On a simple numerical method for estimation of an unknown matrix. Application to adaptive filtering based innovation approach

HONG SON HOANG
SHOM
HOM/REC
42 Av Gaspard Corriolis
FRANCE
hhoang@shom.fr

REMY BARAILLE
SHOM
HOM/REC
42 Av Gaspard Corriolis
FRANCE
remy.baraille@shom.fr

Abstract: We describe a simple algorithm for estimating the elements of a matrix as well as its product decomposition under the condition that only the matrix-vector product is accessible. This algorithm is based on application of the stochastic simultaneous perturbation (SSP) method. Such problems arise frequently in solving inverse problems, nonlinear filtering and control of dynamical systems, especially in data assimilation in high dimensional systems where the numerical model is given by a computer code and the error covariance matrix is to be estimated in order to specify the filter gain for state estimation. Theoretical results on the convergence of the proposed algorithm are proven, its efficiency is demonstrated in numerous engineering estimation problems, especially for the design of an adaptive filter for data assimilation in a high dimensional ocean model.

Key–Words: Numerical differentiation, Stochastic simultaneous perturbation, Statistical simulation, Dynamical system, Real Schur vectors, ocean numerical model, data assimilation.

1 Introduction

Consider the following linear system of equations

$$\Phi x = b, \tag{1}$$

where $\Phi \in R^{m\times n}, b \in R^m, x \in R^n$. We assume in (1) that the matrix Φ is unknown, but the vector y', $y' = \Phi x'$, is known once x' is given. The problem we are interested in is to estimate the matrix Φ, and, when needed, to decompose it into the product of two matrices, $\Phi = AB$.

The motivation for solving the aforementioned problem arises in many engineering inverse problems. As an example, consider the filtering problem

$$\begin{aligned} x_{k+1} &= \phi(x_k) + w_k, \\ z_{k+1} &= h(x_{k+1}) + v_{k+1}. \end{aligned} \tag{2}$$

where $\phi(.)$ and $h(.)$ may be linear or nonlinear functions. Based on a set of observations $z_l, l = 1, 2, ..k$, the filtering problem is to estimate the system state x_k as precisely as possible. For the linear $\phi(x) = \Phi x$, $h(x) = Hx$ and under standard conditions related to the model and observation noise sequences w_k, v_k, the minimum mean squared (MMS) estimate $\hat{x}_k$ can be obtained by the well-known Kalman filter (KF) [19]. In the KF algorithm, the exact values of Φ, H and statistics of the noises w_k, v_k are required.

For nonlinear $\phi(.), h(.)$, the Taylor series expansions are used to linearize the model about a current estimate and the standard KF formalism is applied to obtain the extended KF (EKF) [17].

If the system model is non-linear or not well known, or simply inaccurate, the Monte Carlo methods, especially particle filters, are good candidates [6]. Particle filter techniques provide a methodology for generating samples from the required distribution. In this context, Sampling Importance Resampling (SIR), introduced in [8] and developed in [7], is an original particle filtering algorithm. The distribution is approximated with importance weights, which are approximations to relative posterior densities of the particles, and the sum of the weights is one. A main point is that, most of the times, SIR is used in a sequential setting and this is why one needs to reallocate particles to best deal with the next time integration. Resampling allows to reallocate particles from low density regions into high density regions, making thus to more optimal use of available particles.

However, these methods do not perform well when applied to very high-dimensional systems. One of the reasons is that they require ensembles of large size to estimate posterior densities.

One class of filters, more adapted for solving filtering problems in the high dimensional environment, is an ensemble Kalman filter (EnKF) [9]. The EnKF is

a Monte Carlo approximation of the KF, in which the filtered and forecast error covariance matrices (ECM) are evolved using an ensemble of error samples for the state estimate. The most important difference between the particle filter and the EnKF is lying in the proposition that all probability distribution functions in the EnKF are Gaussian. Using the KF formalism for linear systems, starting from an ensemble of samples from the initial state, the EnKF is performed on the basis of the Bayesian update, combined with advancing the model in time and incorporating new data from time to time. In the EnKF, the ECM is replaced by a sample covariance computed from the ensemble of samples. One of the major disadvantages of the EnKF is the rank deficiency of the sample ECM since usually one can generate the number L of samples only of order $O(100)$ which is too small compared to the dimension of the system state $O(10^6)$. The localized EnKF is proposed in [25] to overcome such difficulty.

In many engineering applications, in particular in data assimilation in meteorology and oceanography, generally speaking, we do not know about $\phi(.)$ (and possible about $h(.)$) : they are given only in the form of the computer code. It means that we can obtain the value $\phi(x)$ (or $h(x)$) for a given x using the computer code. To see the idea to follow in this paper, let us consider the situation $\phi(x) = \Phi x$, $h(x) = Hx$ with the unknown Φ. As the code Φx is given, the question we are interested in is how one can compute numerically Φ in order to perform, for example, the KF. One of the widely used methods for numerically computing Φ in this situation is a componentwise technique. It has been used in [10] to compute the fundamental matrix of a linearized system for updating the Riccati equation in the EKF to estimate the circulation in an idealized Gulf Stream model: the elements of Φ are obtained by computing the product $\Phi e_i, i = 1, ..., n$, where $e_i = (0, ..., 1, ..., 0)^T$ - the vector with all zero components except the i^{th} equal to 1. This method is applicable only if the dimension n is moderate. For numerical models with n, being in the range$[10^6 - 10^7]$, such a method is inapplicable. As an example, one integration Φx in the HYCOM (Hybrid Coordinate Ocean Model) for the Bay of Biscay at SHOM (parallel version, 62 processors) requires about 1h for simulating 5 day prediction at the supercomputer Beaufix (Météo France). It means that during 5 days, it is possible to make maximally 100 model integrations. Clearly, the described method is impossible to apply for the actual model HYCOM at SHOM.

In such situations, it is important to have a procedure capable of estimating the transition matrix independently on its dimension.

Another question, being addressed in this paper, concerns the estimation of the ECM. This matrix plays an essential role in providing a high performance of the adaptive filter (AF). It will be shown that the numerical method, developed in this paper for matrix estimation, can be applied also to estimate the ECM by generating the prediction error (PE) samples. As the size of an ensemble of samples is insufficient for approximating the ECM, to avoid the rank deficiency, the optimal ECM will be found as a solution of an optimization problem under the hypothesis on separability of the horizontal and vertical structure (SeVHS) of the ECM [5]. This new strategy for generating an ensemble of PE samples is different from the method described in [9] for estimating the ECM in the EnKF as well as from that proposed in [12] based on Schur decomposition. The efficiency of this method will be compared with that based on dominant Schur vectors [12].

The paper is organized as follows. In section 2 the algorithm for estimating an unknown matrix is presented (Algorithm 2.1). The proof of convergence of the estimation procedure is given (Theorem 2.1). It will be seen that for a matrix of given dimensions, convergence to true unknown matrix is guaranteed as the number of iterations tends to infinity. For practical applications with high dimensional matrices, as the matrix-product (or model integration) operation is time-consuming, we can run only the algorithm for a very limited number of iterations. As a consequence, it is expected to obtain good estimates only if the required matrix has a sparse structure [3]. In section 3, the algorithm for estimating unknown parameters in a matrix product is described which is based on solving a minimization problem by the SPSA (Simultaneous Perturbation Stochastic Approximation) [29]. As a particular case, it will be proved (Theorem 3.1) that the procedure yields the solution which is equivalent to a singular value decomposition (SVD) for a given matrix [11]. Before going to the problem of estimation of the gain matrix in a high dimensional AF, in section 4 we outline the AF and variational methods (VM) approaches widely used in data assimilation for high dimensional systems. We will show the principal differences between these two approaches from which follow the advantages of the AF. In section 5 the algorithm for solving the parameter estimation problem, closely related to Nearest Kronecker Product (NKP) problem, is presented. This algorithm is important for estimating the ECM which participates in the construction of the AF. Here the SeVHS hypothesis is introduced for the estimated ECM, with the "data" matrix generated by integrating the numerical model with all state components perturbed by SSP method. Numerical examples and simulation results are presented

in section 5 for systems of small and moderate dimensions. Application of the numerical algorithms (matrix estimation, product decomposition) for generating the PE samples as well as for estimation of the ECM are demonstrated here. The obtained ECM will be used in the experiment on the sea surface height (SSH) data assimilation in a high dimensional ocean model MICOM by the AF is presented in section 7. The conclusion is given finally in section 8.

2 Estimation of matrix

Return to Eq. (1). For $b := (b_1, ..., b_m)^T$, taking the derivative of b_1 with respect to (wrt) x yields

$$db_1/dx = (\partial b_1/\partial x_1, ..., \partial b_1/\partial x_n) = (\phi_{11}, ..., \phi_{1n})$$

where $\phi_{i,j}$ are the i, j element of Φ. One can write then

$$db/dx = [(db_1/dx)^T, ..., (db_1/dx)^T]^T = \Phi$$

Thus, if we are able to find a procedure to approximate the derivative of b wrt to x, independently of the dimension of x and at a low cost, it is possible to estimate the elements of the high dimensional Φ.

2.1 Stochastic simultaneous perturbation (SSP) and gradient approximation

In [29] Spall proposes a simultaneous perturbation stochastic approximation (SPSA) algorithm for finding optimal unknown parameters by minimizing some objective function. The main feature of this new algorithm resides in the way to approximate the gradient vector : a sample gradient vector is estimated by perturbing simultaneously all components of the unknown vector in a stochastic way (SSP). This method requires only two or three measurements of the objective function, regardless of the dimension of the vector of unknown parameters. For details on this algorithm, see [29].

2.2 Algorithm for estimation of Φ

Let $\bar{\Delta} := (\Delta_1, ..., \Delta_n)^T$, $\Delta_i, i = 1, ..., n$ be Bernoulli independent and identically distributed (i.i.d.) variables assuming two values +/- 1 with equal probability 1/2. Introduce $[\bar{\Delta}]^{-1} := (1/\Delta_1, ..., 1/\Delta_n)^T$, $\bar{\Delta}_c := c\bar{\Delta}, c > 0$ is a small positive value.

In the context of estimation of Φ, the proposed algorithm looks as follows :

Algorithm 2.1. Suppose it is possible to compute the product $\Phi x = b(x)$ for a given x. At the beginning let $l = 1$. Let the value u be assigned to the vector x, i.e. $x := u$, L be a (large) fixed integer number.

Step 1. Generate $\bar{\Delta}^{(l)}$ whose components are l^{th} samples of the Bernoulli i.i.d. variables assuming two values +/- 1 with equal probabilities 1/2;

Step 2. Compute $\delta b^{(l)} = \Phi(u + \bar{\Delta}_c^{(l)}) - \Phi u$, $\bar{\Delta}_c^{(l)} = c\bar{\Delta}^{(l)}$, c is a small positive value;

Step 3. Compute $g_i^{(l)} = \delta b_i^{(l)}[\bar{\Delta}_c^{(l)}]^{-1}$, δb_i is the i^{th} component of δb, $g_i^{(l)}$ is the column vector consisting of derivative of $b_i(u)$ wrt to u, $i = 1, ..., m$.

Step 4. Go to Step 1 if $l < L$. Otherwise, go to Step 5.

Step 5. Compute

$$\hat{g}_i = \frac{1}{L}\sum_{l=1}^{L} g_i^{(l)}, i = 1, ..., m,$$
$$\hat{\Phi}(L) := D_x b = [\hat{g}_1, ..., \hat{g}_m]^T$$

Comment 2.1. For a fix L, this algorithm yields an unbiased estimation with mean square error proportional to $(1/L)$ (see below).

2.3 Demonstration of convergence of algorithm 2.1

Let us analyse a convergence of Algorithm 2.1. We have

$\delta b := \Phi(u + \delta u) - \Phi u = \Phi\delta u, \delta u = \bar{\Delta}_c$.

Let $\delta b_i^{(l)}$ denote the i^{th} component of $\delta b^{(l)}$. Then

$$\frac{\delta b_i^{(l)}}{\delta u_j^{(l)}} = \sum_{k=1}^{n} \phi_{ik}\delta u_k^{(l)}/\delta u_j^{(l)} =$$
$$\sum_{k=1}^{n} \phi_{ik}\Delta_k^{(l)}/\Delta_j^{(l)} =$$
$$\phi_{i,j} + \sum_{k\neq j} \phi_{ik}\xi^{(l)}(k,j), \xi^{(l)}(k,j) = \Delta_k^{(l)}/\Delta_j^{(l)}$$

We have then

$$\frac{1}{L}\sum_{l=1}^{L}\frac{\delta b_i^{(l)}}{c\Delta_j^{(l)}} = \phi_{i,j} + \sum_{k\neq j}^{n}\phi_{ik}[\frac{1}{L}\sum_{l=1}^{L}\xi^{(l)}(k,j)]. \quad (3)$$

However, the sequence (by l) of $\xi^{(l)}(k,j) = \Delta_k^{(l)}/\Delta_j^{(l)}$ is also a sequence of Bernoulli i.i.d variables assuming two values +/- 1 with equal probabilities 1/2, the sum $[\frac{1}{L}\sum_{l=1}^{L}\xi^{(l)}(k,j)]$ will tend to zero as $L \to \infty$. We have then

$$e_{i,j}(L) := \frac{1}{L}\sum_{l=1}^{L}\frac{\delta b_i^{(l)}}{c\Delta_j^{(l)}} - \phi_{i,j} = \eta_L,$$
$$\eta_L := \sum_{k\neq j}^{n}\phi_{ik}[\frac{1}{L}\sum_{l=1}^{L}\xi^{(l)}(k,j)], \quad (4)$$

and

$$|\eta_L| \leq C(i)|\frac{n}{L}\sum_{l=1}^{L}\xi^{(l)}(k,j)|,$$
$$C(i) = max|\phi_{i,k}|, k = 1, ..., j, j+1, ..., n, \quad (5)$$

which will tend to zero as $L \to \infty$ for fix n and finite $C(i)$. Mention that for $\mu_L := \frac{1}{L}\sum_{l=1}^{L}\xi^{(l)}(k,j)$, $E(\mu_L^2) = \frac{1}{L}$ hence $e_{i,j}(L)$ will tend to $\phi_{i,j}$ (in a mean squared sense) with the error proportional to $(1/L)$. We have

Theorem 2.1 Consider Algorithm 2.1 for estimation of the elements of the matrix Φ. Then this algorithm will yield the estimates for the elements of Φ with the mean squared error (MSE) proportional to $1/L$ where L is the number of samples used in the estimation procedure.

Comment 2.2

The estimation in (3) shows for a fix n the convergence of the estimate $\phi_{i,j}$ is proportional to $1/L$. It is seen that for large n, if many elements of Φ are of nearly the same magnitude (and not to small), the convergence will be slow and a large number of samples (L) will be required. Fortunately, for numerical models resulting from discretization of the set of partial differential equations (PDEs) (they are just what we expect to do with) the things look quite different ([3], p. 509). The local character of the PDE (contains only low-order derivatives), as well as the local character of the discretization schemes applied to the differential operators (implying only neighboring mesh points) cause the numerical models to have the so-called *sparse matrix*, with non-zero elements on just a few diagonals. A moderate number of samples L then is possibly sufficient to provide good estimates for $\phi_{i,j}$.

3 Estimation of decomposition of Φ

3.1 Problem statement

In practice, if the dimensions of Φ are too high, there is no interest to store Φ by its elements (even if they are known). There is a need to approximate this matrix by some matrix in a subspace of fewer dimensions, which in some sense is the best estimate among the members of a class of matrices of a given structure (for example, a class of matrices of given rank).

Let Φ be a matrix of dimensions $(m \times n)$. For definiteness, let $m \leq n$ with rank$(\Phi) = m$. We want to find the best approximation for Φ among members of the class of matrices

$$\Phi_e = AB^T, A \in R^{m\times r}, B \in R^{n\times r}. \quad (6)$$

under the constraint

Condition (C)

A, B are matrices of dimension $m \times r, r \times n$, $r \leq m$, rank$(AB^T) = r$.

Under the condition (C) the optimization problem is formulated as

$$J(A,B) = ||\Phi - \Phi_e||_F^2 = ||\Phi - AB^T||_F^2 \to \min_{(A,B)}, \quad (7)$$

where $||.||_F$ denotes the Frobenius matrix norm.

In the context of the problem of estimation Φ, the problem (7) sometimes is replaced by

$$J(A,B) = ||\hat{\Phi}(L) - AB^T||_F^2 \to \min_{A,B}, \quad (8)$$

where $\hat{\Phi}(L)$ is some estimate of Φ obtained, for example, by application of Algorithm 2.1. The problem (7)(8) is a particular case of the parameter estimation when A, B are parametrized by θ - vector of unknown parameters, i.e. $A := A(\theta), B := B(\theta)$, and

$$J(\theta) = ||\Phi - A(\theta)B(\theta)^T||_F^2 \to \min_{\theta}, \quad (9)$$

It is evident that (7) is a particular case of (8) for θ consisting of all elements of A and B.

Consider Φ and let $U\Sigma V^T$ be SVD of Φ [11], i.e.

$$\Phi = U\Sigma V^T, U \in R^{m\times m}, V \in R^{n\times n},$$
$$\Sigma = [\Sigma_m|0],$$
$$\Sigma_m = \text{diag}[\sigma_1, ..., \sigma_m], \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_m \geq 0. \quad (10)$$

Let

$$\tilde{\Phi} = \Phi + \Delta\Phi, \tilde{\Phi} = \tilde{U}\tilde{\Sigma}\tilde{V}^T. \quad (11)$$

and $\tilde{\sigma_1} \geq \tilde{\sigma_2} ... \geq \tilde{\sigma_m}$, $\tilde{\sigma_k}$ be the k^{th} singular value of $\tilde{\Phi}$.

Then we have

Lemma 3.1 (Mirsky' Theorem [30])

The following inequality holds

$$\sum_{i=1}^{m}(\tilde{\sigma_i} - \sigma_i)^2 \leq ||\Delta\Phi||_F^2 \quad (12)$$

where $||.||_F$ denotes the matrix Frobenius norm.

Theorem 3.1 below characterizes a solution of the problem (9)(C).

Theorem 3.1. Suppose $A_o B_o^T$ is a solution to the problem (7) subject to Condition (C). Then

$$J(A_o, B_o) = \sum_{k=r+1}^{m} \sigma_k^2 \quad (13)$$

where σ_k is the k^{th} singular value of Φ, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_m$.

Theorem 3.1 implies that $\Phi_e^o := A_o B_o^T$ is equal to the matrix formed by truncating the SVD of Φ to its first r singular vectors and singular values.

Proof.
Consider $\Phi, \tilde{\Phi}$ defined in (10), (11).
Introduce

$$\Phi_r := U_r \Sigma_r V_r^T = U \Sigma' V^T, \Sigma' = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix},$$
$$U := [U_r, U'], V := [V_r, V'], \Sigma_r = \text{diag}[\sigma_1, ..., \sigma_r],$$

or it can be rewriten as

$$\Phi_r = AB^T, A := U[\Sigma']^{1/2}, B := V[\Sigma']^{1/2}.$$

One sees that Φ_r is the matrix formed by truncating the SVD of Φ to its first r singular vectors and singular values. As Φ_r has the rank equal to r hence J attains the minimum at Φ_r,

$$J(X) = ||\Phi - X||_F^2 \to \min_X,$$

with

$$J(\Phi_r) = \sum_{k=r+1}^{m} \sigma_k^2$$

(see [30]).

This proves that the problem (7)(C) has Φ_r as its solution and the value $J(\Phi_r)$ is equal to the right hand side of (13).

Let now A', B' be such that $A' \in R^{m \times r}, B' \in R^{n \times r}$, and rank $(A'B'^T) = r$. We show that

$$J(A', B') \geq J(\Phi_r).$$

Let the singular values of $A'B'^T$ be $\sigma_1' \geq \sigma_2' ... \geq \sigma_m'$. Then $\sigma_{r+1}' = ... = \sigma_m' = 0$. By Lemma 3.1,

$$||\Phi' - \Phi_r||_F^2 \geq \sum_{i=1}^{m} |\sigma_i' - \sigma_i|^2 \geq \sigma_{r+1}^2 + ... + \sigma_m^2 = ||\Phi_r - \Phi||_F^2$$

which proves the theorem. (End of Proof)

3.2 Decomposition algorithm

Theorem 2.1 allows to avoid storing elements of the estimate $\hat{\Phi}(L)$ of Φ (their number is of order $O(10^{m \times n})$). In fact, it is our interest to consider the estimate $\hat{\Phi}(L)$ (see Algorithm 2.1) as composed from two following ensembles of vectors-elements :

$$En_L(\delta x) := [\delta x^{(1)}, ..., \delta x^{(L)}], \delta x^{(l)} = c\bar{\Delta}^{(l)},$$
$$En_L(\delta b) := [\delta b^{(1)}, ..., \delta b^{(L)}],$$
$$\delta b^{(l)} = (\delta b_1^{(l)}, ..., \delta b_m^{(l)})^T, l = 1, ..., L. \quad (14)$$

The ensemble $En_L(\delta x)$ is composed of all vectors of random perturbations and the ensemble $En_L(\delta b)$ - from all perturbed output vectors. It is seen that the number of all elements of these two ensembles is proportional to $(L(m+n))$ which is much less than $O(10^{m \times n})$ - the number of elements of $\hat{\Phi}(L)$ - for large m and n.

As an example, computing the product $z = \hat{\Phi}(L)y$ with a vector $y \in R^n$ can be performed as follows : as $z_i = \sum_{k=1}^{n} \hat{\phi}_{ik} y_k = \sum_{k=1}^{n} [\frac{1}{L} \sum_{l=1}^{L} \frac{\delta b_i^{(l)}}{\delta x_k^{(l)}}] y_k$,

$$z_i = \frac{1}{L} \sum_{l=1}^{L} \delta b_i^{(l)} [\sum_{k=1}^{n} \frac{y_k}{\delta x_k^{(l)}}], i = 1, ..., m$$

where z_i is the i^{th} component of z. In a more compact form we have

$$z = \frac{1}{L} \sum_{l=1}^{L} \alpha_l \delta b^{(l)}, \alpha_l := \sum_{k=1}^{n} \frac{y_k}{\delta x_k^{(l)}}. \quad (15)$$

Thus, from the computational point of view, Eq. (15) allows to perform $z = \hat{\Phi}(L)y$ with $L(m+2n)+1$ elementary operations. In contrast, if $\hat{\Phi}(L)$ is stored by its elements, it requires mn elementary operations to fulfill the same product.

Similarly, it is not hard to calculate z_i of $z = \hat{\Phi}^T(L)y, y \in R^m$ as

$$z_i = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{\delta x_i^{(l)}} \sum_{k=1}^{m} \delta b_k^{(l)} y_k, i = 1, ..., n \quad (16)$$

The formula (16) is of extreme importance : it allows us to perform a matrix-vector product of the transpose (adjoint) Φ^T by a vector. It means that if Φ is well approximated, the product $\Phi^T y$ can be easily performed without the need to construct an adjoint code. As it is known, writing an adjoint code for high dimensional systems is time consuming and hard task, especially when the direct model code has discontinuity elements.

3.2.1 Algorithm 3.1

For simplicity, let the elements of Φ (or $\hat{\Phi}$) be available (may be in an algorithmic form). At the beginning let $l = 1$. Let us choose $\hat{A}^{(1)} \in R^{m\times r}$, $\hat{B}^{(1)} \in R^{n\times r}$, rank $[\hat{A}^{(1)}\hat{B}^{(1),T}] = r$.

Step 1. For a given l, generate $\Delta_A^{(l)} \in R^{m\times r}, \Delta_B^{(l)} \in R^{n\times r}$ whose elements are samples of Bernoulli i.i.d. variables assuming two values +/- 1 with equal probabilities 1/2;

Step 2. Compute

$$\delta J^{(l)} := J[\hat{A}^{(l)} + c^{(l)}\Delta_A^{(l)}, \hat{B}^{(l)} + c^{(l)}\Delta_B^{(l)}] - J[\hat{A}^{(l)}, \hat{B}^{(l)}],$$

$c^{(l)}$ is a small positive value, the sequence $\{c^{(l)}\}$ satisfies the conditions for convergence of SPSA procedure [29].

Step 3. Compute $G_A^{(l)}(i,j) = \delta J^{(l)}/\delta a_{i,j}^{(l)}, i = 1,...,m; j = 1,...,r$, $G_B^{(l)}(i,j) = \delta J^{(l)}/\delta b_{i,j}^{(l)}$, $i = 1,...,n; j = 1,...,r$ where $\delta a_{i,j}^{(l)}, \delta b_{i,j}^{(l)}$ are the (i,j) elements of $c^{(l)}\Delta_A^{(l)}, c(l)\Delta_B^{(l)}$ respectively;

Step 4. Compute the elements of $\hat{A}^{(l)}, \hat{B}^{(l)}$ by

$$\hat{a}_{i,j}^{(l+1)} = \hat{a}_{i,j}^{(l)} - \gamma^{(l+1)}G_A^{(l)}(i,j), i = 1,...,m; j = 1,...,r,$$
$$\hat{b}_{i,j}^{(l+1)} = \hat{b}_{i,j}^{(l)} - \gamma^{(l+1)}G_B^{(l)}(i,j), i = 1,...,n; j = 1,...,r.$$

$\gamma^{(l)}$ is a sequence of positive values guaranteeing a convergence of the SPSA algorithm (for the conditions for both sequences $\{\gamma^{(l)}\}, \{c^{(l)}\}$, see [29]).

Step 5. Go to Step 1 if $l \leq L$. Otherwise, go to Step 6.

Step 6. Compute

$$\hat{\Phi}(L) := \hat{A}^{(L)}\hat{B}^{(L)}.$$

Step 7. Stop.

Comment 3.1. To accelerate a convergence of the estimation procedure, one can use, instead of the estimate in *Step 6*, the averaging procedure [27] as follows

$$\hat{\Phi}_a(L) := \frac{1}{L}\sum_{l=1}^{L}\hat{A}^{(l)}\hat{B}^{(l)}. \quad (17)$$

3.2.2 Algorithm 3.2

Another way to decompose the matrix Φ is to apply iteratively Algorithm 3.1 as follows

Algorithm 3.2.
At the beginning let $i = 1$.

Step 1. For $i = 1$, solve the minimization problem

$$J_1 = ||\Phi_1 - ab^T||_F^2 \to \min_{a,b},\ a \in R^m, b \in R^n.$$
$$\Phi_1 := \Phi,\ \text{rank}(ab^T) = 1.$$

Its solution is denoted as $\hat{a}(1), \hat{b}(1)$.

Step 2. For $i < r$, put $i := i + 1$ and solve the problem

$$J_i = ||\Phi_i - ab^T||_F^2 \to \min_{a,b},\ a \in R^m, b \in R^n.$$
$$\Phi_i := \Phi - \sum_{k=1}^{i-1}\hat{a}(k)\hat{b}^T(k),\ \text{rank}(ab^T) = 1$$

Its solution is denoted as $\hat{a}(i), \hat{b}(i)$.

Step 3. If $i = r$, compute

$$\hat{\Phi} = \hat{A}(r)\hat{B}^T(r),$$
$$\hat{A}(r) = [\hat{a}(1),...,\hat{a}(r)],\ \hat{B}(r) = [\hat{b}(1),...,\hat{b}(r)].$$

and stop. Otherwise, go to *Step 2*.

Theorem 3.2. The couple $\hat{A}(r), \hat{B}(r)$ produced by Algorithm 3.2 is a solution for the problem (8)(C).

Proof. By Theorem 3.1, after *Step 1* one obtains $\hat{a}(1)\hat{b}^T(1) = u_1v_1^T\sigma_1$ where u_1, v_1 are the left- and right-singular vectors associated with σ_1 (see (10)). The matrix Φ_2 then has σ_2 as its biggest singular value with u_2, v_2 as associated left and right singular vectors. Solving $J_2 \to \min_{a,b}$ then yields $\hat{a}(2)\hat{b}^T(2) = u_2v_2^T\sigma_2$ and so on for $i = 3,...,r$. It means that

$$\hat{A}(r)\hat{B}^T(r) = \sum_{k=1}^{r}\hat{a}(k)\hat{b}^T(k) = \sum_{k=1}^{r}u_kv_k^T\sigma_k = U_r\Sigma_rV_r^T.$$

(End of proof)

Comment 3.2. Algorithm 3.2 requires to solve r optimization problems of the type $J_i \to \min_{a,b}$ compared with one optimization problem in Algorithm 3.1. However, as a and b are vectors, the number of iterations should be much fewer in solving $J_i \to \min_{a,b}$ compared with that of (8).

4 Adaptive filter and variational method

In this section, we first describe the AF based innovation approach and the VM widely used for data assimilation in high dimensional systems. The differences between these two approaches are presented

from which it follows clear advantages of the AF over the use of the VM.

Consider the problem of estimation of the system state x_k,

$$x_{k+1} = \Phi x_k + w_k, k = 0, 1, ... \quad (18)$$

given the observations z_k

$$z_{k+1} = Hx_{k+1} + v_{k+1}, k = 0, 1, 2, ... \quad (19)$$

here $x_k \in R^n$ is the system state at k instant, $\Phi \in R^{n\times n}$ is the fundamental matrix, $z_k \in R^p$ is the observation vector, $H \in R^{p\times n}$ is the observation operator, w_k, v_k are the model and observation uncorrelated noise sequences which are mutually uncorrelated and uncorrelated with x_0.

For the today's ocean (or meteorological) numerical models, the system state x_k has the dimension lying in the range $[10^6 : 10^8]$ and there is uncertainty in statistics of the initial state, model and observational noises. Due to very large n, it is impossible to apply traditional optimal procedures to estimate the system state and for that reason there exist different approximation algorithms for solving this estimation problem.

4.1 Variational method (VM) [31]

The VM consists of minimizing

$$J[x_0, ..., x_N] = e_0 M_0^{-1} e_0 + \sum_{k=1}^{N} (z_k - Hx_k)^T R^{-1} (z_k - Hx_k) \to \min_{[x_0,...,x_N]}, \quad (20)$$

under the constraints (18), where $e_0 := x_0 - \bar{x}_0$. Thus, the VM seeks an optimal solution in the functional space (space of functions $\{x_k\}$). For systems of high dimension, this task is impossible to perform. The simplification is required. Suppose the system (18) is perfect, i.e. $w_k = 0$. Expressing all x_k as functions of the initial state x_0,

$$x_k = \Phi(k, 0)x_0,$$
$$\Phi(k, l) = \Phi^{k-l}, (k > l), \Phi(k, k) = I, \quad (21)$$

$I$is the identity matrix of appropriate dimension

and substituting $x_k, \forall k$ (21) into (19), at each k^{th} observation instant we have

$$z_k = H_k^1 x_0 + \epsilon'_k, k = 1, 2, .. \quad (22)$$
$$H_k^1 := [(H_1\Phi(1, 0))^T, ..., (H\Phi(k, 0))^T]^T,$$
$$v_k^1 = [v_1^T, ..., v^T]^T.$$

The optimization problem (20) now is simplified,

$$J[x_0] \to \min_{[x_0]}, \quad (23)$$
$$J[x_0] := e_0^T M_0^{-1} e_0 + (1/N) \sum_{k=1}^{N} (z_k - H'_k x_0)^T R_k^{-1} (z_k - H'_k x_0), \quad (24)$$
$$H'_k := H\Phi(k, 0).$$

We have now the unconstrained optimization problem (23)(24) with the control vector $\theta := x_0$ - the initial state.

4.2 Adaptive filter

The main underlying principle in the construction of the AF concerns the choice of the innovation representation [18] instead of the original input-output system (18)(19) to formulate the optimization problem [16]. The innovation process, associated with z_k, is written as $\zeta_k = z_k - E[z_k|z_k^1]$ where $E[z_k|z_k^1]$ is conditional expectation, and under standard conditions (gaussianness, uncorrelated noise sequences ...), we have $E[z_k|z_{k-1}^1] = H\hat{x}_{k/k-1}$ hence

$$\zeta_k = z_k - H\hat{x}_{k/k-1}, \hat{x}_{k/k-1} = \Phi\hat{x}_{k-1}, \quad (25)$$

where $\hat{x}_{k/k-1}$ is an optimal in MSE one-step ahead prediction for x_k given z_{k-1}^1. Using ζ_k instead of z_k, under standard conditions, one can write out the formula for the optimal (in minimum mean squared - MMS) estimate $\hat{x}_k$ using the KF

$$\hat{x}_k = \Phi\hat{x}_{k-1} + K_k\zeta_k, \quad (26)$$
$$K_k = M_k H^T [HM_k H^T + R]^{-1} \quad (27)$$

where M_k is the ECM for the prediction $\hat{x}_{k/k-1}$ which can be computed recursively as a solution to the Algebraic Riccati equation (ARE) [19], R is the covariance of v_k. For high dimensional systems, no computational capacity and memory storage are sufficient to solve the ARE. The AF is introduced to overcome these difficulties [13]. From (25) one can rewrite

$$z_k = H\Phi\hat{x}_{k-1} + \zeta_k = H_p\hat{x}_{k-1} + \zeta_k, \quad (28)$$

Considering the problem of estimation $\hat{x}_k$ for the input-output system (26)(28), one can formulate here the optimization problem (like (23))

$$J[x_0] \to \min_{[x_0]}, \quad (29)$$
$$J[\theta] := e_0^T M_0^{-1} e_0 + (1/N) \sum_{k=1}^{N} (z_k - H_p\hat{x}_{k-1})^T \Sigma_k^{-1} (z_k - H_p\hat{x}_{k-1}), \quad (30)$$
$$H_p := H\Phi.$$

As the system (26) must be stable by its construction [13], the error e_0 in the estimate for the initial state $\hat{x}_0$ decreases as assimilation progresses, the term $e_0^T M_0^{-1} e_0$ in (30) can be neglected, and as a consequence, the choice of $\hat{x}_0$ as a control vector (as in the VM), loses its meaning. The matrix Σ_k in principle can be estimated using the innovation sequence ζ_k. For simplicity, let $\Sigma_k = I$. On the other hand, from Eqs (26)(28) it is evident that, the behavior of the system (26)(28) depends on the choice of the gain K_k. As the innovation ζ_k is of minimal variance if the filter is optimal, the AF in [13] is designed to minimize (30) subject to θ consisting of some pertinent parameters of the filter gain $K_k = K_k(\theta)$. For more details on the stabilizing structures for $K_k(\theta)$, see [13]. The optimization problem (30) now takes the form

$$J_N[\theta] \rightarrow \min_\theta,$$
$$J_N[\theta] = (1/N) \sum_{k=1}^{N} \Psi(\zeta_k), \Psi(\zeta_k) := ||\zeta_k||^2. \quad (31)$$

under the constraints (26)(28).

One can see that (31) represents a sample version of the following optimization problem

$$J[\theta] = E[\Psi(\zeta_k)] \rightarrow \min_\theta, \quad (32)$$

where $E(.)$ is the mathematical expectation.

From a practical point of view, there is a less interest in formulating the AF in the form (31) because the filter loses then its sequential character and, as a consequence, the amount of computational burden and memory storage remains still too high. In contrast, it is possible to greatly simplify the implementation of the AF on the basis of (32).

As an example, the problem (32) can be solved by applying a stochastic approximation (SA) algorithm,

$$\theta_{k+1} = \theta_k - a_k \nabla_\theta \Psi(\zeta_{k+1}) \quad (33)$$

where $\{a_k\}$ is a sequence of positive scalars satisfying some conditions to guarantee a convergence of the estimation procedure. The standard conditions are

$$a_k > 0, a_k \rightarrow 0, \sum_{k=1}^{\infty} a_k = \infty, \sum_{k=1}^{\infty} a_k^2 < \infty \quad (34)$$

Another advantage of the formulation (32) concerns the gradient computation. Writing out the gradient of the objective function (24) for the VM (or even for $\Psi(\zeta_k)$ in (33)) one sees that their computation requires the product $\Phi^T y$, $y \in R^n$. It means that it is absolutely necessary to have, at least, the code of the adjoint Φ^T to compute the product $\Phi^T y$ (since it is impossible to stock Φ^T of very high dimensions). That adjoint equation (AE) approach constitutes a key tool in the implementation of the VM.

With the formulation (32), one can achieve the optimality of the AF by using the SPSA algorithm [29]. The main feature of the SPSA algorithm is the gradient approximation that requires only two measurements of the objective function, regardless of the dimension of the optimization problem. By this way, there is no need in the construction of the tangent linear (if the model is nonlinear) and adjoint models. For more details, see [14].

4.2.1 Differences between VM and AF

We list here the main differences between two approaches VM and AF from which it becomes clear what are the advantages of the AF over the VM.

(D1) Dynamical system (DS): if in (24), the DS is the initial system (18), in (30) the DS is the filtering equation (15). This difference has an interesting consequence : if in practice, there is very little known about statistics of w_k, the sequence ζ_k is observed and hence it is possible to estimate its statistics.

(D2) w_k in (18) is white, while ζ_k in (26) is a white only if the filter is optimal : This allows to apply different statistical tests for verifying the optimality of the assimilation procedure.

(D3) Control variable x_0 in the VM is the initial state, whereas the control variable in the AF is the gain parameters.

This difference has an important consequence : as x_0 has to be of precise physical meaning (depending, for example, on the ocean domain of interest), the structure for the guess $\theta_0 := \hat{x}_0^0$ - initial state, as well as its correction $\delta\hat{x}_0^\nu$, must be chosen carefully so that $\hat{x}_0^\nu$, $\hat{x}_0^\nu = \hat{x}_0^{\nu-1} + \delta\hat{x}_0^\nu$, must be of physically realistic structure. That is not an easy task. As for the AF, the parameters usually are immaterial hence the choice of structure for θ is of no importance.

(D4) Suppose (18) is unstable. It implies the error growth in estimating x_0 during integration of the direct and AE. As for the AF, by its construction, the filtering system remains stable. It implies that the filtering error is bounded during model integration since the parameters θ_i are lying in the interval guaranteeing a stability of the filter (26).

(D5) Taking the derivative of (24) wrt x_0 one sees that one needs to compute N terms, the k^{th} term is associated with the assimilation instant k and one needs to compute first $\mu_k := \Phi(k, 0)e_0^\nu$ - i.e. to integrate k times the direct model Φ^k and next to integrate backward (k times also) the AE Φ^T. The larger k, the bigger amplification of the initial error e_0^ν and of the observation error v_k. The error e_0^ν is amplified doubly since it is integrated by the direct and adjoint models.

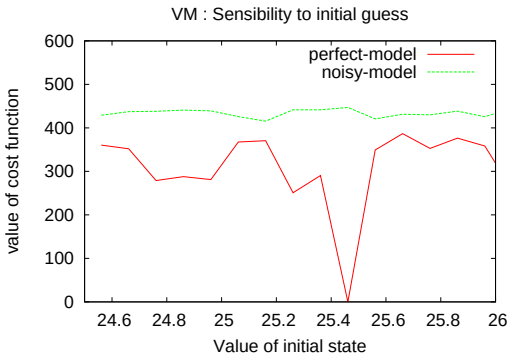

Figure 1. Time averaged variance between the true trajectory and model trajectory in the VM as a function of perturbed third component of the initial state. The global minimum is attained at the true initial condition, but there is no guarantee for the VM to approach the true initial state even in the perfect model case. For the noisy model, the global minimum is not attained at the true initial state. The curve "noisy-model" is scaled by the factor $C = 1/15$.

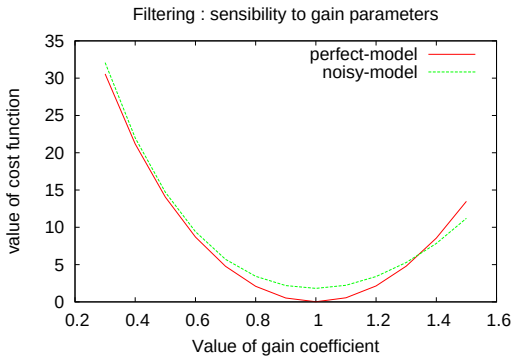

Figure 2. Cost function in the PEF as a function of perturbed third gain parameter θ_3. It is seen that in the PEF the cost function is quadratic wrt to the gain parameter in both situations of the perfect and noisy models. The curve "noisy-model" is scaled by the factor $C = 1/50$.

But the amplification of v_k (and w_k when $w_k \neq 0$) is most worrying since it is integrated in the gradient estimate, making the gradient direction to be, possibly, completely erroneous.

In Figure 1 we show the time averaged variances of the difference between the true trajectory and model trajectory, denoted as $AV(x^*, \hat{x})$ in the data assimilation experiment with the Lorenz system [23] based on the VM. The Lorenz system is 3 dimensional chaotic system. In this experiment the estimate for the initial state is the same as that for the true system except for the third component which varies in the interval $[24.5 : 26.5]$. The true $x_3^*(0) = 25.46091$. The curve "perfect-model" corresponds to the situation $w_k = 0$ whereas the curve "noisy-model" is computed for $w_k \neq 0$ (with variance equal to 2). It is seen that for the perfect model, the global minimum is attained at $x_0^*(3) = 25.46091$ as expected. However, if the system is initialized by the estimate in a vicinity, even not so far from the true $x_3^*(0)$, there is no guarantee that the VM can approach to the true initial state and the resulting estimation error may be very large. For the noisy model, the global minimum is not attained at $x_0^*(3)$.

Figure 2 shows the objective function (32) resulting from the filter (26) whose gain (27) is computed on the basis of $M_k = M$ using PE samples generated on the basis of Schur decomposition [12] (denoted as PEF). Here only the third gain component is varying the interval [0:2]. One sees that the function $AV(x^*, \hat{x})$ is quadratic wrt to the gain parameter, for both situations of the perfect and noisy models. It means that by optimization one can approach the optimal filter for both situations of the perfect and noisy models. as seen in Figure 4.2.1 : here the sample cost function is averaged over all assimilation period, by varying the third parameter θ_3 in the gain (related to the third observed component of the system state).

5 Decomposition in Kronecker product

5.1 Decomposition in Kronecker product and Prediction Error Filter (PEF)

The decomposition algorithm, studied in the previous section, can be applied to solve the problem of decomposition of some matrix into a Kronecker product of two matrices [11]. We show in this section that estimation of parameters of the matrix, given in the form of a Kronecker product, is of particular interest in the design of a filter for high dimensional system. This concerns the prediction error ECM M (or "background" covariance) which plays an important role in determining the filter' gain. The difficulty encountered here is that for high dimensional systems it is impossible to solve the Algebraic Riccati equation to find M and hence to apply the KF. An alternative adaptive filtering (AF) approach is developed in [13] to overcome this difficulty. According to [13], first a class of PEFs is constructed and next the filter performance is optimised by tuning some parameters of the filter gain to minimise the PE of the system outputs (innovation vector [18]). As the PEF requires to specify the ECM, this matrix is estimated in two steps. First a sample "data" matrix M^d is obtained on the

basis of an ensemble of PE samples (generated, for example, by using dominant Schur vector approach ($En(SCH)$ [12], ...). The second step is to solve the optimisation problem to estimate the parameters of the Kronecker product matrix [15].

The objective of this section is to show there exists an another, potentially efficient way to generate an ensemble of PE samples (denoted as $En(SSP)$). This approach is based on perturbing the system state using SSP perturbations. Once having the $En(SSP)$, two step procedure, described above, will be applied to obtain the data matrix M^d and to estimate the parameters of the Kronecker product matrix. The performances of two filters, PEF(SCH) and PEF(SSP), which are designed on the basis of two ensembles of samples $En(SCH)$ and $En(SSP)$ respectively, will be presented and compared with the standard Cooper-Haines filter (CHF). Mention that the CHF is widely used in the oceanic data assimilation [4].

5.2 Estimation of parameters in error covariance matrix M

One of the classical problems in multivariate statistics is to estimate the covariance matrix. For the n-dimensional zero mean error vector δx, given an ensemble of L independent samples, the sample covariance matrix for x is estimated through

$$M^d(L) = \frac{1}{L-1}\sum_{l=1}^{L}\delta x^{(l)}\delta x^{(l),T}. \qquad (35)$$

The properties of the estimate (35) are well studied in classical setting when the dimension of x is small (see [1], [24]).

For the problem of high dimensional covariance matrix, there is no possibility to produce a large number of samples L compared with the state dimension of the numerical model. This happens in today practice of data assimilation in ocean models: the numerical dimension of the problems to be solved is extremely large. As an example, the state dimension of the present ocean models lies in the range $10^6 - 10^7$ and the number of L samples is of the order $O(100)$. With the large number of elements of $M^d(L)$, it is critical to exploit the sparse structure of the covariance, resulting from numerical models. Using the estimate (35) does not take advantage of the sparsity and, as is known, performs poorly under usual matrix norms for large n. For approaches dealing with sparse models in the high dimensional covariance estimation, see [21], [22], [20]. When the dimension n is larger than the number of samples L, the sample covariance matrix (35) is not of full rank, so its inverse will not exist. Secondly, even if the sample covariance matrix is invertible, the expected value of its inverse is a biased estimator for the theoretical inverse since (see [2]),

$$E([M^d(L)]^{-1}) = \tfrac{L}{L-n-2}M^{-1}.$$

The idea to follow here is to use $M^d(L)$ in (35) only as a "data" matrix and the ECM to be used in the filter is estimated by solving an optimization problem. More precisely, a class of parametrized ECMs will be introduced and we will fit a parameterized ECM to the data $M^d(L)$. Concretely, let the estimated ECM $M_e \in R^{n\times n}$ be $M_e = M_e(\mathbf{s},\mathbf{s}')$, $s := (i,j,lr)$ where (i,j,lr) represents a grid point in the three dimensional space. The hypothesis on the separability of vertical and horizontal structure (SeVHS) for M_e (widely used in meteorology and oceanography, [5]) implies

$$M_e(s,s') = M_v(s_v,s'_v)\otimes M_h(s_h,s'_h), \quad s_v := l, s_h := (i,j), \qquad (36)$$

where $\otimes$ denotes the Kronecker product between two matrices,

$$M_v(s_v,s'_v)\otimes M_h(s_h,s'_h) = M(i,j,l;i',j',l') = \begin{pmatrix} m_v(1,1)M_h & m_v(1,2)M_h & ... & m_v(1,n_v)M_h \\ m_v(2,1)M_h & m_v(2,2)M_h & ... & m_v(2,n_v)M_h \\ ... & ... & ... & ... \\ m_v(n_v,1)M_h & m_v(n_v,2)M_h & ... & m_v(n_v,n_v)M_h \end{pmatrix} \qquad (37)$$

In (36) $M_v(s_v,s'_v)$ represents the vertical ECM whereas $M_h(s_h,s'_h)$ - the horizontal ECM. There are two main advantages of the parametrized structured ECM (36)(37): (i) it is easy to ensure a full rank of M_e (by ensuring a positiveness of M_h and M_v) and to avoid rank deficiency in covariance M_e; (ii) the number of parameters to be estimated in the covariance matrix M_e is reduced drastically (see below) compared to the number of elements of M. As the elements $c_{lm} := m_v(l,m)$ of M_v represent horizontal averaged covariances between l^{th} and m^{th} layers, one ensemble (of samples) with small size L constitutes a large data set (the similar averaging procedure is applied to estimate the parameters of M_h). This is equivalent to noise cancelling and provides a fast convergence of the parameter estimation procedure (see section 4).

Represent the estimated matrix M_e (37) in the form

$$M_e(s,s') = M_v(s_v,s'_v) \otimes M_h(s_h,s'_h),$$
$$M_v(s_v,s'_v) \otimes M_h(s_h,s'_h) = M(i,j,l;i',j',l') = \begin{pmatrix} c_{11}M_h & c_{12}M_h & ... & c_{1n_v}M_h \\ c_{21}M_h & c_{22}M_h & ... & c_{2n_v}M_h \\ ... & ... & ... & ... \\ c_{n_v1}M_h & c_{n_v1}M_h & ... & c_{n_vn_v}M_h \end{pmatrix} \quad (38)$$

At the present, in meteorological and oceanic models, the number of vertical layers $n_v < 100$. It is therefore possible to estimate all the elements c_{km} of the vertical ECM, without necessity to introduce additional constraints like homogeneity or isotropy. As to M_h, one assumes that it is homogeneous and is well determined analytically up to a vector of unknown parameters. For example, M_h is usually assumed to have the structure like Gaussian, first-order (second-order) auto-regressive models (FOAR, SOAR, ... [26]). In what follows, for illustration purpose, let $M_h = C_h$,

$$C_h(s_h,s'_h) = exp[-d/L_h], d = d(s_h,s'_h) \quad (39)$$

where $d = d(i,j;i'j') = \sqrt{(i-i')^2+(j-j')^2}$, L_h has the meaning of correlation length. Thus, for the model (38)(39) the vector of parameters θ has $\frac{(n_v+1)n_v}{2}+1$ parameters to be estimated.

The procedure related to estimation of vertical and horizontal covariance matrices M_v, M_h is outlined as follows

Let

$$En_L[.] := [\delta x^{(1)}, ..., \delta x^{(L)}] \quad (40)$$

be an ensemble of PE samples $\delta x^{(l)}, l = 1, ..., L$. The data matrix M^d is obtained by applying (35),

$$M^d(L) = \frac{1}{L-1}\sum_{l=1}^{L} M^{(l)}, M^{(l)} := \delta x^{(l)}\delta x^{(l),T} \quad (41)$$

The reason for generating $\delta x^{(l)}$ in directions of dominant real Schur vectors of the transition matrix of the (linearized) linear dynamical system is given in [12].

Define the vector of unknown parameters as

$$\theta := (\theta_v^T, \theta_h^T)^T,$$
$$\theta_v := (c_{11}, ..., c_{1n_v}, c_{21}, ..., c_{2n_v}, c_{n_v1}, ..., c_{n_vn_v})^T,$$
$$\theta_h := L_h. \quad (42)$$

Considering $M^{(l)}, l = 1, ..., L$ as a sequence of samples for M, the SPSA algorithm (see [29]) can be used to solve the following optimization problem for determining the vector θ,

$$J[\theta] = E[\Psi(M^{(l)},\theta)] \rightarrow \min_\theta,$$
$$\Psi(M^{(l)},\theta)] := ||M^{(l)} - M_v(s_v,s'_v) \otimes M_h(s_h,s'_h)||_F^2, \quad (43)$$

where $||A||_F$ denotes the Frobenius norm of the matrix A [11].

Comment 5.1. Compared to the Nearest Kronecker Problem (NKP) [11], the problem (43) is different : (i) it is aimed at minimizing the cost function which is a mathematical expectation of the squared Frobenius norm of the difference between the data matrix and estimated matrix (which is the Kronecker product of two matrices); (ii) not all the elements of M_v, M_h are estimated but only a few parameters of these matrices are adjusted to minimise the cost function. Of course, this procedure can be applied to solve the traditional NKP problem described in [11].

6 Numerical examples

6.1 Small dimension case

Let us consider the problem of estimating elements of the following matrix $\Phi \in R^{4\times5}$ (see https://en.wikipedia.org/wiki/Singular-value-decomposition)

$$\Phi := A = [a_{i,j}] = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix} \quad (44)$$

In the SVD-decomposition the matrix A has 3 non-zero singular values $\sigma_1 = 4$, $\sigma_2 = 3$, $\sigma_3 = \sqrt{5}$. In the experiment, first we have applied Algorithm 2.1 to obtain the estimate A_e and next Algorithm 3.1 subject to different approximation subspaces corresponding to $r = 1, 2, 3, 4$.

Figure 3 displays the cost function during the estimation process by Algorithm 2.1. It is seen that the process converges after about 120 iterations.

Figure 4 shows the values of sample cost functions (SCFs) resulting from experiments with Algorithm 3.1 on the basis of results produced by Algorithm 2.1, subject to different subspaces with $r = 1, 2, 3$. To see the performance of Algorithm 3.1, remember (13)

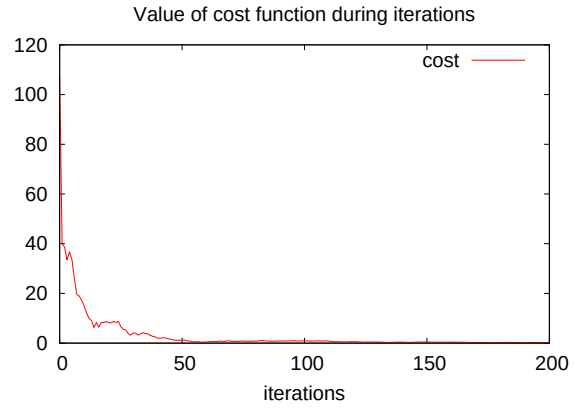

Figure 3. Cost function resulting from Algorithm 2.1: convergence is observed after about 120 iterations

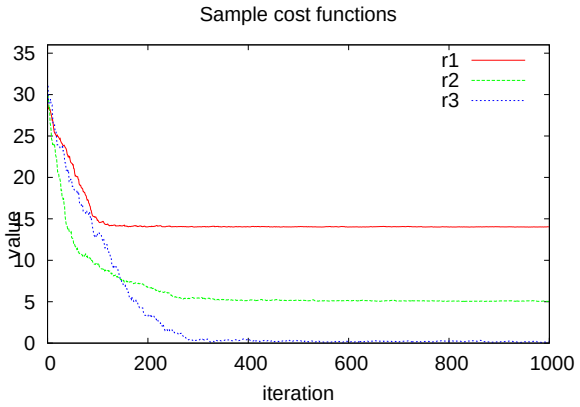

Figure 4. Cost function resulting from experiments subject to different subspaces with $r = 1, 2, 3$. One sees the convergence of the estimation procedure and the value of cost function gives the information on the singular values of the true matrix

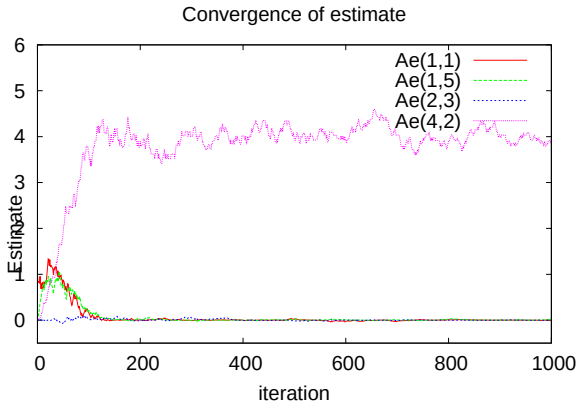

Figure 5. Estimates of matrix elements by one-dimensional subspace approximation : The biggest element $a_{4,2} = 4$ is well estimated, whereas all others are assigned to 0

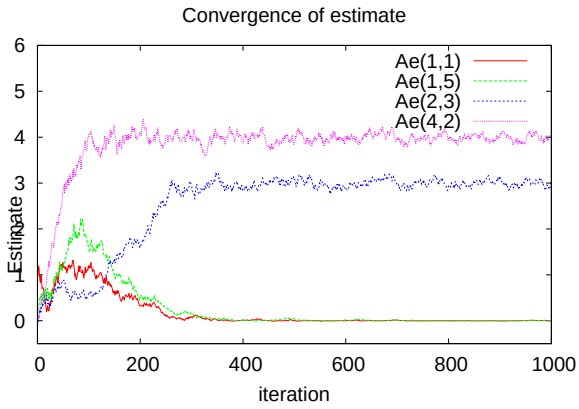

Figure 6. Estimates of matrix elements by two-dimensional subspace approximation : Only two elements $a_{4,2} = 4, a_{2,3} = 3$ are well estimated

$$||A||_F^2 = \sum_{k=1}^{m} \sigma_k^2 \tag{45}$$

where σ_k are the singular values of A. In the sequel we assume $\sigma_1 \geq \sigma_1 \geq ... \geq \sigma_m \geq 0$.

In the first experiment we solve the problem (7) subject to $\Phi_e := A_1 B_1^T, A_1 := [a_1], B_1 := [b_1]$, $a_1 \in R^m, b_1 \in R^n$ which corresponds to setting $r := r_1 = 1$ (for Φ_e, see Eq. (6)). The curve $r1$ shows values of SCF as a function of iteration. It is seen that SCF attains a stationary regime for less than 200 iterations and reaches the value $J_{r1} = 14$. As $||A||_F^2 = 30$ and $\min J_{r1} = \sum_{k=2}^{m} \sigma_k^2 = 14$, one concludes that Algorithm 2.1-3.1 allow to find the optimal a_1^o, b_1^o (since $\sigma_1^2 = 16$).

The second experiment solves the problem (7) subject to $r := r_2 = 2$, i.e. $\Phi_e := A_2 B_2^T, A_2 := [a_1, a_2], B_2 := [b_1, b_2]$, $a_i \in R^m, b_i \in R^n, i = 1, 2$. The theoretical minimal value of $J_{r2} = \sum_{k=3}^{m} \sigma_k^2$ is equal to 5 that is well approached by Algorithms 2.1-3.1

In the experiment 3, the problem (7) is solved subject to $r := r_3 = 3$ hence $\Phi_e := A_3 B_3^T, A_3 \in R^{m \times 3}, B_3 \in R^{n \times 3}$. Figure 4 shows that the SCF tends to 0 which is the minimal value of J_{r3}.

To check whether increasing the subspace dimension can lead to other estimation results, we have performed the experiment 4 which solves (7) subject to $\Phi_e := A_4 B_4^T, A_4 \in R^{m \times 4}, B_4 \in R^{n \times 4}$. It is found (not shown here) in this case that the cost function decreases more quickly than J_{r3} makes. This happens

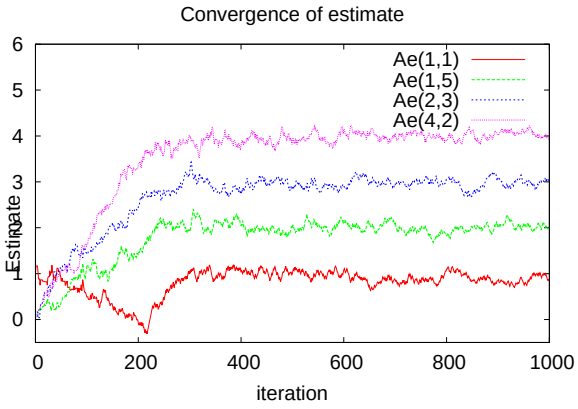

Figure 7. Estimates of matrix elements by three-dimensional subspace approximation : All elements $a_{4,2} = 4, a_{2,3} = 3, a_{1,5} = 2, a_{1,1} = 1$

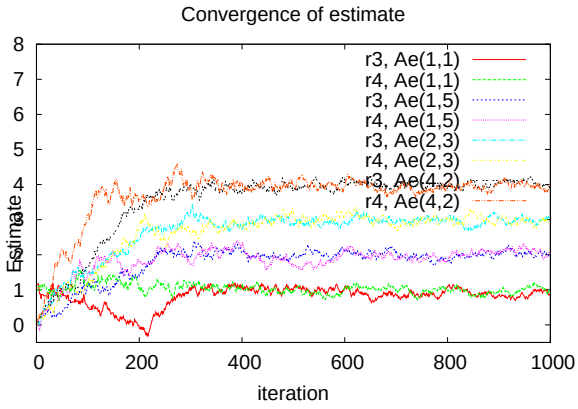

Figure 8. Estimates of matrix elements by three- and four-dimensional subspace approximations : All elements $a_{4,2} = 4, a_{2,3} = 3, a_{1,5} = 2, a_{1,1} = 1$ are well estimated. A more quick convergence is observed for the estimates produced by the four-dimensional subspace approximation.

only up to about 200 iterations and remains almost the same as does J_{r3} thereafter, tending to the minimal value 0. This result is correct since rank(Φ) = 3.

Figure 5 shows the estimates for four elements $A_e(1,1), A_e(1,5), A_e(2,3), A_e(4,2)$ produced by Algorithms 2.1-3.1 in the experiment 1. As $r = 1$, the algorithms are capable of well estimating only the biggest element $a_{4,2} = 4$ using one dimensional subspace approximation. All other elements are assigned to the value 0. The two dimensional subspace approximation allows to well estimate the two largest elements $a_{4,2} = 4, a_{2,3} = 3$ (see Figure 6). As to the three dimensional subspace approximation, all four elements $a_{4,2} = 4, a_{2,3} = 3, a_{1,5} = 2, a_{1,1} = 1$ are well estimated (see Figure 7). One observes here that the choice $r = 3$ is sufficient to well estimate all four non-zero elements of the matrix. Figure 8 displays the estimates produced in two experiments subject to $r = 3$ and $r = 4$. As happened with the cost functions, a more quick convergence is obtained for $r = 4$. We conclude that by involving initially a subspace of dimension higher than the rank of the estimated matrix, it is possible to accelerate convergence of the estimates, but asymptotically they produce the estimates of the same precision.

Comment 6.1. In the experiments presented above, the algorithms are implemented under the condition that all the elements of A are unknown and they are all estimated. It results in a slow convergence of the estimation procedure. If the algorithms estimate only the 4 nonzero elements, the convergence must be much faster (see also *Comment 2.2*).

6.2 Moderate dimension case

Consider the nonlinear transport problem [28]

$$\begin{aligned} \frac{dh}{dt} &= 2\frac{dh}{dx} - h^2 + e^{4t+2x}, \\ h(1,t) &= e^{2t+1}, 0 \le t \le 1, \\ h(x,0) &= e^x, 0 \le x \le 1. \end{aligned} \tag{46}$$

Introduce $x_i = (i-1)dx, i = 1, ..., n, x_1 = 0, x_n = 1$. The numerical model is obtained using the upwind difference formula,

$$\frac{dh}{dt} \approx \frac{h(x_i,t_{k+1}) - h(x_i,t_k)}{dt}, \frac{dh}{dx} \approx \frac{h(x_{i+1},t_k) - h(x_i,t_k)}{dx},$$

Define the system state at the $k := t_k$ instant,

$$s_k := s(t_k) = [s_1(k), ..., s_n(k)]^T, s_i(k) = h(x_i, t_k).$$

Then one can write a discretized version of (46) as

$$\begin{aligned} s_{k+1} &= f(k, s_k) + u_k + w_k, k = 0, 1, 2, ... \\ f_i(k, s_k) &= s_i(k) + c(s_{i+1}(k) - s_i(k)) - s_i^2(k)dt, \\ c &:= 2dt/dx, u_k := e^{4t_k+2x_i}. \end{aligned} \tag{47}$$

Suppose at each time instant t_k we are given the observations at the points $z_i(k) := x_j(k) + v_i, i = 1, ..., 25, j = 2i$. The model and observation errors $\{w_k\}$ are assumed to be random sequences of uncorrelated Gaussian variables with zero mean and variance $\sigma_w^2 = 1$ and $\sigma_v^2 = 1$ respectively. The following parameters are used in modeling the transport problem

$$n = 51, \delta x = 1/(n-1), \delta t = 0.00833$$

To apply Algorithm 2.1, $f(k, s_k)$ is linearized around the nominal state $\hat{s}_i^* = e^{x_i}, i = 1, .., n$ and the obtained linearized model has the transition matrix denoted as Φ. From (47) one has

$$\phi_{i,i} = 1 - 2\frac{\delta t}{\delta x} - 2\delta t \hat{s}^*$$
$$\phi_{i,i+1} = 2\frac{\delta t}{\delta x}, \phi_{n,n} = e^{2\delta t} \quad (48)$$

It is seen that the matrix Φ has a diagonal structure with non-zero diagonal $\phi_{i,i}$ and up-diagonal elements $\phi_{i,i+1}$. Such (large) sparse matrices often appear in scientific or engineering applications when solving partial differential equations. Despite the fact that Theorem 2.1 guarantees a convergence of estimates, Eq. (4) says that the speed of convergence depends on the number of non-zero elements of Φ. Hence, if Φ is dense (i.e. not sparse) and high dimensional, convergence will be extremely slow : the speed of convergence is linearly dependent on n/L where n is the state dimension and L is the number of samples. With the state dimension of order 10^6 it is unimaginable to produce a sufficiently large number of iterations L to yield a small n/L. Operations using standard dense-matrix structures and algorithms are slow and inefficient when applied to large sparse matrices : From Eq. (4), the estimates of zero elements remain significantly non-zero (see below) which contribute to increase of estimation errors. That is why any information about a particular structure of the estimated matrix (for example, the sparseness of the matrix structure as in this experiment), should be exploited maximally to accelerate a convergence of estimates and to save memory and CPU time.

To illustrate this fact, Algorithm 2.1 has been applied subject to 3 following assumptions :

A1. Nothing is known about the structure of Φ (hence all elements of Φ will be estimated),

A2. It is known approximately that all elements of Φ are zero except for $\phi_{i,j}$ such that $|i - j| \leq 1$,

A3. We know exactly the structure of Φ.

Figure 9 shows squared Frobenius norm of estimation error $\delta\Phi := \Phi - \Phi_e$ where Φ_e is the estimate. The curves "A1", "A2", "A3" correspond to the errors resulting from applying Algorithm 2.1 subject to three assumptions A1, A2, A3 respectively. It is seen that for A2, and especially, for A3, one can reduce significantly the estimation errors and accelerates convergence of the algorithm.

To see how the estimated matrices Φ_e are useful for filtering problem, we have applied the EKF to estimate the system state using the observations. The

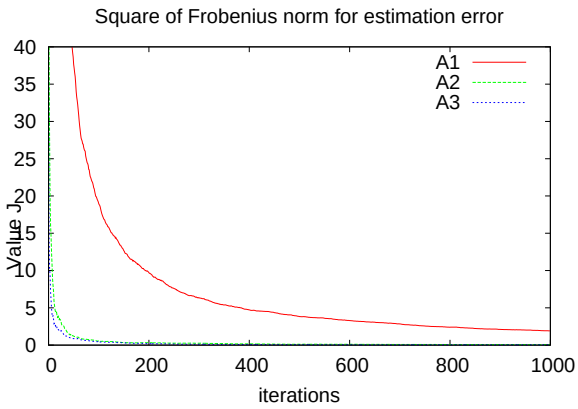

Figure 9. Squared Frobenius norm, resulting from the experiment subject to different assumptions A1, A2, A3. It is seen that convergence of an estimate depends significantly on our knowledge on the structure of the true matrix

different estimates for Φ_e are obtained as described above subject to the assumptions A1-A3 and after 1000 iterations. The "TEKF" (True EKF) is obtained using the formulas (48) for computing the transition matrix of the linearized system. Mention that the transition matrix is calculated at each assimilation instant subject to the last filtered estimate for s_k. Figure 10 shows the performances of different EKFs (in term of root mean square error (rms) of prediction error for the system state).

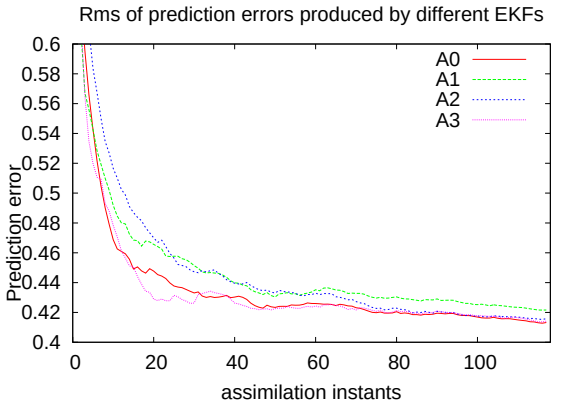

Figure 10. Rms of prediction error (PE) resulting from different EKFs. The error is calculated as an average (time and component-wise). It is seen that the best performance is produced by TEKF (A0) and EKF(A3) (they have near the same performance). It follows by EKF(A2) and the EKF(A1) has less favorable performance.

7 Assimilation in high dimensional ocean model MICOM

In this section the results in section 5 will be applied to choose a class of the ECMs in the form of the Kronecker product. The decomposition algorithm will be implemented after to estimate its unknown parameters in the ECM.. The "data" ECM is obtained from an ensemble of PE samples (denoted as $En(SSP)$) which is generated on the basis of the SSP method. The efficiency of the PEF constructed on the basis of $En(SSP)$ (denoted as PEF(SSP)) will be compared with the CHF (Cooper-Haines filter, [4]) and PEF(SCH) (based on an ensemble of Schur PE samples [12]) for the experiment on sea the sea surface height (SSH) data assimilation with the ocean model MICOM.

7.1 Ocean model MICOM

The ocean model used here is the MICOM (Miami Isopycnal Ocean Model) which is identical to that described in [12]. The model configuration is a domain situated in the North Atlantic from 30^0 N to 60^0 N and 80^0 W to 44^0 W; for the exact model domain and some main features of the ocean current (mean, variability of the SSH, velocity ...) produced by the model, see [12]. The grid spacing is about 0.2^0 in longitude and in latitude, requiring N_h = $N_x \times N_y$ = 25200 (N_x = 140, N_y = 180) horizontal grid points. The number of layers in the model is N_z = 4. It is configured in a flat bottom rectangular basin ($1860km \times 2380km \times 5km$) driven by a periodic wind forcing. The model relies on one prognostic equation for each component of the horizontal velocity field and one equation for mass conservation per layer. We note that the state of the model is $x := (h, u, v)$ where $h = h(i, j, lr)$ is the thickness of lr^{th} layer, $u = u(i, j, lr), v = v(i, j, lr)$ are two velocity components. The layer stratification is made in the isopycnal coordinates, i.e. the layer is characterized by a constant potential density of water. Thus, with three variables $x := (h, u, v)$, the state of the discretized model has the dimension $n = 302400$.

The model is integrated from the state of rest during 20 years. Averaging the sequence of states over two years 17 and 18 gives a so-called *climatology*. During the period of two years 19 and 20, every ten days, we calculate the SSH from the layer thickness h which are considered as observations in assimilation experiments (totally 72 observations are available). To be closer to realistic situations with the observations available only at along-track grid points, assume that we are given the observations not at all grid points at the surface, but only at the points $i = 1, 11, ..., 131$, $j = 1, 11, ..., 171$ and they are noise-free.

7.2 Filter and gain structure

As seen above, the system state x in the MICOM is composed from three variables $x := (h, u, v)$. The filter used in assimilating SSH observations is a reduced-order filter where the component h is estimated from SSH and two velocity components (u, v) are estimated from h using geostrophy hypothesis. For estimating x, the following filter will be used

$$\hat{x}(k) = F[\hat{x}(k-1)] + [I, Eq^T]^T K\zeta(k), \quad k = 0, 1, ... \tag{49}$$

where $\hat{x}(k)$ is the filtered estimate for $x(k)$, at $k := t_k$, $t_{k+1} - t_k = \Delta t = 10\ ds$ (days), $F(.)$ represents the integration of the MICOM nonlinear model over 10 ds, K is the gain of the reduced-order filter, $\zeta(k)$ is the innovation vector. Thus the corrections for three variables (h, u, v) are $dh = K\zeta(k), (du, dv) = Eq[dh]$ where $Eq(.)$ is the operator, computing geostrophic velocity correction from dh.

As SSH observation is a linear function with respect to h, the observation operator H is of the form

$$H = [H_1, ..., H_{N_z}], \tag{50}$$

where $H_1 = H_2 = ... = H_{N_z} = H_h$, $H_h := I_{p \times N_h} \in R^{p \times N_h}$ is the matrix whose elements are equal to 1 at the horizontal points where SSH is observed and 0 otherwise, p is the number of observations available at the surface.

Substituting $M := M_e$ into the gain

$$K = MH^T[HMH^T + R]^{-1} \tag{51}$$

for $R = \sigma_r^2 I$, one can prove that the gain K is of the form

$$\begin{aligned} K = K_v \otimes K_h, K_v = [k(1), ..., k(N_z)]^T, \\ K_h = M_h H_h^T [H_h M_h H_h^T + R]^{-1}, \\ k(lr) = \frac{\sum_{l=1}^{N_z} c_{lr,l}}{\sum_{l,m=1}^{N_z} c_{lm} + \sigma_r^2}, lr = 1, ..., N_z, \end{aligned} \tag{52}$$

substituting (52) into $K\zeta$ results in

$$K\zeta = K_v \otimes K_h\zeta \tag{53}$$

Looking at $K_h\zeta$ one recognizes that K_h plays the role of the interpolation operator which interpolates

the innovation ζ, available at the observation points, over all surface grid points.

Mention that the elements c_{lm} may be chosen a priori, from different physical considerations ... For example, in the Cooper-Haines filter (CHF, see [4], [12]), c_{lm} participate indirectly in deducing the gain coefficients from several physical constraints (conservation of potential vorticity, no motion at the bottom layer ...).

7.3 Assimilation results

7.3.1 Generating ensemble of PE samples

The gain K (51) will be completely defined if the ECMs M and R are given. While R is more or less well known, there is a great difficulty in specification of M. To do that, assume M is of the form of Kronecker product (38)(39) and our task is to estimate optimally the parameters c_{lm} and L_h by solving the problem (43). This is possible if the data matrix M^d is available. In [15] the matrix M^d is obtained using the dominant Schur vector approach. Here the SSP approach will be used to generate an ensemble of PE samples $En(SSP)$ and the produced assimilation results will be compared to those obtained in [15]. The PE samples are simulated as follows : let

$$\delta h^{(l)}(i,j,lr) = \delta h^o(i,j,lr) + c\Delta h^{(l)}(i,j,lr), c > 0,$$

be l^{th} sample where $\Delta h^{(l)}(i,j,lr)$ are random Bernoulli i.i.d. variables assuming 2 values +/- 1 with the equal probability 1/2 and $c > 0$ is small value. For $h^{(l)}(i,j,lr) = h(i,j,lr) + \delta h^{(l)}(i,j,lr)$ we compute the velocity components $u^{(l)}(i,j,lr)$ and $v^{(l)}(i,j,lr)$ by applying the geostrophy to $h^{(l)}(i,j,lr)$. The component $\delta h^o(i,j,lr) = h(i,j,lr;k^o{+}1){-}h(i,j,lr;k^o)$ where k^o is a fix time instant. This component is added to the perturbation in order to guarantee $\delta h^{(l)}(i,j,lr)$ to be of "well defined physical structure".

At the moment t_0, let the model be integrated from the state $x_0 := x_{clim}$ where x_{clim} represents a climatology (see section 6.1). Let $x_p := F(x_0)$ be the prediction resulting from forwarding the MICOM from x_0 over the interval (t_0, t_1). In the same way, for $l = 1$, forwarding MICOM from $x_0^{(l)} = x_0 + \delta x^{(l)}$, $\delta x^{(l)} := (\delta h^{(l)}(i,j,lr), \delta u^{(l)}(i,j,lr), \delta v^{(l)}(i,j,lr))$ over the interval (t_0, t_1) will produce $x_p^{(l)} = F(x_0^{(l)})$. Here $\delta u^{(l)}(i,j,lr)$, $\delta v^{(l)}(i,j,lr)$ are also computed from $\delta h^{(l)}(i,j,lr)$ be gepstrophy. One sees that

$$\delta x_p^{(l)} = (\delta h_p^{(l)}, \delta u_p^{(l)}, \delta v_p^{(l)}) = x_p^{(l)} - x_p$$

represents the PE sample, resulting from forwarding the state $x := x_0$ and its perturbed state $x' := x + \delta x^{(l)}$ over 10 days, by the model,

$$\delta x_p^{(l)} = F(x') - F(x), x' := x + \delta x^{(l)}$$

By drawing an ensemble of samples $\Delta h^{(l)}(i,j,lr), l = 1,2,...,L$ and repeating the sample procedure, we obtain the ensemble of samples $En_L[\delta h_p^{(l)}], l = 1,...,L$ and it can be used to generate the sample ECM

$$M^d(L) = \frac{1}{L}\sum_{l=1}^{L} M_l^d, M_l^d := \delta h_p^{(l)} \delta h_p^{(l),T} \qquad (54)$$

It is seen that each sample is generated by two integrations of the model, one is started from the state x, another - from x'.

In parallel, for the comparison purpose, one ensemble of PE samples $En(SCH)$ is also generated using the sampling procedure in [12]. According to [12], the PE samples from $En(SCH)$ are simulated in the direction of the most rapid growth of the prediction error (the first dominant Schur vector or Krylov vector [12]) associated with the linearized transition matrix over time period (t_k, t_{k+1}) (as MICOM is a nonlinear model).

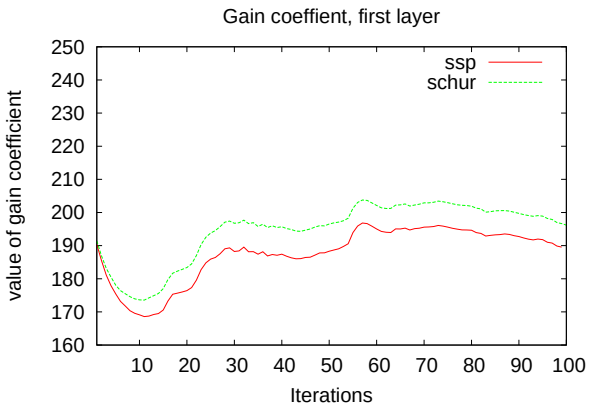

Figure 11. Evolution of estimates for the gain coefficients at the first layer, computed from $\hat{c}_{kl}$ on the basis of $En(SSP)$ and $En(SCH)$ during model integration

7.3.2 Estimation of gain elements

The vector of parameters θ (see Eq. (42)) is estimated by applying the SPSA algorithm to solve the optimization problem (43). For a comparison purpose, the correlation length L_h is fixed, $L_h = 25$ and we estimate

Table 1: Gain coefficients in different filters

$En[.]$	$k(1)$	$k(2)$	$k(3)$	$k(4)$
CHF	185.97	0	0	-184.97
PEF(SCH)	196.65	-70.49	-59.07	-66.09
PEF(SSP)	148.235	-28.161	-31.389	-87.686

Table 2: rms of prediction error for ssh, and u, v velocity components

rms	CHF	PEF(SCH)	PEF(SSP)
ssh(fcst) (cm)	6.455	4.091	3.704
u(fcst) (cm/s)	7.501	5.255	4.966
v(fcst) (cm/s)	7.618	5.599	5.331

only the elements of M_v, i.e. $\theta := [c_{lm}]$ since the CHF (see below) is also applied subject to $L_h = 25$.

In Figure 11 we show the evolution of the gain coefficient $k(1)$ (for the first layer) as a function of the number of samples which are obtained from two ensembles $En(SCH)$ and $En(SSP)$ (see (52)). In (52) we put $\sigma_r^2 = 0$ which corresponds to the situation when the observations are noise-free.

Table 1 displays the values of gain coefficients obtained from two ensembles $En(SCH)$ and $En(SSP)$. Here we show also the gain coefficients of a so-called Cooper-Haines filter (CHF) [4] which projects the PE of the surface height data by lifting or lowering of water columns. We observe that all $k(lr), lr = 1, .., 4$ in the three filters are of the same sign. Compared to PEF(SSP), corrections in PEF(SCH) are bigger at the three upper layers and smaller at the bottom. In the CHF, no corrections are assigned in the intermediate thickness layers (2 and 3). As $\sigma_r^2 = 0$, $\sum_{l=1}^{4} k(l) = 1$, the magnitude of correction in the 4^{th} layer in the CHF is more important compared to that in PEF(SCH) and PEF(SSP).

7.3.3 Performance of different filters

In Table 2 the performances of the three filters are displayed. The errors are averaged (spatially and temporally) rms of prediction error for the SSH and two velocity components u and v.

The results in Table 2 show that two filters PEF(SCH) and PEF(SSP) largely overperform the CHF, with a slightly better performance for the PEF(SSP). We note that as the PEF(SCH) is constructed on the basis of an ensemble of samples tending to the first dominant Schur vector, its performance must be theoretically better than that of the PEF(SSP). The slightly better performance of PEF(SSP) (compared to that of PEF(SCH)) may be explained by the fact that the best theoretical performance of PEF(SCH) can be obtained only if the model is linear, stationary and the number of PE samples in $En(SCH)$ at each iteration must be large enough. The ensemble size of $En(SCH)$ in the present experiment is too small compared with the dimension of the MICOM model. It is therefore possible that in such situations, a more efficient way to simulate the PE samples is to perturb randomly all components of the system state. Detailed study of this question is of undoubted interest and is left for a future study.

Table 3: Error reduction (rms prediction error)

	$\frac{chf-pef(sch)}{chf}$	$\frac{chf-pef(ssp)}{chf}$	$\frac{pef(sch)-pef(ssp)}{pef(sch)}$
ssh	37 %	43 %	9 %
u	30 %	34 %	6 %
v	26 %	30 %	5 %

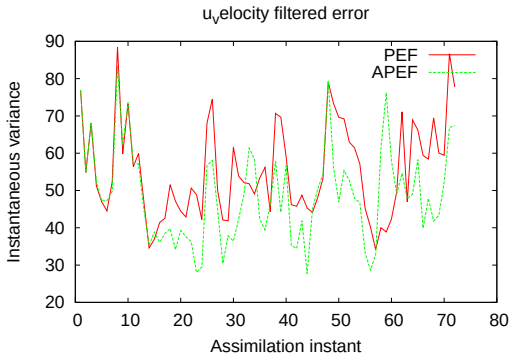

Figure 12. Filtered errors for the u-velocity component estimate, resulting from PEF and APEF (AF based on PEF). Optimization is performed by SPSA. By tuning the parameters in the filter gain, one can improve considerably the performance of the PEF

Error reductions, produced by different filters, are presented Table 3. For example, in average, the SSH forecast errors in PEF(SCH) and PEF(SSP) are 37 % and 43 % lower than that produced by the CHF.

For the improvement of performance of the AF compared to its nonadaptive version, see [12]-[14]. As illustration, Figure 12 displays the filtered errors for the u-velocity component estimates at the surface, produced by the PEF(SCH) and APEF respectively.

The APEF is an AF, which is an adaptive version of the PEF. Here the tuning parameters are optimized by the SPSA method. From the computational point of view, the SPSA requires much less time integration and memory storage compared with the traditional AE method. At each assimilation instant, we have to make only two integrations of the MICOM for approximating the gradient vector. From Figure 12 one sees that the adaptation allows to reduce significantly the estimation errors produced by the PEF. Mention that the effect of adaptation is more important if the corresponding nonadaptive version is less performant as it happens, for example, with the CHF.

8 Conclusion

In this paper a simple algorithm for estimating the elements of an unknown matrix as well as the way to decompose the estimated matrix into a product of two matrices, under the condition that only the matrix-vector product is accessible, has been proposed. This approach is beneficial for manipulating matrices of high dimensions encountered frequently in solving filtering and estimation engineering problems.

As it is seen, the proposed algorithm is simple to implement, it requires two times matrix-vector product, one from a nominal state, another from a perturbed state, to generate one sample-estimate for all elements of the unknown matrix, independently on its dimensions. The final estimate is obtained by an averaging procedure. For high dimensional systems, despite the fact that the estimated matrix cannot be stored directly in element-wise form, it is possible to manipulate this matrix by considering it as a collection of two ensembles of vectors, one consists of perturbed output vectors, another - from vectors of random perturbations. For the other purposes (storage compression, seeking directions of most rapid growth of prediction error ...), based on this algorithm, the SPSA procedure can be applied to solve different numerical problems like SVD decomposition, Nearest Kronecker Problem (NKP) of high dimension, parameter estimation ... in a simple and efficient way, compared with the classical algorithms (see the algorithms in [11] for solving NKP, for example).

Although the theoretical results on the convergence of the algorithm are established, it would be emphasized that for high dimensional systems, such algorithm will be efficient only if the estimated matrix has a sparse structure. Fortunately, such type of structure takes a place in the majority of the numerical systems resulting from discretization of the system of partial differential equations.

Numerical experiments in section 6 (low and moderate systems) and section 7 (high dimensional system) well illustrate the theoretical results and the efficiency of the proposed algorithms.

References:

[1] T.W. Anderson, *An Introduction to Multivariate Statistical Analysis*. Wiley-Interscience, London, 2003.

[2] J. Bai and S. Shi, Estimating High Dimensional Covariance Matrices and its Applications, *Annals of Economics and Finances*, 12-2, 2011, pp. 199-215.

[3] T.A. Beu, *Introduction to numerical programming*, Taylor and Francis Group, 2015.

[4] M. Cooper and K. Haines, Altimetric assimilation with water property conservation. *J. Geophys. Res.*, 101, 1996, pp. 1059-1077.

[5] R. Daley, The effect of serially correlated observation and model error on atmospheric data assimilation. *Monthly Weather Review*, 120, 1992, pp. 165-177

[6] P. Del Moral, Non Linear Filtering: Interacting Particle Solution. *Markov Processes and Related Fields*, 2 (4), 1996, pp. 555-580.

[7] P. Del Moral, A. Doucet and A. Jasra, On Adaptive Resampling Procedures for Sequential Monte Carlo Methods. *Bernoulli*, 18 (1), 2012, pp. 252-278, doi:10.3150/10-bej335.

[8] N.J. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing*, IEE Proceedings, ISSN 0956-375X, F 140 (2), 1993, pp. 107-113.

[9] G. Evensen, *Data assimilation : The ensemble Kalman filter*, Springer, Berlin, 2007.

[10] I. Fukumori and P. Malanotte-Rizzoli, An approximate Kalman filter for ocean data assimilation: An example with an idealized Gulf Stream model. *J. Geophys. Resear.*, V.100, Issue C4, 1995, pp. 6777-6793.

[11] G.H. Golub and C.F. van Loan, *Matrix Computations*, Cambridge University Press, 1996.

[12] H.S. Hoang and R. Baraille, Prediction error sampling procedure based on dominant Schur decomposition. Application to state estimation

in high dimensional oceanic model, *Applied Mathematics and Computation*, Vol. 218, No 7, 2011, pp. 3689-3709.

[13] H.S. Hoang, R. Baraille and O. Talagrand. On the design of a stable adaptive filter for state estimation in high dimensional system. *Automatica*, Vol 37, No 3, pp. 2001, pp. 341-359.

[14] H.S. Hoang and R. Baraille, On Efficiency of Simultaneous Perturbation Stochastic Approximation Method for Implementation of an Adaptive Filter. *Comput. Tech. Appl.*, 2, 2011, pp. 948-962.

[15] H.S. Hoang and R. Baraille, A Low Cost Filter Design for State and Parameter Estimation in Very High Dimensional Systems, *Proceedings of the 19th IFAC World Congress*, Vol. 19, Part 1, 2014, pp. 3256-3261.

[16] H.S. Hoang and R. Baraille, Adaptive filter based innovation approach for state and parameter estimation, *J. WSEAS Trans. on Syst.*, Vol 15, 2016, pp. 197-206.

[17] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. New York, Academic Press, 1970.

[18] T. Kailath, An Innovations Approach to Least-Squares Estimation, Pt. I: Linear Filtering in Additive Noise, *IEEE Trans. Autom. Contr.*, 13(6), 1968, pp. 646-655.

[19] R.E. Kalman, A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, V. 82, D, 1960, pp. 35-45.

[20] N. El Karoui, Operator norm consistent estimation of large dimensional sparse covariance matrices. *Annals of Statistics*, 36, 2008, pp. 2717-2756.

[21] O. Ledoit and M. Wolf, A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2), 2004, pp. 365-411.

[22] E. Levina, A.J. Rothman and J. Zhu, Sparse estimation of large covariance matrices via a nested lasso penalty. *Annals of Applied Statistics*, 2, 2007, pp. 245-263.

[23] E.N. Lorenz, Deterministic non-periodic flow. *J. Atmos. Sci.*, 20, 1963, pp. 130-141.

[24] R. Muirhead, *Aspects of Multivariate Statistical Theory*. Wiley, London, 2005.

[25] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. A. Yorke, A local ensemble Kalman filter for atmospheric data assimilation, *Tellus A*, 56, 2004, pp. 415-428.

[26] O. Pannekoucke, L. Berre and G. Desroziers, Background-error correlation length-scale estimates and their sampling statistics. *Quarterly J. Royal Meteorological Society*, V. 134, Iss 631, 2008, pp. 497-508.

[27] B.T. Polyak and A.B. Juditsky, Acceleration of stochastic a pproximation by averaging. *SIAM Journal on Control and Optimization*, 30(4), 1992, pp. 838-855.

[28] G. Sewell, *The numerical solution of ordinary and partial differential equations*, Acad. Press, 1988.

[29] C.S. Spall (1998). An Overview of the Simultaneous Perturbation Method for Efficient Optimization. *Johns Hopkins Apl Tech. Digest*, V. 19, No 4, 1998, pp. 482-492.

[30] G.W. Stewart and Ji guang Sun, *Matrix Perturbation Theory*. Academic Press, Boston, 1990.

[31] O. Talagrand and P. Courtier, Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. *Quart. J. Roy. Meteor. Soc.*, 113, 1987, pp. 1311-1328.