Centrum voor Wiskunde en Informatica

Kalman filtering for nonlinear atmospheric chemistry models:
first experiences

M. van Loon and A.W. Heemink

Kalman Filtering for Nonlinear Atmospheric Chemistry Models: first experiences

M. van Loon*

CWI

P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

A.W. Heemink

Delft University of Technology

Department of Applied Mathematics

Mekelweg 4, Delft, The Netherlands

ABSTRACT

The aim of the EU project RIFTOZ is to analyse regional differences in tropospheric ozone over Europe. One of the key activities within RIFTOZ therefore involves recovering ozone concentrations from available measurements. This will be done by running the atmospheric chemistry model LOTOS over the selected period using a data assimilation technique to incorporate the measurements. A commonly used data assimilation technique is the (extended) Kalman filter. This filter has proved to be very useful in many applications. However, the models involved in these applications are usually only weakly nonlinear, whereas atmospheric models, like LOTOS, are often highly nonlinear.

The paper presents first results on data assimilation with a highly nonlinear test model using the (extended) Kalman filter algorithm. The test model has been designed such that the essential characteristics of the LOTOS model, including stiff (photo-)chemistry, have been retained. Application of the standard algorithm for Kalman filtering is infeasible because of the huge computational and storage requirements. Instead, a reduced rank approximation of the covariance matrix is used, which reduces the computational burden to an acceptable amount of CPU time. Also attention is paid to reducing the number of noise parameters in the filter algorithm in order to further restrict the number of model evaluations that is required to solve the filtering problem. The results of the tests are very promising and show that Kalman filtering may be successfully applied to atmospheric chemistry models.

1. INTRODUCTION

The final goal of EU project RIFTOZ (Regional dIFferences in Tropospheric OZone) is to produce a reliable data set of ozone concentrations for the summer of 1997 and to analyse why ozone behaves differently in different parts of Europe, using this data set. Of course ground measurements as well as satellite observations will be available. However, these ground level data are irregularly distributed over the horizontal domain whereas the satellite observations will only be available once in three days and only in cloud-free situations with a low aerosol loading. Therefore, a model simulation with the model LOTOS will be performed. In order to incorporate the available measurements, data assimilation will be performed. The present paper examines whether the (extended) Kalman filter algorithm is a suitable data assimilation technique for atmospheric chemistry models. This filter technique has proved to be very useful in many applications. However, the models involved in these applications are usually only weakly nonlinear, whereas atmospheric chemistry models, like LOTOS, are often highly

---

*Corresponding author. E-mail: vanloon@cwi.nl

nonlinear. In order to test the Kalman filter for this kind of applications, a 3D atmospheric chemistry test model is used. This test model has been designed such that the essential characteristics of the LOTOS model, including stiff chemistry, have been retained. Attention will be paid to ways to restrict the computational work as much as possible. In that connection, it will be shown how the operator splitting, that is applied in LOTOS and also in the test model, enables further restriction of the computational work. The implementation of the filter is based on the Reduced Rank Square Root algorithm, presented in [7].

The paper is organized as follows. In Section 2 a short description of the Kalman filter for linear and non linear models is given. Section 3 describes the Reduced Rank Square Root (RRSQRT) implementation. In Section 5 the test model and the data assimilation experiments are defined and their results are presented. The conclusions drawn from the experiments are summarized in Section 7.

2. THE KALMAN FILTER

2.1 The linear case

Suppose we have a linear deterministic model

$$x^{k+1} = A^k x^k, \quad x^k \in R^n, \ A^k \in R^{n \times n}, \tag{2.1}$$

that describes the approximate (discrete) time behavior of a state vector $x(t)$ where x^k denotes the approximation for $x(t^k)$ with $t^k = t^{k-1} + \tau_k$. The time interval between two successive approximations need not be constant. Suppose also observations y^k of (linear combinations of the elements of) the state vectors at (some of) the time levels $t = t^k$ are available. Hence,

$$y^k = C^k x^k, \quad y^k \in R^m, \ C^k \in R^{m \times n}, \tag{2.2}$$

In practice, models are often far from perfect, not so much because the state vectors are numerical approximations of the solution of the model equation but in particular because often a number of parameters, including the initial and boundary conditions, are unknown or only approximately known. Also, measurements will not be perfect. An observation error will sometimes be present. In addition, an observation may be not be representative for the volume considered by the model. To obtain an optimal estimate of the state using both sources of information, the model and the observations, a filter technique is used. In order to be able to use a filter technique stochastic descriptions of the model and the measurements are necessary instead of deterministic ones. Therefore, we replace the model (2.1) by its stochastic extension

$$x^{k+1} = A^k x^k + F w^k, \quad w^k \in R^p, \ F^k \in R^{n \times p}, \tag{2.3}$$

where x now is a stochastic variable. The vector w^k represents the system noise with zero mean and covariance matrix $Q^k = E[w^k (w^k)^T]$. A discussion on how to identify the system noise is postponed until later. Similarly, the measurements are represented as

$$y^k = C^k x^k + v^k, \quad v^k \in R^m, \tag{2.4}$$

where v^k denotes the measurement noise. The measurements are supposed to be independent, i.e. the covariance matrix $R^k = E[v^k (v^k)^T]$ is diagonal.

An optimal estimate $\hat{x}^{k+1}$ of x^{k+1} in (2.3) is given by the Kalman filter equations

$$\hat{x}_f^{k+1} = A^k \hat{x}^k \tag{2.5}$$

$$P_f^{k+1} = A^k P^k (A^k)^T + F^k Q^k (F^k)^T \tag{2.6}$$

$$K^{k+1} = P_f^k (C^k)^T \left[C^{k+1} P_f^{k+1} (C^{k+1})^T + R^k \right]^{-1} \tag{2.7}$$

$$\hat{x}^{k+1} = \hat{x}_f^{k+1} + K^{k+1} (y^{k+1} - C^{k+1} \hat{x}_f^{k+1}) \tag{2.8}$$

$$P^{k+1} = P_f^{k+1} (I - K^{k+1} C^{k+1}) \tag{2.9}$$

The first step, given by equation (2.5), is just a model evaluation. In the second step (2.6), the covariance matrix $P^k = E[(x^k - \hat{x}^k)(x^k - \hat{x}^k)^T]$ is updated. The third and fourth step form the measurement update. In the last step, the covariance matrix is updated again because of measurement update of the state vector.

From the filter equations, it is seen that the initial estimate $\hat{x}^0$ and the covariance matrix $P^0 = E[(x^0 - \hat{x}^0)(x^0 - \hat{x}^0)^T]$ for the initial conditions must be specified. Since usually nothing is known about the covariances, P^0 is often set to zero. Note that the covariance matrix P is always symmetric and positive semi definite.

It is also seen from the filter equations, that the computational work involved with the measurement update is quite large. In the first place, the computation of $A^k P^k (A^k)^T$ in (2.6) is $2n$ times as expensive as one model evaluation. For small problem dimensions, this may not be much of a problem, but for large n this factor is infeasible. In the Reduced Rank Square Root algorithm (see Section 3) this problem is circumvented. In the second place, in case the number of measurements m is large, computing the Kalman gain K^{k+1} is also expensive, since it involves n times the solution of a $m \times m$ matrix-vector equation.

Fortunately, the computational effort for the measurement update can significantly be reduced in case the measurements are uncorrelated. In that case it can be shown (see e.g. [2, 6]) that the following iterative procedure may be applied (dropping the superscripts $k + 1$):

$$\left. \begin{array}{rl} a_i = & (c_i P_{i-1} c_i^T + R_{ii})^{-1} \\ b_i = & (1 + \sqrt{a_i R_{ii}})^{-1} \\ K_i = & a_i P_{i-1} c_i^T \\ P_i = & P_{i-1} - a_i P_{i-1} c_i^T c_i P_{i-1} \\ x_i = & x_{i-1} + K_i (y_i - c_i x_{i-1}) \end{array} \right\} \quad i = 1, \ldots, m \tag{2.10}$$

with $P_0 = P_f^{k+1}$ and $x_0 = \hat{x}_f^{k+1}$. The row vector c_i is equal to the ith row of the matrix C^{k+1}. As final result we have $P^{k+1} = P_m$ and $x^{k+1} = x_m$. From the measurement update (2.10) the following can be seen. Suppose $C_i^{'T}$ is the unit vector ε_k, i.e. the kth entry is one and all others are zero. It is then easily seen that the kth element of the Kalman gain (which is a vector) K_i is a nonnegative number bounded by one, so that the kth entry of x_i will be closer to the measurement than the kth entry of x_{i-1}. Further, if the different measurement sites are uncorrelated (the corresponding entries of P_i are zero), this will also be the case after the measurement updates.

2.2 The nonlinear case

We now suppose that the model is nonlinear and can be written as

$$x^{k+1} = f(t, x^k, w^k). \tag{2.11}$$

The measurements are still supposed to be of the form (2.4). The filter equations (2.6)-(2.9) may now be applied to the linearized model with

$$A^k = \frac{\partial f}{\partial x^k}(\hat{x}^k, 0), \quad F^k = \frac{\partial f}{\partial w^k}(\hat{x}^k, 0). \tag{2.12}$$

In the first step (2.5) of course the model itself is applied, i.e.

$$\hat{x}_f^{k+1} = f(t, \hat{x}^k, 0).$$

The procedure for the non linear case is called the Extended Kalman Filter (EKF). Even though we now have a procedure that deals with non linear models, for models with a large dimension n it is computationally still infeasible to apply the EKF. The computational burden has even become much havier than in the linear case, because now two large Jacobian matrices have to be computed.

2.3 Filter divergence

Filter divergence occurs if the computed error variances (in the measurement points) are small whereas the residues, i.e. the difference between the measured and computed solution, are large. Filter divergence often happens when the dynamics of the model are (too) far from reality. For more details we refer to [4], where also a nice example of filter divergence in the linear case is worked out. In atmospheric models filter divergence may also occur. For example, the wind fields may be in error, the chemical mechanism does not describe the true chemical process accurately etc.

In the literature various ways are described to try to prevent filter divergence. Here, we briefly discuss some possibilities. The emphasis is on errors caused by deviations between the true and modeled wind fields. Note that in atmospheric models advection is to a large extent responsible for the horizontal coupling between grid cells. This implies that correlations between grid cells are mainly caused by horizontal advection.

- Memory reduction. The idea behind this is that information present in the filter is not valid any more after some time, either because of modelling errors or due to the nonlinearity of the model. In the context of deviations in the wind field the information is, so to speak, just transported into the wrong direction and hence becomes more and more inaccurate. Therefore it makes sense to only take into account recent information and to "forget" old information. A systematic approach is given in [4], section 7.10. This approach, however, is rather expensive because the filtering procedure has to be applied twice and, in addition, two extra matrix inversions are involved. Therefore, we do not consider this approach for the present application. A very simple approach to achieve memory reduction is followed by Cañizares et al. [1]: after each time step the covariance matrix is multiplied by a memory reduction factor $\zeta < 1$. In this way old information becomes exponentially less dominant. There is, however, the risk that "false certainty" is introduced into the system, for (co)variances are decreased by the reduction factor. Hence ζ may not be chosen too small.

- Overweighting recent data. This approach is based on a similar assumption as memory reduction. By multiplying the standard deviation of the measurements and the covariances by an appropriate factor, it is achieved that the influence of old information decreases exponentially in time. Instead of equation (2.7) we apply

$$K^{k+1} = P_f^k (C^k)^T \left[C^{k+1} P_f^{k+1} (C^{k+1})^T + e^{-\xi} R^k \right]^{-1} \tag{2.13}$$

and instead of (2.9)

$$P^{k+1} = e^\xi P_f^{k+1} (I - K^{k+1} C^{k+1}) \tag{2.14}$$

with $\xi > 0$, see [4] for details.

- Increase the noise input, thus increasing the uncertainty in the model. This in turn causes the filter to relatively more weight to the measurements.

  The first possibility to increase the noise input is simply add a noise vector W with specified statistics directly to the stochastic extention of the model equation (2.3). In the Kalman filter equations we then only have to add the covariance matrix $E[WW^T]$ to the right hand side of (2.6). The covariance between two grid points P_1 and P_2 is then often described by

  $$Cov(P_1, P_2) = \sigma^2 \exp\{-\alpha \cdot \text{dist}(P_1, P_2)\},$$

  where σ and α need to be specified. Dist(P_1, P_2) denotes a measure for the distance between the points P_1 and P_2.

  The second possibility to increase the noise input is to introduce a few parameters that describe the the deviations in e.g. the wind field. The vector w in (2.3) is then extended with these

parameters. In contrast to the first possibility, this acts on directly on the part of the dynamics in which errors are assumed. A complication, however, is that adding noise to wind fields should preserve the divergence-freeness.

3. THE REDUCED RANK SQUARE ROOT ALGORITHM

It is convenient not to work with the convariance matrix P, but with its square root S (i.e. $SS^T = P$). A major advantage is that P then always is semi positive definite, no matter how S is approximated. Another important advantage is that S need not have n columns, but can have $q < n$ columns. The trivial example is $P = 0$ in which case one column with all entries zero suffices. This property will be exploited in the Reduced Rank Square Root algorithm (RRSQRT) that is presented in [7]. For the moment being we just assume $S \in R^{n \times q}$. The algorithm consists of three steps.

1. Time step

The time step performs the time propagation of the state vector x^k and the square root of the covariance matrix.

$$\hat{x}_f^{k+1} = A^k \hat{x}^k \tag{3.1}$$

$$S_f^{k+1} = [A^k S^k, F^k (Q^k)^{1/2}], \tag{3.2}$$

where the notation $[A_1, A_2]$ means that A_1 is extended with the columns of A_2.

2. Reduction step

Since due to (3.2) the number of columns of S will grow rapidly, the corresponding computation time for the filter evaluations will grow as well. Therefore, the number of colums of S_f^{k+1} is reduced to $q \ll n$. Supposing that S^k has already q columns, S_f^{k+1} will have $q + m$ columns, hence S_f^{k+1} has to be approximated such that it has only q columns. This is done by taking only the q leading singular values of $S_f^{k+1}(S_f^{k+1})^T$ into account. Let

$$S_f^{k+1} = U \Sigma V^T$$

denote the singular value decomposition of S_f^{k+1}. Then

$$(S_f^{k+1})^T S_f^{k+1} = V \Sigma^T \Sigma V^T,$$
$$S_f^{k+1}(S_f^{k+1})^T = U \Sigma \Sigma^T U^T,$$

from which we conclude that the first $(q + m)$ singular values of $S_f^{k+1}(S_f^{k+1})^T$ and $(S_f^{k+1})^T S_f^{k+1}$ are the same. Since the latter is only a $(q + m) \times (q + m)$ matrix, it is much more efficient to compute the singular value decomposition of $(S_f^{k+1})^T S_f^{k+1}$. The square root of $S_f^{k+1}(S_f^{k+1})^T$ is now given by $U \Sigma$ which is approximated by its first q columns, i.e. the last m entries of Σ are neglected. Since U is not known, we use the identity

$$U \Sigma = S_f^{k+1} V.$$

By multiplying both sides by V^T, it is easily seen that this equality holds, since $V V^T = I$.

3. Measurement step

Assuming that the measurements are uncorrelated, the iterative procedure may be applied. Formulated in terms of S, this procedure is given by

$$\left.\begin{array}{rl} H_i = & S_{i-1}^T c_i^T \\ a_i = & (H_i^T H_i + R_{ii})^{-1} \\ b_i = & (1 + \sqrt{a_i R_{ii}})^{-1} \\ K_i = & a_i S_{i-1} H_i \\ S_i = & S_{i-1} - b_i K_i H_i^T \\ x_i = & x_{i-1} + K_i(y_i - c_i x_{i-1}) \end{array}\right\} \quad i = 1, \ldots, m, \tag{3.3}$$

6

with $x_0 = \hat{x}_f^{k+1}$ and $S_0 = S_f^{k+1}$.

3.1 Approximating the Jacobian
Since really computing the Jacobian matrices according to (2.12) is far too expensive, we follow the approach proposed in [7]. Let s_i^k denote the ith column of S^k. Then the ith column of $A^k S^k$ may be approximated by

$$\frac{\partial f}{\partial x^k} s_i^k \approx \frac{f(x^k + \epsilon s^i, 0) - f(x^k, 0)}{\epsilon}. \tag{3.4}$$

This procedure requires only q instead of n model evaluations. In a similar way $F^k (Q^k)^{1/2}$ is approximated, which requires another m model evaluations.

4. Description of the test model
As horizontal domain we take an area of 3000 × 3000 km^2. In vertical direction, two layers of 500m depth each are modeled. On this domain we solve for each species $c_{i,j}$, denoting the concentration of species i in layer j the following equation.

$$\frac{\partial c_{i,j}}{\partial t} + \frac{\partial u_j c_{i,j}}{\partial x} + \frac{\partial v_j c_{i,j}}{\partial y} = D_j(c_{i,j}) + K_j(c_{i,1}, c_{i,2}) + F_i(c_{1,j}, \ldots, c_{s,i}) + E_{i,j}(x,y), \tag{4.1}$$

where (u_j, v_j) denotes the velocity field in layer j, D_j the deposition, K_j the vertical exchange between the two layers, $F_i(c)$ the chemical reactions and $E_{i,j}(x,y)$ the source term. The number of species taken into account is four, i.e. $s = 4$. The species are NO_2, NO, O_3 and OH.

4.1 Chemistry
The species are coupled by the chemical reactions. From the chemical mechanism in [5], we took the following reactions

$$
\begin{array}{lll}
NO_2 & \xrightarrow{k_1} & NO + O_3 \\
NO + O_3 & \xrightarrow{k_2} & NO_2 \\
NO_2 + OH & \xrightarrow{k_3} & NO_3^a \\
O_3 & \xrightarrow{k_4} & b_1 OH + b_2 O_3 \\
CO + OH & \xrightarrow{k_5} & a_1 O_3 \\
CH_4 + OH & \xrightarrow{k_6} & a_2 O_3
\end{array}
\tag{4.2}
$$

These reactions form the basis for many ozone chemistry models. For the corresponding reaction rates we refer to [5]. For the second order reactions, the rates from [5] are multiplied with $2.46 \cdot 10^{10}$ to obtain rates in $(ppb\,s)^{-1}$. For most of these reaction rates either the temperature T_k (in Kelvin) or the cosine of the solar angle $\cos\gamma$ is necessary. They are modeled by

$$
\begin{aligned}
T_k &= 293.1 + 5\cos(\frac{\pi}{12}tod - 12), \\
\cos\gamma &= 0.6\cos(\frac{\pi}{12}tod - 12),
\end{aligned}
$$

where *tod* denotes the time-of-day in hours. This choice for $\cos\gamma$ corresponds with a midsummer day at about 53^0 Northern latitude. If $\cos\gamma \leq 0$, the corresponding reaction rates are zero. The concentrations of CO and CH_4 are given a constant value of 150 ppb and 1700 ppb, respectively. The coefficients b_1 and b_2 are related through the relation $b_2 = 1 - b_2/2$. In [5] these parameters depend on the water vapour concentration. Here, they are taken constant and given the values 0.8 and 0.6, respectively.

4.2 Deposition

Depostion from layer one to the surface is modeled by

$$D_1(c_{i,j}) \;=\; -v_i^d c_{i,j},$$

where v_i^d denotes the net deposition velocity for species i. The deposition velocity for NO_2 is $2 \cdot 10^{-6}$ and for ozone $5 \cdot 10^{-6}$.

4.3 Vertical diffusion

The vertical exchange between the two layers is modeled by

$$K_1(c_{i,1}, c_{i,2}) \;=\; -K_2(c_{i,1}, c_{i,2}) \;=\; \alpha(tod)(c_{i,2} - c_{i,1}),$$

where $\alpha(tod)$ is chosen such that its value at 12:00h would cause the difference between $c_{i,1} - c_{i,2}$ to be decimated in 15 minutes, $\alpha(12) \approx 3.85 \cdot 10^{-4} s^{-1}$ if no other processes were present. A similar time behavior as for the temperature is taken for α

$$\alpha = 1.925 \cdot 10^{-4}(1 \;+\; \cos^{0.2}(\frac{\pi}{12}(tod - 12))) \; s^{-1}.$$

This parametrization is meant to represent the fact that the vertical mixing is strong during day time and weak at night. The cosine is raised to the power 0.2 in order to let the profile of α more look like a block function.

4.4 Emission

At four different locations in the domain NO emissions take place. Two of them take place in the layer 1, the other two (stronger) ones in the second layer. The locations and source strenghts will be specified in Section 4.7.

4.5 Initial and boundary conditions

The initial conditions are summarized in Table 1 For the inflow boundaries, time dependent boundary

	NO_2	NO	O_3	OH
layer 1	10	1	40	0
layer 2	20	0	30	0

Table 1: Initial conditions in ppb

conditions for NO_2 and O_3 are specified. In the first layer they are modeled by

$$NO_2 \;=\; 30 + 20\cos(\frac{\pi}{12}(tod - 12)), \quad O_3 \;=\; 20\cos(\frac{\pi}{12}(tod - 12))$$

and by

$$NO_2 \;=\; 30 + 15\cos(\frac{\pi}{12}(tod - 12)), \quad O_3 \;=\; 15\cos(\frac{\pi}{12}(tod - 12))$$

in the second layer. All other boundary conditions are zero.

4.6 The wind fields

For both layers, the well-known rotational wind field is taken, but with different centers of rotation and different scaling. They are given by

$$\begin{aligned}
u_1(x,y) &= 2\pi U_1(y - y_1) & v_1 &= -2\pi U_1(x - x_1) \\
u_2(x,y) &= 2\pi U_2(y - y_2) & v_2 &= -2\pi U_2(x - x_2)
\end{aligned}$$

where $x_1 = y_1 = y_2 = 1.5 \cdot 10^6$, $x_1 = 2.25 \cdot 10^6$, $U_1 = (2/3) \cdot 10^{-6}$ and $U_2 = 10^{-6}$.

4.7 Numerical aspects

The horizontal domain is divided into 30×30 grid cells. This results into a dimension of the state vector equal to 7200. (For the LOTOS model this will be about $2.5 \cdot 10^4$.) NO emissions take place in the cells (10,20) and (20,10) of the first layer. The strength of the emission is such that the NO concentration would increase with 2.5 and 5.0 ppb per hour if no other processes are taken into account. In the second layer, NO emission takes place in the cells (10,10) and (20,20) such that the concentrations would increase by 10.0 and 7.5 ppb per hour. The time interval considered is a period of 4 days, starting at 12:00h at day 1 and ending at 12:00h at day 4.

To generate measurements a model run is performed in which the model equations are solved very accurately. This has been done by using the implicit-explicit scheme (see [9]) with very small time steps of one minute. Each hour the O_3 concentrations at 25 different locations are stored, that will serve as measurements $M_{i,j}$ for the data assimilation runs. The indices i and j refer to the grid cells and may take the values 5, 10, 15, 20 and 25.

In the data assimilation runs, so-called operator splitting is applied, similar as in LOTOS. This means that the model equation is split up in a number of subprocesses. Each subprocess is then solved separately using the outcome of the previous subprocess as initial condition. For the present test model three subprocesses are identified: chemistry, advection and emission. The vertical exchange between the layers and the deposition are included in the chemistry. The chemistry is numerically solved by the method TWOSTEP [8, 10, 5]. The advection operator is spatially disretized by the 3rd order limited $\kappa = \frac{1}{3}$ discretization, integrated in time with a second order explicit Runge-Kutta formula [3, 5].

The different solution methods prevent that the assimilation runs are able to follow the "reality" exactly.

5. EXPERIMENTS WITH THE TEST MODEL

For NO and OH the initial conditions are set to zero in both layers. For O_3 and NO_2 they are taken 10 ppb in both layers. The boundary conditions are unchanged. It is supposed that the reaction rates k_1 and k_2, the parameters b_1 and b_2, the emissions and the diffusion parameter α are not exactly known. This is simulated by putting

$$k_1 = k_1 \cdot (0.75 + w_1) \tag{5.1}$$

$$k_2 = k_2 \cdot (0.75 + w_2) \tag{5.2}$$

$$b_1 = b_1 \cdot (0.75 + w_3) \tag{5.3}$$

$$E_{10,10} = E_{10,10} \cdot (0.75 + w_4) \tag{5.4}$$

$$E_{10,20} = E_{10,20} \cdot (0.75 + w_4) \tag{5.5}$$

$$E_{20,10} = E_{20,10} \cdot (0.75 + w_5) \tag{5.6}$$

$$E_{20,20} = E_{20,20} \cdot (0.75 + w_5) \tag{5.7}$$

$$\alpha = \alpha \cdot (0.75 + w_6). \tag{5.8}$$

The parameter b_2 follows from the relation $b_2 = 1 - b_1/2$. The w_i are supposed to be independent of each other and their standard deviations are given the value 0.5. The measurements have standard deviation of 1.0.

The experiments will be done with different values for the maximum number of modes. As measure for the error between the exact solution and the solution produced by the data assimilation run, the following norms are used

$$ERR_2 = \left(\frac{1}{900} \sum_{i,j} (c_{ij}^{ex} - c_{ij}^{ass})^2 \right)^{\frac{1}{2}}, \tag{5.9}$$

$$INF = max|c_{ij}^{ex} - c_{ij}^{ass}|, \tag{5.10}$$

where c^{ex} and c^{ass} denote the solutions for O_3 in the first layer from the reference run and from the assimilation run, respectively. The norms ERR_2 and INF are computed using the concentration field at the end of the time interval considered. Also a norm is introduced to measure the error in the measurement points at time $t = t^k$

$$ EM_k = \left(\frac{1}{|\mathcal{J}|} \sum_{(ij) \in \mathcal{J}} (c_{ij}^{ex}(t^k) - c_{ij}^{ass}(t^k))^2 \right)^{\frac{1}{2}} , \tag{5.11} $$

with $\mathcal{J}$ the set of indices of the measurement points. The total error in the measurement points is measured by the average EMA over the EM_k

$$ EMA = \frac{1}{96} \sum_{i=1}^{96} EM_i. \tag{5.12} $$

In order to be able to identify the occurence of filter divergence the norm VAR is introduced, which denotes the average computed variance over measurement points and time steps.

5.1 Experiment 1
This experiment consists of 3 cases, A, B and C. In case A all 25 measurements are taken into account. In case B and C only the measurements M_{ij} with indices $i, j \in \{5, 15, 25\}$ and $i, j \in \{5, 25\}$ are taken into account. The results for this experiment are summarized in Table 2. The colums "split" gives values of the error norms for the results of a model run with the correct parameters and right initial conditions in which operator splitting is applied in the way indicated above. From the numbers in the column "split" it can be concluded that the numerical solution is not very close to the exact solution of the model problem. For our purpose this is a nice situation to have, because usually models do not describe the modeled reality very accurately. The column "wrong" gives values of the error norms in case data assimilation is not applied and the wrong initial values and parameters are used. The numbers in this column clearly show that the results of this simulation are far from "reality". An important observation from Table 2 is that the error norms for all number of modes used are

case	norm	split	wrong	maximum #modes				
				10	25	50	75	100
	ERR_2	6.71	16.31	1.67	1.56	1.56	1.56	1.56
1A	INF	15.75	31.27	8.33	7.84	7.91	7.90	7.90
	EMA	3.24	22.85	1.36	0.94	0.94	0.94	0.94
	VAR	-	-	0.27	0.28	0.28	0.28	0.28
	ERR_2	6.71	16.31	2.06	1.85	1.82	1.82	1.82
1B	INF	15.75	31.27	8.18	9.28	9.68	9.73	9.92
	EMA	2.83	22.55	1.28	1.14	1.11	1.12	1.12
	VAR	-	-	0.24	0.25	0.25	0.25	0.25
	ERR_2	6.71	0.54	2.49	2.40	2.37	2.37	2.37
1C	INF	15.75	31.27	7.41	7.41	7.37	7.40	7.37
	EMA	2.22	20.18	1.63	1.59	1.58	1.58	1.58
	VAR	-	-	0.35	0.35	0.35	0.35	0.35

Table 2: Values of the error norms for experiment 1

significantly smaller than in the column "split". This shows that the extended Kalman filter is able to deal with the model problem. Another observation from Table 2 is that increasing the maximum

number of modes taken into account leads to some improvement of the simulations. Figure 1 shows the average error (absolute averaged residue) and the computed covariance for experiment 1A for 10 and 100 modes. In general, the computed covariances are smaller than the residues but filter divergence does not seem to occur in this case. Recall that the measurements were given a standard deviation equal to one. From Figure 1 it is seen that the computed residues have about the same value. Figure
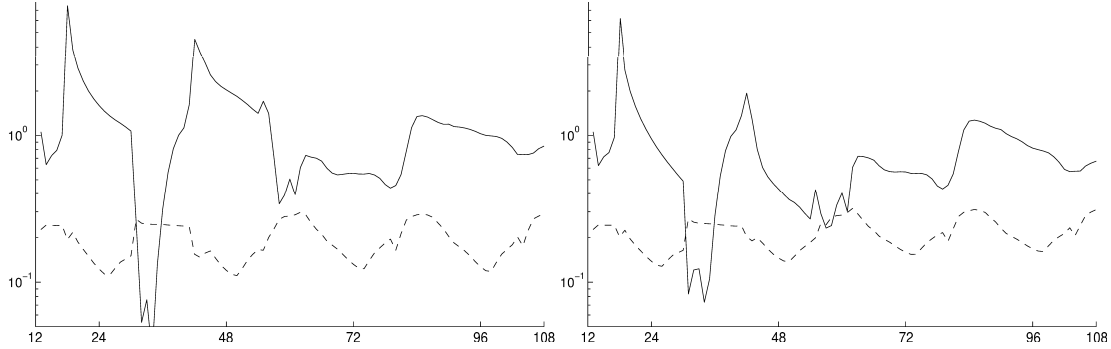

Figure 1: Average difference at the measurement locations (solid) and average computed covariance (dashed) for experiment 1A. Left: 10 modes Right: 100 modes

2 shows a plot of the reference solution (at the end of the model run) and of the solution of the data assimilation runs with 100 modes. The patterns in the plot corresponding to experiment 1A, 1B and 1C are all similar as the pattern of the reference solution. This indicates again proper functioning of the filter. Moreover, since in experiment 1C only 4 measurements per hour are used, Figure 2 shows that the filter is able to produce good approximations if only little information is available. Although both computations give good results, Figure 1 clearly shows that using only 10 modes is by far not enough to approximate the covariance matrix very accurately and causes loss of relevant information.

5.2 Experiment 2

In order to see if the filter is able to perform well if the dynamics of the model are "wrong", the wind field is changed. The centers of the rotational wind field have been moved four grid cells in eastern direction and three in northern direction in the first layer and four grid cells in western direction and three in southern direction in the second layer. In addition all wind speeds are multiplied by $\frac{2}{3}$. All 25 measurements are used. In case A, the filter is applied without doing anything extra to prevent filter divergence, which is expected to occur in this case. In Table 4 the error norms for the present experiment are listed.

In case B, C and D, measurements are taken to prevent filter divergence as described in Section 2.3. In case B a memory reduction factor $\zeta = 0.9$ is applied. In case C overweighting recent data is applied with $\xi = 0.25$. Finally, in case D the vector w is extended with ten parameters describing uncertainty in the wind fields. In order to obtain divergence-free wind fields, a stream function $\Psi(x, y)$ is introduced

$$\Psi_1(x, y) = w_7 x + w_8 y + w_9 xy + w_{10} x^2 + w_{11} y^2, \tag{5.13}$$
$$\Psi_2(x, y) = w_{12} x + w_{13} y + w_{14} xy + w_{15} x^2 + w_{16} y^2, \tag{5.14}$$

from which parametrized deviations in the wind fields are derived. As a result, the wind fields are
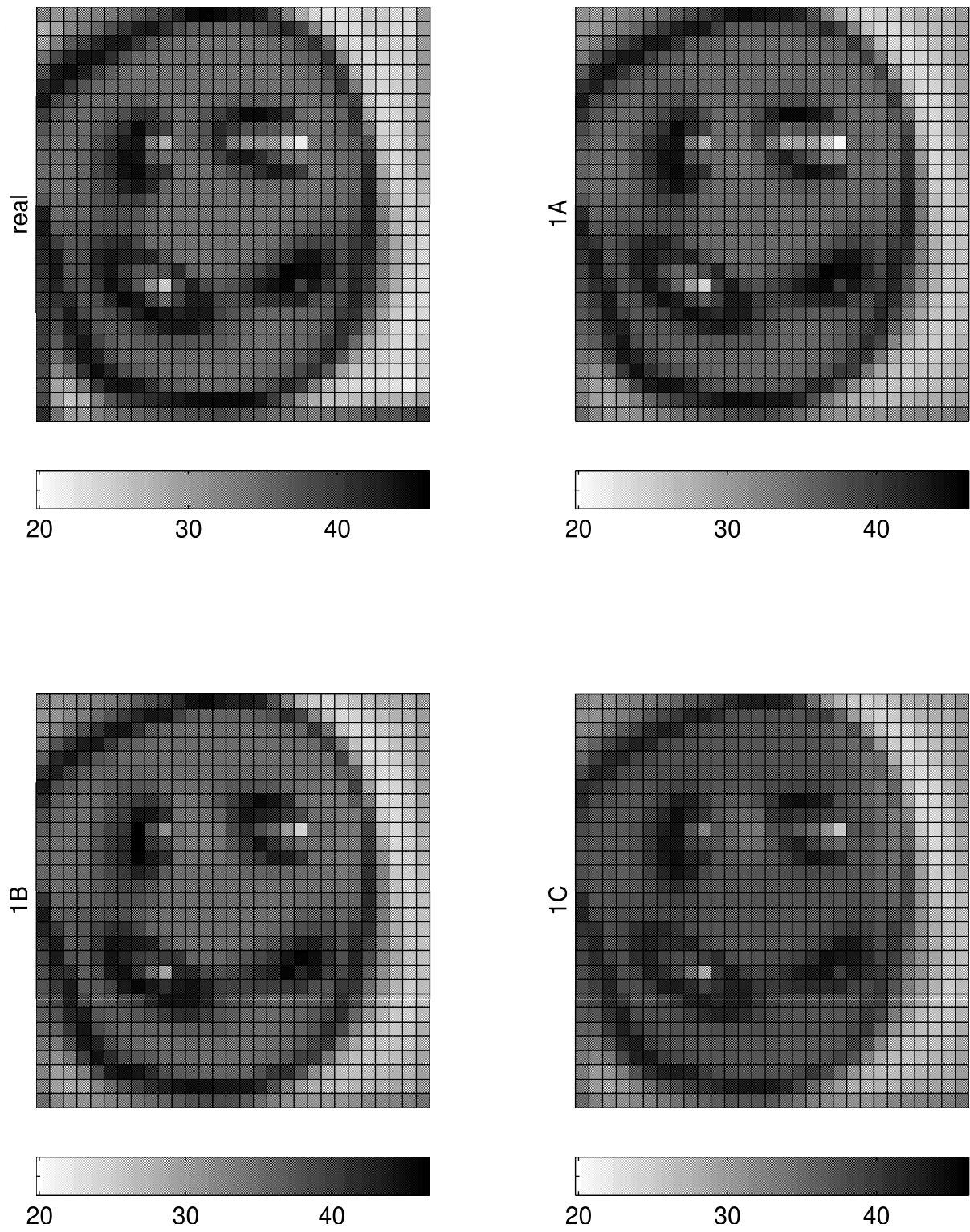
Figure 2: Real solution and the solutions for the experiments 1A, 1B and 1C from the data assimilation runs using 100 modes

specified as

$$
\begin{aligned}
u_1(x,y) &= u_1(x,y) + w_8 + w_9 x + 2w_{11} y, \\
v_1(x,y) &= v_2(x,y) - w_7 - w_9 y + 2w_{10} x, \\
u_2(x,y) &= u_1(x,y) + w_{13} + w_{14} x + 2w_{16} y, \\
v_2(x,y) &= v_2(x,y) - w_{12} - w_{14} y - 2w_{15} x.
\end{aligned}
$$

The standard deviation of $w_7, \ldots, w_{16}$ is taken 0.5, where the $x-$ and y-coordinate are both scaled to [0,1]. The results for this experiment are summarized in Table 4. In all cases sometimes negative

	case A #modes			case B #modes		
	10	25	50	10	25	50
ERR_2	6.01	6.36	6.36	10.29	10.09	10.11
INF	14.31	15.11	15.11	26.73	25.92	26.12
EMA	5.22	4.88	4.88	5.28	5.28	5.28
VAR	0.33	0.34	0.34	0.29	0.29	0.29
	case C #modes			case D #modes		
	10	25	50	10	25	50
ERR_2	0.20	0.19	0.19	0.14	0.13	0.14
INF	14.45	13.64	13.63	16.96	19.12	20.05
EMA	5.16	4.67	4.66	2.39	2.32	2.27
VAR	0.30	0.31	0.31	0.40	0.40	0.40

Table 3: Values of the error norms for experiment 2 using only 4 measurements

	case A #modes			case B #modes		
	10	25	50	10	25	50
ERR_2	5.11	5.11	5.09	5.11	5.11	5.11
INF	12.50	12.46	12.27	12.59	12.53	12.48
EMA	3.22	3.18	3.17	3.54	3.53	3.54
VAR	0.27	0.27	0.27	0.26	0.26	0.26
	case C #modes			case D #modes		
	10	25	50	10	25	50
ERR_2	5.07	5.01	5.02	8.93	2.82	2.35
INF	12.44	11.82	11.73	145.3	14.90	12.14
EMA	3.18	3.10	3.09	2.07	1.69	1.38
VAR	0.22	0.22	0.22	0.29	0.32	0.27

Table 4: Values of the error norms for experiment 2 using all 25 measurements

values were encountered after the processing of the measurements. In order to prevent unphysical solutions and (numerical) instabilities negative values are cut off to zero. The occurence of negative values indicates that the filter sometimes may introduce oscillations into the model leading to under- and overshoots. The latter can be seen from the relative high values for the INF error norm compared
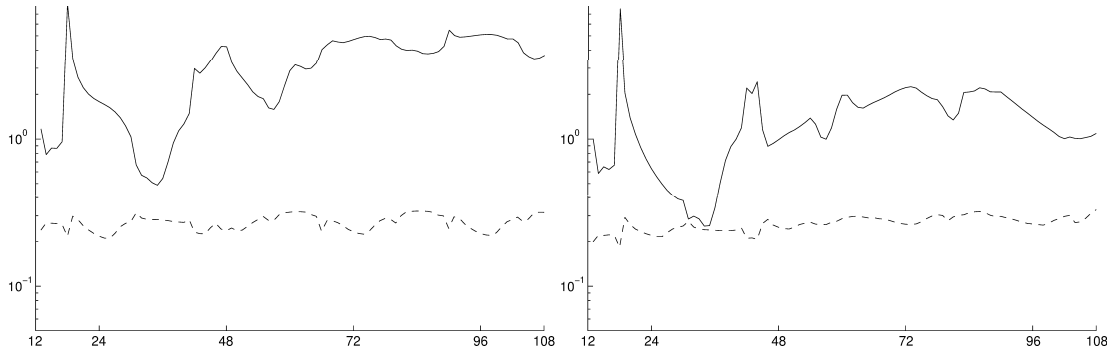
Figure 3: Average difference at the measurement locations (solid) and average computed covariance (dashed) for experiment 2A (left) and 2D (right) using 50 modes

to the values for ERR_2.

The results for case A cleary show, as expected, that due to the modification in the wind fields filter divergence occurs: the computed covariances are about 0.6 whereas the average residues are about ten times larger. Application of the memory reduction factor (case B) makes the situation even worse. Overweighting recent data (case C) gives only very little improvement. In Figure 4 shows solution plots of the real solution and the solution of experiment 2B, 2C and 2D. In case D more significant improvement is achieved. The error norms, except for INF, decrease considerably compared to the case A, B and C. The computed covariances increase, as they should, whereas the average residues decrease, though the differences between the two quantities are still too large. From Figure 4 it is seen that in this case the solution pattern of the real solution is recovered in contrast to case B and C.

5.3 Experiment 3
In experiment 2 we have seen that application of a memory reduction factor or overweighting recent data does not lead to improvements in case perturbations in the wind fields are present. Only if the wrong dynamics are also taken into account by means of adding noise, improvements were obtained (experiment 2D). Meanwhile the basic assumptions for memory reduction and overweighting recent data are still valid, so it seems natural to consider a combination of one of these two with case D of the previous experiment. The results for this experiment are summarized in Table 5. The error norms printed in italics in Table 5 indicate that in the corresponding experiment (3.4) has been applied with $\epsilon = 0.25$ because of instability in case of $\epsilon = 1.0$. It has been verified that the instability was caused by blow up in the chemistry routine due to negative initial concentrations. The latter are caused by the fact that the initial vectors $x^k + \epsilon s^i$ for a model evaluation in (3.4) may have negative components. In fact, this often happens without blow up occuring in the chemistry.

6. COMPUTATIONAL ASPECTS
Application of the RRSQRT algorithm to realistic atmospheric models is computationally very expensive. Even for the present simple test model a data assimilation run of five days using 50 modes already takes approximately ten hours of CPU time on a workstation (SGI, Indy). Since the aim is to apply the RRSQRT algorithm to LOTOS for the summer of 1997, it is clear that further reduction of the
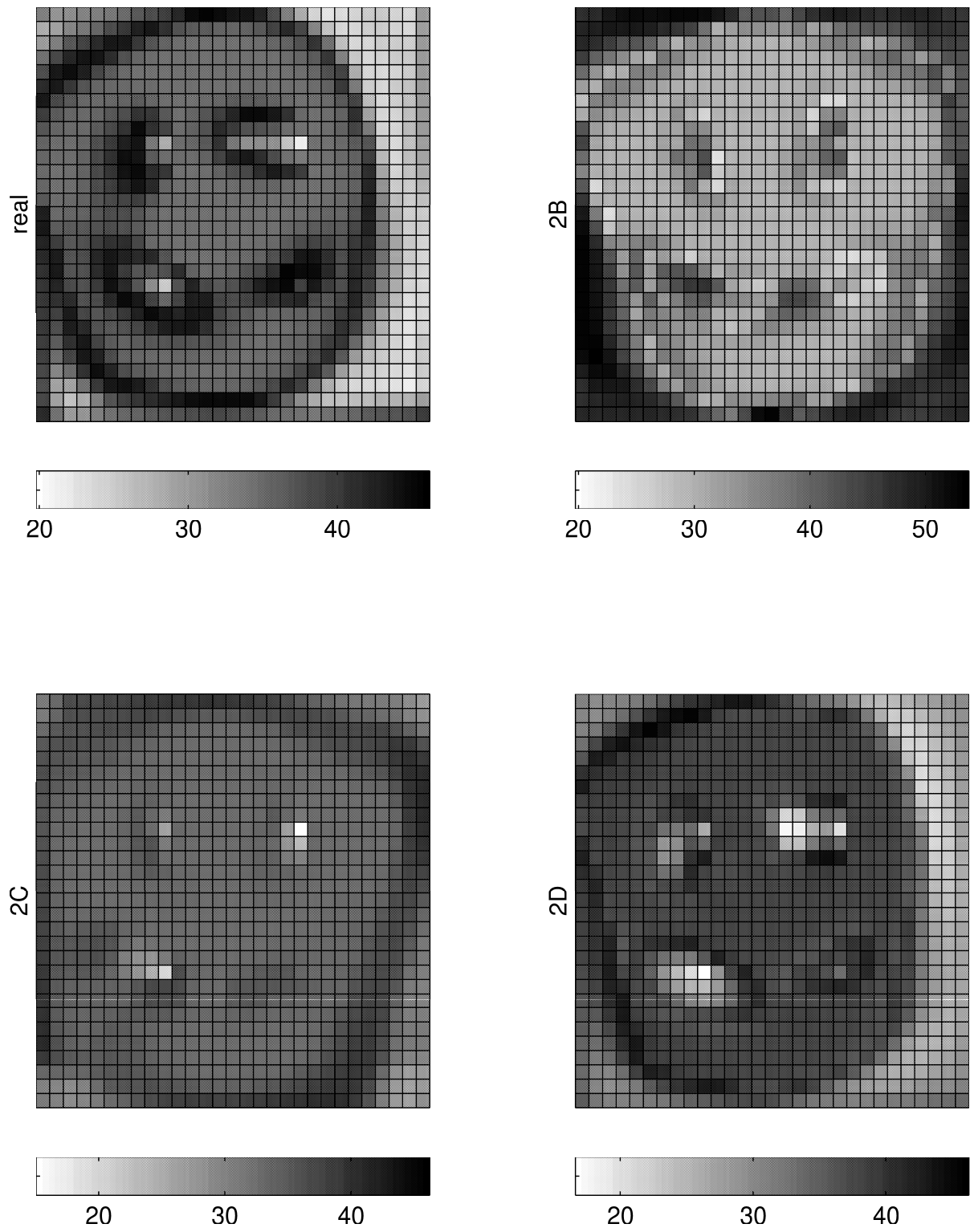
Figure 4: Real solution and the solutions for the experiments 2B, 2C and 2D from the data assimilation runs using 50 modes

σ		4 measurements				25 measurements			
		$\xi = 0.0$	$\xi = 0.1$	$\xi = 0.25$	$\xi = 0.5$	$\xi = 0.0$	$\xi = 0.1$	$\xi = 0.25$	$\xi = 0.5$
0.5	ERR_2	0.14	0.13	0.13	0.13	2.35	2.55	2.59	2.35
	INF	16.96	19.63	19.00	16.22	12.14	8.48	10.83	12.33
	EMA	2.39	2.12	2.00	2.16	1.60	1.55	1.52	1.38
	VAR	0.40	0.39	0.37	0.22	0.32	0.31	0.29	0.27
1.0	ERR_2	0.12	0.13	0.13	3.50	3.73	3.73	*3.52*	3.68
	INF	14.00	13.84	14.04	12.91	21.37	26.67	*23.16*	25.51
	EMA	1.59	1.49	1.36	1.17	1.34	1.34	*1.34*	1.21
	VAR	0.47	0.46	0.43	0.40	0.36	0.34	*0.33*	0.30
1.5	ERR_2	0.13	0.13	0.12	0.12	3.59	3.43	2.92	3.10
	INF	11.45	10.46	10.18	9.86	25.09	16.34	20.24	21.48
	EMA	1.46	1.37	1.26	1.19	1.24	1.47	1.16	1.19
	VAR	0.49	0.47	0.44	0.40	0.38	0.32	0.34	0.31

Table 5: Values of the error norms for experiment 3 using 50 modes

CPU time is necessary. In this section we briefly discuss two possibilities to achieve this reduction. In future work we will investigate these possibilities more thoroughly.

6.1 Further reducing sequential CPU time

The filter algorithm requires a number ($\sim$ maxmodes) of extra model evaluations per time step for the time propagation of the covariance matrix. If it is possible to reduce the cost of these extra model evaluations, a considerable reduction of the total computational work can be achieved. One possibility is to perform the extra model evaluations on a coarser grid. This requires all input data to be available on a coarser grid as well, which is of course possible but requires a lot of work for a large atmospheric model like LOTOS. Therefore we suggest a different approach. Since the model equations are solved numerically using operator splitting, it is possible to try to solve some of the subprocesses in a cheaper way. For example, the chemical equations can be integrated less accurately. This may lead to a significant reduction in CPU time, because the chemistry takes relatively a large part of the total CPU time. In Table 6 some preliminary results are shown. The preliminary results

norm	maximum #modes			
	10	25	50	100
ERR_2	2.16	2.01	2.12	2.06
INF	9.93	11.21	11.35	10.89
EMA	0.82	0.54	0.41	0.31
VAR	0.28	0.30	0.31	0.31

Table 6: Results for experiment 1A using a cheaper model evaluation

indicate that it may be possible to perform the model evaluations in a cheaper way when updating the covariance matrix. This seems to result into somewhat larger computed (co)variances, causing the filter to give relatively more weight to the measurements. This explains why values for EMA for the present experiment are smaller than the corresponding values from the original experiment (see Table 2

6.2 Efficient Parallel Implementation

The filter algorithm can be very efficiently implemented on a shared memory parallel machine because of the natural parallelism in the time step (3.1) and (3.2) of the RRSQRT algorithm. This time step requires $q+1$ independent model evaluations, q being the number of modes used. For the test model a parallel implementation (fortran 90) on a Cray C90 with four processors has been made. The singular value decomposition in the reduction step is performed by the routine SGESVD from the LAPACK library which exhibits very good speed up factors. Since the processing of the measurements is an essentially sequential process, parallelism has been achieved within each measurement update on the linear algebra level. On four processors the code running with 50 modes attains a speed up factor of about 3.5. Although this is only an indication of the possible performance of an implementation of the the the extended Kalman Filter applied to LOTOS, we think that good parallel performance can be achieved indeed. We plan to report on this in the near future.

7. CONCLUSIONS

Summarizing, the following conclusions can be drawn

- The Extended Kalman Filter (EKF) has been succesfully applied to an atmospheric chemistry test model. This indicates that it should be possible to apply the EKF to the model LOTOS.

- The Reduced Rank Square Root (RRSQRT) implementation, which approximates the EKF, works well in the present application where only 50 modes suffice for good results. Since LOTOS is a larger model and probably more noise parameters will be introduced, we expect that in applications with LOTOS 50-100 modes will be needed.

- Preventing filter divergence by some means has been shown to be necessary. Memory reduction gave worse results and should therefore not be considered. Overweighting recent data seems to improve the results, but is not able to prevent filter divergence without modelling the uncertainty in the dynamics.

- On a share memory machine excellent paralellization for the test model has been achieved by parallelisation over the calls to the model for all modes.

REFERENCES

1. R. Cañizares, A.W. Heemink, and H.J. Vested. Sequential data assimilation in a fully non-linear hydronamic model. *Unknown*, ?: ? – ?, 1996.

2. C.K. Chui and G. Chen. *Kalman Filtering with real time applications.* volume 17 of *Springer Series in Information Sciences.* Springer Verlag, Berlin, 1987.

3. W. Hundsdorfer, B. Koren, M. van Loon, and J.G. Verwer. A Positive Finite-Difference Advection Scheme. *J. Comp. Phys.*, 117: 35 – 46, 1995. Revision of CWI report NM–R9309.

4. A.H. Jazwinski. *Stochastic Processes and Filtering Theory.* volume 64 of *Mathematics in Science and Engineering.* Academic Press, New York and London, 1970.

5. M. van Loon. *Numerical Methods in Smog Prediction.* PhD thesis, University of Amsterdam, 1996.

6. P.S. Maybeck. *Stochastic Models, Estimation, and Control.* volume 141-1 of *Mathematics in Science and Engineering.* Academic Press, New York, 1979.

7. M. Verlaan and A.W. Heemink. Tidal Flow Forecasting using Reduced Rank Square Root Filters. Technical report, Delft University of Technology, Delft, The Netherlands, 1996.

8. J.G. Verwer. Gauss-Seidel iteration for stiff ODEs from chemical kinetics. *SIAM J. Sci. Comput.*, 15: 1243 – 1250, 1994.

9. J.G. Verwer, J.G. Blom, and W.H. Hundsdorfer. An implicit-explicit approach for atmospheric transport-chemistry problems. *Appl. Numer. Math.*, 20: 191 – 209, 1996.

10. J.G. Verwer and D. Simpson. Explicit methods for stiff ODEs from atmospheric chemistry. *Appl. Numer. Math.*, 18: 413 –430, 1995.