BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică "Gheorghe Asachi" din Iaşi
Tomul LVII (LXI), Fasc. 1, 2011
Secţia
MATEMATICĂ. MECANICĂ TEORETICĂ. FIZICĂ

Numerical procedures for data assimilation in air pollution models – An overview

BY

Gabriel Dimitriu and Răzvan Ştefănescu

Abstract. We present an overview on several procedures for data assimilation applied to models in atmospheric chemistry, oceanography and air pollution. The data assimilation schemes are applied to models defined by ordinary and partial differential equations. Emphasis is laid on the performance evaluation of an adaptive proper orthogonal decomposition (POD) procedure – with respect to the solution obtained with the classical 4D-Var (full model), and POD 4D-Var data assimilation, as well as that of two types of low-rank Kalman filters. Numerical simulations are also included.

Key words: keyword1, keyword2,...

2010 MSC: 00J00, 00J00

1. Introduction

Data assimilation represents a methodology to combine the results of a large-scale numerical model with the measurements available to obtain an optimal reconstruction of the state of the system. The basic components of forward data assimilation were introduced operationally already 50 years ago, together with barotropic forecast models by Gilchrist and Cressman ([18]), and Bergthorsson and Döös ([3]).

In this paper we present an overview on several procedures for data assimilation applied to models in atmospheric chemistry, oceanography and air pollution. There are analyzed data assimilation schemes applied to models defined by ordinary and partial differential equations. Emphasis is laid on the performance evaluation of an adaptive proper orthogonal decomposition (POD) procedure – with respect to the solution obtained with the classical 4D-Var (full model), and POD 4D-Var data assimilation – as well as that of two types of low-rank Kalman filters.

The outline of this paper is as follows. Section 2 presents three sequential data assimilation schemes (successive corrections, optimal interpolation and analysis correction) applied to a model in atmospheric chemistry and oceanography defined mathematically by a system of ordinary differential equations. Section 3 is dedicated to a model given by the one-dimensional diffusion-advection equation. It is studied the 4D-Var assimilation procedure, together with the POD 4D-Var method and its adaptive variant. Section 4 describes some Kalman filters in factorized form, applied to data assimilation problems using a two-dimensional advection-diffusion model. The performance of the algorithms is illustrated by numerical tests. The paper ends with some conclusions.

2. Sequential data assimilation schemes applied to a model in atmospheric chemistry and oceanography

The model developed by Walker ([30]) simulates the interaction of the various forms of carbon that are stored in three regimes: the atmosphere, the shallow ocean, and the deep ocean. The five principal variables of the model $(p, \sigma_s, \sigma_d, \alpha_s, \alpha_d)$ are all functions of time: p represents the partial pressure of carbon dioxide in the atmosphere, σ_s is the total dissolved carbon concentration in the shallow ocean, σ_d is the total dissolved carbon concentration in the deep ocean, α_s is the alkalinity in the shallow ocean, and α_d represents the alkalinity in the deep ocean.

Three additional quantities are involved in the equilibrium equations in the shallow ocean: h_s - hydrogen carbonate in the shallow ocean, c_s - carbonate in the shallow ocean, and p_s - partial pressure of gaseous carbon dioxide in the shallow ocean.

the atmosphere and the shallow ocean involves a constant characteristic transfer time d, and a source term f.

$$\frac{dp}{dt} = \frac{p_s - p}{d} + \frac{f}{\mu_1}\,. \tag{1}$$

The equations describing the exchange between the shallow and deep oceans involve ν_s and ν_d, the volumes of the two regimes.

$$\begin{aligned} \frac{d\sigma_s}{dt} &= \frac{1}{\nu_s}\left((\sigma_d - \sigma_s)w - k_1 - \frac{p_s - p}{d}\mu_2\right), \quad \frac{d\sigma_d}{dt} = \frac{1}{\nu_d}(k_1 - (\sigma_d - \sigma_s)w), \\ \frac{d\alpha_s}{dt} &= \frac{1}{\nu_s}((\alpha_d - \alpha_s)w - k_2), \quad \frac{d\alpha_d}{dt} = \frac{1}{\nu_d}(k_2 - (\alpha_d - \alpha_s)w). \end{aligned} \tag{2}$$

The equilibrium between carbon dioxide and the carbonates dissolved in the shallow ocean is described by three nonlinear algebraic equations:

$$h_s = \frac{\sigma_s - (\sigma_s^2 - k_3\alpha_s(2\sigma_s - \alpha_s))^{1/2}}{k_3}, \quad c_s = \frac{\alpha_s - h_s}{2}, \quad p_s = k_4\frac{h_s^2}{c_s}.$$

We used in the numerical simulations the following values for the constants involved in the model equations: $d = 8.64$, $\mu_1 = 4.95\text{E}{+}2$, $\mu_2 = 4.95\text{E-}2$, $\nu_s = 0.12$, $\nu_d = 1.23$, $w = 10^{-3}$, $k_1 = 2.19\text{E-}4$, $k_2 = 6.12\text{E-}5$, $k_3 = 0.997148$, and $k_4 = 6.79\text{E-}2$. The source term f describes the burning of fossil fuels in the modern industrial era. We will use a time interval that starts about a thousand years ago and extends a few thousand years into the future: $1000 \leq t \leq 5000$. The initial values at the moment of time $t = 1000$ are: $p = 1$, $\sigma_s = 2.01$, $\sigma_d = 2.23$, $\alpha_s = 2.20$, $\alpha_d = 2.26$. These values characterize preindustrial equilibrium and remain nearly constant as long as the source term $f \equiv 0$.

The values contained in Table 1 describe a scenario for a source term f that models the release of carbon dioxide from burning fossil fuels, especially gasoline. The amounts begin to be significant after 1850, peak near the end of this century, and then decrease until the supply is exhausted.

The numerical simulations were carried out with the interpolated values in Table 1 using piecewise cubic Hermite interpolation method. The two alkalinity variables of the model (α_s and α_d) are almost constant throughhout this entire simulation. Initially, the carbon in the three regimes is nearly at equilibrium and so the amounts hardly change before 1850.

In the following, we introduce three sequential data assimilation schemes: successive corrections, optimal interpolation and analysis correction. The general sequential data assimilation problem can be considered as the minimization of the cost function

$$J = [y(i) - H(x(i))]^{-1}R^{-1}[y(i) - Hx(i))] + [x(i) - x_b(i)]^{-1}B^{-1}[x(i) - x_b(i)], \tag{3}$$

with respect to $x(i)$. Here $x_b(i)$ is a background state with error covariance matrix B^{-1}, $y(i)$ is a vector of observations with covariance matrix R^{-1}, and

H is the (possibly nonlinear) observation operator which converts the model field into an equivalent model observation value. The solution to the data assimilation problem, which we call *the analysis* is denoted by x_a. Each of the implemented data assimilation methods approximates the solution to this minimization problem. Here we do not aim to give a full explanation of the schemes, but just briefly outline how each is implemented. Further details can be found in [24], [11].

Successive corrections scheme. The successive corrections method is an iterate algorithm, which can be written as

$$x^{j+1}(i) = x^j(i) + W[y(i) - H(x^j(i))], \tag{4}$$

where $x^0(i) = x_b(i)$ is the background state, j is the iteration or correction index, and W is a weighting matrix. The algorithm is stopped after k corrections, after which the analysis is given by $x_a(i) = x^k(i)$. For the experiments in these programs the weighting matrix is given by $W = 0.5 \times I$, where I is the identity matrix.

Optimal interpolation scheme. The optimal interpolation analysis is given by

$$x_a(i) = x_b(i) + K[y(i) - H(x_b(i))], \tag{5}$$

with $K = BH^T(HBH^T + R)^{-1}$. If H is a nonlinear operator then it should be linearized around a background state.

Analysis correction scheme. The analysis correction algorithm is written as

$$x^{j+1}(i) = x^j(i) + WQ[y^j(i) - H(x^j(i))], \; y^{j+1}(i) = y^j(i) - Q[y^j(i) - H(x^j(i))], \tag{6}$$

where $y^0(i)$ is the vector of observations, $W = BH^TR^{-1}$, and $Q = (HW+I)^{-1}$. If H is a nonlinear operator, then its linearization around a background state should be used. The algorithm is stopped after k corrections, after which the analysis is given by $x_a(i) = x^k(i)$. The source code of the program was performed using Matlab language, version 6.1. The data assimilation results for the components p and σ_s are illustrated in Figures 1–6.

3. The four-dimensional variational data assimilation using POD procedures

The four-dimensional variational data assimilation (4D-Var) method has been a very successful technique used in operational numerical weather prediction at many weather forecast centers ([17], [27], [28], [31], [32]).

Proper orthogonal decomposition technique has been used to obtain low dimensional dynamical models of many applications in engineering and science

([1], [5], [6], [12], [16], [23]). Basically, the idea starts with an ensemple of data, called *snapshots*, collected from an experiment or a numerical procedure of a physical system. The POD technique is then used to produce a set of basis functions which spans the snapshots collection. The goal of the approach is to represent the ensemble of data in terms of an optimal coordinate system. That is, the snapshots can be generated by a smallest possible set of basis functions.

The drawback of the POD 4D-Var consists of the fact that the optimal solution can only be sought within the space spanned by the POD basis of background fields. When observations lay outside of the POD space, the POD 4D-Var solution may fail to fit observations sufficiently. The above limitation of the POD 4D-Var can be improved by implementing an adaptive POD 4D-Var scheme. In this section an adaptive proper orthogonal decomposition (A-POD) procedure is applied to set up a reduced-order control space for the one-dimensional diffusion-advection equations.

The model under study is a one-dimensional diffusion-advection equation defined by the following partial differential equation:

$$\text{(7)}\quad \frac{\partial \mathbf{c}}{\partial t} = a\frac{\partial^2 \mathbf{c}}{\partial x^2} + b\frac{\partial \mathbf{c}}{\partial x} + f(x,t)\,, \quad (x,t)\in\Omega\times(0,T), \quad \mathbf{c}(x,0)=x(1-x)\,.$$

Here, the spatial domain $\Omega = [0,1]$, and the coefficients a and b are positive constants. The function f is chosen to be $f(x,t) = (-x^2+(1+2b)x+2a-b)e^t$. Then, the exact (analytical) solution of (7) is given by $\mathbf{c}_{exact}(x,t) = x(1-x)e^t$. Details about the numerical implementation of a data assimilation algorithm for a similar model of (7) is presented in [32]. For a successful implementation of the POD 4D-Var in data assimilation problems, it is of most importance to construct an accurate POD reduced model. In what follows, we briefly present a description of this procedure (see [5], [6], [23]).

For a temporal-spatial flow $\mathbf{c}(x,t)$, we denoted by $\mathbf{c}_1,\ldots,\mathbf{c}_n$ a set adequately chosen in a time interval $[0,T]$, that is $\mathbf{c}_i = \mathbf{c}(x,t_i)$. Defining the mean $\overline{\mathbf{c}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{c}_i$, we expand $\mathbf{c}(x,t)$ as

$$\text{(8)}\qquad \mathbf{c}^{\mathrm{POD}}(x,t) = \overline{\mathbf{c}}(x) + \sum_{i=1}^{M}\beta_i(t)\Phi_i(x)\,,$$

where $\Phi_i(x)$ – the ith element of POD basis –, and M are appropriately chosen to capture the dynamics of the flow as follows:

(1) Calculate the mean $\overline{\mathbf{c}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{c}_i$;
(2) Set up the correlation matrix $K = [k_{ij}]$, where $k_{ij} = \int_\Omega(\mathbf{c}_i-\overline{\mathbf{c}})(\mathbf{c}_j-\overline{\mathbf{c}})\,dx$;
(3) Compute the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ and the corresponding orthogonal eigenvectors $v_1, v_2, \ldots, v_n$ of K;
(4) Set $\Phi_i := \sum_{j=1}^{n} v_j^i(\mathbf{c}_i - \overline{\mathbf{c}})$.

Now, we introduce a relative information content to select a low-dimensional basis of size $M \ll n$, by neglecting modes corresponding to the small eigenvalues. Thus, we define the index

$$I(k) \quad = \quad \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

and choose M, such that $M = \text{argmin}\{I(m) : I(m) \geq \gamma\}$, where $0 \leq \gamma \leq 1$ is the percentage of total information captured by the reduced space $X^M = \text{span}\{\Phi_1, \Phi_2, \ldots, \Phi_M\}$. The tolerance parameter γ must be chosen to be near the unity in order to capture most of the energy of the snapshot basis. The reduced order model is then obtained by expanding the solution as in (8).

Generally, an atmospheric or oceanic model is usually governed by the following dynamic system

$$\frac{d\mathbf{c}}{dt} = F(\mathbf{c}, t)\,, \qquad \mathbf{c}(x, 0) = \mathbf{c}^0(x)\,. \tag{9}$$

To obtain a reduced model of (9), we first solve (9) for a set of snapshots and follow above procedures, then use a Galerkin projection of the model equations onto the space X^M spanned by the POD basis elements (replacing $\mathbf{c}$ in (9) by the expansion (8), then multiplying Φ_i and integrating over spatial domain Ω):

(10)

$$\frac{d\beta_i}{dt} =< F\left(\overline{\mathbf{c}} + \sum_{i=1}^{M} \beta_i \Phi_i,\, t\right), \Phi_i >, \quad \beta_i(0) =< \mathbf{c}(x, 0) - \overline{\mathbf{c}}(x),\, \Phi_i(x) > .$$

Equation (10) defines a reduced model of (9). In the following, we will analyze applying this model reduction to 4D-Var formulation. In this context, the forward model and the adjoint model for computing the cost function and its gradient represent the reduced model and its corresponding adjoint, respectively.

At the assimilation time interval $[0, T]$, a prior estimate or ‘background estimate’, $\mathbf{c}_b$ of the initial state $\mathbf{c}_0$ is assumed to be known and the initial random errors $(\mathbf{c}_0 - \mathbf{c}_b)$ are assumed to be Gaussian with covariance matrix $\mathbf{B}$.

The aim of the data asimilation is to minimize the square error between the model predictions and the observed system states, weighted by the inverse of the covariance matrices, over the assimilation interval. The initial state $\mathbf{c}_0$ is treated as the required control variable in the optimization process. Thus, the objective function associated with the data assimilation for (9) is expressed by

$$J(\mathbf{c}_0) = (\mathbf{c}_0 - \mathbf{c}_b)^T \mathbf{B}^{-1} (\mathbf{c}_0 - \mathbf{c}_b) + (\mathbf{H}\mathbf{c} - \mathbf{y}^o)^T \mathbf{R}^{-1} (\mathbf{H}\mathbf{c} - \mathbf{y}^o)\,. \tag{11}$$

Here, $\mathbf{H}$ is an observation operator, and $\mathbf{R}$ is the observation error covariance matrix.

In POD 4D-Var, we look for an optimal solution of (11) to minimize the cost function $J(\mathbf{c}_0^M) = J(\beta_1(0), \ldots, \beta_M(0))$ given by

$$(12)\quad J(\mathbf{c}_0^M) = (\mathbf{c}_0^{\mathrm{POD}} - \mathbf{c}_b)\mathbf{B}^{-1}(\mathbf{c}_0^{\mathrm{POD}} - \mathbf{c}_b) + (\mathbf{H}\mathbf{c}^{\mathrm{POD}} - \mathbf{y}^{\mathrm{o}})\mathbf{R}^{-1}(\mathbf{H}\mathbf{c}^{\mathrm{POD}} - \mathbf{y}^{\mathrm{o}})\,,$$

where $\mathbf{c}_0^{\mathrm{POD}}$ is the control vector.

In (12), $\mathbf{c}_0^{\mathrm{POD}}(x) = \mathbf{c}_0^{\mathrm{POD}}(x,0)$ and $\mathbf{c}^{\mathrm{POD}}(x) = \mathbf{c}^{\mathrm{POD}}(x,t)$ are expressed by

$$\mathbf{c}_0^{\mathrm{POD}}(x) = \bar{\mathbf{c}}(x) + \sum_{i=1}^{M} \beta_i(0)\Phi_i(x), \qquad \mathbf{c}^{\mathrm{POD}}(x) = \bar{\mathbf{c}}(x) + \sum_{i=1}^{M} \beta_i(t)\Phi_i(x)\,.$$

Therefore, in POD 4D-Var the control variables are $\beta_1(0), \ldots, \beta_M(0)$. As explained later, the dimension of the POD reduced space could be much smaller than that the original space. As a consequence, the forward model is the reduced model (10) which can be very efficiently solved. The adjoint model of (10) is then used to calculate the gradient of the cost function (12) and that will significantly reduce both the computational cost and the programming effort.

The POD model in POD 4D-Var assimilation is established by construction of a set of snapshots, which is taken from the background trajectory, or integrate original model (9) with background initial conditions. The A-POD algorithm used in our numerical experiments is presented below:

A-POD Algorithm:

Step 1: Set $k = 1$, the iteration level for POD procedure, and the initial guess controls $\mathbf{c}_0^k$.

Step 2: Set up the snapshots ensemble from the solution of the full forward model, with the controls $\mathbf{c}_0^k$.

Step 3: Compute the POD bases (the number of POD bases is chosen to capture a prescribed energy level, γ, mentioned above).

Step 4: Project the controls $\mathbf{c}_0^k$ on the reduced space $\beta_{k,iter}$ $(iter = 1)$.

Step 5: Optimize the initial controls $\beta_{k,iter}$ (here, *iter* denotes the iteration of the optimization process, completely carried out on the reduced space).

Step 6: (i) Check the value of the cost function (11). If $|J_{iter}| < \varepsilon$ (where ε is the tolerance for the optimization), then STOP;

(ii) If $|J_{iter}| > \varepsilon$ and $|J_{iter} - J_{iter-1}| > 10^{-3}$, then set $iter = iter + 1$ and go to Step 5;

(iii) If $|J_{iter}| > \varepsilon$ and $|J_{iter} - J_{iter-1}| < 10^{-3}$, then update the POD bases:

(a) Find the new controls $\mathbf{c}_0^{k+1}$ by projecting the optimization controls $\beta_{k,iter}$ onto the original domain.

(b) Set $k = k + 1$ and go to Step 2.

In the following we present some numerical results of several data assimilation experiments carried out to examine the performance of A-POD 4D-Var procedure, by comparing it with the full 4D-Var and POD 4D-Var data assimilation. All the performed tests have used as 'true' (exact) solution $\mathbf{c}_0$ of the assimilation problem, that one computed from the analytical solution $\mathbf{c}_{exact}$ of (7).

We set in our approach $T = 1$, and used 31 discretization points in space, and 69 points in time interval [0, 1]. By means of a perturbed initial condition we generated the observed state $\mathbf{y}^o$. We assumed that the correlation matrices in (11) and (12) are diagonal matrices, given by $\mathbf{B} = \sigma_b^2 I$ and $\mathbf{R} = \sigma_o^2 I$, with I denoting the identity matrix of appropriate order. We set $\sigma_b^2 = 0.05^2$ and $\sigma_o^2 = 0.1^2$, representing the variances for the background and observational errors, respectively.

The numerical tests have been performed with the following values of the diffusion and advection parameters: $a = 0.0001$ and $b = 1$. The background value $\mathbf{c}_b$ and the observation $\mathbf{y}^o$ were obtained by perturbing the exact solution of the state equation.

Figure 7 illustrates the decay of the eigenvalues, in case when the spatial correlation matrix, K, was calculated using 12, 18, 23 and 35 snapshots, respectively. The dimension of the POD reduced model depends on the number of basis functions. We found that only few basis functions are required to capture a high percentage from the dynamic of the system. For the presented cases, we can remark in Table 2 that more than 90% from the full variability characterizing the model dynamics can be captured with 4, 16 and 18 POD modes, respectively. Thus, choosing $M = 16$ modes and $n = 36$ snapshots, the captured energy represents 98.53% of the total energy, while when $M = 18$, the captured energy percentage is 99.16%.

We performed several numerical tests by implementing the three distinct procedures for data assimilation problem (7), (11): the standard (full) 4D-var, the POD 4D-Var, and the adaptive POD (A-POD) scheme ([12], [13], [14]). The numerical solution of the optimal control problem was obtained using *fminunc* – the Matlab unconstrained minimization routine. Its algorithm is based on the BFGS quasi-Newton method with a mixed quadratic and cubic line search procedure.

The comparative assimilation results associated with our experiments for certain selections of M and n are depicted in Figures 8, and 9. Obviously, one can conclude that better results are obtained when one is using more POD modes and more snapshots.

We also remark that the variance of the observational errors chosen twice bigger than the variance of the background errors caused a relative high instability for the assimilation solution (even for the best case, where the reduced model was constructed with $n = 36$ snapshots and $M = 22$ modes, see Figure 9, bottom plot). A certain contribution to this oscillatory behaviour of the solution could also be attributed to the fact the advection strongly dominates diffusion in the model (7).

4. Data assimilation using Kalman filters

Kalman filtering represents a powerful framework for solving data assimilation problems ([17], [19], [20], [21], [22]). For the implementation of a Kalman filter the evolution of the state and observation of measurements can be described with the stochastic system:

$$\mathbf{x}^t[k+1] = \mathbf{A}[k]\mathbf{x}^t[k] + \boldsymbol{\eta}[k]\,, \qquad \mathbf{y}^o[k] = \mathbf{H}[k]'\mathbf{x}^t[k] + \boldsymbol{\nu}[k] \tag{13}$$

with $\mathbf{x}^t[k] \in \mathbb{R}^n$ the true state vector at time $t[k]$, $\mathbf{A}[k]$ a deterministic model, $\boldsymbol{\eta}[k] \in \mathbb{R}^n$ a Gaussian distributed model error (zero mean, covariance $\mathbf{Q}$), and $\mathbf{y}^o[k] \in \mathbb{R}^r$ a vector of observations with $\boldsymbol{\nu}[k]$ the representation error (Gaussian with zero mean and covariance $\mathbf{R}$). The superscripts t, o, and later on f and a refer to the true, observed, forecasted and analyzed entities, respectively. We also mention that the time indices for $\mathbf{A}$ and $\mathbf{H}'$ will be omitted in coming equations, assuming that the time is implied by the state where the operators act on.

The aim of the filter operations is to obtain the mean $\hat{\mathbf{x}}^a$ and covariance $\mathbf{P}^a$ for the probability density of the true state. The filter equations for this system contain the *forecast* stage given by:

$$\hat{\mathbf{x}}^f[k+1] = \mathbf{A}[k]\hat{\mathbf{x}}^a[k]\,, \qquad \mathbf{P}^f[k+1] = \mathbf{A}\mathbf{P}^a[k]\mathbf{A} + \mathbf{Q}[k] \tag{14}$$

and *analysis* stage expressed by

$$\mathbf{x}^a = \hat{\mathbf{x}}^f + \mathbf{K}(\mathbf{y}^o - \mathbf{H}'\hat{\mathbf{x}}^f) \tag{15}$$

$$\mathbf{P}^a = \begin{cases} (\mathbf{I} - \mathbf{K}^{MV}\mathbf{H}')\mathbf{P}^f\,, & \mathbf{K}^{MV} = \mathbf{P}^f\mathbf{H}'(\mathbf{H}'\mathbf{P}^f\mathbf{H} + \mathbf{R})^{-1} \\ (\mathbf{I} - \mathbf{K}\mathbf{H}')\mathbf{P}^f(\mathbf{I} - \mathbf{K}\mathbf{H}')' + \mathbf{K}\mathbf{R}\mathbf{K}'\,, & \text{arbitrary gain } \mathbf{K}\,. \end{cases} \tag{16}$$

In the case of a large model, the propagation of the covariance matrix in (14) represents the most expensive part in the full rank filter. If $\mathbf{A}$ is defined by an $n \times n$ matrix, then the dynamical model is called $2n$ times to perform the operation $\mathbf{A}(\mathbf{A}\mathbf{P})'$. Limiting both the number of model evaluations as well as the storage requirements will be achieved in this study by reducing the rank of the covariance matrix.

Bierman ([4]) proposed to write the equations for the Kalman filter using the factorization $\mathbf{P} = \mathbf{S}\mathbf{S}'$. Numerical inaccuracies made in computation and storage of the matrix $\mathbf{S}$ will never affect the property of positive definiteness of $\mathbf{P}$. Inaccuracies will even be reduced since the condition number of $\mathbf{S}$ is only the square root of the condition number of $\mathbf{P}$.

The idea of factorization is useful to reduce the storage requirements of $\mathbf{P}$. Consider a covariance matrix $\mathbf{P}$ written as the product of a rectangular matrix square root $\mathbf{S}$ and its transpose:

$$\mathbf{P}_{n\times n} \quad = \quad \mathbf{S}_{n\times m}\mathbf{S}'_{m\times n}\,.$$

In order to obtain the Kalman filter in square root form, apart from the previous factorization $\mathbf{P} = \mathbf{S}\mathbf{S}'$ for the covariance of the true state, we also introduce the factorizations $\mathbf{Q} = \mathbf{T}\mathbf{T}'$ and $\mathbf{R} = \mathbf{U}\mathbf{U}'$ for the covariance of the forecast and representation error, respectively. Further, a matrix $\mathbf{\Psi}' = \mathbf{H}'\mathbf{S}$ is introduced for the mapping of the forecast covariance root to the observation space. After (14), the forecast of mean and covariance become:

$$\begin{aligned}
(17) \qquad \hat{\mathbf{x}}^f[k+1] \quad &= \quad \mathbf{A}\hat{\mathbf{x}}^a[k] \\
(\mathbf{S}^f\mathbf{S}^{f'})[k+1] \quad &= \quad \mathbf{A}(\mathbf{S}^a\mathbf{S}^{a'})[k]\mathbf{A} + \mathbf{T}\mathbf{T}'[k] \\
(18) \qquad \text{or} \qquad (\mathbf{S}^f[k+1] \quad &= \quad [\mathbf{A}(\mathbf{S}^a[k]\,, \mathbf{T}[k]\,]
\end{aligned}$$

The second formula in (18) is able to reduce both the computational complexity and the numerical inaccuracies, since the condition number of $\mathbf{S}^f$ or $\mathbf{S}^a$ is only the square root of the condition number of $\mathbf{P}^f$ and $\mathbf{P}^a$, respectively. The introduction of a forecast error leads to extension of the square root with the columns of $\mathbf{T}$. Each new column introduces a new direction for the uncertainty of the state vector. To preserve the number of modes from growing to infinity, filter algorithms based on factorizations include approximations or mechanism to avoid the growth, for example avoiding the use of dynamic noise completely, projection of $\mathbf{T}$ on the base spanned by $\mathbf{AS}$ or reduction of the number of columns whenever necessary. If $\mathbf{T}$ is to be added to the covariance square root, the degree of freedom in the system noise (rank of $\mathbf{T}$) should be of order $10-100$ to keep storage and propagation of the covariance square root feasible.

In the following we introduce two type of factorized Kalman filters.

RRSQRT filter. In the *Reduced Rank SQuare RooT* (RRSQRT) formulation of the Kalman filter, the covariance matrix is expressed in a limited number of (orthogonal) modes, which are re-orthogonalized and truncated to a fixed number during each time step. The basic formulation is a direct translation of the linear Kalman filter into square root formulation, leading to:

$$\begin{aligned}
(19) \qquad \hat{\mathbf{x}}^f[k+1] \quad &= \quad \mathbf{A}\hat{\mathbf{x}}^a[k]\,, \qquad S^f[k+1] = [AS^a[k], T[k]] \\
(20) \qquad \mathbf{\Psi} \quad &= \quad \mathbf{H}'\mathbf{S}^f[k+1]\,, \\
(21) \qquad \mathbf{K} \quad &= \quad \mathbf{S}^f[k+1]\mathbf{\Psi}[\mathbf{\Psi}'\mathbf{\Psi} + \mathbf{R}[k+1]]^{-1}\,, \\
(22) \qquad \hat{\mathbf{x}}^a[k+1] \quad &= \quad \hat{\mathbf{x}}^f + \mathbf{K}(\mathbf{y}^o[k+1] - \mathbf{H}'\hat{\mathbf{x}}^f[k+1])\,, \\
(23) \qquad \mathbf{S}^a[k+1] \quad &= \quad \mathbf{S}^f[k+1][\mathbf{I} - \mathbf{\Psi}(\mathbf{\Psi}'\mathbf{\Psi} + \mathbf{R}[k+1])^{-1}\mathbf{\Psi}']^{\frac{1}{2}}\,, \\
(24) \qquad \mathbf{V}\mathbf{\Lambda}\mathbf{V}' \quad &= \quad \mathbf{S}^a[k+1]'\mathbf{S}^a[k+1]\,, \qquad \tilde{\mathbf{S}}^a[k+1] = \mathbf{S}^a[k+1]\tilde{\mathbf{V}}\,.
\end{aligned}$$

Generally, the algorithm is initialized with an empty covariance square root; new columns are added every time step due to the introduction of system noise **T** in (19). For each of the m modes stored in **S**, the forecast of the covariance requires one evaluation of the model **A**. The analysis steps (21)-(23) are usually implemented in the form of a sequential update for scalar measurements. An important part of the RRSQRT algorithm is the reduction of the covariance square root (24). With the introduction of system noise in (19), the number of modes has grown from m to $m + q$, where q is the number of columns in **T** (rank of **Q**). The reduction step reduces the size to m again. Matrix $\tilde{\mathbf{V}}$ contains the eigenvectors of $(\mathbf{S}^a)'\mathbf{S}^a$ corresponding with the largest m eigenvalues. The new matrix $\mathbf{S}^a\tilde{\mathbf{V}}$ represents an approximation of **S**, maintaining the largest singular vectors. In term of computational costs, the most expensive part of the RRSQRT filter is formed by the propagation of the modes in (19), when for each mode the model should be called once. The reduction should therefore reduce the number of modes as far as possible.

Ensemble filter. In comparison with the RRSQRT filter which is based on the factorization of the covariance matrix, the ENsemble Kalman Filter (ENKF) is based on convergence of large numbers. Both approaches lead to a low-rank approximation of the covariance matrix. The ensemble filter was introduced by Evensen ([15]) for assimilation of data in oceanographic models.

The essential idea behind the ensemble filter is to express the probability function of the state in an ensemble of possible states $\{\xi_1, \ldots, \xi_N\}$. Each ensemble member is assumed to be a single sample out of the distribution of the true state. Whenever necessary, statistical moments are approximated with sample statistics:

$$(25)\qquad \hat{\mathbf{x}} \approx \frac{1}{m}\sum_{j=1}^{m}\xi_j\,, \qquad\qquad \mathbf{P} \approx \frac{1}{m-1}\sum_{j=1}^{m}(\xi_j - \hat{\mathbf{x}})(\xi_j - \hat{\mathbf{x}})'\,, \quad \ldots$$

The sample statistics will always converge to the true values with increasing ensemble size. However, the convergence is rather slow (order $1/\sqrt{m}$), and this is the only serious disadvantage of the ensemble filter.

An important difference between the pair $(\hat{\mathbf{x}}, \mathbf{P})$ of the Kalman or factorized filter and the ensemble statistics (25) is that the later are much more connected with each other. In the traditional Kalman filters, $\hat{\mathbf{x}}$ and **P** are processed more or less independent of each other. The mean $\hat{\mathbf{x}}$ is analyzed using a gain matrix computed from **P**, but **P** is never affected by $\hat{\mathbf{x}}$; the covariance and gain could even be computed off-line.

It is possible to reformulate the ensemble in terms of a (sample) covariance square root:

$$\mathbf{P} = \sum_{k=1}^{m} \mathbf{e}_k \mathbf{e}_k' = \mathbf{E}\mathbf{E}' \qquad , \qquad \mathbf{e}_k = \frac{\boldsymbol{\xi}_k - \overline{\boldsymbol{\xi}}}{\sqrt{m-1}} \,. \tag{26}$$

Each ensemble member defines a rank one covariance matrix $\mathbf{e}_k\mathbf{e}_k'$. We notice that at least two ensemble members are required to provide a sample mean and sample covariance. This is not different for the filters based on factorizations which require at least two states for the mean and covariance too: the mean itself and one mode for a rank-one covariance matrix.

The filter equations for the ensemble filter are different from the previous described factorized filters in operating on an ensemble of states instead of a mean and covariance factor. Given an initial ensemble of states describing a range of possible true states, a forecast of the statistics for the true state at a future time is simply obtained from propagated ensemble members. In case of a non-linear model, the propagation becomes:

$$\xi_k^f[k+1] = \mathbf{M}[\xi_k^a[k]) + \eta_k[k] \,, \qquad \eta_k[k] \sim \mathcal{N}(0, \mathbf{Q}[k]) \,, \tag{27}$$

where a sample of the system noise is obtained from a random generator. Whenever measurements are available, each of the ensemble members is analyzed with a linear gain:

$$\xi_j^a[k+1] = \xi_j^f[k+1] + \mathbf{K}(\mathbf{y}^o[k+1] + \nu_j - \mathbf{H}'\xi^f[k+1]) \,, \tag{28}$$

where $\nu_j \sim \mathcal{N}(0, \mathbf{R}[k+1])$. The vectors ν_j denote samples of the representation error, drawn from a random generator. With $\mathbf{P}^e$ and $\mathbf{R}^e$, the sample covariances of the vectors $\boldsymbol{\xi}_j$ and $\boldsymbol{\nu}_j$ respectively, this analysis scheme leads to an analyzed mean and covariance given by (a bar denotes an ensemble mean):

$$\begin{aligned} \hat{\mathbf{x}}^a &= \overline{\boldsymbol{\xi}_j^a} = \overline{\boldsymbol{\xi}_j^f} + \mathbf{K}(\mathbf{y}^o + \overline{\nu_j} - \mathbf{H}'\overline{\boldsymbol{\xi}_j^f}) \\ \mathbf{P}^{e,a} &= \overline{(\boldsymbol{\xi}_j^a - \overline{\boldsymbol{\xi}_j^a})(\boldsymbol{\xi}_j^a - \overline{\boldsymbol{\xi}_j^a})'} \\ &= [\mathbf{I} - \mathbf{K}\mathbf{H}]\mathbf{P}^{e,f}[\mathbf{I} - \mathbf{K}\ \mathbf{H}]' + \mathbf{K}^e\mathbf{R}^e\mathbf{K}' \\ &\quad + \mathcal{O}\left(\overline{(\nu_j - \overline{\nu_j})(\nu_j - \overline{\nu_j})} - \mathbf{R}\right) + \mathcal{O}\left(\overline{(\boldsymbol{\xi}_j^a - \overline{\boldsymbol{\xi}_j^a})(\nu_j - \overline{\nu_j})}\right) \,. \end{aligned}$$

The last two terms converge to zero with order $1/\sqrt{m}$. If these terms are omitted, the analysis scheme produces what is expected from (16) for analysis of covariance $\mathbf{P}^e$ with an arbitrary gain matrix $\mathbf{K}$. The ensemble analysis (28) is independent of the gain matrix used. Under the assumption that the probability densities of both state and measurements are close to Gaussian, a gain matrix for the ensemble filter might be formed using the ensemble covariance:

$$\mathbf{K}^e = \mathbf{P}^e\mathbf{H}\,[\mathbf{H}'\mathbf{P}^e\mathbf{H} + \mathbf{R}]^{-1} \,. \tag{29}$$

The performance of two types of low-rank filters (RRSQRT and ENK) was tested during a filter experiment with simulated data. We used some slightly modified Matlab routines carried out by Heemink et al ([20]).

In comparison with the study of Heemink et al ([20]), in which the locations of the measurement and emission points are spreaded on the whole domain (Case III below), here we also take into consideration other two particular cases. Thus, we will distinguish on the whole domain two straight lines of points: a horizontal line containing only emission points placed in the lower part of the domain and another horizontal line constructed only with measurement points situated in the upper part of the domain. The situation, where the distance between the two horizontal lines is small, is analyzed in Case I. The increase of this distance (the distance between measurements and emission points) is studied in Case II. The motivation of the introduction in our analysis of Case I and Case II comes from real-life applications, when not the whole area of the space domain is qualified as accessible for measurement points.

As a model under investigation, we consider the 2-D advection-diffusion equation:

$$\frac{\partial \mathbf{c}}{\partial t} = \nu \left(\frac{\partial^2 \mathbf{c}}{\partial x^2} + \frac{\partial^2 \mathbf{c}}{\partial y^2} \right) - u \frac{\partial \mathbf{c}}{\partial x} - v \frac{\partial \mathbf{c}}{\partial y} \, , \tag{30}$$

with a square domain and zero initial conditions. The concentration at the boundary is zero for inflow. We set the diffusion coeficient $\nu = 0.2$ and define the wind velocities u and v as follows:

$$u = -vel.scale * (y_{grid} - yc_{vortex}) \, , \qquad v = vel.scale * (x_{grid} - xc_{vortex}) \, .$$

Here, *vel.scale* denotes the velocity scale computed at the value 0.06 ($2 * 0.9$ divided by 30, the maximum number of grid points in one direction), (x_{grid}, y_{grid}) denotes the current grid point, and by $(xc_{vortex}, yc_{vortex})$ we specify the center of the vortex.

We used a backward Lagrangian scheme to discretize these equations on a 30×30 grid. The velocity field is considered known and constant in time with a vortex located in the middle of the domain.

In Case I and Case II a reference solution was generated by inserting constant emissions at five pollution locations, equally spaced on a straight horizontal line at the grid cells $\{(5, 10), (10, 10), (15, 10), (20, 10), (25, 10)\}$. The increase of concentration per timestep for these location was set to $\{0.2, 0.1, 0.1, 0.2, 0.2\}$, respectively.

In the same manner, the reference solution for Case III was generated by inserting constant emissions at grid cells $\{(6, 6), (8, 10), (20, 9), (7, 19), (23, 20)\}$. The increase of concentration per timestep for these location was maintained at the same values as in Cases I and II.

We want to point out that the pollution locations within a certain analysed case (I, II or III) have different mean pollution rates, but identical values of these rates for the corresponding pollution points in all three cases. Thus, the third point (with the coordinates (15, 10) in Cases I and II, and (20, 9) in Case III) has the biggest mean pollution rate among all the five points. It follows, in this order, the second, the fifth, the first and finally, the fourth point. We can remark in Figures 10–17 that the pollution location with the biggest mean pollution rate is surrounded by many streamlines shifted to the right by the wind velocity.

In the Case I, the measurements were generated from simulated true concentrations, which were computed by adding fluctuations to the mean emissions, according to $\tilde{z}_j(k+1) = \gamma_j \tilde{z}_j(k) + z_j(k)$, with independent Gaussian white noise processes with $E\{z_j(k)\} = 0$ and $Var\{z_j(k)\} = 1$. The index j refers to measurement locations (3, 18), (6, 18), (9, 18), (12, 18), (15, 18), (18, 18), (21, 18), (24, 18), (27, 18), and γ_j is a decay per step corresponding to the measurement location. The 9 measurement locations are placed on a horizontal line at a distance (measured vertically) of 8 grid points from the line defined by the pollution locations.

The measurement locations in Case II are defined by the following coordinates of the grid cells: (3, 28), (6, 28), (9, 28), (12, 28), (15, 28), (18, 28), (21, 28), (24, 28), (27, 28). Therefore, the distance between the emission points and measurement locations is now greater than that in Case I (a distance of 18 grid points measured vertically).

In the last case under study (Case III) the measurement locations are given by the grid points: (3, 10), (12, 4), (27, 18), (14, 11), (22, 3), (10, 10), (14, 21), (22, 11), (6, 24). Finally, white observational noise with variance 0.1 was added to the true concentrations. To compare the performance of the different filters with each other, the root mean square (RMS) errors were computed:

$$RMS = \sqrt{\frac{1}{N_x^2 N_t} \sum_{m,n,k} (\mathbf{c}_{m,n}(k) - \hat{\mathbf{c}}_{m,n}(k))^2}\,, \tag{31}$$

where $\mathbf{c}_{m,n}(k)$ are the exact generated concentrations and $\hat{\mathbf{c}}_{m,n}(k)$ are the estimates computed, N_x is the number of gridpoints in one direction and N_t is the number of timesteps.

If the RMS errors of all experiments are compared (see Tables 3 and 4), the RRSQRT algorithm seems to have a robust behaviour for this particular application. The filter provides an accurate and constant result at a level of required model evaluations where the other algorithms still suffer from random fluctuations. Even for small numbers of modes, the results are more accurate than what could be achieved with an ENKF approach with comparable ensemble size. These results show that the convergence of the RRSQRT filter is much faster than the convergence of the ensemble filter. In Figures 10–17 the concentration fields of the truth-run and the reference-run are shown after $N_t = 100$ timesteps. The $+$ signs indicate measurement locations and the diamond-signs the locations of the emissions. It can be noticed clearly that the true fields are perturbed with time-varying fluctuations, while the reference solutions only contain a steady emission which is advected and spreading smoothly.

In Figures 10–13 we show the concentrations calculated using RRSQRT filter at the final time $k = 100$ for 30 ensemble members. We present the assimilation results both for 5 and 25 modes as well. We remark that RRSQRT filter for the settings $(q, N) = (5, 30)$ do not perform well, creating spurious streamlines inside the domain. Its RMS error is 1.447 for the Case I, 1.068 for Case II and 0.9295 for Case III (see Table 3). Increasing the number of modes to $q = 25$, we obtain better results, with RMS errors diminished at the values 0.482, 0.562 and 0.351 for the Cases I, II and III, respectively. An improved assimilation result can be seen in Figures 11 and 13. We notice that the RMS error obtained in Case I is smaller than the RMS error in Case II. This fact strongly motivates the decision that the positions of the measurement locations to be settled as close as possible (or in any case, not far) from the pollution locations.

The assimilation results of the concentrations calculated with ENK filter are contained in Figures 14–17. The ensemble filter suffers from statistical noise due to the use of a random number generator. Large values of RMS errors are obtained for small number of ensemble members (See Table 4). Other numerical results for this model, obtained with hybrid Kalman filters can be found in ([13]).

5. Conclusions

In this study we presented an overview on several procedures for data assimilation applied to models in atmospheric chemistry, oceanography and air pollution.

Three sequential data assimilation schemes (successive corrections, optimal interpolation and analysis correction), were applied to a model in atmospheric chemistry and oceanography that simulates the interaction of the various forms of carbon that are stored in three regimes: the atmosphere, the shallow ocean, and the deep ocean. The equations of the model are mildly stiff, because the various chemical reactions take place on very different time scales. Efficient numerical schemes applied to different stiff ODEs from atmospheric chemistry are described in [29] and [31]. Implementation aspects and numerical results from some preliminary investigations were presented.

We also applied to a simple diffusion-advection model, both a standard 4D-Var assimilation scheme and a reduced order approach to 4D-Var assimilation using an adaptive POD procedure. The numerical results from several POD models (constructed with different numbers of snapshots and POD modes) were compared with those of the original model. Our numerical tests showed that variability of the original model could be captured reasonably well by applying an adaptive POD scheme to a low dimensional system set up with 36 snapshots and 22 leading POD basis functions.

Table 1. Possible scenario for the source term f that models the release of carbon dioxide from burning fossil fuels, especially gasoline.

Year	1000	1850	1950	1980	2000	2050	2080	2100	2120	2150	2225	2300	2500	5000
Rate	0.0	0.0	1.0	4.0	5.0	8.0	10.0	10.5	10.0	8.0	3.5	2.0	0.0	0.0

Table 2. The values of the index $I(k)$ for different numbers of snapshots (n), and POD modes (M).

Number of snapshots (n)	Number of modes (M)	Index $I(M)$
12	4	91.74%
18	16	99.89%
36	16	98.53%
36	18	99.16%

TABLE 3. Numerical results using RRSQRT filter – RMS errors and standard deviation (STD) values for concentration and system noise.

Modes q	Ensemble N	RMS conc. (Case I)	RMS conc. (Case II)	RMS conc. (Case III)	STD conc. (Case I)	STD conc. (Case II)	STD conc. (Case III)
5	30	1.4479	1.0680	0.9295	0.1752	0.2080	0.1578
10	30	1.4941	0.6154	0.6582	0.2715	0.2875	0.2533
15	30	1.4412	0.6129	0.3970	0.3320	0.3587	0.3082
20	30	1.0725	0.6652	0.3500	0.3842	0.4171	0.3197
25	30	0.4826	0.5620	0.3519	0.4102	0.4599	0.3225
30	30	0.4356	0.5822	0.3522	0.4144	0.4785	0.3231
Modes q	Ensemble N	RMS noise (Case I)	RMS noise (Case II)	RMS noise (Case III)	STD noise (Case I)	STD noise (Case II)	STD noise (Case III)
5	30	6.8737	3.0845	4.9441	1.4422	1.2871	1.4431
10	30	4.7941	2.1529	2.9814	1.6663	1.4653	1.7231
15	30	3.1432	2.1278	2.1266	1.8626	1.6315	1.9407
20	30	3.3308	2.2157	2.0632	1.9631	1.8059	1.9663
25	30	2.0726	2.0803	2.0653	1.9981	1.9503	1.9706
30	30	2.0533	2.1084	2.0655	2.0022	2.0151	1.9715

There is a recent tendency in data assimilation to find the optimal location of the observations by using the adjoint sensitivity method ([25]), as well as to combine the advantage of the numerical splitting schemes ([2]), variational methods ([8], [9], [10], [25]), and Kalman filter techniques ([7], [19]-[22], [26]).

Finally, two low-rank filters have been implemented in a 2-D advection-diffusion model: based on factorization (RRSQRT filter), and ensemble statistics (ENKF). The methods were found to be suitable to assimilate data with stochastic varying emissions. The ensemble filter suffers from statistical noise due to the use of a random number generator; the results still show a large spread where an RRSQRT filter with comparable costs already converged.

The results also indicated that it is favorable for data assimilation that the observation points be located as close as possible to the emission points, since the RMS errors of all experiments showed a decrease of their values. In the near future we intend to study the behaviour of such filters when one takes into consideration a vortex (or several vortices) with time depending positions inside the assimilation domain, thus capturing much better features of the complex atmospheric turbulences.

TABLE 4. Numerical results using ENSEMBLE filter – RMS errors and standard deviation (STD) values for concentration and system noise.

Ensemble	RMS	RMS	RMS	STD	STD	STD
N	conc.	conc.	conc.	conc.	conc.	conc.
	(Case I)	(Case II)	(Case III)	(Case I)	(Case II)	(Case III)
5	6.0610	1.4746	4.2539	0.1742	0.2693	0.1146
10	1.4567	0.7936	0.7806	0.2002	0.3147	0.1441
15	0.7683	0.6937	0.6210	0.2585	0.3577	0.1877
20	0.7284	0.7651	0.5692	0.2813	0.3614	0.1958
25	1.0245	0.6061	0.4529	0.2940	0.3772	0.2102
30	0.9070	0.7215	0.4410	0.3079	0.4112	0.2405

Ensemble	RMS	RMS	RMS	STD	STD	STD
N	noise	noise	noise	noise	noise	noise
	(Case I)	(Case II)	(Case III)	(Case I)	(Case II)	(Case III)
5	31.598	5.0479	18.245	1.6084	1.7061	1.3832
10	7.2784	2.6368	6.0983	1.6648	1.7845	1.5585
15	3.4405	0.3822	2.8188	1.7767	1.9152	1.6857
20	2.7913	2.3777	2.6913	1.7554	1.8361	1.6657
25	3.6213	2.1128	2.2937	1.7718	1.8672	1.6922
30	3.0147	2.3221	2.2190	1.9056	1.9779	1.7992

Acknowledgement. *The paper was supported by the project ID 342/2008, CNCSIS, Romania.*

Received: March 23, 2011

*"Gr. T. Popa" University of Medicine and Pharmacy, Department of Mathematics and Informatics
700115 Iaşi
Romania
e-mail:* dimitriu.gabriel@gmail.com

**"Gr. T. Popa" University of Medicine and Pharmacy, Department of Mathematics and Informatics
700115 Iaşi
Romania
e-mail:* rastefanescu@yahoo.co.uk

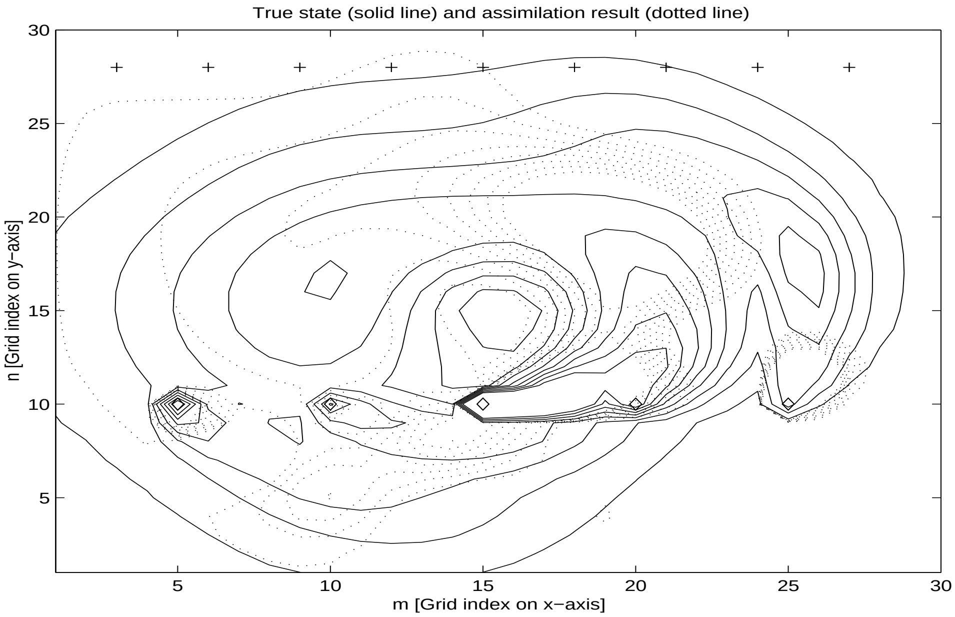

FIGURE 1. The concentrations calculated using ENK filter with $(q, N) = (5, 30)$ at time step $k = 100$ (CASE II).

References

1. A. Antoulas, S. Sorensen, K.A. Gallivan, *Model Reduction of Large-Scale Dynamical Systems*, Lecture Notes in Computer Science, **3038**, 740–747, **(2004)**.
2. J. Bartholy, I. Faragó, Á. Á. Havasi, *Splitting Method and its Application in Air Pollution Modeling*, Idöjárás **105** 39–58, **(2001)**.
3. P. Bergthorsson, B. Döös, *Numerical weather map analysis*, Tellus **7** 329–340, **(1955)**.
4. G.J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, Vol. **128** of Mathematical in Science and Engineering, Academic Press, New York, **(1977)**.
5. Y. Cao, J. Zhu, Z. Luo, I.M. Navon, *Reduced Order Modeling of the Upper Tropical Pacific Ocean Model Using Proper Orthogonal Decomposition*, Accepted for publication in Computer & Mathematics with Applications, **(2006)**.
6. Y.H. Cao, J. Zhu, I.M Navon, *A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition*, Int. J. Numer. Meth. Fluids, **53 (10)**, 1571–1583, **(2007)**.

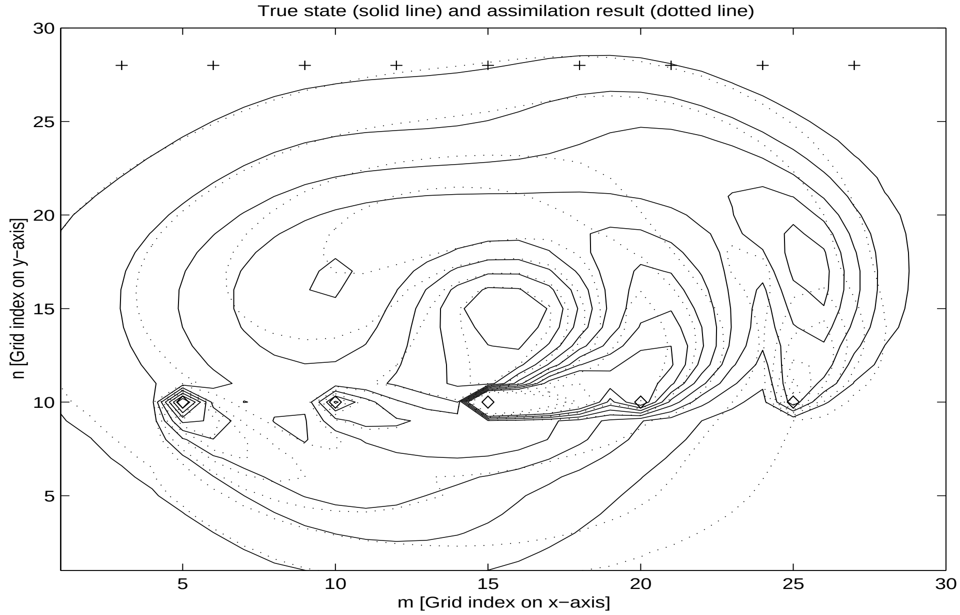

FIGURE 2. The concentrations calculated using ENK filter with $(q, N) = (25, 30)$ at time step $k = 100$ (CASE II).

7. C.K. Chui, G. Chen, *Kalman Filtering with real time applications*, Vol. 17 of Springer Series in Information Sciences, Springer Verlag, Berlin, **(1987)**.
8. Daescu, D., Carmichael G.R., Sandu, A.: Adjoint Implementation of Rosenbrock Methods Applied to Variational Data Assimilation Problems. *J. Comput. Phys.* **165** (2) (2000), 496–510.
9. Dimitriu, G.: *Parameter identification for some classes of nonlinear systems*. PhD thesis (1999) (in Romanian).
10. Dimitriu, G., Cuciureanu, R. (2005): Mathematical aspects of data assimilation for atmospheric chemistry models. In: *Advances in Air Pollution Modeling for Environmental Security*. I. Faragó et al. (eds.) (2005), 93–104.
11. Dimitriu, G.: Data assimilation for a model in atmospheric chemistry and oceanography. In: *Proceedings of the International Symposium on Systems Theory*, SINTES 12, XIIth Edition, October 20-22, Craiova, Romania, Vol. IV: Computer Engineering (2005), 717–722.
12. Dimitriu, G.: Using Singular Value Decomposition in Conjunction with Data Assimilation Procedures. LNCS **4310** (2007), 435–442.
13. Dimitriu, G.: A numerical comparative study on data assimilation using Kalman filters. *Computers and Mathematics with Applications* **55** (2008),

2247-2265.

14. Dimitriu, G., Apreutesei, N., Ştefănescu, R.: Numerical simulations with data assimilation using an adaptive POD procedure. LNCS **4310** (2009), 435–442.
15. Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *J. Geophys. Res.*, **99**(C5) (1994), 10143–10162.
16. Fang, F., Pain, C.C., Navon, I.M., Piggot, M.D., Gorman, G.J., Allison, P.A., Goddard, A.J.H.: Reduced-order modelling of an adaptive mesh ocean model. *Int. J. Numer. Meth. Fluids* **59** (2009), 827–851.
17. Ghil, M., Malanotte-Rizzoli, P.: In *Data Assimilation in Meteorology and Oceanography. Advances in Geophysics.* R. Dunowska, B. Saltzman (eds.) **33** (1991), 141–266.
18. Gilchrist, B., Cressman, G.P.: An experiment in objective analysis. *Tellus* **6** (1954), 97–101.
19. Heemink, A.W., Segers, A.J.: Modeling and prediction of environmental data in space and time using Kalman filtering, *Stochastic Environmental Research and Risk Assessment* **16** (3) (2002), 225–240.
20. Heemink, A.W., Verlaan, M., Segers, A.J.: Variance reduced ensemble Kalman filtering. *Mon. Weather Rev.*, **129** (7) (2001), 1718–1728.
21. Houtekamer, P.L., Mitchell, H.L.: Data assimilation using an Ensemble Kalman Filter technique, *Mon. Weather Rev*, **126** (1998), 796–811.
22. Loon, M. Van, Heemink, A.W.: *Kalman Filtering for Nonlinear Atmospheric Chemistry Models: first experiences*, Report MAS-R9711, Amsterdam (1997).
23. Luo, Z., Zhu, J., Wang, R., Navon, I.M.: Proper Orthogonal Decomposition Approach and Error Estimation of Mixed Finite Element Methods for the Tropical Pacific Ocean Reduced Gravity Model. *Computer Methods in Applied Mechanics and Engineering* **196**, (41-44) (2007), 4184–4195.
24. Martin, M.J., Nichols, N.K., Bell, M.J.: Treatment of Systematic Errors in Sequential Data Assimilation. Met Office Ocean Applications. Technical Note, No. **21** (1999).
25. Sandu, A., Daescu, D.N., Carmichael, D.N., Chai, T.: Adjoint sensitivity analysis of regional air quality models. *J. Comput. Phys.* **204** (1) (2005), 222–252.
26. Segers, A.: *Data assimilation in atmospheric chemistry models using Kalman filtering*, PhD thesis, Delft University Press (2002).
27. Qui, C.J., Zhang, L., Shao, A.M.: An Expicit four-dimensional variational data assimilation method. *Sci. China Ser. D-Earth Sci.*, **50**(8) (2007), 1232–1240.
28. Thépaut, J.N., Hoffman R.N., Courtier, P.: Interactions of dynamics and observations in a 4D variational assimilation. *Mon. Weather Rev.*, **121**

(1993), 3393–3414.

29. Verwer, J.G., Blom, J.G., Hundsdorfer, W.H.: An implicit-explicit approach for atmospheric transport-chemistry problems. *Appl. Numer. Math.* **20** (1996), 191–209.
30. Walker, J.C.G.: *Numerical adventures with geochemical cycles*, Oxford University Press, New York (1991).
31. Zlatev, Z.: *Computer treatment of large air pollution models*, Kluwer Academic Publishers, Dordrecht (1995).
32. Zlatev, Z., Brandth, J., Havasi, À.: Implementation issues related to variational data assimilation: some preliminary results and conclusions, *Working Group on Matrix Computations and Statistics*, Copenhagen, Denmark, April 1-3, (2005).