ASSIMILATION OF SATELLITE DATA IN BETA-PLANE OCEAN CIRCULATION MODELS

Amir Asif and José M. F. Moura

Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890

ABSTRACT

The paper discusses a scheme based on Kalman–Bucy filters for the assimilation of satellite data in equatorial beta plane ocean circulation models. The state equation of the Kalman–Bucy filter is obtained by decoupling the non-linearities from the Navier-Stokes equations by assuming an inviscid isentropic shallow water motion. Direct application of the Kalman-Bucy filter leads to a computationally intensive algorithm which precludes its application to meaningful sized domains. By imposing a Gauss Markov random field (GMRF) structure on the error covariance matrix, we obtain an efficient recursive algorithm, capable of estimating the velocity fields and the sea surface height.

1. INTRODUCTION

The trend to achieve higher resolution in underwater applications is to couple detailed ocean general circulation models (OGCM) [1] to the signal processing algorithms, e.g., matched field approaches and acoustic tomography. These models of ocean circulation are based on the Navier-Stokes equations, but they do not provide high accuracy at all scales and at every latitude. Increased accuracy is obtained by coupling into these models data collected through a variety of measurement programs, e.g., satellite derived data, or data from moored or drifting buoys. In physical oceanography, combining the ocean model dynamics with ocean measurements is referred to as data assimilation [2].

Data assimilation in physical oceanography is challenging due to several reasons. We need to handle nonlinear models and couple in randomness introduced for example by the boundary conditions, forcing terms like wind stresses and evaporation, or other physical quantities which are assumed known. The volumes of data involved are large and highly sparse.

In this paper, we develop recursive processing algorithms based on discrete Kalman-Bucy filtering [3] for the assimilation of satellite data. The nonlinearities of the model are decoupled from the random aspects by assuming these to lead to small perturbations of the nominal nonlinear solution to the primitive equations. We consider an inviscid, isentropic, shallow water motion on a Beta plane. The state equation for the Kalman-Bucy filter is obtained by discretizing the Beta plane approximation to the Navier-Stokes equations and the continuity equations using finite-difference approximations. The resulting state transition matrix is block tridiagonal and the observation equation based on the data-patterns obtained from the satellite scans has a highly sparse observation matrix. This leads to efficient recursive algorithms, capable of computing both the estimates and the error statistics for the velocity fields and the sea surface height.

The paper is structured in five major sections. In Section 2, we discuss briefly the discretization of the Navier-Stokes equation, simplified over a Beta plane. In Section 3, based on Kalman-Bucy Filtering, we present an efficient and recursive algorithm to estimate the ocean circulation. Finally in Sections 4 and 5, we provide experimental results and summarize the paper.

This work was partially supported by ONR grant #N00014-91-J-1001.

2. MODEL FORMULATION

2.1. Equatorial Beta Plane

We start by considering the set of the Navier-Stokes (NS) equations (eqns. 1–3) and the continuity equation (eqn. 4), [4], for the sake of completeness and to introduce notation.

$$\frac{du}{dt} - (2\Omega + \frac{u}{r\cos\phi})(v\sin\phi - w\cos\phi) = -\frac{1}{r\cos\phi}\frac{\partial p}{\partial\lambda} \quad (1)$$

$$\frac{dv}{dt} + \frac{wv}{r}(2\Omega + \frac{u}{r\cos\phi})u\sin\phi = -\frac{1}{\rho}\frac{\partial p}{\partial\phi} \quad (2)$$

$$\frac{dw}{dt} - \frac{v^2}{r}(2\Omega + \frac{u}{r\cos\phi})u\cos\phi = -\frac{1}{\rho}\frac{\partial p}{\partial z} - g \quad (3)$$

$$\frac{1}{\rho}\frac{d\rho}{dt} + \frac{1}{r} + \frac{1}{r\cos\phi}\frac{\partial}{\partial\phi}(v\cos\phi) + \frac{\partial w}{\partial z} = 0 \quad (4)$$

where λ, ϕ, and r are the longitude, the latitude and the radial co-ordinate, u, v, and w the velocity components associated with the co-ordinates λ, ϕ, and r respectively, ρ the density, Ω the Earths angular velocity, and p the pressure. In equations (1–4) we have assumed an inviscid flow and ignored any forcing components.

We are interested in developing a numerical model for assimilating observed data into the ocean model dynamics. A model based on equations (1–4) is difficult and computationally intensive. We can however decouple the nonlinearities in equations (1–4) by considering an inviscid, isentropic, shallow water motion on an equatorial beta plane – the region which lies at latitudes of less than 30 degrees. In such a plane, equations (1-4) can be approximated by a set of linearized equations, given below (see [5] for details)

$$\frac{\partial u}{\partial t} - \beta y v = -g\frac{\partial \eta}{\partial x} + \frac{X}{\rho H} \quad (5)$$

$$\frac{\partial v}{\partial t} + \beta y v = -g\frac{\partial \eta}{\partial y} + \frac{Y}{\rho H} \quad (6)$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}(Hu) + \frac{\partial}{\partial y}(Hv) = -\frac{E}{\rho} \quad (7)$$

where β is a constant given by $2.3 \times 10^{-11}/\mathrm{ms}$, η the vertical displacement of the ocean, H the ocean depth, x and y the eastward and the northward distances, given by $r\lambda$ and $r\phi$ respectively, and (X,Y) and E the forcing terms i.e., the surface stress, and the evaporation rate respectively. Any form of forcing can be added to the right side of equations (5–7), so X, Y, and E can be given a wider interpretation.

By using an equatorial beta plane approximation, we have reduced the primitive NS equations and the continuity equation to a set of linear partial differential equations (5–7), which are easier to discretize and implement. These equations will be used in our data assimilation model.

2.2. Finite Difference Approximation

There is a rich class of numerical schemes to solve partial differential equations based on finite difference schemes, finite element schemes, or spectral schemes. We use the Lax Friedrich method to discretize equations (5–7). Lax Friedrich method is a second order finite difference scheme, which approximates the first order time and the spatial derivatives of a function, say Ψ, in the (x,y) plane at time t as follows

$$\frac{\partial \Psi}{\partial t} = \frac{\Psi_{ijk+1} - \frac{1}{4}(\Psi_{i-1jk} + \Psi_{i+1jk} + \Psi_{ij-1k} + \Psi_{ij+1k})}{\Delta t} \quad (8)$$

$$\frac{\partial \Psi}{\partial x} = \frac{\Psi_{i+1jk} - \Psi_{i-1jk}}{2\Delta x} \quad (9)$$

$$\frac{\partial \Psi}{\partial y} = \frac{\Psi_{ij+1k} - \Psi_{ij-1k}}{2\Delta y} \quad (10)$$

where we have divided the (x,y) plane into a uniform grid of width Δx and Δy. The time duration t is divided in intervals of Δt. The indices i and j represent the point (x_i, y_j), which correspond to the point $(i\Delta x, j\Delta y)$ on the grid. The index k represents the time instant t_k or $k\Delta t$. Ψ_{ijk} is the discretized value of Ψ at the coordinate (x_i, y_j) at time instant t_k and equals $\Psi(x_i, y_j)$, and similarly for the other terms in equations (8–10). For simplicity, we assume a rectangular (x,y) plane of finite dimensions so that i and j have an upper limit, say I and J respectively.

Following standard procedure, we stack the values of u for row i at time index k into the $J \times 1$ vector $U_i^k = [u_{i1k} \quad u_{i2k} \ldots u_{iJk}]^T$. We do a similar procedure on v, η, X, Y, and E so that their ith rows at time index k are represented by $J \times 1$ vectors V_i^k, $\mathcal{N}_i^k$, X_i^k, Y_i^k and E_i^k respectively. Finally U_i^k, V_i^k, and $\mathcal{N}_i^k$ are stacked together into the $3J \times 1$ vector $\mathcal{X}_i^k = [U_i^{k^T} \quad V_i^{k^T} \quad \mathcal{N}_i^{k^T}]^T$, and X_i^k, Y_i^k, and E_i^k stacked together into the $3J \times 1$ vector $W_i^k = [X_i^{k^T} \quad Y_i^{k^T} \quad E_i^{k^T}]^T$. Equations (5–7), after discretizing by the Lax Friedrich scheme, are expressed in a matrix form by stacking the values of u_{ijk}, v_{ijk}, and η_{ijk}. Taking into account appropriate boundary conditions, we obtain

$$\begin{bmatrix} \mathcal{X}_1 \\ \mathcal{X}_2 \\ \mathcal{X}_3 \\ \vdots \\ \mathcal{X}_I \end{bmatrix}^{k+1} = \begin{bmatrix} B_1 & C & \underline{0} & . & . & . \\ D & B & C & \underline{0} & . & . \\ \underline{0} & D & B & C & \underline{0} & . \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ . & . & . & \underline{0} & D & B_2 \end{bmatrix} \begin{bmatrix} \mathcal{X}_1 \\ \mathcal{X}_2 \\ \mathcal{X}_3 \\ \vdots \\ \mathcal{X}_I \end{bmatrix}^{k} + \begin{bmatrix} F & \underline{0} & . & . & . & . & . \\ \underline{0} & F & \underline{0} & . & . & . & . \\ . & \underline{0} & F & \underline{0} & . & . & . \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ . & . & . & . & . & \underline{0} & F \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \vdots \\ W_I \end{bmatrix}^{k} \quad (11)$$

where B, B_1, B_2, C, D, and F are all $3J \times 3J$ blocks. The blocks B_1, and B_2 are slight modifications of blocks B accounting for the boundary conditions. The blocks B, C, D, and F have the following structure

$$B = \begin{bmatrix} \frac{1}{4}H_J^1 & \beta\Delta t\Delta y\Phi_J & \underline{0} \\ -\beta\Delta t\Delta y\Phi_J & \frac{1}{4}H_J^1 & -g\lambda_y K_J^1 \\ \underline{0} & -H\lambda_y K_J^1 & \frac{1}{4}H_J^1 \end{bmatrix} \quad (12)$$

$$C = \begin{bmatrix} \frac{1}{4}I_J & \underline{0} & -g\lambda_x I_J \\ \underline{0} & \frac{1}{4}I_J & \underline{0} \\ -g\lambda_x I_J & \underline{0} & \frac{1}{4}I_J \end{bmatrix} \quad (13)$$

$$D = \begin{bmatrix} \frac{1}{4}I_J & \underline{0} & g\lambda_x I_J \\ \underline{0} & \frac{1}{4}I_J & \underline{0} \\ g\lambda_x I_J & \underline{0} & \frac{1}{4}I_J \end{bmatrix} \quad (14)$$

$$F = \begin{bmatrix} \frac{\Delta t}{\rho H}I_J & \underline{0} & \underline{0} \\ \underline{0} & \frac{\Delta t}{\rho H}I_J & \underline{0} \\ \underline{0} & \underline{0} & -\frac{\Delta t}{\rho}I_J \end{bmatrix} \quad (15)$$

where λ_x and λ_y are given by $\frac{\Delta t}{2\Delta x}$ and $\frac{\Delta t}{2\Delta y}$. I_J is the identity matrix of order J. H_J^1 and Φ_J are $J \times J$ matrices given by

$$H_J^1 = \begin{bmatrix} 0 & 1 & 0 & . & . \\ 1 & 0 & 1 & . & . \\ 0 & 1 & 0 & 1 & . \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ . & . & 0 & 1 & 0 \end{bmatrix} \qquad \Phi_J = \begin{bmatrix} 0 & 0 & 0 & . & . \\ 0 & 1 & 0 & . & . \\ . & 0 & 2 & 0 & . \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ . & . & 0 & 0 & J-1 \end{bmatrix}$$

K_J^1 is the same as H_J^1 except the entries of the lower diagonal, which are -1. In the following discussion, we will refer to equation (11) as the state equation. Solutions of the state equation are stable only if Δt, Δx, and Δy, satisfy Courant, Friedrich, and Lewy (CFL) conditions [6]. For the above discretization, the CFL conditions are of the type

$$(\beta J \Delta y)^2 + \frac{gH}{\Delta x^2} + \frac{gH}{\Delta y^2} \leq \frac{1}{\Delta t^2} \quad (16)$$

3. DATA ASSIMILATION ALGORITHM

We will now describe the data assimilation algorithm. The vector $[W_1 \quad W_2 \ldots W_J]$ in the state equation contains the forcing terms, for example the wind stress (X,Y) and the evaporation rate E. Each of the forcing terms can be decomposed into deterministic and random components. The deterministic components are used in an ocean general circulation model (OGCM) to construct an ocean field, referred to as the mean ocean field. The random component drives the data assimilation model. We assume that the random components are white and Gaussian, and has a known covariance, i.e., $\mathcal{E}\{W_i W_j^T\} = Q_i \delta_{ij}$, where $\mathcal{E}$ denotes the expectation operator.

This paper is concerned with the assimilation of satellite data to ocean circulation models. Satellite data assimilation is extremely challenging since satellite measurements are at spatially separated points along the satellite track and are corrupted by noise. For simplicity, consider a rectangular basin discretized into a mesh of I rows and J columns, the rows oriented towards north and the columns towards east. It is assumed that the satellite scans one row during each satellite scan. During the duration of each satellite line scan, we further assume that the ocean characteristics do not change significantly. The satellite measurement model is given by

$$\mathcal{Y}_m^k = \Theta_m \mathcal{X}_m^k + \mathcal{W}^k \tag{17}$$

where $\mathcal{X}_m^k$ is the $3J \times 1$ vector defined earlier, $\mathcal{Y}_m^k$ the observation vector, and $\mathcal{W}^k$ the observation noise, assumed Gaussian with a covariance matrix denoted by R_m. The index m denotes the row scanned by the satellite.

Direct application of the Kalman–Bucy filter using equations (11) and (17) is computationally intense and difficult. We simplify the algorithm by using the tridiagonal block structure of the state transition matrix, i.e., the matrix containing blocks B, C, and D in equation (11), and the sparseness of the observation. We denote the error covariance matrix $\mathcal{P}$ to be $\{P_{ij}\}$ for $1 \le i, j \le I$. Substituting in the Kalman filtering equations [3], each vector and matrix by blocks, i.e., substitute $\mathcal{P}$ by $\{P_{ij}\}$ and similarly for the other variables, we get the recursive algorithm given below

<u>Predictor update:</u>

$$\widehat{\mathcal{X}}_i(k+1\,|\,k) = D\widehat{\mathcal{X}}_{i-1}(k\,|\,k) + B\widehat{\mathcal{X}}_i(k\,|\,k) + C\widehat{\mathcal{X}}_{i+1}(k\,|\,k) \qquad 1 \le i \le I \tag{18}$$

<u>Predictor covariance update:</u>

$$\begin{aligned} P_{ij}(k+1\,|\,k) &= Q\delta_{ij} + \\ &(DP_{i-1j-1}(k\,|\,k) + BP_{ij-1}(k\,|\,k) + CP_{i+1j-1}(k+1\,|\,k))D^T + \\ &(DP_{i-1j}(k\,|\,k) + BP_{ij}(k\,|\,k) + CP_{i+1j}(k+1\,|\,k))D^T + \\ &(DP_{i-1j+1}(k\,|\,k) + BP_{ij+1}(k\,|\,k) + CP_{i+1j+1}(k+1\,|\,k))D^T \\ &\qquad 1 \le i, j \le I \end{aligned} \tag{19}$$

<u>Filter update:</u> if $\text{rem}(k\Delta t, t_s) = 0$

$$\begin{aligned} \widehat{\mathcal{X}}_i(k+1\,|\,k+1) &= \widehat{\mathcal{X}}_i(k+1\,|\,k) + P_{im}(k+1\,|\,k)\Theta_m^T(k) \\ &(R_m + \Theta_i(k)P_{mm}\Theta_m^T(k))^{-1}(\mathcal{Y}_m - \Theta_m(k)\mathcal{X}_i(k+1\,|\,k)) \\ \text{else} \quad \widehat{\mathcal{X}}_i(k+1\,|\,k+1) &= \widehat{\mathcal{X}}_i(k+1\,|\,k) \qquad 1 \le i \le I \end{aligned} \tag{20}$$

<u>Filter covariance update:</u> if $\text{rem}(k\Delta t, t_s) = 0$

$$\begin{aligned} P_{ij}(k+1\,|\,k) &= P_{ij}(k+1\,|\,k) - \\ &P_{im}(k+1\,|\,k) + \Theta_i^T(R_m + \Theta_m P_{mm}\Theta_m^T)^{-1}\Theta_i P_{mj}(k+1\,|\,k) \\ \text{else} \quad P_{ij}(k+1\,|\,k+1) &= P_{ij}(k+1\,|\,k) \qquad 1 \le i, j \le I \end{aligned} \tag{21}$$

where $\text{rem}(k\Delta t, t_s)$ is the remainder of $k\Delta t/t_s$, and t_s is the duration inbetween two satellite scans. The OGCM provides estimates of the mean ocean fields, say fields for u, v, and η. To construct accurate maps, we need to couple the satellite data into these mean ocean fields. This is done by the data assimilation algorithm as follows.

For simplicity, we consider a rectangular basin, divided into a mesh of width Δx and Δy, resulting in say I rows and J columns. The update time Δt is evaluated using the CFL conditions of the type given in equation (16). The index k represents the update time t_k which equals $k\Delta t$. We start at $k = 0$. $\widehat{\mathcal{X}}_i(0|0)$ and $P_{ij}(0|0)$ are the initial conditions. Using equations (18) and (19), we form estimates of $\widehat{\mathcal{X}}_i(1|0)$ and $P_{ij}(1|0)$ for $1 \le i, j \le I, J$. After prediction, the predictor checks if satellite data is available. Assume that the satellite scans row m and measures u, v, and η. Mean ocean fields, u, v, and η, for row m are subtracted from the measurements and the difference is fed to the filter. Both $\widehat{\mathcal{X}}_i(1|1)$ and $P_{ij}(1|1)$ for $1 \le i, j \le I, J$ are then updated using equations (20) and (21). The predicted values $\widehat{\mathcal{X}}_i(1|1)$ for $1 \le i \le I$ are added to the mean fields i.e., to the input to the OGCM. If the satellite does not provide data, $\widehat{\mathcal{X}}_i(1|1)$ and $P_{ij}(1|1)$ are set equal to $\widehat{\mathcal{X}}_i(1|0)$ and $P_{ij}(1|0)$. This completes one iteration. The value of k is then incremented and the process repeated.

The data assimilation algorithm of equations (18–20) has one implementational issue. For the example considered above, the error covariance matrix $\mathcal{P}$ is of the order $3IJ \times 3IJ$. For meaningful sized domains, this will cause a serious storage problem. We circumvent the problem by imposing a first order Gauss Markov structure on $\mathcal{P}$. In [7], it is shown that a first order GMRF has a covariance matrix, of the form $(U^T U)^{-1}$ where U has the bidiagonal structure given below

$$U = \begin{bmatrix} U_1 & O_1 & \underline{0} & \underline{0} & . & . \\ \underline{0} & U_2 & O_2 & \underline{0} & . & . \\ . & \underline{0} & U_3 & O_3 & \underline{0} & . \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ . & . & . & . & \underline{0} & U_I \end{bmatrix} \tag{22}$$

By expressing $\mathcal{P}U = U^{-T}$, substituting from equation (22) for U and $\{P_{ij}\}$ for $\mathcal{P}$, and equating the lower triangular entries we get

$$P_{II} = (U_I^T U_I)^{-1} \tag{23}$$

$$P_{ij}U_j^T + P_{ij+1}O_j^T = (U_j)^{-1}\delta_{ij} \quad 1 \le i \le I, 1 \le j \le i \tag{24}$$

Equations (23–24) show that any P_{ij} can be evaluated recursively from O_is and U_is and vice versa. In our data assimilation model, instead of propagating P_{ij} we propagate U_i and O_i. This reduces significantly the storage requirements and the computational effort associated with the filter.

4. EXPERIMENT

To test the data assimilation scheme presented in Section (3), an equatorial channel version of the Pacific ocean was simulated. We considered a rectangular basin of length 1500

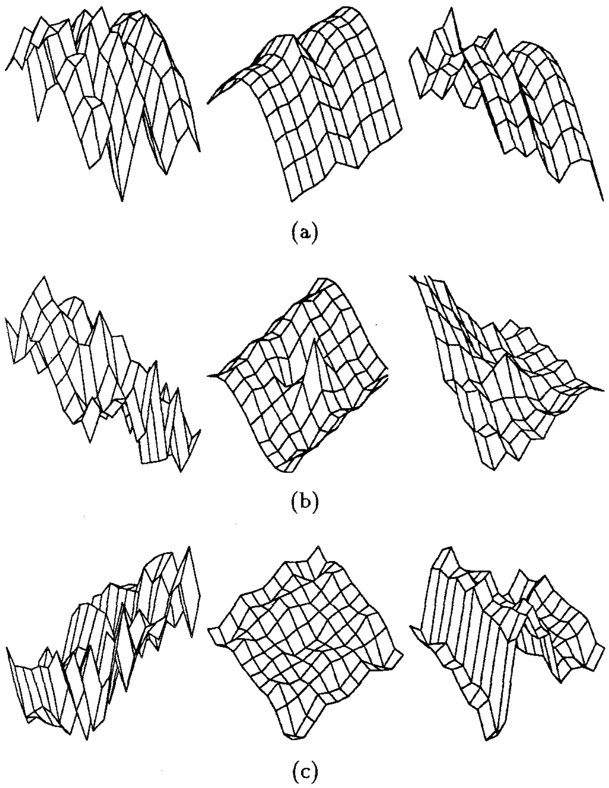

Fig. 1: Mesh plots of (a) latitudional component of velocity (b) longitudional component of velocity (c) sea surface height, shown at same scale. In (a) - (c), the left plot displays simulated actual field, the center displays mean OGCM field, and the right displays satellite data assimilated field using the algorithm described in the paper.

km in x-direction and 750 km in y-direction. The ocean coastline was omitted. The resolution in x-direction was chosen to be 150 km and in y-direction 75 km. The time step was 12 minutes, which satisfies the CFL conditions. To simulate real conditions, we started with an initial disturbance of height 20 cms at the center of the basin. The shape of the disturbance was chosen to be normal, i.e., $K\exp(-\frac{x^2}{2\sigma_x^2}-\frac{x^2}{2\sigma_y^2})$ with σ_x and σ_y selected to be 400 km and 100 km respectively. The horizontal wind stress was assumed to be sinusoidal, i.e., $X = X_m \sin\frac{\Pi y}{b}$ with X_m set to 0.2 N$/m^2$ and b set to 750 km. The evaporation rate E was made zero. We ran two OGCMs based on equations (1-4) for 10 days to simulate fields for the velocity components u and v, and displacement η. In the first setup, we introduced 20% randomness in X. The second had X as the wind stress. The first setup was assumed to simulate real oceanographic conditions, while the second constructed the mean or the deterministic components for u, v, and η.

To construct the random components for u, v, and η, the data assimilation model, described in Section 3, was run on the above data. The initial conditions were set to $\widehat{\mathcal{X}}_i(0\,|\,0) = \underline{0}$ and $P_{ij} = I_J$ for $1 \le i, j \le I$. It was assumed that the satellite scans the region every 4 hours and provides observations for u, v and η. During each scan, the satellite scans one complete row. 10% randomness was added to the observations to simulate noise.

In Fig. (1), we provide mesh plots for the three fields u (Fig. 1(a)), v (Fig. 1(b)), and η (Fig. 1(c)) 410 hours after the initialization, constructed using the three techniques described above, i.e., actual fields simulated under the aforementioned conditions, mean fields constructed from OGCM, and data assimilated fields constructed using the proposed algorithm. It is apparent that the data assimilated mean fields provide us with better estimates of the actual conditions than the mean field delivered by OGCM.

5. SUMMARY

The paper proposes a framework for the assimilation of satellite data in ocean circulation models. The framework is obtained by expressing the Kalman–Bucy filtering in a sub–block structure. This and the approximation of the error covariance as a first order Gauss Markov random field led to fast recursive algorithms capable of estimating the velocity fields and the sea surface height. We showed that fields obtained from assimilating satellite data into the the mean fields, using the recursive algorithm outlined above, provides us with better understanding of the ocean circulation and yields efficient estimates.

6. ACKNOWLEDGMENTS

The authors would like to acknowledge the interactions with Prof. Arthur Mariano from the Division of Meteorology and Physical Oceanography, Rosenstiel School of Marine and Atmospheric Science, University of Miami.

7. REFERENCES

[1] A. F. Bennett, *Inverse Methods in Physical Oceanography.* Cambridge University Press, 1992.

[2] M. Ghil, "Meteorological Data Assimilation for Oceanographers. part 1: description and theoretical framework," in *Dynamics of Atmospheres and Oceans*, 1989, pp. 13(3–4):171–218.

[3] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82D, pp. 35–45, March 1960.

[4] G. Neuman and J. W. J. Pierson, *Principles of Physical Oceanography.* Englewood Cliffs, NJ: Prentice Hall, Inc., 1966.

[5] A. E. Gill, *Atmosphere-Ocean Dynamics*, Vol. 30 of *International Geophysics Series.* New York: Academic Press, Inc., 1982.

[6] R. Courant, K. O. Friedrichs, and H. Lewy, "Uber die partiellen differenzengleichungen der mathematischen physik," *Mathematische Annalen*, pp. 32–74, 1928.

[7] J. M. F. Moura and N. Balram, "Recursive structure of noncausal Gauss Markov random fields," in *IEEE Transactions on Information threoy.*, March 1992, pp. IT–38(2):334–354.