} is block diagonal with blocks $\Delta t \mathcal{I}_J$, i.e.,

$$\mathcal{C} = \mathcal{I}_I \otimes F \quad \text{with} \quad F = \Delta t \mathcal{I}_J. \quad (12)$$

Equation (7) is the dPDE model. We refer to it as the state equation. Borrowing terminology from systems theory and the Kbf, we call \mathcal{A} the state matrix and \mathcal{C} the input matrix. The state matrix \mathcal{A} relates the current state $\Psi^{(k+1)}$ with the previous state $\Psi^{(k)}$. Collectively, \mathcal{A} , \mathcal{C} , B_1 , B_2 , and B_3 are referred to as the system matrices.

The input vector $W^{(k)}$ in (7) is modeled as a zero mean, temporally white Gaussian noise. The covariance of its constituent vectors $W_i^{(k)}$ is $\mathcal{E}[W_i^{(k)} W_j^{(l)}] = Q \delta_{ij} \delta_{kl}$ where δ is the Kronecker symbol and \mathcal{E} the expectation operator.

C) Structure of the System Matrices

The dPDE model, (7)–(12), corresponds to the diffusion model (6) with Dirichlet boundary conditions, using the FE finite difference scheme. In applications, we may have higher order PDE operators, different boundary conditions may be more appropriate, or we may use other discretization schemes. The following remarks regarding the dPDE models will hold true in general.

First, the state matrix \mathcal{A} and the input matrix \mathcal{C} are block banded, block Toeplitz, and sparse. The blocks B_1 , B_2 , and B_3 are themselves sparse—in the above example, B_2 has only 3 nonzero diagonal entries, and B_1 and B_3 are diagonal. The sparseness and the banded structure of these matrices is a consequence of the local coupling in the field ψ introduced by the finite difference representation of the PDE operator \mathcal{L} of (5).

Second, the block banded structure of the system matrices is described in terms of its bandwidth which is the farthest lower or upper diagonal (counting out from the main diagonal) containing one or more nonzero block elements. A tridiagonal block matrix, for example, has a bandwidth of two blocks and a pentadiagonal block matrix has a bandwidth of three blocks. The bandwidth of the state matrix \mathcal{A} depends on the order of the spatial differential operator \mathcal{L} and on the order of the numerical differencing scheme used. It can be shown, using arguments similar to those in [8] and [9] that, for PDE's with up to second-order partial derivatives in space and using the numerical approximations in (1)–(3), the state matrix \mathcal{A} is block tridiagonal; with up to fourth order spatial differential operators, \mathcal{A} is block pentadiagonal, and so on for higher orders. The bandwidth of the input matrix \mathcal{C} depends likewise on the order of the spatial differential operator acting on the input forcing terms and on the schemes that discretize it. Hence, PDE models with higher order derivatives lead to dPDE models with a structure similar to the state equation (7). The state vector Ψ may be of higher dimension and the blocks B_1 , B_2 , and B_3 may vary considerably but the system matrices \mathcal{A} and \mathcal{C} will still be sparse block banded matrices. We exploit the block structure of \mathcal{A} and \mathcal{C} when developing the *block* and *scalar* implementations of the Kalman-Bucy filter.

D) Multiple Fields

Physical oceanography applications usually deal with coupled fields like the ocean circulation velocity components, the sea surface height, the ocean density, the sea surface temperature, and the salinity field. When there are p fields,

we have p coupled dPDE equations. For multiple fields, the structure of the modeling equations (7)–(12) remains essentially the same, except the entries in the matrices B_1 , B_2 , and B_3 are now $pJ \times pJ$ blocks. The size of the square blocks H_I^\dagger , \underline{Q} , and \mathcal{I}_I changes from J to pJ . The field vector Ψ becomes a pIJ -dimensional vector. To avoid cluttering the notation, we will assume that the field under study is scalar, i.e., $p = 1$, but, in Section VII, we will actually apply our results to a multidimensional field.

E) Convergence

Numerical schemes should be convergent, i.e., they must approximate well the solution of the corresponding PDE, the approximation improving as the time and grid spacings Δt , Δx , and Δy tend to zero. Convergence is checked by von Neumann analysis [11]. This leads to inequality bounds between the grid meshes $(\Delta x, \Delta y)$, and the time spacing Δt . These inequalities are the Courant–Friedrich–Lewy (CFL) conditions [15]. The choice of the grid sizes is made based on these bounds. In the sequel, we assume that care has been taken in selecting an appropriate numerical procedure and that the grid meshes and time steps do satisfy the CFL conditions.

III. MEASUREMENT MODELS

Equation (7) is the dPDE model describing in time and space the evolution of the field $\Psi^{(k)}$. We assume that we have available noisy measurements of the field. The observed field $\mathcal{Y}^{(k)}$ is

$$\mathcal{Y}^{(k)} = \mathcal{H}^{(k)}\Psi^{(k)} + \Xi^{(k)}. \quad (13)$$

In (13), $\Xi^{(k)}$ is a zero mean white Gauss noise field independent of the field $\Psi^{(k)}$ with covariance $\mathcal{E}[\Xi^{(k)}\Xi^{(t)T}] = R\delta_{kt}$ where δ_{kt} is the Kronecker symbol. The matrix $\mathcal{H}^{(k)}$ is the observation matrix.

The structure of $\mathcal{H}^{(k)}$ depends on the type of measurement program. Usually, at each time instant, it is practical to observe only a small portion of the field, so that the dimension of $\mathcal{Y}^{(k)}$ is much smaller than that of $\Psi^{(k)}$. This makes the observation matrix $\mathcal{H}^{(k)}$ highly sparse. We consider two types of measurements that occur often in physical oceanography: *scanned measurements* when at each particular time instant only a few rows of the discretized field are observed; and *point measurements* when only isolated grid points of the field are observed. We determine next how these measurements are reflected on $\mathcal{H}^{(k)}$.

1) *Scanned Measurements*: Assume N rows are scanned. say row m, n, \dots, p with $m < n < \dots < p$. The measurement

model is then given by (14), shown at the bottom of the page, where the ξ 's represent the measurement noise, assumed Gaussian and independent, i.e., $\mathcal{E}[\xi_m \xi_n^T] = R_{mn} \delta_{mn}$.

While collecting altimetric data, the measurement arcs are curved. We assume that the discretizing basin has rows aligned with the even (or odd) satellite scans and that the columns are approximately aligned with the odd (or even) satellite scans. The state model (7) and the measurements (14) may be modified to account for this pattern. Scanning along rows leads to (7) and (14), as explained before. For scans along columns, permute the state vector with a permutation of stride J , and modify accordingly the system matrices. Stride permutations are obtained by simple pre-multiplication with a suitable elementary matrix. At each filtering step of the KBf, a check on the type of scan will dictate the appropriate state and measurement models to be used, the error covariance matrix \mathcal{P} being reordered accordingly to incorporate these changes. In the following discussion, for the sake of convenience and simplicity, we assume data to be measured along the rows only.

2) *Point Measurements*: Besides onboard satellite instrumentation, oceanographers use a panoply of measurement devices in drifting or moored buoys. The buoys provide measurements of ocean physical quantities at their location. Let $\mathcal{S}^{(k)}$ represent the collection of N sites (points in the lattice) at which we have buoys at time k . We assume these locations to be known. For moored buoys, this set does not change with time. For drifting buoys, this set collects the buoys positions at time k . The pointwise measurement model is

$$\mathcal{Y}_{no}^{(k)} = \Theta_{no} \Psi_{no}^{(k)} + \xi_{no}^{(k)} \quad (n, o) \in \mathcal{S}^{(k)}. \quad (15)$$

We collect the N measurements in vector $\mathcal{Y}^{(k)}$. The measurement equation is then (16), shown at the bottom of the next page.

As before, the measurement noise vector is assumed independent of the input noise sequence $\{W^{(k)}\}$. It is a zero mean, temporally white Gaussian noise with covariance $\mathcal{E}[\xi_{tm}^{(k)} \xi_{no}^{(t)}] = R_{tm} \delta_{tn} \delta_{mo} \delta_{kt}$.

In (15) and (16), like in (14), the observation matrix $\mathcal{H}^{(k)}$ is highly sparse. We exploit these sparse observations in our signal processing algorithms. We consider the scanned and the point measurements separately.

IV. KALMAN–BUCY FILTER: SCANNED AND POINT MEASUREMENTS

To reconstruct the field, we assimilate the measurements in Section III to the dynamical model (7) in Section II, via the KBf, [3]. Direct application of the KBf is computationally

$$\begin{bmatrix} \mathcal{Y}_m \\ \mathcal{Y}_n \\ \vdots \\ \mathcal{Y}_p \end{bmatrix}^{(k)} = \begin{bmatrix} \underline{0} & \cdot & \underline{0} & \Theta_m & \underline{0} & \cdot & \underline{0} & \underline{0} & \underline{0} & \cdot & \underline{0} & \underline{0} & \underline{0} & \cdot & \underline{0} & \underline{0} \\ \underline{0} & \cdot & \underline{0} & \underline{0} & \underline{0} & \cdot & \underline{0} & \Theta_n & \underline{0} & \cdot & \underline{0} & \underline{0} & \underline{0} & \cdot & \underline{0} & \underline{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \underline{0} & \cdot & \underline{0} & \underline{0} & \underline{0} & \cdot & \underline{0} & \underline{0} & \underline{0} & \cdot & \underline{0} & \Theta_p & \underline{0} & \cdot & \underline{0} & \underline{0} \end{bmatrix}^{(k)} \Psi^{(k)} + \begin{bmatrix} \xi_m \\ \xi_n \\ \vdots \\ \xi_p \end{bmatrix}^{(k)} \quad (14)$$

prohibitive. For a lattice of size $I \times J$, the state vector $\Psi^{(k)}$ has dimension IJ . The Kbf computational cost grows with the third power of the state dimension, in this case, it is of $O((IJ)^3)$. We present in the next subsections implementations of the Kbf that use the dPDE block structure and the structure of the measurements to reduce the computational cost of the Kbf. In Section IV-A, we develop a block Kbf (bKbf) for scanned measurements. In Section IV-B, we design a scalar Kbf (sKbf) for pointwise measurements. The bKbf and sKbf are *exact* efficient implementations of the Kbf that simplify the Kbf by $O(I)$. In Section V we make a simplifying assumption on the error of the Kbf. This approximation enables us to derive localized (Gauss–Markov) implementations of the Kbf which reduce the computational costs by an $O(I)$.

A. Scanned Measurements: Block Kbf

We partition all vector and matrix quantities in the Kbf according to the block structure of the system matrices. We expand the filter equations into these partitioned quantities and exploit the block sparse structure of the measurements. Let $\hat{\Psi}_i$ be the i th row of the estimated field $\hat{\Psi}$, P_{ij} the error cross-covariance between rows i and j of the error field, $P_{ij} = \mathcal{E}[(\Psi_i - \hat{\Psi}_i)(\Psi_j - \hat{\Psi}_j)^T]$, and $\mathcal{P} = \{P_{ij}\}$ for $1 \leq i, j \leq I$, the error covariance matrix. The bKbf equations are as follows.

1) Predictor Update: $1 \leq i \leq I$

$$\hat{\Psi}_i(k+1|k) = B_1 \hat{\Psi}_{i-1}(k|k) + B_2 \hat{\Psi}_i(k|k) + B_3 \hat{\Psi}_{i+1}(k|k). \quad (17)$$

2) Predictor Covariance Update: $1 \leq i, j \leq I$

$$\begin{aligned} P_{ij}(k+1|k) = & [B_1 P_{i-1j-1}(k|k) + B_2 P_{ij-1}(k|k) \\ & + B_3 P_{i+1j-1}(k|k)] B_1^T \\ & + [B_1 P_{i-1j}(k|k) + B_2 P_{ij}(k|k) \\ & + B_3 P_{i+1j}(k|k)] B_2^T \\ & + [B_1 P_{i-1j+1}(k|k) + B_2 P_{ij+1}(k|k) \\ & + B_3 P_{i+1j+1}(k|k)] B_3^T + F Q F^T \delta_{ij}. \end{aligned} \quad (18)$$

3) Filter Update: $1 \leq i \leq I$

$$\begin{aligned} & \text{if } \text{rem}(k\Delta t, t_s) = 0 \\ & \quad \hat{\Psi}_i(k+1|k+1) \\ & \quad = \hat{\Psi}_i(k+1|k) + \chi(k+1|k)\Gamma(k+1) \\ & \quad \quad \cdot \begin{bmatrix} \mathcal{Y}_m(k+1) - \Theta_m \hat{\Psi}_m(k+1|k) \\ \mathcal{Y}_n(k+1) - \Theta_n \hat{\Psi}_n(k+1|k) \\ \vdots \\ \mathcal{Y}_p(k+1) - \Theta_p \hat{\Psi}_p(k+1|k) \end{bmatrix} \\ & \quad \text{else } \hat{\Psi}_i(k+1|k+1) = \hat{\Psi}_i(k+1|k). \end{aligned} \quad (19)$$

4) Filter Covariance Update: $1 \leq i, j \leq I, I$

$$\begin{aligned} & \text{if } \text{rem}(k\Delta t, t_s) = 0 \\ & \quad P_{ij}(k+1|k+1) \\ & \quad = P_{ij}(k+1|k) - \chi(k+1|k)\Gamma(k+1) \\ & \quad \quad \cdot \begin{bmatrix} \Theta_m P_{mj}(k+1|k) \\ \Theta_n P_{nj}(k+1|k) \\ \vdots \\ \Theta_p P_{pj}(k+1|k) \end{bmatrix} \\ & \quad \text{else } P_{ij}(k+1|k+1) = P_{ij}(k+1|k) \end{aligned} \quad (20)$$

where $\chi(k+1|k)$ is a $J \times NJ$ matrix given by

$$\chi(k+1|k) = [P_{im}(k+1|k)\Theta_m^T \quad P_{in}(k+1|k)\Theta_n^T \quad P_{ip}(k+1|k)\Theta_p^T] \quad (21)$$

and $\Gamma(k+1)$ is the $NJ \times NJ$ matrix (22), shown at the bottom of the page, with P_{ij} on the right side of (22) measured at $(k+1|k)$. The term $\text{rem}(k\Delta t, t_s)$ represents the remainder of the division of the sampling interval $k\Delta t$ by the time t_s between scans. The interval Δt is the time interval in between two consecutive updates of the field Ψ . Its value is determined from numerical considerations by the CFL condition. The interval t_s is the time duration in between two consecutive satellite scans. In practice, t_s is usually much greater than Δt . We choose Δt according to CFL and such that t_s is an integer multiple of Δt . At update k , we check to see if measurements are available or not. Measurements are available

$$\begin{aligned} \mathcal{Y}^{(k)} = & [\mathcal{Y}_{tm}^{(k)} \quad \mathcal{Y}_{no}^{(k)} \quad \dots \quad \mathcal{Y}_{rs}^{(k)}]^T = \mathcal{H}^{(k)}\Psi^{(k)} + \Xi^{(k)} \\ = & \begin{bmatrix} 0 & \cdot & 0 & \Theta_{tm} & 0 & \cdot & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 \\ 0 & \cdot & 0 & 0 & 0 & \cdot & 0 & \Theta_{no} & 0 & \cdot & 0 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & 0 & 0 & \cdot & 0 & 0 & \cdot & 0 & \Theta_{rs} & 0 & \cdot & 0 \end{bmatrix} \Psi^{(k)} + \begin{bmatrix} \xi_{tm}^{(k)} \\ \xi_{no}^{(k)} \\ \vdots \\ \xi_{rs}^{(k)} \end{bmatrix} \end{aligned} \quad (16)$$

$$\Gamma(k+1) = \begin{bmatrix} \Theta_m P_{mm} \Theta_m^T + R_{mm} & \Theta_m P_{mn} \Theta_n^T + R_{mn} & \dots & \Theta_m P_{mp} \Theta_p^T + R_{mp} \\ \Theta_n P_{nm} \Theta_m^T + R_{nm} & \Theta_n P_{nn} \Theta_n^T + R_{nn} & \dots & \Theta_n P_{np} \Theta_p^T + R_{np} \\ \vdots & \vdots & \ddots & \vdots \\ \Theta_p P_{pm} \Theta_m^T + R_{pm} & \Theta_p P_{pn} \Theta_n^T + R_{pn} & \dots & \Theta_p P_{pp} \Theta_p^T + R_{pp} \end{bmatrix}^{-1} \quad (22)$$

when $\text{rem}(k\Delta t, t_s) = 0$. If there are measurements, both the prediction and filtering stages of the KBf are used. If not, the field is updated with the prediction equations until a new measurement becomes available.

B. Point Measurements: Scalar KBf

The error covariance \mathcal{P} is now partitioned into its scalar elements, i.e., $P_{ipjq} = \mathcal{E}[(\Psi_{ip} - \hat{\Psi}_{ip})(\Psi_{jq} - \hat{\Psi}_{jq})]$ where (i, p) and (j, q) are the field sites. As in the bKBf, the $\text{rem}(k\Delta t, t_s)$ determines when measurements are available. The prediction step remains the same as in the bKBf, (17) and (18). We do not repeat them here and write only the filtering step of the sKBf.

1) *Filter Update:* $1 \leq i \leq I$ and $1 \leq p \leq J$

$$\begin{aligned} &\text{if } \text{rem}(k\Delta t, t_s) = 0 \\ &\quad \hat{\Psi}_{ip}(k+1|k+1) \\ &\quad = \hat{\Psi}_{ip}(k+1|k) + \Lambda^{(k+1)}\Upsilon^{(k+1)} \\ &\quad \cdot \begin{bmatrix} \mathcal{Y}_{im}^{(k+1)} - \Theta_{im}\hat{\Psi}_{im}(k+1|k) \\ \mathcal{Y}_{no}^{(k+1)} - \Theta_{no}\hat{\Psi}_{no}(k+1|k) \\ \vdots \\ \mathcal{Y}_{rs}^{(k+1)} - \Theta_{rs}\hat{\Psi}_{rs}(k+1|k) \end{bmatrix} \\ &\text{else } \hat{\Psi}_{ip}(k+1|k+1) = \hat{\Psi}_{ip}(k+1|k). \end{aligned} \quad (23)$$

2) *Filter Covariance Update:* $1 \leq i, j \leq I$ and $1 \leq p, q \leq J$

$$\begin{aligned} &\text{if } \text{rem}(k\Delta t, t_s) = 0 \\ &\quad P_{ipjq}(k+1|k+1) \\ &\quad = P_{ipjq}(k+1|k) - \Lambda^{(k+1)}\Upsilon^{(k+1)} \\ &\quad \cdot \begin{bmatrix} \Theta_{im}P_{imjq}(k+1|k) \\ \Theta_{no}P_{nojq}(k+1|k) \\ \vdots \\ \Theta_{rs}P_{rsjq}(k+1|k) \end{bmatrix} \\ &\text{else } P_{ipjq}(k+1|k+1) = P_{ipjq}(k+1|k) \end{aligned} \quad (24)$$

where $\Lambda^{(k+1)}$ is a row vector of size N and $\Upsilon^{(k+1)}$ is a square matrix of order N with the structure in (25) and (26), shown at the bottom of the page. In (25) and (26), we did not write explicitly the time index. The covariances are the predicted covariances $P(k+1|k)$. The sKBf, (23)–(26), has a structure similar to that of the bKBf, (17)–(22). With the sKBf, the observed grid points need not lie on the same, or close by, rows, they can be isolated and lie anywhere on the grid.

.	.	.	6	.	.	.
.	5	4	3	4	5	.
.	4	2	1	2	4	.
6	3	1	o	1	3	6
.	4	2	1	2	4	.
.	5	4	3	4	5	.
.	.	.	6	.	.	.

Fig. 1. First- to sixth-order neighborhoods.

V. LOCALIZED KBf: GAUSS-MARKOV APPROXIMATION

The bKBf and sKBf presented in Sections IV-A and IV-B are exact implementations of the KBf for scanned or pointwise measurements, respectively. No approximations were made, they reflect the block banded structure of the dPDE models and the sparsity of the measurements. To reduce further the computational effort and the storage requirements, we develop here *approximate* implementations for the bKBf and sKBf which we call the localized bKBf (lbKBf) and the localized sKBf (lsKBf). In the following discussion, we derive the localized structure for the block KBf. At the end of the section, we consider briefly the localized scalar KBf.

Underlying the lbKBf is an approximation to the inverse of the error covariance matrix, i.e., to the information matrix of the filter. We approximate the information matrix by a sparse block banded matrix. Banded approximations to information matrices correspond in essence to modeling the error field in our spatial estimates at each point in time as a reduced-order MRF [4], [8]. For a first-order MRF, site (i, j) , labeled as “o” in Fig. 1, interacts with the sites $\{(i, j-1), (i, j+1), (i-1, j), (i+1, j)\}$ which are marked as sites “1” in the figure. Second-order interactions involve the sites marked as “1” and “2,” and similarly for higher order MRF’s.

We emphasize that the fields modeled by the dPDE models are local fields—see the block structure of the system matrices, but the corresponding error field associated with the KBf is not in general an MRF. However, it is this localized or MRF property that we impose on the error field as an approximation to reduce the computational complexity of the bKBf and its memory requirements. In [16], it is proved that these are Gauss–Markov approximations that optimize the Kullback–Leibler mean information distance criterion under a certain constraint.

A. MRF Property

Under the MRF property, the blocks $\{P_{ij}\}$ of the covariance matrix are not independent of each other. For simplicity, we

$$\Lambda^{(k+1)} = [\Theta_{im}P_{iplm} \quad \Theta_{no}P_{ipno} \quad \cdots \quad \Theta_{rs}P_{iprs}] \quad (25)$$

$$\Upsilon^{(k+1)} = \begin{bmatrix} R_{im} + \Theta_{im}^2 P_{imlm} & \Theta_{im}\Theta_{no}P_{imno} & \cdots & \Theta_{im}\Theta_{rs}P_{imrs} \\ \Theta_{im}\Theta_{no}P_{nolm} & R_{no} + \Theta_{no}^2 P_{nono} & \cdots & \Theta_{no}\Theta_{rs}P_{nors} \\ \vdots & \vdots & \ddots & \vdots \\ \Theta_{im}\Theta_{rs}P_{rslm} & \Theta_{no}\Theta_{rs}P_{rsno} & \cdots & R_{rs} + \Theta_{rs}^2 P_{rsrs} \end{bmatrix}^{-1} \quad (26)$$

illustrate our results by using a first order MRF approximation. The information matrix \mathcal{P}^{-1} is local and block tridiagonal, [8]. Since \mathcal{P} is a covariance matrix, \mathcal{P}^{-1} is symmetric, and so its upper and lower diagonal blocks are the transpose of each other, i.e., $P_{ij} = P_{ji}^T$. The Cholesky factor U of the information matrix in the Cholesky's factorization $\mathcal{P}^{-1} = U^T U$ has the upper bidiagonal block structure

$$U = \begin{bmatrix} U_1 & O_1 & \underline{0} & \underline{0} & \cdot & \underline{0} \\ \underline{0} & U_2 & O_2 & \underline{0} & \cdot & \underline{0} \\ \underline{0} & \underline{0} & U_3 & O_3 & \cdot & \underline{0} \\ \cdot & \cdot & \cdot & \ddots & \ddots & \cdot \\ \underline{0} & \cdot & \cdot & \underline{0} & U_{I-1} & O_{I-1} \\ \underline{0} & \cdot & \cdot & \underline{0} & \underline{0} & U_I \end{bmatrix}. \quad (27)$$

Since U is an upper bidiagonal block matrix, its inverse U^{-1} is upper block triangular. The diagonal block entries of U^{-1} are the inverse of the block diagonal entries of U . Specifically, U^{-1} is

$$U^{-1} = \begin{bmatrix} U_1^{-1} & \ast & \ast & \ast & \cdot & \ast \\ \underline{0} & U_2^{-1} & \ast & \ast & \cdot & \ast \\ \underline{0} & \underline{0} & U_3^{-1} & \ast & \cdot & \ast \\ \cdot & \cdot & \ddots & \ddots & \cdot & \cdot \\ \underline{0} & \cdot & \cdot & \underline{0} & U_{I-1}^{-1} & \ast \\ \underline{0} & \cdot & \cdot & \underline{0} & \underline{0} & U_I^{-1} \end{bmatrix}. \quad (28)$$

The notation \ast stands for nonzero, not known entries, and will be referred to as don't care entries. The lower triangular block entries of U^{-1} are all zero block entries. This feature is used to derive the following results, where we show how to obtain: 1) the block entries $\{U_i, O_i\}$ of the Cholesky factor U from the blocks $\{P_{ii}\}$ and $\{P_{ii+1}\}$ of the diagonal and first upper diagonal of the error covariance matrix \mathcal{P} ; 2) the block entries $\{P_{ij}\}$ of \mathcal{P} from the block entries $\{U_i, O_i\}$ of the Cholesky factors of \mathcal{P}^{-1} ; and finally 3) the off-diagonal entries P_{ij} , $1 \leq i \leq I$, $i+2 \leq j \leq I$ from the main diagonal blocks P_{ii} and the first upper main diagonal blocks P_{ii+1} of \mathcal{P} . References [17] and [18] study how to compute a portion of the inverse of a sparse matrix. The algorithms in [17] and [18] differ from our results because they mix entries of the inverse and entries of the original sparse matrix.

In the following results, we introduce the following compact notation for the product of matrices:

$$\prod_{\tau=1}^N (A_\tau) = A_1 A_2 \cdots A_N. \quad (29)$$

Result 1: Given the blocks $\{P_{ii}\}$ and $\{P_{ii+1}\}$ of the error covariance matrix \mathcal{P} of a first order MRF, the blocks $\{U_i, O_i\}$ of the Cholesky factor U are given by

$$U_I = \text{chol}((P_{II})^{-1}) \quad (30)$$

$$\left. \begin{aligned} U_i &= \text{chol}((P_{ii} - P_{ii+1} P_{i+1}^{-1} P_{ii+1}^T)^{-1}) \\ O_i &= -(P_{i+1}^{-1} P_{ii+1}^T U_i^T)^T \end{aligned} \right\} \quad (31)$$

$$1 \leq i \leq I-1$$

where $\text{chol}(\cdot)$ means Cholesky's factorization. \blacksquare

Note that Result 1 expresses the block entries $\{U_i, O_i\}$ of the Cholesky factor U of the inverse of \mathcal{P} in terms of the blocks of the main diagonal and the upper main diagonal. By transposition, it is clear that we can also obtain relations involving the block entries of the main diagonal and the first lower diagonal of \mathcal{P} .

Proof: From the equality $\mathcal{P} = (U^T U)^{-1}$, we get $\mathcal{P} U^T = U^{-1}$, where we replace U by its value in (27), substitute U^{-1} from (28), and express \mathcal{P} in terms of its constituent blocks, i.e., $\mathcal{P} = \{P_{ij}\}$, for $1 \leq i, j \leq I$. After this substitution, we multiply out the left hand side of (5) and equate the block entries of the diagonal and lower diagonals of the left and the right hand sides. We obtain

$$P_{II} U_I^T = U_I^{-1} \quad (32)$$

$$\begin{aligned} P_{ij} U_j^T + P_{ij+1} O_j^T \\ = (U_j)^{-1} \delta_{ij} \quad 1 \leq i < (I-1), 1 \leq j \leq i. \end{aligned} \quad (33)$$

Equations (32) and (33) involve the lower triangular block entries in \mathcal{P} . Since \mathcal{P} is symmetric, i.e., $P_{ij} = P_{ji}^T$, we could as well obtain equivalent relations involving the upper triangular block entries.

Result (30) is obtained directly by rearranging terms in (32).

In (33), substitute $i = \ell + 1$ and $j = \ell$. The resulting expression is

$$P_{\ell+1\ell} U_\ell^T + P_{\ell+1\ell+1} O_\ell^T = 0. \quad (34)$$

Move $P_{\ell+1\ell} U_\ell^T$ to the right hand side, left multiply on both sides by $P_{\ell+1\ell+1}^{-1}$, and take the transpose to get

$$O_\ell = -(P_{\ell+1\ell+1}^{-1} P_{\ell\ell+1}^T U_\ell^T)^T. \quad (35)$$

which verifies the bottom equation in Result 1.

To prove the expression for U_i , we substitute $i = j = \ell$ in (33), giving

$$P_{\ell\ell} U_\ell^T + P_{\ell\ell+1} O_\ell^T = (U_\ell)^{-1}. \quad (36)$$

The equality for U_ℓ , (31), can now be derived by substituting for O_ℓ from (35), right multiplying by U_ℓ , and finally taking the Cholesky factors on both sides of the resulting equation.

Result 2: Given the blocks $\{U_i, O_i\}$ of the Cholesky factor U , we have the following.

- 1) The main diagonal blocks P_{ii} , $1 \leq i \leq I$ of \mathcal{P} of an MRF, can be obtained recursively from the following expressions

$$P_{II} = (U_I^T U_I)^{-1} \quad (37)$$

$$\begin{aligned} P_{ii} &= (U_i^T U_i)^{-1} + (U_i^{-1} O_i) P_{i+1} (U_i^{-1} O_i)^T \\ &\text{for } (I-1) \geq i \geq 1. \end{aligned} \quad (38)$$

- 2) The remaining upper triangular blocks P_{ii+k} , $1 \leq i \leq (I-1)$, $1 \leq k \leq (I-i)$, are given by

$$P_{ii+k} = \left(\prod_{\tau=i}^{i+k-1} (-U_\tau^{-1} O_\tau) \right) P_{i+k} \quad (39)$$

or alternatively, by

$$P_{ij} = \left(\prod_{\tau=i}^{j-2} (-U_{\tau}^{-1} O_{\tau}) \right) P_{j-1j} \quad \text{for } 1 \leq i \leq (I-1), (i+1) \leq j \leq I. \quad (40)$$

Result 2 determines the block entries of \mathcal{P} from the block entries of the Cholesky factor of the inverse \mathcal{P}^{-1} . ■

Proof: Equation (37) follows directly by rearranging terms in (32).

We prove now Results (39) and (40), and then come back to expression (38). We prove by induction.

Case 1—Equation (39): With $i = \ell + 1$ and $j = \ell$ in (33), we get (34). In (34), right multiplying by U_{ℓ}^{-T} , take the transpose on both sides, and rearrange, to get

$$P_{\ell\ell+1} = (-U_{\ell}^{-1} O_{\ell}) P_{\ell+1\ell+1} \quad (41)$$

which is (39) for $k = 1$.

By the induction step, we assume (39) is valid for $k = \wp$, i.e.,

$$P_{ii+\wp} = \left(\prod_{\tau=i}^{i+\wp-1} (-U_{\tau}^{-1} O_{\tau}) \right) P_{i+\wp i+\wp} \quad (42)$$

and prove (39) for $k = \wp + 1$.

With $i = \ell + \wp + 1$ and $j = \ell$ in (33) and following the steps that led to (41), we get

$$P_{\ell\ell+\wp+1} = (-U_{\ell}^{-1} O_{\ell}) P_{\ell+1\ell+\wp+1}. \quad (43)$$

Substituting for $P_{\ell+1\ell+\wp+1}$ from (42) with $i = \ell + 1$, proves (39) for $k = \wp + 1$.

Case 2—Equation (40): Expression (40) is different from Result (39). We prove it directly by expressing (39) in the following alternate form:

$$P_{ij} = \left(\prod_{\tau=i}^{j-1} (-U_{\tau}^{-1} O_{\tau}) \right) P_{jj} \quad \text{for } 1 \leq i \leq (I-1), (i+1) \leq j \leq I. \quad (44)$$

Equation (40) follows by expanding the product term in the above equation as

$$\prod_{\tau=i}^{j-1} (-U_{\tau}^{-1} O_{\tau}) = \left(\prod_{\tau=i}^{j-2} (-U_{\tau}^{-1} O_{\tau}) \right) (-U_{j-1}^{-1} O_{j-1}) \quad (45)$$

and realizing that $(-U_{j-1}^{-1} O_{j-1}) P_{jj}$ equals P_{j-1j} .

Case 3—Equation (38): We start by substituting $i = \ell$ and $j = \ell$ in (38), giving

$$P_{\ell\ell} U_{\ell}^T + P_{\ell\ell+1} O_{\ell}^T = (U_{\ell})^{-1}. \quad (46)$$

Result (38) follows by right multiplying the above expression by U_{ℓ}^T , and substituting for $P_{\ell\ell+1}$ from Result (39). ■

Note that the diagonal blocks, see (37) and (38), can be found directly from the $\{U_i, O_i\}$ without the need for computing any off diagonal blocks P_{ij} , $i \neq j$.

Result 3: Given the main diagonal and the upper diagonal blocks $\{P_{ii}\}$ and $\{P_{ii+1}\}$ of the covariance matrix \mathcal{P} of an MRF, the upper triangular blocks P_{ij} , $1 \leq i \leq I$, $i+2 \leq$

$j \leq I$, are given by

$$P_{ij} = \left(\prod_{\tau=i}^{j-2} (P_{\tau+1\tau+1}^{-1} P_{\tau+1\tau})^T \right) P_{j-1j}. \quad (47)$$

Proof: Result 3 is derived from (40) by expressing the product $(-U_{\tau}^{-1} O_{\tau})$ in terms of the main and upper diagonal blocks of \mathcal{P} . Again starting from (34) with the dummy variable τ rather than ℓ , right multiplying it on both sides by U_{τ}^{T-1} and rearranging, we get

$$(-U_{\tau}^{-1} O_{\tau}) = (P_{\tau+1\tau+1}^{-1} P_{\tau+1\tau})^T. \quad (48)$$

Result 3 follows by substituting in (40) for $(-U_{\tau}^{-1} O_{\tau})$ from the above relationship. ■

Localized bKBf: In the localized bKBf (lbKBf), we update only the diagonal and the upper diagonal blocks P_{ii} 's and P_{ii+1} 's. Any other blocks P_{ij} 's if required may then be obtained directly from the P_{ii} 's and P_{ii+1} 's using Result 3. To elaborate further, we do a stepwise implementation of the lbKBf. We start at time step k . The algorithm has computed the present estimate $\hat{\Psi}(k|k)$ and the block tridiagonal elements, $P_{ii}(k|k)$ and $P_{ii+1}(k|k)$ for $1 \leq i \leq I$, of the error covariance matrix $\mathcal{P}(k|k)$. For the sake of completeness, the initial conditions ($k = 0$) are also specified.

- 1) **Predictor Update:** Update the field $\hat{\Psi}_i(k+1|k)$ for $1 \leq i \leq I$ using (17). This step is straightforward and requires no approximation. *Initial condition:* $\hat{\Psi}(0|0)$.
- 2) **Predictor Covariance Update:** Update the diagonal and the upper diagonal error covariance blocks $P_{ii}(k+1|k)$ and $P_{ii+1}(k+1|k)$ for $1 \leq i \leq I$ using (18). Analyzing (18) for $j = i$ and $j = i+1$, we see that to update $P_{ii}(k+1|k)$ and $P_{ii+1}(k+1|k)$ requires block entries from the main diagonal and from the first, second, and third upper (or lower) diagonals. The main diagonal blocks $P_{ii}(k|k)$ and the first upper diagonal blocks $P_{ii+1}(k|k)$ are known from the initial condition or the previous filter covariance step, see below. To evaluate the offdiagonal blocks $P_{ii+2}(k|k)$ and $P_{ii+3}(k|k)$ that are needed, we use Result 3 which leads to

$$P_{ii+2}(k|k) = P_{ii+1}(k|k) P_{i+1i+1}^{-1}(k|k) P_{i+1i+2}(k|k) \quad \text{for } 1 \leq i \leq (I-2) \quad (49)$$

$$P_{ii+3}(k|k) = P_{ii+1}(k|k) P_{i+1i+1}^{-1}(k|k) P_{i+1i+3}(k|k) \quad \text{for } 1 \leq i \leq (I-3). \quad (50)$$

Note that $P_{i+1i+3}(k|k)$ in (50) is a block entry from the second upper diagonal computed in (49). *Initial conditions:* $\mathcal{P}_{ii}(0|0)$ and $\mathcal{P}_{ii+1}(0|0)$.

- 3) **Filter Update:** To evaluate $\hat{\Psi}_i(k+1|k+1)$, the matrices $\Gamma(k+1)$ and $\chi(k+1)$ are needed, see (19). The matrix $\Gamma(k+1)$ is given by (22) and depends upon the scanned rows. In the remaining discussion, we call it the scan matrix. Computing the scan matrix requires elements of $\mathcal{P}(k+1|k)$ outside the blocks of the main and the upper diagonal. For N scanned rows, say $m < n < o < \dots <$

p , the blocks needed are

$$\begin{array}{cccc} P_{mm}(k+1|k) & P_{mo}(k+1|k) & \cdots & P_{mp}(k+1|k) \\ & P_{no}(k+1|k) & \cdots & P_{np}(k+1|k) \\ & & \cdots & \vdots \\ \cdots & & & P_{op}(k+1|k). \end{array} \quad (51)$$

These blocks are evaluated from the known main and upper diagonal blocks of $\mathcal{P}(k+1|k)$ using Result 3. We now compute the gain $\chi(k+1)$ using (21). In addition to the main diagonal blocks $P_{ii}(k+1|k)$ and the first upper diagonal blocks $P_{ii+1}(k+1|k)$, the following blocks are needed:

$$\left. \begin{array}{l} P_{im}(k+1|k) \\ P_{in}(k+1|k) \\ P_{io}(k+1|k) \\ \vdots \\ P_{ip}(k+1|k) \end{array} \right\} \text{ for } 1 \leq i \leq I. \quad (52)$$

The required blocks are again evaluated from the main and upper diagonal block entries of $\mathcal{P}(k+1|k)$ using Result 3.

- 4) **Filter Covariance Update:** The tridiagonal entries $P_{ii}(k+1|k+1)$ and $P_{ii+1}(k+1|k+1)$ for $1 \leq i \leq I$, are computed with (20). Note that the scan matrix $\Gamma(k+1)$ and the gain matrix $\chi(k+1)$ depend upon the rows scanned and need not be evaluated again. We use their value from step 3. Go back to step 1.

Scalar KBf: With the scalar KBf (sKBf), we update the point covariances $\{P_{ipjq}\}$ for $j = i$ and $j = i + 1$. The subscripts p and q vary from 1 to J , i.e., $1 \leq p, q \leq J$. This implies that we are updating the point cross covariances between field sites located on adjacent rows. This is equivalent to updating all entries P_{ipjq} in blocks P_{ii} and P_{ii+1} . Point covariances other than the ones updated above if needed can then be obtained by evaluating the corresponding block P_{ij} using Result 3, and selecting the appropriate entry.

We refer to the algorithm given in (17)–(22) coupled with Result 3 as the *localized block Kalman–Bucy filter* (lbKBf). Similarly, the algorithm given in (23)–(26) coupled with Result 3 is referred to as the *localized scalar Kalman–Bucy filter* (lsKBf). The significance of this approximation is shown next when we compare the total number of floating point operations (flops) and the storage requirements for these two implementations with the number of flops of the conventional KBf.

VI. COMPUTATIONAL AND STORAGE REQUIREMENTS

In this section, we evaluate the number of floating point operations (flops) of the lbKBf for scanned measurements. We consider a field ψ defined on a square lattice of size $I \times I$. The PDE's describing the field when discretized by the FE method lead to a state vector Ψ of size I^2 . The state blocks B_1 , B_2 , and B_3 are of order I . These state blocks are themselves sparse. In (12), for example, B_1 and B_3 are diagonal and B_2 tridiagonal. The number of flops can be further decreased by using the sparse structure of these state blocks. In the following discussion, we will however assume that B_1 , B_2 , and B_3 are full matrices of order I . For simplicity, we assume that, during

each satellite pass, a single row, say n , is scanned. Later, we comment on multiple N scans. In evaluating the flops, we count only multiplications, ignoring the number of additions.

A. Predictor Update

For the general case, when the state blocks are full matrices, the flops required by (18), are $3I^3$.

B. Predictor Covariance Update

To evaluate the main diagonal blocks $P_{ii}(k+1|k)$, $1 \leq i \leq I$, from (18), the total number of computations is $14I^4$. To compute the upper diagonal blocks $P_{ii+1}(k+1|k)$'s, the first two terms enclosed in $[\cdot]$ in (18) are the same as the last two $[\cdot]$ terms evaluated while updating the blocks $P_{ii}(k+1|k)$'s. The flops required to update the blocks P_{ii+1} 's are therefore reduced to $4I^4$.

In (18), we need in addition the block entries $P_{ii+2}(k|k)$'s and $P_{ii+3}(k|k)$'s of the second and third upper diagonals which are computed using (49)–(50). The flops required are $4I^4$. The number of total flops used to update the predictor covariance, (18), in the lbKBf are then $22I^4$.

C. Filter Update

For a single scan n , the filter update, (19), simplifies to

$$\begin{aligned} \hat{\Psi}_i(k+1|k+1) &= \hat{\Psi}_i(k+1|k) + P_{in}(k+1|k)\Theta_n^T(k+1) \\ &\quad \cdot [R_n + \Theta_n(k+1)P_{nn}(k+1|k)\Theta_n^T(k+1)]^{-1} \\ &\quad \cdot [\mathcal{Y}_n - \Theta_n(k+1)\hat{\Psi}_n(k+1|k)]. \end{aligned} \quad (53)$$

In (53), the term

$$\begin{aligned} &\Theta_n^T(k+1)[R_n + \Theta_n(k+1)P_{nn}(k+1|k)\Theta_n^T(k+1)]^{-1} \\ &\quad \cdot [\mathcal{Y}_n(k+1) - \Theta_n(k+1)\hat{\Psi}_n(k+1|k)] \end{aligned} \quad (54)$$

depends only on n and not on the spatial index i . This term is evaluated once and used for different values of i . During each filter update, off diagonal blocks other than the blocks $P_{ii}(k+1|k)$'s and $P_{ii+1}(k+1|k)$'s are also needed. Which off diagonal blocks are needed depends upon the row scanned by the measurement device or alternatively on the value of n . We consider the case when n equals one. With other values of n , the computations remain the same or get marginally reduced. For $n = 1$, $(I - 2)$ covariance blocks $P_{13}(k+1|k)$, $P_{14}(k+1|k) \cdots P_{1I}(k+1|k)$ are needed. These blocks are computed recursively using Result 3. The subscript i equals one and j varies from three to I . The resulting equations are

$$P_{13}(k+1|k) = P_{12}(k+1|k)P_{22}^{-1}(k+1|k)P_{23}(k+1|k) \quad (55)$$

$$P_{14}(k+1|k) = P_{13}(k+1|k)P_{33}^{-1}(k+1|k)P_{34}(k+1|k) \quad (56)$$

\vdots

$$P_{1j}(k+1|k) = P_{1j-1}(k+1|k)P_{j-1j-1}^{-1}(k+1|k) \cdot P_{j-1j}(k+1|k) \quad \text{for } 5 \leq j \leq I. \quad (57)$$

TABLE I
FLOPS COUNT FOR THE lbKBF

Operation	Flops
Predictor field update	$3I^3$
Predictor covariance update	$22I^4$
Filter field update	$3I^4 - I^3 + 2I^2$
Filter covariance update	$6I^4 - 3I^3$
Total	$31I^4 - I^3 + 2I^2$

Adding up the number of computations needed, the total number of flops for the filter update is $(3I^4 - I^3 + 2I^2)$.

D. Filter Covariance Update

The filter covariance update, (20), for a single scan n , is reduced to

$$\begin{aligned}
 & P_{ij}(k+1|k+1) \\
 &= P_{ij}(k+1|k) - P_{in}(k+1|k) \\
 & \cdot \left[\Theta_n^T(k+1) [R_n + \Theta_n(k+1) P_{nn} \Theta_n^T(k+1)]^{-1} \right. \\
 & \left. \cdot \Theta_n(k+1) P_{nj}(k+1|k) \right] \quad (58)
 \end{aligned}$$

where the term enclosed in the rectangular box was evaluated in the filter update. The additional error covariance blocks needed are $P_{in}(k+1|k)$'s and $P_{ni+1}(k+1|k)$'s which were also evaluated earlier in the filter update. The total number of flops is therefore $(6I^4 - 3I^3)$.

Table I summarizes the total number of floating point operations required by the lbKBF with data from a single row. For N multiple scans, since the predictor equations (17)–(18) are unchanged, the total number of flops (second and third rows in Table I) remains the same. The filter equations (19)–(20) are more complex. Also, several covariance blocks other than the ones specified above, are required. Which blocks are required is determined by the measurement matrix \mathcal{H} which depends on the number and location of rows scanned. By repeating the previous calculations for multiple N rows, it can be verified that the number of computations in the filter equations, (19)–(20), of the lbKBF is roughly N times the computations shown in rows 4 and 5 of Table I. Usually, with satellite altimetry, N is small.

To avoid repetition we do not evaluate the number of flops for the scalar KBf (sKBf). The predictor (17) and (18) is unchanged. The number of flops for the filter field update, (23), and for the filter covariance update, (24), (the last two row entries in the table) are further reduced by approximately a factor of N/I .

In the conventional KBf, the number of computations required are, [19], $7l^3 + (3/2)l^2 + (1/2)l$ where l is the dimension of the state vector. For a grid of size $I \times I$, l equals I^2 , and so the total flop count is $7I^6 + (3/2)I^4 + (1/2)I^2$. This means that the lbKBF provides a simplification of $O(I^2)$.

1) *Storage*: A final note on the storage requirements of the lbKBF. In the lbKBF, instead of storing the error covariance matrix P , which for the lattice size considered earlier is an $I^2 \times I^2$ matrix, we store I blocks P_{ii} and $(I-1)$ blocks P_{ii+1} which are all $I \times I$ matrices. This reduces the storage

requirements from $O(I^4)$ to $O((2I-1)I^2)$ or, a reduction by a factor of I over the conventional KBf.

VII. APPLICATION: OCEAN CIRCULATION

We illustrate the use of the lbKBF in the context of physical oceanography by applying the lbKBF to assimilate satellite altimeter data in ocean circulation. The physics underlying ocean circulation are captured by a set of primitive equations, which under a standard set of assumptions are of the Navier Stokes (NS) type [10], [20]. The primitive equations are further simplified by fine tuning them to the spatial and temporal scales of interest (scale analysis) and to specific domains. Oceanographers may study global oceans or may be interested in coastal waters, or may consider domains of particular interest such as the Gulf Stream, the Sea of Japan, or the Mediterranean Sea. We use a particularly simple model that oceanographers apply to study circulation in equatorial waters—the region within latitudes of less than 30° . This model is commonly referred to as the equatorial beta plane model and, for shallow water, is given by [10]

$$\frac{\partial u}{\partial t} - \beta y v = -g \frac{\partial \eta}{\partial x} + \frac{X}{\rho H} \quad (59)$$

$$\frac{\partial v}{\partial t} + \beta y u = -g \frac{\partial \eta}{\partial y} + \frac{Y}{\rho H} \quad (60)$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}(Hu) + \frac{\partial}{\partial y}(Hv) = 0 \quad (61)$$

where β is a constant to approximate the effects of the earth's rotation and is given by 2.3×10^{-11} /ms; (u, v) are the latitudinal and the longitudinal velocity components of the ocean circulation; η is the vertical displacement about the ocean depth H ; x and y are the eastward and the northward distances; and (X, Y) are the surface wind stresses. Other types of forcing can be added to (59)–(61), so X and Y may be given a broader interpretation.

Numerical Models: The ocean basin in these experiments is rectangular with dimensions 1500 km in the east–west direction and 600 km in the north–south direction. We discretize the beta-plane equations, (59)–(61), by the leap frog method, [11], which uses a second order quantization of the time derivative. The state Ψ of the dynamic equation, (7), at time instant k contains the velocity fields (u, v) and the displacement η at instants k and $(k-1)$. The mean ocean depth, H , is assumed 1 km. The specific volume which is the inverse of the sea water density ρ is $9.7582 \times 10^{-4} \text{ m}^3 \text{ kg}^{-1}$. The model is run with the nominal wind stress pattern (X, Y) , [21], as follows:

$$\begin{aligned}
 X &= -X_m \sin \frac{\pi x}{a} \cos \frac{\pi y}{b} \quad \text{and} \\
 Y &= Y_m \cos \frac{\pi x}{a} \sin \frac{\pi y}{b} \quad (62)
 \end{aligned}$$

with $X_m = Y_m = 1 \text{ m}^2$, $a = 1500 \text{ km}$ and $b = 200 \text{ km}$.

Experimental Results: The experiments presented here are designed to make two major points. First, we examine the effects of our localized approximation and compare the results of the lbKBF to the optimal KBf. Our primary concern is to observe how closely the suboptimal lbKBF approximates the optimal estimates by comparing the estimation errors

associated with the suboptimal and the optimal filter. This is the motivation behind the first set of experiments in which we run a simplified problem with grid of dimensions 11×11 . The state Ψ is of dimension $2 \times 3 \times 11 \times 11 = 726$ and the optimal KBf is of $O(10^8)$ which is still feasible. In the second set of experiments, we use a larger field grid of dimensions 31×31 . The dimension of the state Ψ in the latter case is $2 \times 3 \times 31 \times 31 = 5766$. An exact implementation of the KBf is of $O(10^{11})$ and not practical. These experiments emphasize the usefulness of the lbKBf for large fields and illustrate the improvement it offers over predictions solely based on integrating forward the dPDE models. The results presented here are based on several Monte Carlo runs. Before conducting our study, the discretized model is spun-up from rest in each set of experiments and integrated forward in time. We run the model under two different setups: the “deterministic” run and the “realworld” run.

Deterministic Run: This integration represents the succession of states that are obtained from numerical simulation of the physical model as performed in an oceanography laboratory. Starting from a known initial state, we carry out a controlled forward integration of the beta plane model excited by the wind forcings as defined in (62). We call these fields the deterministic fields.

Realworld Run: This run simulates the groundtruth and provides us with the *real world* succession of the states of the ocean circulation. Random perturbations are added to the initial states and to the external forcing terms used in the deterministic run. The perturbations reflect our lack of complete and accurate knowledge of these initial states and the forcings. We call the resulting fields the realworld fields.

The two models discussed above are integrated forward for the same time duration till stability in the model energies is observed. The resulting fields provide the starting point for our data assimilation experiments. We scan the realworld fields according to a predefined scan pattern. To these measurements we add noise. These noisy measurements of the realworld fields are then assimilated via the KBf in the optimal case or the lbKBf in the suboptimal case to the deterministic fields. The performance of the assimilation algorithms is monitored by calculating the mean square error (MSE) for the various fields. The MSE for SSH η at time iteration k is given by

$$\text{MSE}^{(k)} = \sum_{i=1}^I \sum_{j=1}^J \frac{(\eta_{\text{real}}^{(k)}(i, j) - \eta_{\text{data}}^{(k)}(i, j))^2}{IJ} \quad (63)$$

where η_{real} and η_{data} are the realworld SSH field and the data assimilated SSH field. Similarly, for u and v .

Experiment 1: In this experiment, we scan the sea surface height field, η . At each field update, we assume that a single row of the sea surface height η is scanned. A zero-mean white Gaussian process is added to the observed data to simulate noisy measurements with SNR of about 7 dBs.

Fig. 2 shows the evolution over time of the MSE for the optimal KBf and for the lbKBf. In each case, the MSE is expressed as a percentage of the energy of the realworld sea surface height field, η . The solid line near the bottom represents the estimation error of the KBf and the dotted line

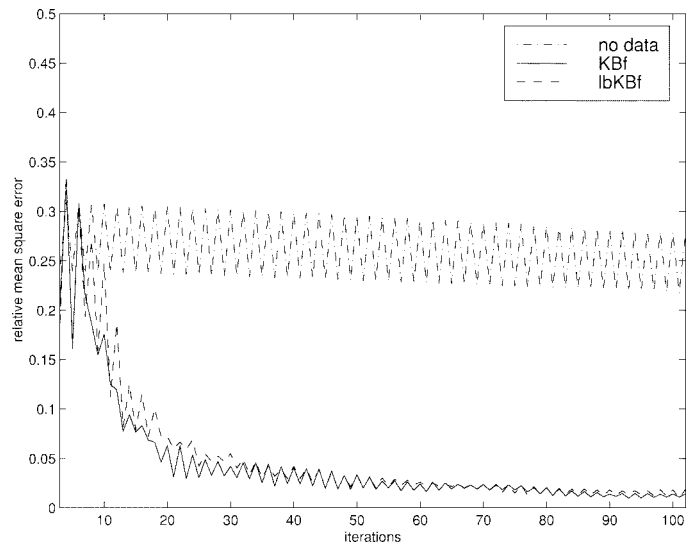


Fig. 2. Results from Experiment 1, total grid size is 11×11 : Comparison of the mean square error (MSE) for the sea surface height for the sub-optimal lbKBf and for the optimal KBf as a function of time. The MSE values on the y -axis are normalized by the energy of the realworld SSH. The x -axis represent time iterations, each time step is 2 min.

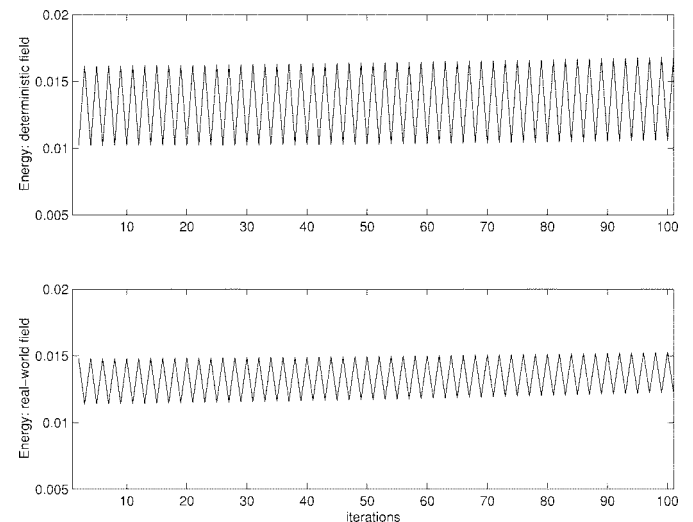


Fig. 3. Results from Experiment 1: Time evolution of the energy for the sea surface height for the deterministic field and for the realworld field. The units for the y -axis are m^2 . The x -axis represent time iterations; each time step is 2 min.

represents the estimation error of the lbKBf. For reference, we include in Fig. 2 the relative MSE when no data is assimilated. This is represented by the semisolid oscillating line near the top. As it can be observed, the KBf and the lbKBf reduce significantly the error between the deterministic and the realworld fields. More importantly, the lbKBf follows closely the optimal KBf showing that the Gauss–Markov approximation is actually a very good approximation for this problem.

The oscillations in Fig. 2, which are more pronounced when no data is assimilated, are explained in Fig. 3. This figure plots the energies of the deterministic and realworld sea surface height fields. As shown, the energies in these two fields are oscillating and these oscillations are reflected in the MSE plots.

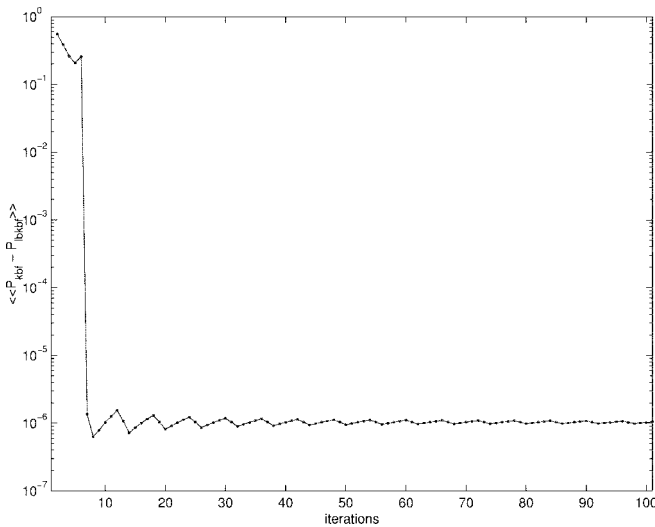


Fig. 4. Results from Experiment 1: comparison of the error covariance matrices for the suboptimal lbKbF and for the optimal KbF as a function of time. The 2-norm difference of the error covariance matrices of the lbKbF and the KbF is shown with respect to time iterations; each time step is 2 min.

To measure the closeness of the approximation of the error covariance matrix to the actual covariance matrix, we plot in Fig. 4 the approximation errors, defined as $\|\mathcal{P}_{\text{kbF}} - \mathcal{P}_{\text{lbkbF}}\|/I^2$, where \mathcal{P}_{kbF} is the optimal error covariance matrix and $\mathcal{P}_{\text{lbkbF}}$ its localized approximation. Both matrices are of dimension I . The 2-norm or Frobenius norm is used as matrix norm. The plot shows that after a short transient the difference between the error covariance matrices of the optimal KbF and its approximation, the lbKbF, is small. In this experiment, we have shown that the lbKbF can be used to efficiently approximate the optimal KbF. Next, we present numerical results to illustrate that data assimilation can significantly improve the prediction of the ocean circulation fields.

Experiment 2: We use a grid size of 31×31 and assimilate noisy measurements of the realworld fields with deterministic fields using the lbKbF. Again, the data observed is the SSH. The primary reason for our interest in the processing of such measurements lies with the availability of similar data due to the launching of the American/French Topex/Poseidon (T/P) altimeter—a satellite based platform that measures the ocean surface height with an accuracy of about 5 cm, [22]. Our measurement model assumes that the altimetry data lies along a single row for each satellite scan. This is only a crude approximation to how T/P scans the ocean surface.

To simulate the errors introduced by miscorrection of the geoid, the observation noise, and other sources of error, we add uncorrelated Gaussian noise with an SNR of 13 dBs. We scan every fifth row starting from the first row. The duration in between two satellite scans is 1 h. For the ocean domain considered above, the field update time, Δt , is 2 min. This implies that the field is updated 30 times in between two successive measurements.

Next, we consider four different patterns for the available altimetry data: i) prediction case: no data assimilation; ii) single recurrent scan pattern; iii) single non-recurrent scan pattern; and iv) dual recurrent scan pattern.

Single Recurrent Scan Pattern: We assume that observations from a single satellite are available. The satellite measures the SSH at spatially separated points along the rows of the ocean domain grid. Starting its measurements from the first row, every fifth row is scanned. Successive line scans are separated by a duration of one hour.

Single Nonrecurrent Scan Pattern: The measurement model is similar to the one considered above except that the satellite follows a nonrecursive or a nonrepetitive pattern. During two consecutive orbiting cycles, different rows are scanned. In the first orbit, for example, the satellite scans rows 1, 6, 11, and so on till the entire ocean basin is covered. In the second orbit, rows 2, 7, 12, and the corresponding higher numbered rows are scanned. Earth orbiting satellites normally follow the same tracks during each orbit. We perform this experiment to determine the improvement in the estimation process provided by measurements that are spread over the entire ocean domain, rather than measurements following the same recurrent pattern.

Dual Recurrent Scan Pattern: Here, measurements from two satellites are assumed available. Combining altimeter data from two satellites faces several problems. The main obstacle to merging data sets from separate altimeter satellites is the different marine geoid sampled along the ground tracks of the various altimeter missions. Recent altimeters (ERS 1, T/P, and Geosat-ERM) have been in different orbits, sampling different geographical points and hence different locations on the marine geoid. In our experiments, we assume appropriate correction has been provided for the respective geoids. Our main concern is to quantify the improvement that data assimilation from multiple satellites offers versus a single satellite. The first satellite begins its scans from row 1. For a grid with dimensions $I \times J$, the second satellite starts from row $J/2 + 1$. The ocean domain that we consider has dimensions 31×31 . Therefore, the second satellite scans first row 16.

The performance of the assimilation algorithm is monitored by calculating the MSE for the various fields. To get a feel for the quality provided by data assimilation, we have generated a movie [23] that displays the time evolution of the *realworld fields*, the *deterministic fields*, and the *data assimilated fields*. In this paper, we focus on comparing the results for the SSH field and include typical contour plots from the video. The velocity components (u, v) follow a similar trend. We discuss the results next.

a) Quantitative comparison: The SSH fields assimilated with the satellite scanned data are better estimates of the actual ocean state conditions, in our case the realworld fields, than the fields predicted by the deterministic run. This is illustrated in Fig. 5, where we plot in units of meters square the MSE for the final 50 data assimilated updates. This corresponds to a duration of 50 h. The plot for the first 350 h of data assimilation is not shown in here but a similar behavior was observed during this initial period. In all cases, the data assimilation fields have a much smaller MSE than the deterministic fields. In the case of a single satellite, a nonrecurrent scan pattern provides a better estimate than a recurrent scan pattern. The former uses data from different rows during successive satellite

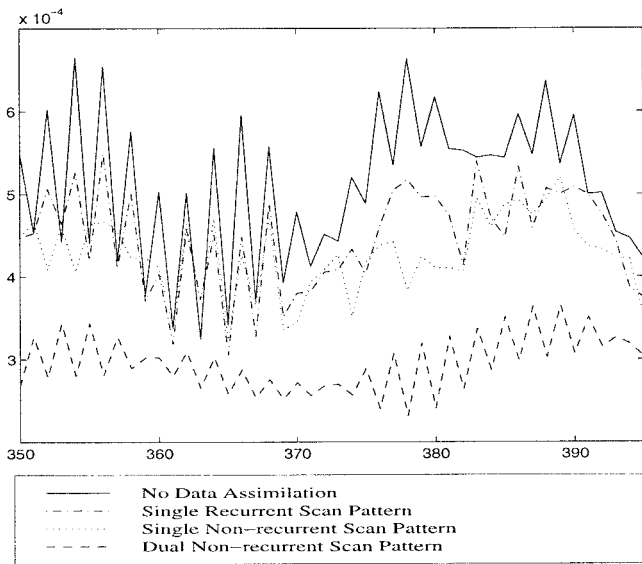


Fig. 5. Results from Experiment 2: comparison of actual MSE error shown in units of m^2 of the sea surface heights versus time. The final duration of 50 h of data assimilation is plotted here. Solid curves are with no data being assimilated. Dashed curves are with assimilation of sea surface height obtained via different schemes described in the above key. The units for the vertical axis are m^2 and for the horizontal axis are hours.

orbits whereas in the latter the same scanning pattern is observed for all orbits over the ocean. This is intuitively pleasing, since better performance is expected from a more diversified measurement pattern.

Using the measurements provided by two satellites results in significant improvement over the fields reconstructed using the measurements of a single satellite. The errors at all updates are half the errors for the deterministic run and at least 15–25% less than the errors for the other two data assimilated fields obtained from assimilating data from a single satellite with recurrent and nonrecurrent scan patterns. The orbits of the two satellites are offset in space so that the second satellite scans a fixed distance apart from the first satellite. There is no offset in time. In our experiments, the distance offset was half of the width of the ocean basin under study. We performed several additional experiments not included in the paper due to lack of space to test the sensitivity of the assimilation to the choice of separation in between the two satellites. The results show that a spacing of half the rectangular ocean basin is optimum.

Next, we compare the subjective quality of the reconstructed fields. Due to lack of space, we include only the contour plots of the sea surface height (SSH) for the dual recurrent scan pattern which provides the minimum MSE. We compare its subjective quality with the realworld SSH field and the deterministic SSH field.

b) Qualitative comparison: The realworld SSH field, the deterministic SSH field, and the data assimilated SSH field are shown from top to bottom by this order in Fig. 6. In these SSH contour plots, the higher elevation regions are shown in white while the deeper regions are shown in different shades of gray. We use the same scale and color-map in all three plots, shown at the top of Fig. 6.

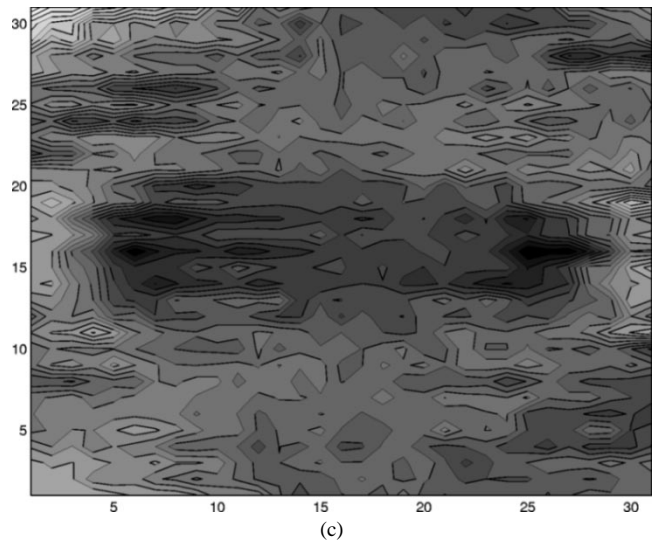
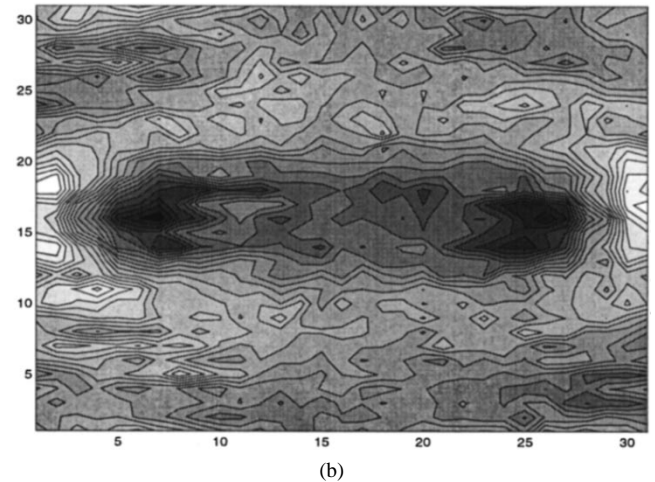
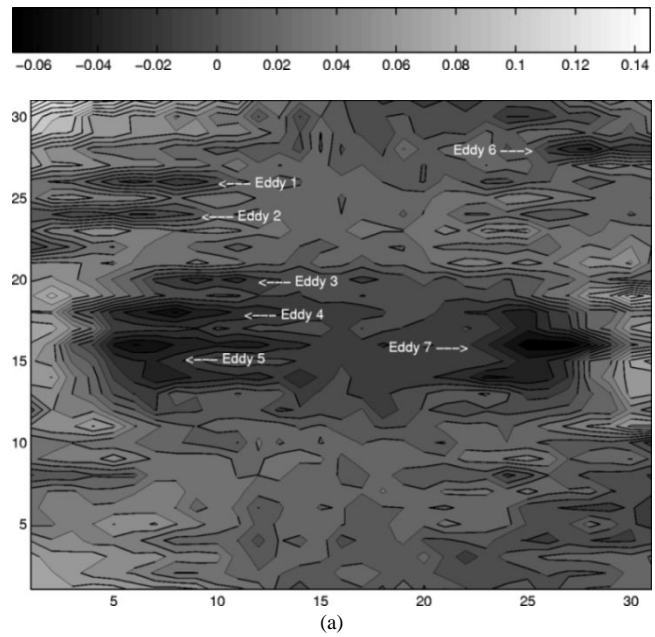


Fig. 6. Results from Experiment 2, ocean basin is $1500 \text{ km} \times 600 \text{ km}$ and the total grid size is 31×31 : contour plots of the sea surface height (SSH). (a) SSH realworld field. (b) SSH deterministic field (no data assimilation). (c) SSH data assimilated fields (altimetry data is assimilated). The fields are shown after seven days and 9 h of data assimilation. Darker regions are deeper, lighter regions are higher (see scale on top in meters).

A close visual comparison of Fig. 6(a) and (b) reveals that there are several major differences in the two contour plots. The deterministic field, Fig. 6(b), does not capture many of the finer details nor the high frequency features such as the eddies seen in the realworld field, see Fig. 6(a). To facilitate comparison, we have labeled in Fig. 6(a) eddies 1–7, which are either absent or masked in Fig. 6(b). Eddies 1, 2, and 6 are completely absent from Fig. 6(b) while eddies 3, 4, and 5 appear merged together into one big eddy. Even eddy 7 though located at the correct coordinates, has a different shape. On the other hand, the SSH field shown in Fig. 6(c), which is produced by assimilating data via the lbKBf, is visually closer to the realworld field, Fig. 6(a). The data assimilated SSH improves on the frequency contents of the deterministic SSH, eddies 1–7 are replicated with higher accuracy.

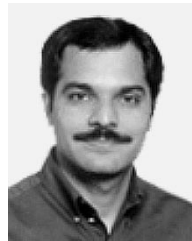
In Experiment 2, we have shown that the lbKBf couples successfully the altimetry data with the ocean circulation fields obtained by direct integration of the Navier Stokes equation and improves considerably the field estimates. We summarize the main contributions of the paper next.

VIII. SUMMARY

The paper considered data assimilation in problems in the physical sciences, in particular, in physical oceanography. In these problems, the underlying models are derived by discretization of partial differential equations (dPDE's). These models are highly structured. A second important characteristic in these data assimilation problems is the sparseness of the measurements available. These measurements arise from satellite scans or from isolated, moored, or drifting buoys. Direct application of KB filtering techniques requires $O(I^6)$ computations for 2-D time-varying fields. By exploiting the structure of the dynamical models and the sparseness of the data, and by approximating the error field in the KBf by an MRF, we derive two approximate implementations for the KBf, the localized bKBf (lbKBf), and the localized sKBf (lsKBf). These localized filters are obtained by an MRF approximation to the error field associated with the KBf. Essentially, this assumption enables us to express the off-diagonal block entries of the covariance matrix in terms of the block entries from the main and first upper diagonals, so that we need only propagate the block entries of these two diagonals. The localized implementations reduce the computational effort to $O(I^4)$ [from $O(I^6)$] and the storage requirements to $O(I^3)$ [from $O(I^4)$]. We showed that the Markov approximation is actually a good approximation, the lbKBf tracking closely the KBf assimilated fields. Finally, we successfully illustrated the application of lbKBf in assimilating altimetry satellite data in a Pacific equatorial basin. The satellite assimilated fields reproduce with good accuracy eddies, troughs, and other structures visible in the realworld fields.

REFERENCES

- [1] P. W. Fieguth, W. C. Karl, A. S. Willsky, and C. Wunsch, "Multiresolution optimal interpolation and statistical analysis of TOPEX/POSEIDON satellite altimetry," *IEEE Trans. Geosci. Remote Sensing*, vol. 33, pp. 280–292, Mar. 1995.
- [2] W. C. Thacker and R. B. Long, "Fitting dynamics to data," *J. Geophys. Res.*, no. 93(C2), pp. 1227–1240, 1988.
- [3] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *ASME J. Basic Eng.*, vol. 83, pp. 95–108, 1961.
- [4] T. M. Chin, W. C. Karl, and A. S. Willsky, "Sequential filtering for multi-frame visual reconstruction," *Signal Process.*, vol. 28, pp. 311–333, Aug. 1992.
- [5] ———, "A distributed and iterative square root algorithm for space-time filtering," *Automatica*, vol. 31, pp. 67–82, Jan. 1995.
- [6] R. Szeliski, *Bayesian Modeling of Uncertainty in Low-Level Visual Reconstruction*. Norwell, MA: Kluwer, 1989.
- [7] A. Asif and J. Moura, "Assimilation of satellite data in beta plane ocean circulation models," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Detroit, MI, May 1995, vol. V, pp. 2789–2792.
- [8] J. M. F. Moura and N. Balram, "Recursive structure of noncausal Gauss Markov random fields," *IEEE Trans. Inform. Theory*, vol. 38, pp. 334–354, Mar. 1992.
- [9] N. Balram and J. M. F. Moura, "Noncausal Gauss Markov random fields: Parameter structure and estimation," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1333–1355, July 1993.
- [10] A. E. Gill, *Atmosphere-Ocean Dynamics*, vol. 30 of *International Geophysics Series*. New York: Academic, 1982.
- [11] J. C. Strikwerda, *Finite Element Schemes and Partial Differential Equations*. Pacific Grove, CA: Wadsworth & Brooks/Cole, 1989.
- [12] O. Zienkiewicz, *The Finite Element Method*. NY: McGraw-Hill, 1977.
- [13] I. Babuska, *Finite Elements, Theory, and Applications*. New York: Wiley, 1988, pp. 199–239.
- [14] A. W. Naylor and G. R. Sell, *Linear Operator Theory in Engineering and Sciences, Vol 42 of Applied Mathematical Sciences*, 2nd ed. Berlin, Germany: Springer-Verlag, 1982.
- [15] R. Courant, K. O. Friedrichs, and H. Lewy, "Über die partiellen differenzengleichungen der mathematischen physik," *Math. Ann.*, pp. 32–74, 1928.
- [16] A. Kavcic and J. M. F. Moura, "L-Band matrix extensions and applications," Tech. Rep., Dept. Elect. Eng., Carnegie Mellon Univ., Pittsburgh, PA, Feb. 1998, submitted for publication.
- [17] A. M. Erisman and W. Tinney, "On computing certain elements of the inverse of a sparse matrix," *Commun. ACM*, vol. 18, pp. 177–179, Mar. 1975.
- [18] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*. Oxford, U.K.: Oxford Univ. Press, 1992.
- [19] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice*, Information and System Science Series. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [20] G. Neuman and W. J. Pierson, *Principles of Physical Oceanography*. Englewood Cliffs, NJ: Prentice-Hall, 1966.
- [21] C. Le. Provost, "On the use of finite element methods for ocean modeling," in *Advanced Physical Oceanographic Numerical Modeling*, J. J. O'Brien, Ed. Boston, MA: Reidel, 1985, pp. 557–580.
- [22] C. S. Ruf, S. J. Keihm, and M. A. Janssen, "Topex/Poseidon microwave radiometer (TMR): I. Instrument description and antenna temperature calibration," *IEEE Trans. Geosci. Remote Sensing*, vol. 33, pp. 125–137, Jan. 1995.
- [23] A. Asif, "Satellite data assimilation in beta plane ocean circulation model," Dept. Elect. Comput. Eng., Carnegie Mellon Univ., Pittsburgh, PA, Aug. 1996, illustrative video.



Amir Asif (M'99) received the B.S. degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, with distinction and the highest honor in 1990 and the M.S. and the Ph.D. in electrical and computer engineering from Carnegie Mellon University (CMU), Pittsburgh, PA, in 1993 and 1996, respectively.

He is presently serving as Assistant Professor at the Technical University of British Columbia, Surrey, B.C., Canada. Prior to this, he was on the faculty of CMU, where he was a Research Engineer (1997–1999) and Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan, where he was an Assistant Professor (1996–1997). His research interests include multidimensional signal processing, image and video coding, multimedia communications, and underwater acoustics.

Dr. Asif is affiliated with several IEEE societies.



José M. F. Moura (S'71–M'75–SM'90–F'94) received the engenheiro electrotécnico degree in 1969 from Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and the D.Sc. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1973 and 1975, respectively.

He is presently a Visiting Professor of electrical engineering at MIT, on sabbatical leave from Carnegie Mellon University (CMU), Pittsburgh, PA, where he has been Professor of electrical and computer engineering since 1986. Prior to this, he was on the faculty of IST, where he was an Assistant Professor (1975), Professor Agregado (1978), and Professor Catedrático (1979). He has held visiting appointments at several Institutions, including MIT (Genrad Associate Professor of Electrical Engineering and Computer Science, 1984–1986) and the University of Southern California (Research Scholar, Department of Aerospace Engineering, Summers 1978–1981). His research interests include statistical signal processing and telecommunications, wavelets and time-frequency transforms, image processing, video representation, video editing, and manipulation. He has over 220 published technical contributions and is the co-editor of two books.

Dr. Moura is the Vice President Elect for Publications of the IEEE Signal Processing Society (SPS) and is a member of the Board of Governors of SPS. He is on the editorial board of the PROCEEDINGS OF THE IEEE, since January 1999. He has been the Editor in Chief for the IEEE TRANSACTIONS ON SIGNAL PROCESSING since October 1995. He is a member of the IEEE SPS Publications Board. He is currently a member of the Multimedia Signal Processing Technical Committee and a founding member of the Sensor Array and Multichannel Processing Technical Committee. He was a member of the Underwater Acoustics Technical Committee of the SPS until 1997, a member of the IEEE Press Board (1991–1995), a technical Associate Editor for the IEEE SIGNAL PROCESSING LETTERS (1993–1995), and an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1988–1992). He was a program committee member for several international conferences, including the IEEE International Conference on Image Processing (ICIP'95) and the IEEE International Symposium on Information Theory (ISIT'93). He has organized and co-directed two international scientific meetings on signal processing theory and applications, and has been a member of the organizing committee for several other international technical meetings on signal processing. He holds three patents in the areas of image and video processing and digital communications with the U.S. Office of Patents and Trade, and has given numerous invited seminars at U.S. and European universities and laboratories. He is a corresponding member of the Academy of Sciences of Portugal (Section of Sciences). He is affiliated with several IEEE societies, Sigma Xi, AMS, IMS, and SIAM.