UDC 004.75

DATA ASSIMILATION USING KALMAN FILTER TECHNIQUES

Gabriel Dimitriu, Rodica Cuciureanu

University of Medicine and Pharmacy "Gr. T. Popa" Iasi, Faculty of Pharmacy,
Department of Mathematics and Informatics, Department of Food and Environment Chemistry
16 Universitatii street, 700115 Iasi, Romania
E-mail: dimitriu@umfiasi.ro, rcuciur@lycos.com

Kalman filtering represents a powerful framework for solving data assimilation problems. Of interest here are the low-rank filters which are computationally efficient to solve large scale data assimilation problems. The low-rank filters are either based on factorization of the covariance matrix (RRSQRT filter), or approximation of statistics from a finite ensemble (ENKF). A new direction in filter implementation is the use of two filters next to each other of the same form or hybrid (POENKF). The factorization approach is based on the linear Kalman filter which can be extended towards nonlinear models. In this paper, the background, implementation and performance of some common used low-rank filters is discussed. Numerical results are presented.

Introduction

Originally designed for guidance problems, the Kalman filter ([5]) has a long history of merging models and measurements in electrical engineering and control. The growing availability of cheap computing power during the last decade made the filter approach feasible for large geophysical models too. Kalman filtering represents a powerful framework for solving data assimilation problems ([3]). For the implementation of a Kalman filter the evolution of the state and observation of measurements can be described with the stochastic system:

$$x^t(k+1) = A(k)x^t(k) + \eta(k), \qquad y^o(k) = H(k)'x^t(k) + \nu(k), \tag{1}$$

with $x^t(k) \in IR^n$ the true state vector at time $t(k)$, $A(k)$ a deterministic model, $\eta(k) \in IR^n$ a Gaussian distributed model error (zero mean, covariance Q), and $y^o(k) \in IR^r$ a vector of observations with $\nu(k)$ the representation error (Gaussian with zero mean and covariance R). Indices 't', 'o', and later on 'f' and 'a' refer to true, observed, forecasted and analyzed entities respectively.

The goal of the filter operations is to obtain the mean $\hat{x}^a$ and covariance P^a for the probability density of the true state. The propagation of the covariance matrix is the most expensive part in the full rank filter. To avoid this problem, Bierman ([1]) proposed to write the equations for the Kalman filter using the factorization $P = SS'$. Numerical inaccuracies made in computation and storage of the matrix S will never affect the property of positive definiteness of P.

In order to obtain the Kalman filter in square root form, apart from the previous factorization $P = SS'$ for the covariance of the true state, we also introduce the factorizations $Q = TT'$ and $R = UU'$ for the covariance of the forecast and representation error, respectively. Further, a matrix $\Psi' = H'S$ is introduced for the mapping of the forecast covariance root to the observation space.

This study presents mathematical aspects of some Kalman filters in factorized form, together with numerical results obtained by applying such filters to data assimilation problems. The paper is organized as follows. In section 2 we briefly describe some factorized filters: Reduced Rank Square Root (RRSQRT) filter, Partially Orthogonal Ensemble Kalman (POENK) filter and its variant (COFFEE), also including the Ensemble Kalman filter. In the following section, the performance of the various algorithms is illustrated by numerical tests carried out with an advection diffusion model application. The last section contains some concluding remarks.

1. Description of some factorized filters

1.1 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:

$$\hat{x}^f(k+1) = A\hat{x}^a(k) \tag{2}$$

$$S^f(k+1) = [AS^a(k)\ ,\ T(k)] \tag{3}$$

$$\Psi = H'S^f(k+1)\,, \quad K = S^f(k+1)\Psi[\Psi'\Psi + R(k+1)]^{-1} \tag{4}$$

$$\hat{x}^a(k+1) = \hat{x}^f + K(y^o(k+1) - H\hat{x}^f(k+1)), \quad S^a(k+1) = S^f(k+1)[I - \Psi(\Psi'\Psi + R(k+1))^{-1}\Psi']^{1/2} \tag{5}$$

$$V\Lambda V' = S^a(k+1)'S^a(k+1), \quad \tilde{S}^a(k+1) = S^a(k+1)\tilde{V}. \tag{6}$$

The algorithm is initialized with an empty covariance square root; new columns are added every time step due to the introduction of system noise (3). For each of the m modes stored in S, the forecast of the covariance requires one evaluation of the model A. The analysis steps (4)-(5) 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 (6). With the introduction of system noise in (3), 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 $\tilde{V}$ contains the eigenvectors of $(S^a)'S^a$ corresponding with the largest m eigenvalues. The new matrix $S^a\tilde{V}$ is 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 (3), when for each mode the model should be called once. The reduction should therefore reduce the number of modes as far as possible.

2.2. Ensemble filter. The RRSQRT filter is based on the factorization of the covariance matrix. The ENsemble Kalman Filter (ENKF) is based on convergence of large numbers. The ensemble filter was introduced in for assimilation of data in oceanographic models ([2]). The basic idea behind the ensemble filter is to express the probability function of the state in an ensemble of possible states $\{\xi_1,...,\xi_N\}$. 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:

$$\xi_k^f(k+1) = M(\xi_k^a(k)) + \eta_k(k) \quad , \qquad \eta_k(k) \sim N(0, Q(k)), \tag{7}$$

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:

$$\xi_j^a(k+1) = \xi_j^f(k+1) + K(y^o(k+1) + v_j - H'\xi^f(k+1)) \quad , \qquad v_j \sim N(0, R(k+1)). \tag{8}$$

The vectors v_j denote samples of the representation error, drawn from a random generator.

2.3. Hybrid approaches: POENK and COFFEE filters. A new direction in implementation of low-rank filters is the use of two filters next to each other. The combination should compensate for errors made in one or both of the individual filters.

The *Partially Orthogonal Ensemble Kalman* filter (POENK) runs a RRSQRT filter next to an ENKF. The basic idea is to let the RRSQRT part compute the bulk of the covariance structure, described in the first modes. The ENKF part should account for the truncation error, by introducing directions in the covariance matrix that have been lost during the reduction. This procedure incorporates the advantages of both filter types, and accounts for their major disadvantages. Ensemble filters suffer from a lack of convergence; many ensembles are required before sample mean and correlations are stable. A variant of POENK filter is the *Complementary Orthogonal subspace Filter For Efficient Ensembles* (COFFEE) algorithm (see [4] for details).

2. Numerical comparative study with an advection diffusion model

The performance of three types of low-rank filters (RRSQRT, ENK, and POENK filters) was tested during a filter experiment with simulated data. We used some slightly modified Matlab routines carried out by Verlaan M. ([4]). As a model under investigation, we consider the 2-D advection diffusion equation:

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} + v\frac{\partial c}{\partial y} = \nu\frac{\partial^2 c}{\partial x^2} + \nu\frac{\partial^2 c}{\partial y^2} \tag{9}$$

with a square domain and zero initial conditions. The concentration at the boundary is zero for inflow. We used a backward Lagrangian scheme to discretize these equations on a 30×30 grid. The velocity field, considered known and constant in time is similar to that of the well-known Molenkamp test. Some twin experiments were carried out. A reference solution 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 {0.2, 0.1, 0.1, 0.2, 0.2} respectively.

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 location {(3, 10), (12, 4), (27, 18), (14, 11), (22, 3), (10, 10), (14, 21), (22, 11), (6, 24)}. The decays per step are $\{\gamma_1,\ldots,\gamma_5\} = \{0.8, 0.9, 0.9, 0.8, 0.8\}$. Finally, white observational noise with variance 0.1 is added to the true concentrations. To compare the performance of the different filters with each other, the root mean square (RMS) errors were computed:

$$RMS = \sqrt{1/(M^2 K_{ts}) \sum_{m,n,k} (c_{m,n}(k) - \hat{c}_{m,n}(k))^2} \qquad (10)$$

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

If the RMS errors of all experiments are compared (see Table 1), the RRSQRT algorithm seems to be the most efficient choice 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. The slow convergence of the ENKF filter is illustrated (see Figure 1) by the large spread in the corresponding STD errors. These results show that the convergence of the RRSQRT filter is much faster than the convergence of the ensemble filter. In the Figures 2-5 the concentration fields of the truth-run and the reference-run are shown after K_{ts} timesteps. The +-signs indicate measurement locations and the diamond-signs the locations of the emissions. It can be seen clearly that the true fields are perturbated with time-varying fluctuations, while the reference solutions only contains a steady emission which is advected and spreading smoothly.

Concluding remarks

In this study four different low-rank filters have been implemented around an 2-D advection diffusion model: based on factorization (RRSQRT filter), ensemble statistics (ENKF), or on hybrid approaches (POENKF combining a RRSQRT and ENKF filter, and its variant COFFEE filter). All four 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 a RRSQRT filter with comparable costs already converged. As a consequence, also the POENKF filter suffers from the statistical noise in its ENKF part. Due to the fast convergence and accurate results reached with the RRSQRT filter, the benefit of additional random directions in the gain of the POENKF is limited. For comparable costs, the RRSQRT filter produces stable and more accurate results than ENKF or POENK and COFFEE filters.

Acknowledgement

The work is partially supported by NATO Collaborative Linkage Grant 980505 / NATO project “Impact of future climate changes on pollution levels in Europe”.

Table 1

Modes/ ensemble	Filter	RMS conc.	STD conc.	RMS noise	STD noise	Modes/ ensemble	RMS conc.	STD conc.	RMS noise	STD noise
	rrsqrt	0.352	0.323	2.065	1.971		1.269	0.212	5.670	1.642
	rrsqrt2	0.351	0.322	2.064	1.971		1.231	0.199	6.651	1.707
30/30	ensemble	0.441	0.240	2.219	1.799	**8/30**	0.441	0.240	2.219	1.799
	poenk	0.375	0.322	2.174	1.971		0.399	0.234	2.221	1.560
	coffee	0.354	0.323	2.069	1.972		0.443	0.234	2.326	1.565
	rrsqrt	0.351	0.322	2.065	1.970		1.398	0.194	6.475	1.527
	rrsqrt2	0.351	0.322	2.064	1.970		1.508	0.195	6.925	1.528
25/30	ensemble	0.441	0.240	2.219	1.799	**6/30**	0.441	0.240	2.219	1.799
	poenk	0.374	0.322	2.173	1.970		0.379	0.200	2.178	1.446
	coffee	0.353	0.323	2.071	1.972		0.410	0.199	2.213	1.447
	rrsqrt	0.350	0.319	2.063	1.966		1.032	0.130	5.220	1.392
	rrsqrt2	0.348	0.319	2.058	1.966		1.017	0.129	5.363	1.390
20/30	ensemble	0.441	0.240	2.219	1.799	**4/30**	0.441	0.240	2.219	1.799
	poenk	0.372	0.319	2.171	1.967		0.399	0.142	2.235	1.284
	coffee	0.355	0.321	2.075	1.969		0.427	0.142	2.254	1.304
	rrsqrt	0.397	0.308	2.126	1.940		0.933	0.078	5.505	1.244
	rrsqrt2	0.409	0.306	2.175	1.938		0.507	0.082	2.893	1.214
15/30	ensemble	0.441	0.240	2.219	1.799	**2/30**	0.441	0.240	2.219	1.799
	poenk	0.386	0.308	2.187	1.935		0.440	0.091	2.222	1.152
	coffee	0.360	0.306	2.073	1.926		0.478	0.088	2.304	1.188
	rrsqrt	0.658	0.253	2.981	1.723		0.862	0.044	4.894	1.085
	rrsqrt2	0.536	0.252	2.573	1.708		0.541	0.057	2.881	1.159
10/30	ensemble	0.441	0.240	2.219	1.799	**1/30**	0.441	0.240	2.219	1.799
	poenk	0.391	0.258	2.209	1.698		0.471	0.052	2.182	1.020
	coffee	0.398	0.260	2.153	1.719		0.440	0.057	2.077	1.058

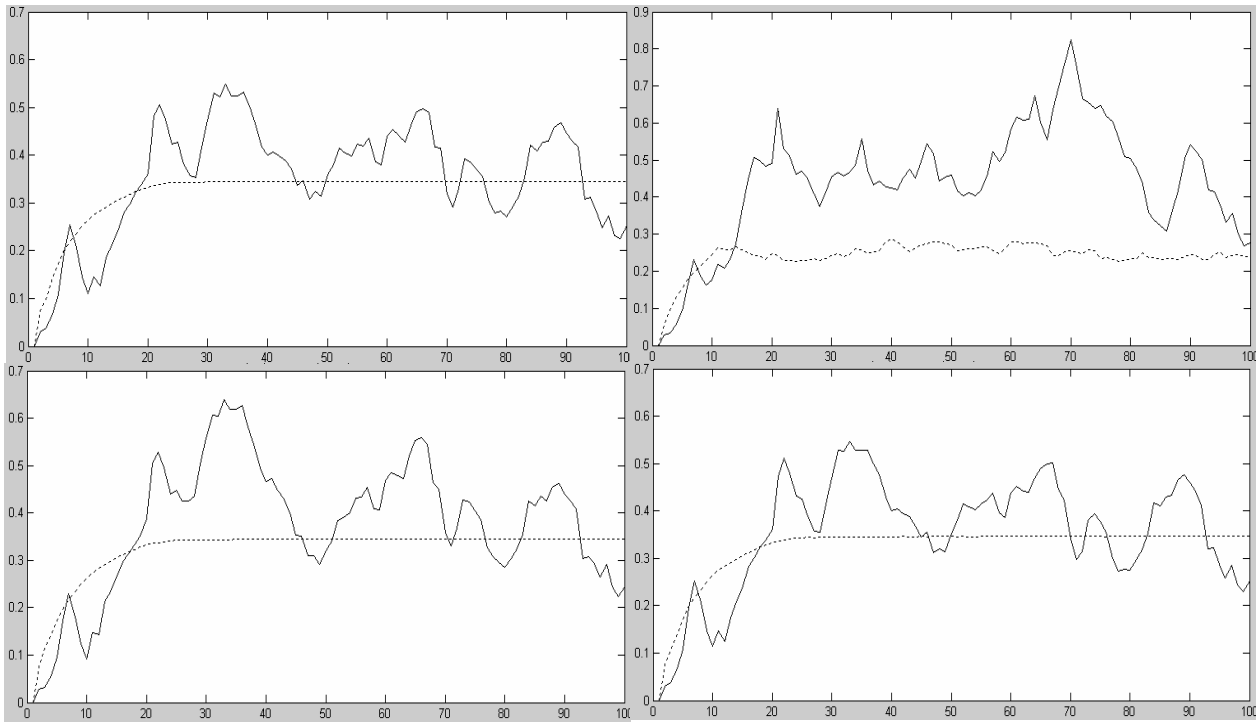

Fig. 1. RMS errors (line) and STD (dotted) of the concentrations for 30 modes and 30 ensemble members (RRSQRT and ENKF filters: upper row subplots, respectively; POENK and COFFEE filters: lower row subplots, respectively)

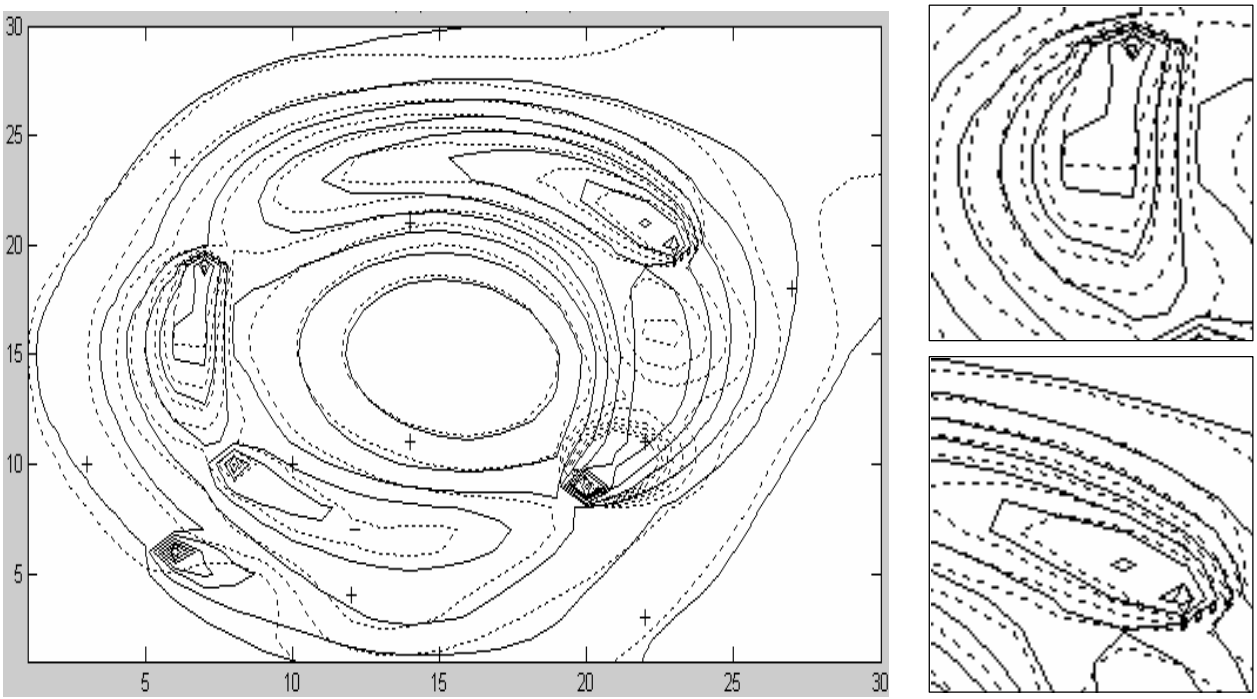

Fig. 2. The concentrations calculated with RRSQRT filter with $(q, N) = (6,30)$ at $k = 100$. Contours plotted with increment 1 for true concentrations '-' and filter solutions '…' (left plot). The subplots on the right side represent zoomed patterns of the assimilation results for the upper two pollution sources.

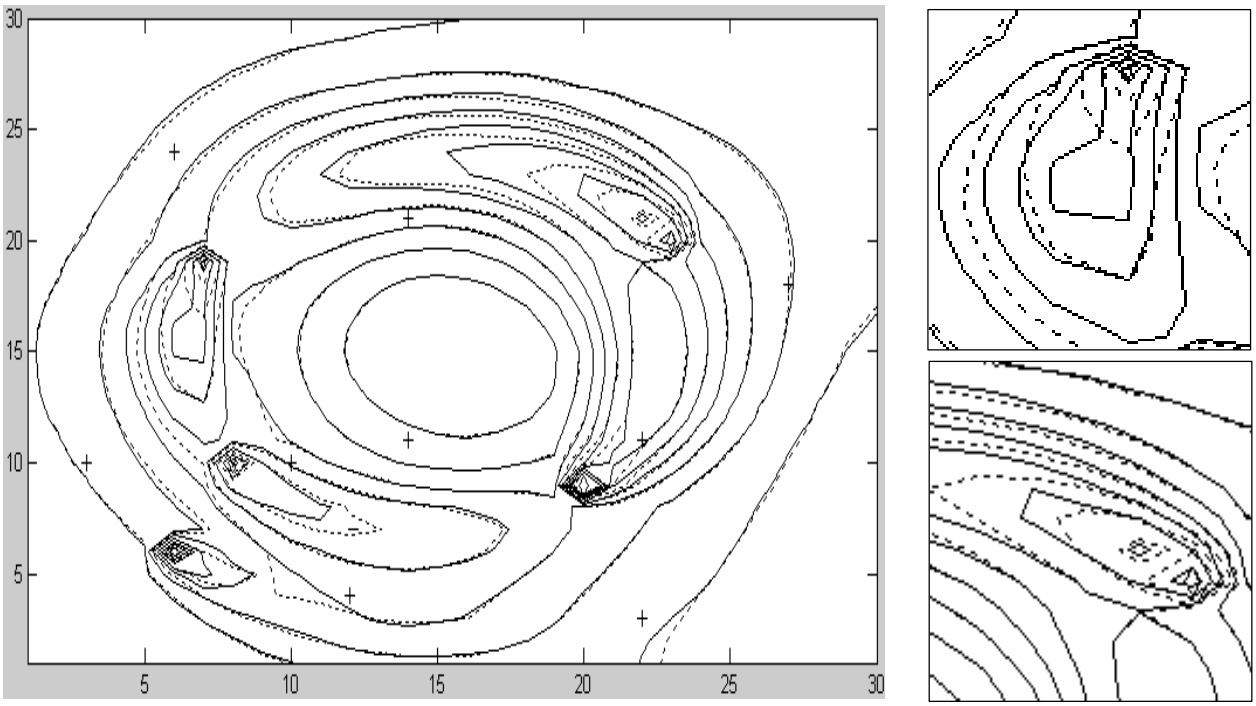

Fig. 3. The concentrations calculated with RRSQRT filter with $((q, N) = (30,30)$ at $k = 100$. Contours plotted with increment 1 for true concentrations (-) and filter solutions (…). The subplots on the right side represent zoomed patterns of the assimilation results for the upper two pollution sources.

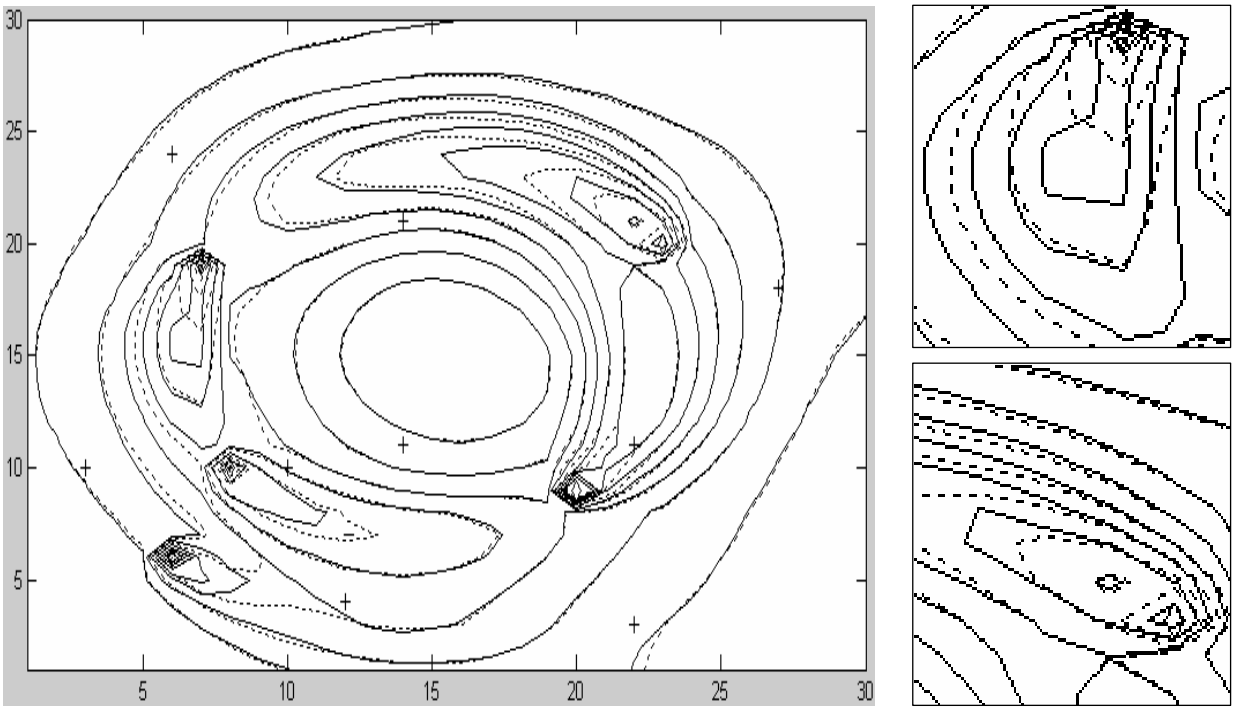

Fig. 4. The concentrations calculated with POENK filter with $((q, N) = (30,30)$ at $k = 100$. Contours plotted with increment 1 for true concentrations (-) and filter solutions (…).The subplots on the right side represent zoomed patterns of the assimilation results for the upper two pollution sources.

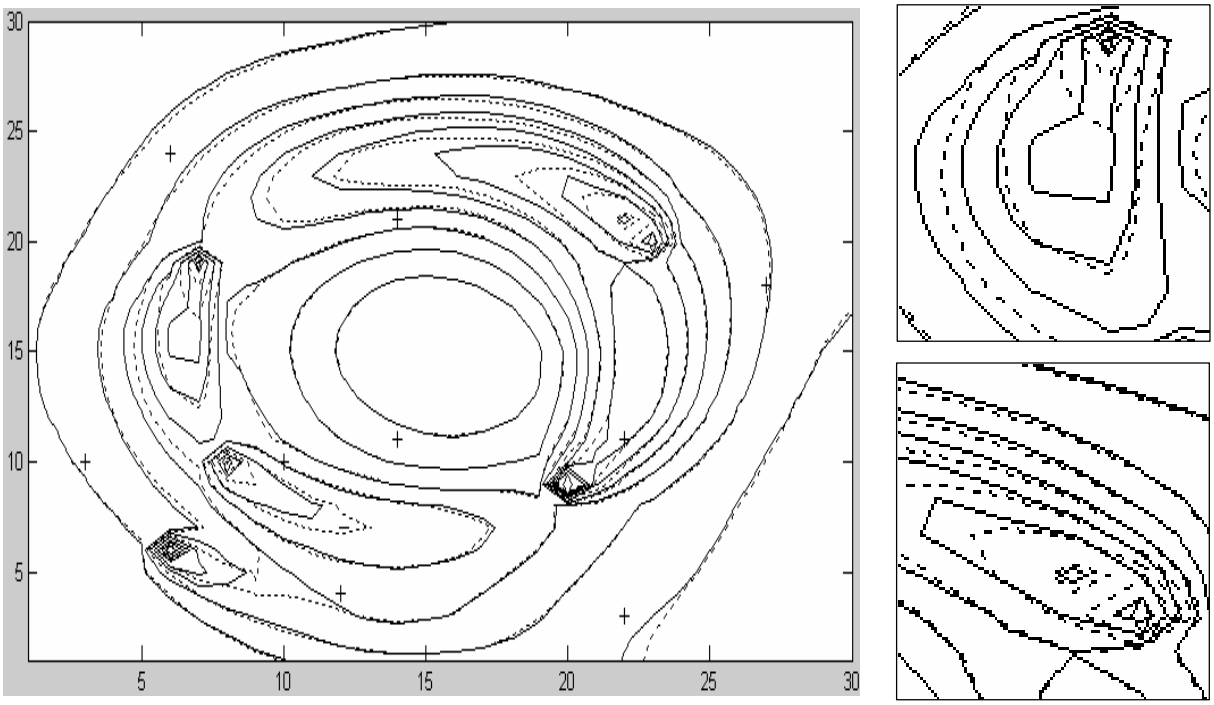

Fig. 5: The concentrations calculated with COFFEE filter with $(q, N) = (30,30)$ at $k = 100$. Contours plotted with increment 1 for true concentrations (-) and filter solutions (…). The subplots on the right side represent zoomed patterns of the assimilation results for the upper two pollution sources.

1. *Bierman, G.J.* (1977). *Factorization Methods for Discrete Sequential Estimation,* Vol. 128 of *Mathematical in Science and Engineering,* Academic Press, New York.
2. *Evensen, G.* (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.*, 99(C5): 10143-10162.
3. *Ghil, M. and P. Malanotte-Rizzoli* (1991). Data assimilation in meteorology and oceanography. Advances in Geophysics 33, 141-266. Academic Press, San Diego, Calif.
4. *Heemink, A., Verlaan, M., and Segers, A.* (2001). Variance reduced Ensemble Kalman filtering. Mon. Weather Rev., 129(7): 1718-1728.
5. *Kalman, R.E.* (1960). A new approach to linear filter and prediction theory. *J. of Basic Engineering*, 82D:35-45.