



Laan van Westenenk 501
Postbus 342
7300 AH Apeldoorn
The Netherlands

www.mep.tno.nl

T +31 55 549 34 93

F +31 55 541 98 37

info@mep.tno.nl

TNO-report

R 2003/251

**Data assimilation for the chemical transport model
REM3/CALGRID based on Kalman Filtering**

Contribution of TNO to the FUB-project "Entwicklung eines Modell-
systems für das Zusammenspiel von Messung und Rechnung für die
bundeseinheitliche Umsetzung der EU-Rahmenrichtlinie Luftqualität"

Date	May 2003
Authors	M. van Loon J. Flemming (Free University of Berlin)
Order no.	31291
Keywords	Data assimilation, Kalman Filter, Optimum Interpolation
Intended for	Freie Universität Berlin Institut für Meteorologie

All rights reserved.

No part of this publication may be reproduced and/or published by print, photoprint, microfilm or any other means without the previous written consent of TNO.

In case this report was drafted on instructions, the rights and obligations of contracting parties are subject to either the Standard Conditions for Research Instructions given to TNO, or the relevant agreement concluded between the contracting parties.

Submitting the report for inspection to parties who have a direct interest is permitted.

© 2003 TNO

Berichts-Kennblatt

BerichtsnummerUBA-FB	2.	3.
4. Titel des Berichts Datenassimilation für das Chemie Transport Modell REM3/CALGRID mit Kalman Filter		
5. Autor(en), Name(n), Vorname(n) Van Loon, Maarten (TNO) Flemming, Johannes (FUB)	8. Abschlußdatum 18.06.2003	9. Veröffentlichungsdatum 31.05.2003
	10. UFOPLAN-Nr. 29943246	
6. Durchführende Institution (Name, Anschrift) TNO-MEP Business Park E.T.V. Laan van Westenenk 501 P.O. Box 342 7300 AH Apeldoorn The Netherlands FU-Berlin Institut für Meteorologie Carl-Heinrich-Becker Weg 6-10 12165 Berlin	11. Seitenzahl 51	12. Literaturangaben 16 13. Tabellen und Diagramme 1 14. Abbildungen 16
	7. Fördernde Institution (Name, Anschrift) Umweltbundesamt, Postfach 33 00 22, D-14191 Berlin	
15. Zusätzliche Angaben		
16. Kurzfassung Der Bericht beschreibt die Implementation und Anwendung eines Kalman Filters für das Modell REM3/CALGRID der FU-Berlin im Rahmen des Projektes "Entwicklung eines Modellsystems für das Zusammenspiel von Messung und Rechnung für die bundeseinheitliche Umsetzung der EU-Rahmenrichtlinie Luftqualität		
17. Schlagwörter Datenassimilation, Kalman Filter, Optimale Interpolation		
18. Preis	19.	20.

Report Cover Sheet

Report No.UBA-FB	2.	3.
4. Report Title Data assimilation for the chemical transport model REM3/CALGRID based on Kalman Filtering		
5. Autor(s), Family Name(s), First Name(s) Van Loon, Maarten (TNO) Flemming, Johannes (FUB)	8. Report Date 31.05.2003	
	9. Publication Date 18.06.2003	
6. Performing Organisation (Name, Address) TNO-MEP Business Park E.T.V. Laan van Westenenk 501 P.O. Box 342 7300 AH Apeldoorn The Netherlands FU-Berlin Institut für Meteorologie Carl-Heinrich-Becker Weg 6-10 12165 Berlin	10. UFOPLAN-Ref. No. 29943246	
	11. No. of Pages 51	
	12. No. of Reference 16	
7. Sponsoring Agency (Name, Address) Umweltbundesamt, P.O. Box 33 00 22, D-14191 Berlin, Germany	13. No. of Tables, Diagrams 1	
	14. No. of Figures 16	
15. Supplementary Notes		
16. Abstract This report describes the implementation and the application of Kalman Filter routines around the model REM3/CALGRID employed at the Free University of Berlin (FUB) in the framework of the FUB-project "Entwicklung eines Modellsystems für das Zusammenspiel von Messung und Rechnung für die bundeseinheitliche Umsetzung der EU-Rahmenrichtlinie Luftqualität".		
17. Keywords Data assimilation, Kalman Filter, Optimum Interpolation		
18. Price	19.	20.

Summary

This report describes the implementation and the application of Kalman Filter routines around the model REM3/CALGRID employed at the Free University of Berlin (FUB) in the framework of the FUB-project “Entwicklung eines Modellsystems für das Zusammenspiel von Messung und Rechnung für die bundeseinheitliche Umsetzung der EU-Rahmenrichtlinie Luftqualität”.

The report is joint work of TNO and FUB. The implementation of the Kalman Filter with REM/CALGRID and preliminary testing has been carried out by TNO. FUB was responsible for the application and its evaluation. Chapter 4 (Modelling noise in REM/CALGRID RRSQRT – KF) and chapter 5 (Assimilation performance) have been prepared by FUB.

A theoretical description of the Kalman Filter is given and the special version that is employed in this project is described, the so-called Reduced Rank Square Root algorithm (RRSQRT). Also the routines that have been developed are described.

Results are presented for a number of simulations for July 2001:

1. Assimilation of only O_3 measurements
2. Assimilation of only NO_2 measurements
3. Assimilation of both O_3 and NO_2 measurements

In these three simulations noise was applied to a limited set of model input fields, namely the total NO_x emissions, the total VOC emissions and the vertical turbulent exchange coefficient K_z .

It is shown that in case O_3 (NO_2) measurements are assimilated the statistics for O_3 (NO_2) improve. If only NO_2 is assimilated for some stations the performance for O_3 decreases. For NO_2 the performance increases in all three cases.

Comparing the results covariances as computed by the Kalman Filter to empirical covariance modelling shows similar magnitudes, but with different spatial distribution. This is due to the very simple way the noise parameters are chosen in this study. In view of that it is very promising that the Kalman Filter results are already slightly better than results obtained with a computationally much cheaper method (Optimum Interpolation) using the empirical covariances. Hence it is concluded that the performance of the Kalman Filter may further increase with improving noise description.

The noise parameters have been included in the analysis process. They seem to have systematic daily patterns, which suggest that a more turbulent exchange during the night and less during the day may improve the performance of the model.

Table of contents

Berichts-Kennblatt.....	2
Report Cover Sheet	3
List of Figures	7
List of Tables.....	9
1. Introduction.....	10
1.1 Preliminaries.....	10
1.2 The current project.....	11
1.3 Structure of the report.....	13
2. The Kalman Filter	14
2.1 Introduction	14
2.2 Statistical framework.....	14
2.3 The Kalman Filter.....	14
2.4 The RRSQRT algorithm.....	15
2.4.1 Initial conditions.....	17
2.5 Optimum Interpolation	17
3. Implementation aspects of the RRSQRT algorithm.....	18
3.1 Introduction	18
3.2 Schematic overview of the data assimilation system	18
3.2.1 General requirements to air quality models	18
3.2.2 Setting and re-setting noise	19
3.2.3 The state vector	20
3.2.4 Measurements	20
3.2.5 Output.....	21
3.2.6 Preparation by the user.....	21
3.3 Structure of the system around REM3/CALGRID.....	21
4. Modelling noise in REM/CALGRID RRSQRT – KF	23
4.1 Introduction	23
4.2 Reminder of numerical realization of P modelling	24
4.3 Choice of noise factors and their implementation	25
4.4 Comparison with empirical modelling of observation errors (OI).....	26
4.5 Systematic patterns of analysed model noise parameters.....	28

5.	Assimilation performance	31
5.1	Set-up of assimilation runs	31
5.2	Methodology of performance evaluation.....	32
5.3	Assimilation performance for Ozone and or NO ₂ in a run for July 2001	34
5.4	The Kalman Filter versus Optimum Interpolation.....	38
6.	Conclusions and recommendations.....	40
7.	References.....	41
8.	Authentication.....	43
Appendix A Technical description of the KF code		
A.1	Description of the Kalman Filter routines as implemented around REM3/CALGRID	
A.2	REM3/CALGRID code preparation for connection to the Kalman Filter routines	
A.3	Description of output options REM/CALGRID KF	

List of Figures

Figure 4.1	Daily cycle of the standard deviation of the model error variance of ground level O ₃ obtained by empirical covariance modelling for OI (left) and by the Kalman filter assimilating NO ₂ and O ₃ (right)	27
Figure 4.2	Spatial structure of covariance with station BB001 of ground layer O ₃ -field at 13.00 UTC for July 2001 obtained by empirical covariance modelling for OI (left) and by the Kalman filter assimilating NO ₂ and O ₃ (right)	28
Figure 4.3	Daily cycle of the mean of the noise parameters for the Knox-emission (left), the VOC-emission (right) and vertical turbulent exchange coefficient KZ1 (right) for a KF run assimilating O ₃ and NO ₂ -observations	29
Figure 4.4	Daily cycle of the mean of the noise parameters for the NO _x -emission (left), the VOC-emission (right) and vertical turbulent exchange coefficient KZ1 (right) for a KF run assimilating O ₃ -observations	30
Figure 4.5	Daily cycle of the mean of the noise parameters for the NO _x -emission (left), the VOC-emission (right) and vertical turbulent exchange coefficient KZ1 (right) for a KF run assimilating NO ₂ -observations	30
Figure 5.1	Estimated Observation error standard deviation in ppb vs. air quality regime (TYP) for all stations in Germany (left NO ₂ , right O ₃). B = mountain, R = rural, U1 = suburban, U2 = urban, U3 = urban, polluted, S = street, s2= Street extreme	32
Figure 5.2	RMSE of ozone observations for the base run (BAS), the KF assimilation of NO ₂ -observations (KF(NO ₂)), the KF assimilation of O ₃ - observations (KF(O ₃)) and the KF assimilation of O ₃ and NO ₂ – observations (KF(O ₃ NO ₂)) at rural, suburban, urban and polluted urban stations	34
Figure 5.3	Bias (model – observation) of ozone observations for the base run (BAS), the KF assimilation of NO ₂ -observations (KF(NO ₂)), the KF assimilation of O ₃ - observations (KF(O ₃)) and the KF assimilation of O ₃ and NO ₂ – observations (KF(O ₃ NO ₂)) at rural, suburban, urban and polluted urban Stations	35
Figure 5.4	RMSE of NO ₂ observations for the base run (BAS), the KF assimilation of NO ₂ -observations (KF(NO ₂)), the KF assimilation of O ₃ - observations (KF(O ₃)) and the KF assimilation of O ₃ and NO ₂ – observations (KF(O ₃ NO ₂)) at rural, suburban, urban and polluted urban Stations	35

Figure 5.5	Ratio between RMSE and observation error standard deviation SDOBS of ozone for the base run (BAS), the KF assimilation of NO ₂ -observations (KF(NO ₂)), the KF assimilation of O ₃ - observations (KF(O ₃)) and the KF assimilation of O ₃ and NO ₂ – observations (KF(O ₃ NO ₂)) at rural, suburban, urban and polluted urban Stations.	36
Figure 5.6	Average of O ₃ - ground layer concentration for July 2001 from the base run, i.e. no assimilation	37
Figure 5.7	Average of O ₃ - ground layer concentration for July 2001 from the KF assimilation of O ₃ - observations in Germany	37
Figure 5.8	Average of NO ₂ - ground layer concentration for July 2001 from the base run, i.e. no assimilation	38
Figure 5.9	Average of NO ₂ - ground layer concentration for July 2001 from the KF assimilation of O ₃ and NO ₂ - observations in Germany	38
Figure 5.10	Mean daily cycles of the measured, modelled and assimilated (with OI and KF) ground level concentrations at rural (left) and urban (right) stations for Ozone.....	39
Figure 5.11	Mean daily cycles of the measured, modelled and assimilated (with OI and KF) ground level concentrations at rural (left) and urban (right) stations for NO ₂	39

List of Tables

Table 3.1	List of KF routine within the data assimilation system around REM3/CALGRID.....	22
-----------	---	----

1. Introduction

1.1 Preliminaries

Historically, one of the first large applications of data assimilation took place in a field related to air quality hind- and forecasting: weather prediction. In meteorology, the aim was and still is to improve the initial conditions for the forecast. It is well known that the initial conditions for a weather forecast largely determine the prediction, in particular on the longer term (i.e. after the second day of the forecast period). For this reason, much effort is put into improving the description of the present (meteorological) state, which in turn improves the performance of the forecast. During the mid 80s data assimilation procedures were introduced, aimed at improving the initial conditions for the forecast. In this field usually so-called variational data assimilation was used in which the initial conditions for the hind-casting period (typically an interval of a few days before the start of the forecast interval) were optimised using all available observations during this period.

In more recent years also air quality forecasts have become operational. Some of these forecast systems apply a form of data assimilation as well for the same reason as in weather prediction: improving the initial state for the forecast period. Given the complex nature and the large amount of computing time required by the variational assimilation, simpler and computationally less demanding methods are often used, usually a form of Optimum Interpolation (OI). Another useful application of data assimilation in connection with simulating air quality is in the assessment of ambient air quality, for example in the framework of the EU directives. Each calendar year, Member States have the obligation to report concentration levels and indicator values of regulatory compounds. This is where combining models and measurements through data assimilation may come into play. By applying data assimilation for a calendar year using all available, relevant measurements, air quality models can be used to act as an ‘intelligent interpolator’ to describe the air quality throughout the area of interest and to examine compliance with AQ standards.

Hence, data assimilation techniques aim to combine two sources of information: the observations and the (here: air quality) model. Both contain information: the measurements on the state of the atmosphere at certain times and locations, the AQ models on the processes that govern the time evolution of air pollutants throughout the model domain during the whole simulation period. Both information sources have weaknesses and strengths.

Apart from instrumental errors (or: uncertainties), the obvious weakness of measurements is that limited spatial information is obtained. Operational air quality networks are mainly designed to monitor high concentration levels in polluted areas. The spatial representativeness of these observations appears to be limited and is not well known. Therefore spatial information from Eulerian models is useful for interpolating and mapping of the observations (Flemming and Stern, 2002). The

assimilation of satellite observations in chemical transport models may help to interpret the radiance information and to improve their vertical and temporal resolution (van Loon et al., 2002).

The obvious weakness of models is their failure from time to time of reproducing observed concentrations at locations (and times) where they are supposed to show a reasonable agreement. The reasons for these failures are of various kind and include imperfectnesses in the model input (e.g. emissions, meteorological data, land use) and in modelled processes or parameters in these processes.

The basic idea behind data assimilation is to quantify the uncertainty in both sources of information in statistical terms and then, to put it simply, make a weighted average of the model state (i.e. the concentrations) and the observed state. Ideally, by doing this the overall uncertainty is reduced.

Another development that took place in the last six years was the introduction of Kalman Filtering in the field of air quality modelling (see e.g. van Loon et al. 1997, 1999). Although this technique is computationally much more intensive than OI, it has some clear advantages over OI. In fact, OI can be seen as a special form of Kalman Filtering, the difference being that OI constructs the necessary covariance matrix in an (cheap) empirical way, whereas the Kalman Filter constructs this matrix by a number of model evaluations, based on the statistical description of the model uncertainties. Since the number of model evaluations needed by the Kalman Filter is in the order of 50 (this number strongly depends on the model uncertainties), application of the Kalman Filter is approximately in the order of 50 times more expensive than application of OI. The advantage of the Kalman Filter is that it takes into account model uncertainties, like uncertainties in the emissions, meteorological fields, parametrizations and so on, to be specified by the modeller. As a consequence, not only compounds that are observed are changed by the assimilation procedure, but other compounds as well. For example, if the KF adapts the ozone field based on uncertainties in its precursor emissions, it will also adapt these precursors and even a number of other compounds, since these compounds are all interrelated through chemical processes. In addition, the Kalman Filter is able to quantify the adaptations it makes to input fields and parameters, so that by carefully examining the results insight will be obtained into further improvement of the model and its input.

1.2 The current project

Since at the Free University of Berlin (FUB) both air quality forecasts are produced and modelling studies are performed, a (sub)project has been initiated to build a data assimilation system around the Eulerian chemistry transport model REM/CALGRID used at the FUB, within the framework of a larger project entitled “Entwicklung eines Modellsystems für das Zusammenspiel von Messung und

Rechnung für die bundeseinheitliche Umsetzung der EU-Rahmenrichtlinie Luftqualität”. This (sub)project was carried out by TNO-MEP in close collaboration with Dr. R. Stern and Dr. J. Flemming (both from the FUB). The aim of was to build and test this data assimilation system, using already developed routines based on a special implementation of the extended Kalman Filter. The following tasks were defined:

Task 1

Porting the sequential version of the REM3/CALGRID code to TNO, including all necessary input, meteorology, emissions etc, for the Berlin-area, for a period of two weeks. Check if results are equal when the code is run at TNO for the selected period.

Task 2

Building a data assimilation shell around REM3/CALGRID in two stages:

First stage:

- rearrangement of the REM3/CALGRID code
- adding noise to some selected processes in REM/CALGRID
- test if the results are still equal to the original run

Second stage:

- adding the data assimilation shell
- test if the results are still equal to the original run if run with noise input zero

Task 3

Testing the system with measurements provided by the Free University Berlin.

1. creating interface between measurement structure and data assimilation
2. run for the selected two-week period
3. check on the results: the concentrations as well as the estimated values for the noise parameters.
4. if necessary: define additional/different noise parameters and repeat step 2-3

Task 4

Creating a “stand alone” version that uses the outcome of the noise parameters from the data assimilation as input. Check if results are consistent with the full data assimilation run. This version is to be used for the scenario calculations by the Free University Berlin.

Task 5

Porting code back to the Free University Berlin.

During the execution of the project Task 1 (Porting the REM3/CALGRID code to TNO) turned out to be very difficult, mainly because of the structure of the binary input files used by REM3/CALGRID and the different ways various operating sys-

tems handle binary formats. A lot of effort has been put into porting the code the TNO, but after several failures it was finally decided not to port to code to TNO but instead perform the work at FUB. Through remote access most of work on the implementation was done and finalised during a one week visit to FUB by dr. M. van Loon in collaboration with dr. J. Flemming (FUB). Task 2, 3 and 4 were carried out as described above, task 5 had of course become redundant.

1.3 Structure of the report

This report briefly describes the theory of the Kalman Filter and its special implementation, as well as the way the TNO routines are connected to the REM3/CALGRID model (in the remainder denoted as RCG). Technical details on the latter will be done in a technical Annex, appendix A.

The performance of the system is evaluated for a selected period and a comparison is made between the results of the Kalman Filter and those of a much simpler technique, Optimum Interpolation (OI), that is currently applied within the air quality forecasting system at the Free University of Berlin.

The simulations performed concern the month of July 2001, using hourly O₃ and NO₂ observations. The comparison comprises the structure of the obtained model error covariances the analysed concentration fields. In addition, an example of an assessment of model parameters such as turbulent exchange coefficients by means of the Kalman filter is given.

2. The Kalman Filter

2.1 Introduction

Both air quality models and the measurements of chemical constituents contain valuable “knowledge” on the chemical composition of the atmosphere. Data assimilation tries to combine both sources of knowledge. That is, during a model simulation measurement information is processed. This processing of measurements is done in a statistical way: in order to process the observations, one needs to know the statistics of both the observations and the model state. Although not strictly necessary, in Kalman Filtering it is assumed that both have Gaussian distributions and can thus be represented by their means and standard deviations. For vector-valued quantities the latter should be interpreted as a covariance matrix.

The main goal of this chapter is to describe the Kalman Filter and more in particular to provide a description of the special implementation of the so-called RRSQRT algorithm, that is used in this project.

2.2 Statistical framework

Since the statistics of both the model and the observations are needed, a stochastic rather than a deterministic formulation is needed. Such a formulation is given by

$$x^{k+1} = f^k(x^k, w^k) \quad (1)$$

$$y^k = C^k x^k + v^k, \quad (2)$$

where the superscripts denote the time level. The model state vector is denoted by x and the measurements by y . The function f denotes the non-linear model operator which apart from on the state vector acts on a white noise vector w of dimension m with Gaussian distribution and diagonal covariance matrix Q . The measurement vector y is assumed be a linear combination of elements of the state vector and a random, uncorrelated Gaussian error v with (diagonal) covariance matrix R .

2.3 The Kalman Filter

The original Kalman Filter was formulated for linear system, i.e. instead of the function f in (1), a linear operators (matrices) are used:

$$x^{k+1} = A^k x^k + F^k w^k, \quad (3)$$

The operator A denotes the model operator and F maps to noise input vector unto a state-vector. The optimal estimate $\hat{x}$ of the state vector x is then given by

$$\begin{aligned}
 \hat{x}_f^{k+1} &= A^k \hat{x}^k \\
 P_f^{k+1} &= A^k P^k (A^k)^T + F^k Q^k (F^k)^T \\
 K^{k+1} &= P_f^{k+1} (C^{k+1})^T \left[C^{k+1} P_f^{k+1} (C^{k+1})^T + R^{k+1} \right]^{-1} \\
 \hat{x}^{k+1} &= \hat{x}_f^{k+1} + K^{k+1} (y^{k+1} - C^{k+1} \hat{x}_f^{k+1}) \\
 P^{k+1} &= P_f^{k+1} (I - K^{k+1} C^{k+1})
 \end{aligned} \tag{4}$$

where the pair $(\hat{x}, P)$ describe the probability of the state vector x completely if x has a Gaussian distribution. For non-linear problems, one can think of several possibilities to put them in the framework of (4). For example one could linearize f and in this way obtain the operators (matrices) A and F . Another approach is the use the operator f itself (i.e. the model!)

to compute actual steps in (4). However, for large system, formulation (4) cannot be used since the covariance matrix P has a dimension that will not fit into the memory of even the most modern computers. Moreover, the computational burden will be unrealistically high because of the enormous amount of matrix operations in the second step of (4). Therefore, an approximation of (4) needs to be used. We will postpone the exact formulation we use for non-linear problems to the next section on the Reduced Rank Square Root (RRSQRT) algorithm.

2.4 The RRSQRT algorithm

The basic idea behind the RRSQRT algorithm is not to work with the covariance matrix P but instead 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 chosen. Another important advantage is that S need not have n columns, but can have $q < n$ columns. This property is exploited by the RRSQRT algorithm, thus reducing the amount of computational work considerably. From (4) it is possible to rewrite the Kalman Filter in terms of S . The algorithm consists of three steps:

1. Time step

The time step performs the time propagation of the state vector x^k to the next time level and of the matrix S .

$$\begin{aligned}
 \hat{x}_f^{k+1} &= A^k \hat{x}^k \\
 S_f^{k+1} &= \left[A^k S^k, F^k (Q^k)^{1/2} \right]
 \end{aligned} \tag{5}$$

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

2. Reduction step

In case of a non-linear model, the model operator A is just a formal notation: in the first step of (5) the model itself is used and in the second step, the Jacobian matrix of the system (1) is inserted. This Jacobian is not really computed, but instead finite difference approximations are taken, using a model evaluation, to update the covariance matrix. Each column of $A^k S^k$ (second step of (5)) is computed by

$$\left(S_f^{k+1} \right)_i = \left(f(x + \varepsilon s_i^k, 0) - f(x, 0) \right) / \varepsilon \quad (6)$$

and

$$\left(F^k (Q^k)^{1/2} \right)_i = \left(f(x, \varepsilon w_i) - f(x, 0) \right) / \varepsilon \quad (7)$$

This implies that $q+m$ model evaluations need to be done per time step to update the covariance matrix: q times because of the update of the old covariance matrix and m additional times for the introduction of the new noise. Hence, the dimension of S is growing at each time step. Therefore S is reduced to q columns at the end of each time step by means of a singular value decomposition.

3. Measurement step

The measurement update is the step in which the available measurements are taken into account by the assimilation algorithm. Instead of applying the last three equations in (4), in which all measurements are taken into account at the same time, a sequential procedure is followed. This procedure can be proven to be equivalent to the procedure in (4). In terms of S , the procedure reads

$$\begin{aligned} H_i &= S_{i-1}^T c_i^T \\ a_i &= (H_i^T H_i + R_{ii})^{-1} \\ b_i &= (1 + \sqrt{a_i R_{ii}})^{-1} \\ K_i &= a_i S_{i-1} H_i \\ S_i &= S_{i-1} - b_i K_i H_i^T \\ x_i &= x_{i-1} + K_i (y_i - c_i x_{i-1}) \end{aligned} \quad (8)$$

with $S_0 = S_f^{k+1}$ and $x_0 = x_f^{k+1}$.

The index i is the iteration index, ranging from 1 to the number of measurements to be processed. The vector c_i denotes the i -th row of the matrix C^{k+1} .

2.4.1 Initial conditions

As initial condition for the covariance matrix P the zero matrix is taken. This is arbitrary and probably not the best choice. On the other hand, atmospheric chemistry-transport models are such that after one to two days of simulation the influence of the initial state of the system has disappeared. Also, in these first days of the simulation, relatively few adaptations to the measurements will be made, because the covariance structure has not been built up yet. After the “build-up phase”, the covariances will be realistic and adaptations of the state vector due to these covariances and the measurements will be made. Hence, when examining for example concentrations plots of assimilated and measured concentrations, the first, say, two days should be considered as model initialisation.

2.5 Optimum Interpolation

Active data assimilation with Optimal Interpolation (OI) can be considered as a simplification of the KF-algorithm. It relies on the same assumption as KF, such as no bias between observations and the model, uncorrelated and unbiased observation errors and a Gaussian distribution of the errors.

The difference between the described KF-algorithm and OI is the determination of the model error covariance matrix P . In the KF P is propagated like the state vector by an approximation of the tangential linearised operator M of the dynamical model M . The imperfectness of the dynamical model is expressed in a model forecast error covariance matrix Q , which is added at every forecast of P . For OI the covariance of the model error is estimated by empirical covariance modelling of the observation increments.

The empirical covariance modelling of the model error covariance is based on the differences between observations and the modelled values. The ensemble for the estimation is therefore limited to the station locations. In order to obtain a full covariance matrix an empirical covariance model has to be established.

Inhomogeneous covariance modelling has been developed for passive data assimilation with RCG assuming temporal stationarity of the errors (Flemming, 2003). However, in active assimilation the current model state and its error is influenced by the analysis of the previous model time step. That's why a simple form of instantaneous covariance modelling has been applied, namely the fitting of a homogeneous and isotropic covariance function in exponential form. It expresses the covariance of the model error between two points merely as a function of their distance. Such a covariance function of the O_3 and the NO_2 field was estimated for every analysis step in the OI run.

Taking measurements into account for covariance modelling allows to determine and to correct the bias between model state and the observation. Here, the actual bias correction is based on the difference between the mean model state and the mean of the observations at the rural stations.

3. Implementation aspects of the RRSQRT algorithm

3.1 Introduction

In this Section we describe the implementation of the RRSQRT algorithm that is presented in the previous chapter. First this is done in general terms, followed by a description of the connection of the routines developed at TNO to the RCG model. For more detailed technical aspects on the implementation around RCG we refer to Appendix A.

3.2 Schematic overview of the data assimilation system

In Algorithm 1 a schematic overview of the complete structure of the data assimilation system (i.e. both the assimilation shell and the model routines) is given.

1. Initialisation
 - a. Call model initialisation routine
 - b. Pick up model dimensions, set-up of state vector and covariance matrix
2. Prepare (next) time step
3. Add new noise modes
4. Perform time step for:
 - a. The model state vector
 - b. The existing modes
 - c. The new noise modes
5. Assimilate measurements
6. Postprocess time step
7. Repeat 2 - 6 until end of simulation
8. Finish simulation

Algorithm 1 Schematic overview of the data assimilation algorithm

As mentioned earlier, a single time step needs to be repeated many times (in our implementation $q+m+1$ times to be precise) with different initial conditions and/or different values of the noise parameters. Therefore the amount of computations in step 4 of Algorithm 1 should be kept at a minimum. This may require that the source code of an air quality model needs to be rearranged in order to be efficiently connected to the data assimilation routines.

3.2.1 General requirements to air quality models

In order to facilitate the connection of the models to the data assimilation routines, the models should be arranged in a specific form, as explained above. Ideally, the

main program in original model system calls a subroutine that performs a time step with as main argument a model state. This subroutine should not change parameters other than just the model state vector. For example, it should not update meteorological fields at the end of the time step, since they may be needed for the next call to this routine if used in connection with the data assimilation routines. In case the model state vector is part of a common block (Fortran 77), some extra copying may be necessary. This will lower the speed of the code to some extent, but keeps the changes to the original model to a minimum. Otherwise, new version of the model cannot be coupled to the data assimilation routines in a straightforward way.

In order to easily connect the data assimilation routines, it should be possible to split the model into the following parts:

1. Initialisation and set-up
 - a. Perform everything that needs to be done only at model start, such as reading from a run control file.
2. Preparation of a time step
 - a. Process meteo
 - b. Process emissions
 - c. Etc.
3. Execution of a time step
4. Postprocessing of the time step
 - a. Update model time
 - b. Write some output
 - c. Etc.
5. Finish model simulation
 - a. Closing files
 - b. Etc.

Algorithm 2 Model structure

In principle step 3 should contain the minimum necessary, because this step is repeated many times. Everything that can be part of step 2 or 4 should be placed there and not in step 3. In air quality models this means for example that processing meteorological fields are performed in step 2 (see Algorithm 2), since these fields are the same for the model state and each mode. The only exception is when noise is added to one of the meteorological quantities. This should however rather be handled by the routines that set and re-set the noise parameters (see below).

3.2.2 Setting and re-setting noise

Since the data assimilation will act on some of the inputs, extra routines will be necessary to modify these inputs at the beginning of a time step with a specific mode and re-set the values at the end of this time step. Setting and resetting the noise are user defined routines. If one of the noise factors acts on e.g. the NO_x

emissions, in the subroutine “set noise”, the emissions will be multiplied by the noise factor and the nominal values will be set back in subroutine “Reset noise”. In case the NO_x emissions can be changed from outside (e.g. because they are in a common block or module), no changes to the original model subroutines need to take place. The subroutine then simply copies the NO_x emission array and fills the original array with different values. After the time step the copy is put back into the original array. For more “complex” noise definitions, it is possible that some model subroutines need to be modified somewhat. For example, if one wants to let a noise parameter act on the NO_x emissions from traffic, this needs to be done before the aggregation of the emissions into one emission field as is usually done within a models emission routine.

3.2.3 The state vector

A logical choice for the state vector is the concentration array within the model. For reasons of (memory) efficiency, a subset of the state vector could be chosen. For example, if only O₃ measurements are assimilated, it can be expected that fields of some chemical compounds will hardly change, even when noise is added to some precursors. On the other hand, the state vector need not be restricted to (a part of) the concentration array. All parameters that are possibly changed by the data assimilation, are candidates. One could for example add correction factors on the emissions, deposition etc. on which the noise parameters act. By doing so, the filter algorithm will produce estimates for the correction factors as well. Also, when performing the measurement updating within the filter, it is possible to restrict the update when this would result into unrealistic correction factors. An example would be a negative emission, that can be prevented by putting an a priori upper and lower bound on the correction factors that are part of the state vector.

3.2.4 Measurements

The routines that read in and provide the measurements to the data assimilation system are also to be written by the user. The system is set up such that each measurement corresponds to an element of the state vector. Once this information is passed to the KF, it can do the assimilation of the measurements.

In step 3 of algorithm 3, the measurements are assimilated. Assuming that users of the data assimilation system want to keep their own structure of the measurement files, special routines must be written that extract the desired measurements from that measurement files. The easiest way to connect the measurement files to the data assimilation system is by an interface routine that can be called with arguments that specify which data from which station at which time is required.

3.2.5 Output

The data assimilation system can of course produce a number of output variables. Assuming that users prefer their own output formats, also here dedicated routines must be written, preferably calling already existing output routines within the model itself.

3.2.6 Preparation by the user

In view of what is written above, the user (i.e. the owner of the model) could do a lot that will facilitate the connection of the Kalman Filter routines to the model:

1. Bring the model in the form of Algorithm 2
 - a. Provide info on where to catch the model dimensions etc.
 - b. Provide info on the model state and its structure
2. Facilitate the construction of the routines that set and reset the noise (see algorithm 2)
 - a. This should be very easy if you know the model
3. Construct a routine that can be called for retrieving the measurements

3.3 Structure of the system around REM3/CALGRID

The original KF routines are all written in Fortran 90 (filenames have the extension .f90). For the implementation around REM3/CALGRID it was necessary to use Fortran 77 for one of the source files (kf_driver.f, extension .f instead of .f90), because in this file the actual connection with the model sources code is established. Recall that REM3/CALGRID is programmed in Fortran 77. The KF implementation consists of the following modules (all filenames associated with the KF start with kf_), as listed in Table 3.1.

Table 3.1 List of KF routine within the data assimilation system around REM3/CALGRID

Source file name	Short description
kf_driver.f	Performs the connection of the KF with the model. It contains parts of the original source of REM3/CALGRID in order to bring the model in the form of Algorithm 2.
kf_state.f90	Describes the model state, in this case the concentration vector augmented by an array with output for a number of compounds at specific measurement locations and augmented with an array containing the noise parameters.
kf_output.f90	Performs the output of assimilated concentrations as well as KF specific parameters, like the estimated values of the noise parameters.
kf_reduce.f90	Performs the reduction of the covariance matrix (see theory), using a singular value decomposition (SVD). For the SVD routines from the publicly available numerical library LAPACK are used. This library is installed at FUB.
kf_meas.f90	Performs the reading and proper passing of the observations to the KF routines.
kf_noise.f90	Sets and resets the noise parameters. This routine needs to be edited for adding/removal of noise processes
kf_main.f90	The main program.

In Appendix A more details are given on the actual implementation of the KF routines and the necessary changes to the source code of REM3/CALGRID.

4. Modelling noise in REM/CALGRID RRSQRT – KF

4.1 Introduction

The influence of the observation in the assimilation process is determined by the statistical properties of the model error, which is given in terms of its covariance matrix P . In the case of the RRSQRT Kalman Filter P is built up by a specification of uncertainties in the CTM REM/CALGRID and its input data. It means that the user has to specify a range of variation of model parameters, which may be the cause of the misfit between model and the observations. This specification is in general a difficult task because of the many interaction between the different processes involved and the requirement for quantification. For instance the modelling community agrees that there are errors in the emissions inventories but there is discussion about the quantitative range of uncertainty (Van Aardenne, 2002). In the given application the model error covariance relies completely on the specification of the model noise factors since they are the only way to build up a covariance matrix.

The numerical realisation of the noise modelling is explained in section 4.2. The choice of the model noise parameters is the content of section 4.3.

In contrast to the Kalman Filter, the model covariance error P is estimated by means of the observation increments, i.e. the difference between the model and the observations in the case of Optimum Interpolation (OI). Having the empirical covariance modelling available means that there is some reference whether magnitude of the covariance in Kalman Filter is of similar size as the empirical covariance from Optimum Interpolation. This issue is discussed in section 4.4.

The obvious advantage of the Kalman Filter is that its analysis step comprises the whole model state in which the model noise parameters are included. Thereby a linear regression type of relationship, given by the covariance matrix P , between the observation increments (difference between model and observation) and the noise parameters is used to obtain an optimal value for the noise factor. Systematic patterns in the analysed model noise parameters can be used to detect deficiencies in the original model parameters. An investigation of this problem is given in section 4.5

The statistical base of the Kalman Filter assumes no systematic errors, i.e. bias, of the model and the noise parameters. Systematic model deficits should be corrected in the first place before any attempts of data assimilation are carried out. From a theoretical point of view, the Kalman filter treats only the remaining part of the errors with zero mean and a certain variance - covariance structure. However, this limitation does not mean that model with systematic errors must not be used in KF data assimilation. The fact of not fulfilling this assumption leads to a sub optimality of the scheme and not to a complete failure. It could be assumed that after a certain spin up time of the Kalman Filter the model state fulfils the required error characteristics.

The numerical burden of the Kalman Filter requires a fast computation of the several call of the routine `perform_timestep` (see Appendix A2).

Besides attempts to run the Kalman filter in a parallel manner, the proper choice of the length of the sub time step can reduce the CPU time. The newly introduced calculation of the CFL criteria in the routine `windgrdc.f90` (Yamartino, 2003) allows the determination of the maximum time step length, which can in many cases decrease the CPU time by a factor of two to three. However, the practical experience showed that using a time step of maximum length lead in a few cases to artificial “hot spots” especially in coastal areas with high gradients of in model grid heights. These “hot spots” may not be of concern for the long-term model result. However, they will be “interpreted” by the Kalman Filter as response of the model noise and may spoil the physical basis of the calculated covariance matrix. Therefore one has to ensure that no numerical artefacts occurred, either resulting from too large time stepping or unphysical changes of the model state and the modes by the analysis step or the rank reduction or by any other reason.

4.2 Reminder of numerical realization of P modelling

The most prominent feature of the Kalman Filter is the forecast of the covariance matrix P of the model error. According to the Kalman Filter equations (see Flemming, 2003 a) the forecast of P is given by the following equation matrix equation.

$$P_B^{t+1} = AP_A^t A^T + FQF^T \quad (1)$$

M denotes the linearised operator of the Chemical transport model A and FQF^T is a matrix of covariance of the model forecast error, which is added at every time step. It accounts for the noise caused by the deficiency of the model and its linearisation. In the RRSQRT (reduced rank square root) implementation matrix equation (1) is not solved directly (see Flemming, 2003 a for details). Instead the covariance matrix is represented by its square S ($P=SS^T$) and each column of S can be treated as a realization of a model state vector. The columns are called modes. The part APA^T in (1) is then approximated by n forecasts of the modes by the original CT-model code A at a given time step. The contribution of the model forecast error matrix FQF^T is approximated by introducing new modes (noise modes) which are obtained by a model forecast in which model parameters are altered to a certain extent. The steady increase of the number of modes in every time step is limited by means of a SVD of the covariance matrix: by means of a PCA n most important modes are constructed and be used for the next step of the P propagation. The altered parameters are considered to be the cause of the model noise. The degree of the change is expressed by the model noise factors. They can be included into the model state vector. This means that they are also subject of the analysis step.

4.3 Choice of noise factors and their implementation

While choosing the number of model noise parameters one should consider the whole range of important sources for the model uncertainty. However a high number of noise modes increases the number of necessary modes and with it the computational burden. As a rule of thumb, the total number of modes should be 5 to 10 times the number of noise modes (Velders, personal communication). Besides the computational aspects, a large variety of different model noise parameters can make the physical interpretation of the analysed model noise parameters difficult due to the many cross effects involved. For those reasons it was decided to apply not more than 3 to for noise modes in the current application.

The emissions are considered as one of main sources of uncertainty in chemical transport modelling. Therefore it seems to be a natural choice to include them in the specification of model noise parameters. In previous application of the Kalman filter with the LOTOS and EUROS model the changes in the ozone deposition velocity and NO_2 - and O_3 -photolysis rates have been used beside the emissions.

Due the fact that REM/CALGRID has a ground layer of a fixed depth of 20m the processes of vertical turbulent diffusion, which is controlled by turbulent vertical exchange coefficient $K_z(1)$, is of great important for the calculated ground layer concentrations. Moreover the mixing height plays a prominent role due the dynamical vertical resolution of the model.

In the given application, the following model parameters have been selected to represent the model noise:

- Total NO_x -emissions
- Total VOC – emissions
- Vertical turbulent exchange coefficient K_z

Having done this choice one has to decide to what extend a change of the model parameters should be introduced and if the changes should be applied only to a certain part of the model domain. The total emissions for different countries in Europe may have a different uncertainties or the K_z and mixing height may have different deficiencies over land and sea or in cloud free and cloudy conditions. The applications of the Kalman Filter with LOTOS and EUROS discriminated different areas in Europe for the modelling of model noise due to emissions. Due to the premise of limited computer power in the given application no spatial differences in the noise parameters have been applied.

The change of the 3/(4) model parameters was defined arbitrarily to be 25% percent. Although current estimates of the emission error (who to quote) of up to 50-100% a to large deviation from the original values could contradict the assumption about the zero mean error statistics in the Kalman Filter.

The 3/(4) new noise modes were established by taking the propagating the analysed model state by increasing the respective model parameters field, i.e. emissions, and K_z by 25%. Due the linear statistical assumption and “square” character of the covariances an increase or decrease of model parameters field would lead to

the same consequences in the calculation of P . In further studies there should be check if this is really the case since the original non-linear model does the approximation of the P forecast. It is questionable whether the change of the modelled concentration due to an emission increase is of the same size but opposite sign in the case of an equivalent emission decrease. It might be necessary to introduce both an increase and a decrease of the model parameters.

The noise factor for the change in the model parameter, i.e. 1.25 for the new noise modes and 1.0 for the analysed state and the rest of the modes, is included in the model state vector and is therefore subject of the analysis step (see section). The analysis step supplies certain optimal values of the 3 (4) noise parameters. At the present status of the KF implementation the analysed values will not be used a further in the temporal forecast of the modes or the model state. This means that temporal white noise properties of the induced model noise are enforced. It might be fruitful to investigate in further studies the influence of a temporal “memory” of the noise. In this case it could be necessary to introduce some form of dumping and correcting of the noise values because they could become negative, i.e. physical meaningless, or they could lead the analysed model state completely out of the track of the original model.

The problem of obtaining negative analysed noise parameters can be avoided by using the logarithm of the noise factor in the whole analysis procedure. It would increase the Gaussianity of the noise factor since it could now vary in the range between $-\infty$ and $+\infty$. The current noise factor can vary theoretically in the range between 0 and $+\infty$ and is highly skewed since the expectation value of it ought to be 1. From a practical point of view noise factors $\gg 1$ are not sensible.

4.4 Comparison with empirical modelling of observation errors (OI)

The empirical covariance modelling of the model error covariance is based on the differences between observations and the modelled values. The ensemble for the estimation is therefore limited to the station locations. In order to obtain a full covariance matrix an empirical covariance model has to be established. For the sake of simplicity in this application a classical homogeneous and isotropic covariance model has been used. Inhomogeneous covariance modelling has been applied already for passive data assimilation with REM3/CALGRID, which relies on the assumption of temporal stationarity of the error statistics (Flemming, 2003 c). Due to the fact that in active data assimilation the previous analysis step influences the current model state the assumption of the temporal stationarity of the error statistics seems to be not valid.

The KF produces much more comprehensive and temporal and spatial distinguished covariance matrix than the simple covariance function of OI. However, the covariance of the KF depends strongly on the choice and the scope of the variation of the perturbed model parameters.

The empirical covariance modelling in OI uses model errors directly obtained by comparison with the measurements. The measurement seems to be a direct indicator what might be the real atmospheric state. However the difference between observation and a CTM with the resolution of REM/CALGRID does not mean in any case a model failure due the different scales of the model and the observations. The comparison is limited by the simpler results of the OI covariance modelling. It include the of the model error, and a measure of the spatial structure of the covariance by means of a range which describes the decrease of covariance with increasing spatial distance. Figure 4.1 shows box whiskers – plots of the square root of model error variance, i.e. the covariance for zero distance, for every hour of the day. In the homogeneous OI approach this is one value for the whole field at every time step. For KF the variance is heterogeneous and in Figure 4.2 all values at station locations are depicted. The variances have a similar cycle with highest values in the morning and the afternoon. The magnitude of the KF and OI values are in the same range. The variation of the mixing layer height contributes most to modelled variances. KF values are smaller indicating that more model “noise” should be added. The same conclusion can be made by looking at the variances NO_2 .

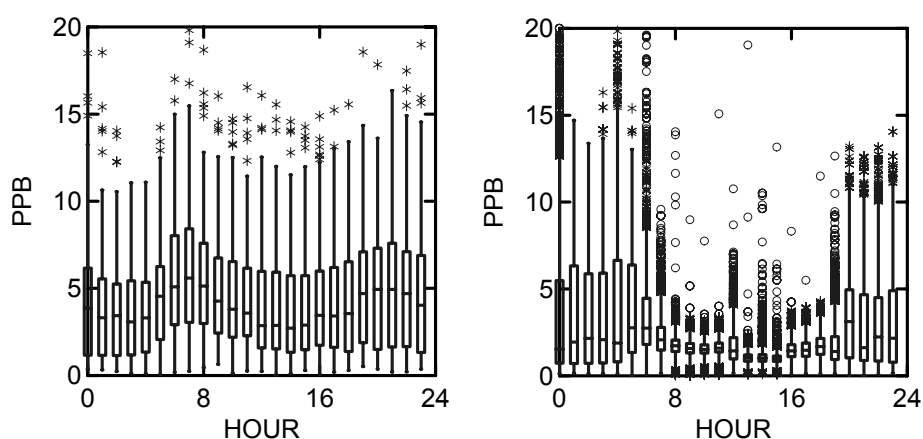


Figure 4.1 Daily cycle of the standard deviation of the model error variance of ground level O_3 obtained by empirical covariance modelling for OI (left) and by the Kalman filter assimilating NO_2 and O_3 (right)

Empirical estimated covariances decrease with increasing distance in a range of about 100 – 200 km. This feature cannot be found in the KF covariances (see Figure 4.2) as a consequence of the uniform variation of the model noise parameters. Assuming that a decrease of covariance with increasing separation has to be a feature of any spatial covariance one could impose this by applying a dumping function. This would lead to a more realistic spatially limited influence of the observations in the analysis step.

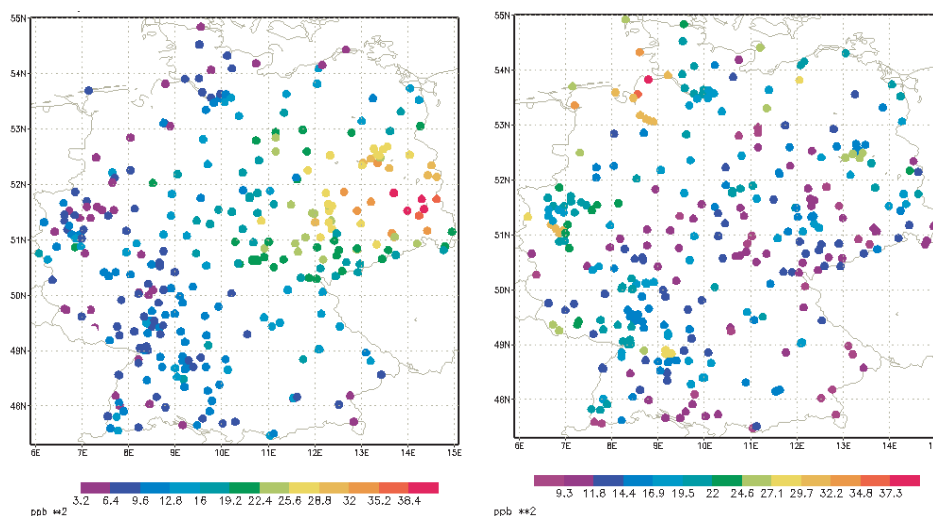


Figure 4.2 Spatial structure of covariance with station BB001 of ground layer O₃-field at 13.00 UTC for July 2001 obtained by empirical covariance modelling for OI (left) and by the Kalman filter assimilating NO₂ and O₃ (right)

4.5 Systematic patterns of analysed model noise parameters

The noise factors have been made part of the model state vector and are therefore analysed according to available concentration observations. This means that a regression type of relationship between the model error at the observation location and the noise factor, given by the covariance matrix of Kalman Filter, is used to find optimum values for the noise factors.

Analysed noise factor greater than 1 indicate that an increase of the model parameter would yield a better agreement with the measurements and vice versa. However this relationship is of pure statistical character and assumes linear responses between noise parameter and changes in the concentration fields and does not account for different levels of the concentrations. Further, the time period might be too short for a significant inference. The scatter of the analysed noise factors is very high.

Despite this limitation, it might be fruitful to investigate systematic features of the analysed model noise parameters. It may help to explore the reasons for the weaknesses of the model and its input data and finally to correct them.

Due to dynamical vertical resolution REM/CALGRID depends strongly on the daily variation of the PBL. Therefore it was examined whether the analysed noise parameters showed any systematic daily pattern. The median of all analysed noise factor (total NO_x-emission, total VOC-emission, vertical turbulent exchange coefficient KZ1) values for every hour of the day in the whole assimilation period (July 2001) is given in Figure 4.3, Figure 4.4 and Figure 4.5 respectively. They show the analysed noise values for the 3 different Kalman Filter runs assimilating either NO₂ and O₃, only O₃ only NO₂ (KF(NO₂,O₃), KF(O₃) and KF(NO₂)), see section 5.1). The most obvious daily pattern exhibits the vertical exchange coefficient in the run KF(NO₂,O₃). A strong decrease in the late morning (8-10 UTC) and the afternoon

(12-16 UTC) is suggested by the KF analysis. During the night the response is not very pronounced but indicates an increase in the vertical turbulent diffusion. The response in the noise parameters for the NO_x - and VOC- emission leads to a proposed increase of both emissions during the day and to a slight decrease during the night. This response point at the same direction as in the case of the vertical diffusion because both corrections would enforce an increase of the concentration of the primary species during the day and a decrease during the night. This conclusion corresponds to the fact known from REM/CALGRIDS performance evaluation that the concentrations of mainly primary species are too high during the night and too low during the day. Given the stronger response in the vertical diffusion, there is evidence that vertical diffusion is the main reason for the overestimation of the daily cycle by the model, i.e. the underestimation of the concentration level during the day. Corrections of the daily emissions time factors in the model or the total NO_x -emission seem to be less important. However, the overall sum of the analysed noise factors for the emissions is biased to values greater 1 which indicates that an increase in the total emissions would lead to a better model performance. The averaged analysed noise parameters depend on the choice of the assimilated species. This fact is a consequence of the non-linear nature of the problem and makes it more difficult to draw general conclusion from the analysed noise factors. Assimilating only ozone ($\text{KF}(\text{O}_3)$) shows weaker and partly different response of the vertical exchange coefficient. The decrease during the day is limited to 12-14 UTC. It is the time when the ozone concentration is overestimated. In the late afternoon strong increase is proposed which corresponds to the time of the underestimation of the ozone concentration. Like in the case of $\text{KF}(\text{NO}_2, \text{O}_3)$, the increase of diffusion is coupled with a decrease in the emissions and vice versa. In the case of $\text{KF}(\text{NO}_2)$ the relevant parts of the covariance matrix are not well developed and it is questionable whether the results are useful for any relevant conclusion. The analysed parameters do not show any strong signal.

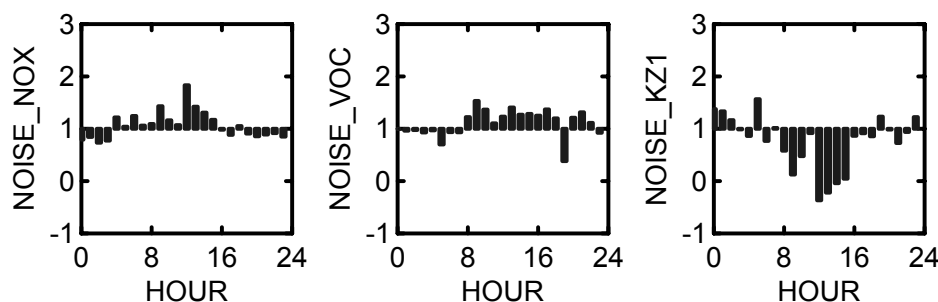


Figure 4.3 Daily cycle of the mean of the noise parameters for the Knox-emission (left), the VOC-emission (right) and vertical turbulent exchange coefficient $KZ1$ (right) for a KF run assimilating O_3 and NO_2 -observations

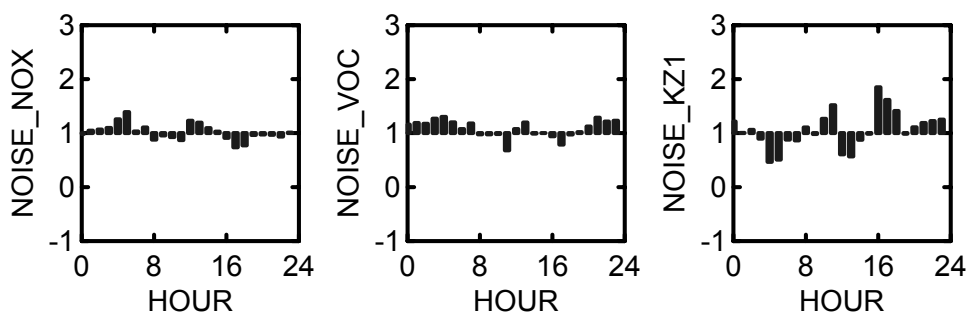


Figure 4.4 Daily cycle of the mean of the noise parameters for the NO_x -emission (left), the VOC-emission (right) and vertical turbulent exchange coefficient KZ1 (right) for a KF run assimilating O_3 -observations

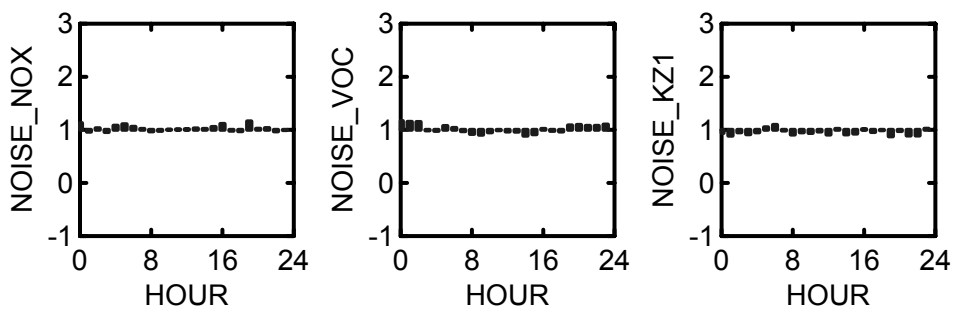


Figure 4.5 Daily cycle of the mean of the noise parameters for the NO_x -emission (left), the VOC-emission (right) and vertical turbulent exchange coefficient KZ1 (right) for a KF run assimilating NO_2 -observations.

5. Assimilation performance

5.1 Set-up of assimilation runs

Three KF runs with REM/CALGRID for July 2001 have been carried out, assimilating ozone, NO₂ or both. The model domain covers the area between -10.0° – 25.0° (Longitude) and 46.5° - 60.0° (Latitude). The horizontal resolution is based on a 0.5° x 0.25° grid. The dynamical 4-layer approach is used for the vertical resolution.

The simulation of the KF assimilation takes about 14 days by a 2000 Mhz- Athlon PC. Due to this high computational burden the number of test runs is very limited. The performances of 3 assimilation runs will be compared to an ordinary model run without assimilation of observations. In chapter 5.4 the Kalman Filter is compared to data assimilation with Optimum Interpolation.

The representation of covariance matrix was based on 27 modes. At every new time step 3 new noise modes were added, which were developed by an increase of 25% of the total NO_x-emission, the total VOC-emissions and the vertical turbulent exchange coefficient K_z between layer 1 (ground layer) and 2 (first half of mixing layer). The change was applied uniformly for in the whole model domain (see section 4.3). Propagating the current analysed state vector by the model with the respective changes in the model parameters developed the noise modes. The first analysis step, i.e. the adaptation to the measurement was executed after all 30 modes had been built up, i.e. after 7 hours. The actual ground layer grid box concentration at the end of the model time step was considered to be the simulated equivalent of the observations in that grid box. After a spin up time of 24 hours, the results of the KF assimilation after 24 hours were subjected to the performance evaluation.

Observations of rural, suburban and urban stations according to the classification of Flemming (2003 b) from the Germany operational air quality network have been assimilated at every hour of the run. This means that data of about 210 stations for Ozone and 190 for NO₂ have been used in the simulations. The assimilation of NO₂ observations is a new development since previous assimilation runs with LOTOS used only ozone observations.

For all assimilation runs, the individual climatological observation error variances for the German network according to Flemming (2003 b) have been used. The estimation of the observation error variance is based on the observational method (Hollingsworth and Lönnberg, 1986). Figure 5.1 shows the range of the estimated observation error standard deviation for NO₂ and O₃ for different air quality regimes. The typical observation error SD for ozone is about 5 ppb for all regimes indicating a smaller relative error for the regimes with higher mean ozone concen-

trations. For NO_2 the observation error increases with the level of pollution from about 2 ppb for rural to about 8 ppb for the traffic sites.

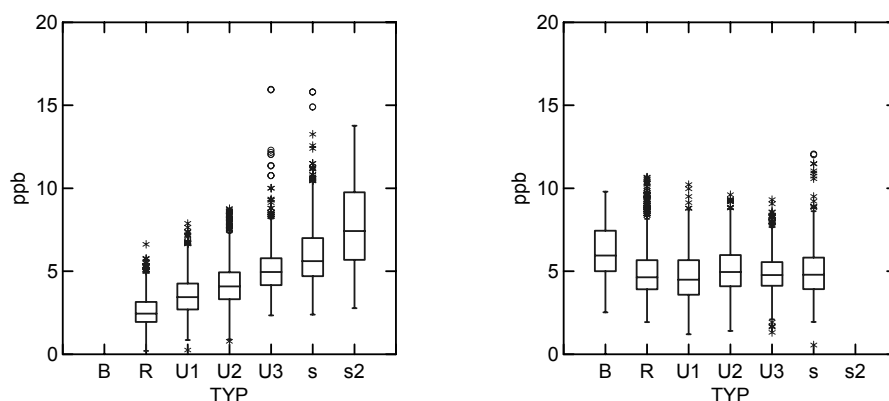


Figure 5.1 Estimated Observation error standard deviation in ppb vs. air quality regime (TYP) for all stations in Germany (left NO_2 , right O_3). B = mountain, R = rural, U1 = suburban, U2 = urban, U3 = urban, polluted, S = street, s2 = Street extreme

5.2 Methodology of performance evaluation

There is no clear methodology for performance evaluation of data assimilation because the classical juxtaposition of model results and observations is not possible any more. In weather forecasting the quality of the assimilated initial fields is examined by the improvement of the forecast, which is the goal of the assimilation efforts. The forecast improvement is not such a distinct criteria for CTM data assimilation, because initial conditions are not very important due to predominant sources (emissions) and sinks (deposition).

Besides noise parameter analysis, obtaining realistic concentration fields maps for air quality assessment is a useful application for CTM data assimilation. In order to check the results of the noise parameter analysis (see section 4.5) one has to apply their systematic patterns to correct, i.e. tune, the current values in the model. An improvement of the tuned stand-alone model performance would prove whether there is a benefit from data assimilation. This task remains for further studies with REM/CALGRID.

A proper assessment of the mapping performance can rely on a cross validation approach. This has been done for the Optimum Interpolation in Flemming (2003 c) and for the Kalman Filter in van Loon et al. (1999). In the cross validation approach the stations are divided in to 2 groups: one is assimilated and the other is used for the performance evaluation. This requires the choice which stations are used for each purpose. Hard core statisticians would argue that this choice has to be made completely randomly and that at least two assimilation runs have to be done

in which the assimilated stations are used for evaluation and vice versa. The performance would have to be of similar quality in both cross test.

The positive outcome of the cross validation is in many cases not more specific than the general statement “It works!” One does not know how good it works and how great the potential for further developments is. Moreover, the assimilation performance is obviously limited since only half of the observational information is used. The fact that the check of the performance is only possible at stations locations, e.g. not over the sea, is common to both data assimilation and classical model evaluation.

The objective of this study is the direct comparison of the different assimilation runs, based on the same observation error variances. Hence, the differences between the assimilation results are completely due to the different response to the assimilation of observations and the complexity of the scheme. That is reason why all stations of rural, suburban and urban characteristics were used in the assimilation and evaluation. The evaluation distinguishes between the different air quality regimes.

The following assimilation performance focuses on Ozone and NO₂. The influence of the assimilation on other species has not been checked. The performance evaluation is based on:

- RMSE
- RMSE/OBSVAR
- Bias/Mean
- Maps

RMSE is the most important quantity since the Kalman filters optimality fulfils a minimum variance criterion. The assumed observation error is the lower limit for this RMSE, which would be obtained by the “perfect” model or assimilation run. In order to get to know the potential of the assimilation effort is helpful to look at the ratio between RMSE and the observation error variance.

Since air quality assessment is based in many cases on temporal averages the average values have to be considered. Maps of the assimilated concentration field are supplied to get an impression of the spatial patterns of the air quality fields.

5.3 Assimilation performance for Ozone and or NO₂ in a run for July 2001

In total four different runs for July 2001 are compared in this chapter:

- Model run with REM/CALGRID **BAS**
- KF assimilation of ozone and NO₂ observations **KF(O₃NO₂)**
- KF assimilation of ozone observations **KF(O₃)**
- KF assimilation of NO₂ observations **KF(NO₂)**

The RMSE in respect to the ozone observations of the Kalman Filter assimilations runs KF(O₃) and KF(O₃NO₂) is reduced by more than 50% compared to the model simulation as its shown in Figure 5.2.

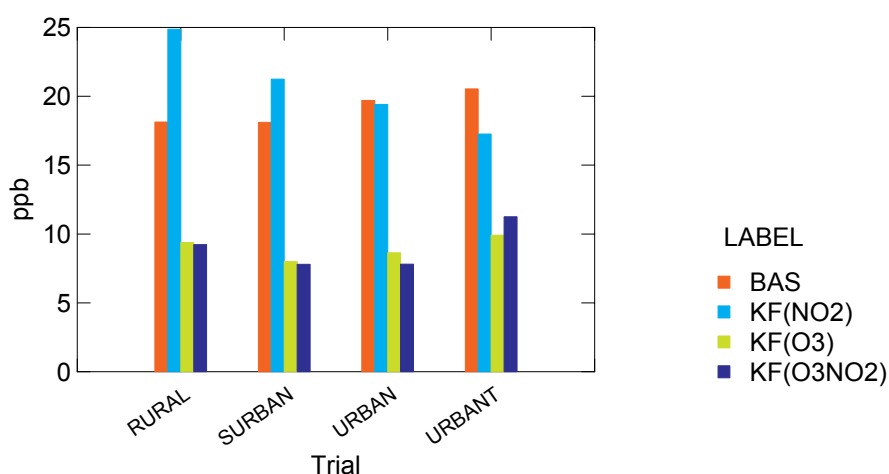


Figure 5.2 RMSE of ozone observations for the base run (BAS), the KF assimilation of NO₂-observations (KF(NO₂)), the KF assimilation of O₃- observations (KF(O₃)) and the KF assimilation of O₃ and NO₂ – observations (KF(O₃NO₂)) at rural, suburban, urban and polluted urban stations

The improvement can also be found for the stations of the air quality regime (“polluted urban”), which were not used for the assimilation. However assimilating only NO₂ does not yield an improvement and is even worse than the base run (BAS) at the rural stations. The observation error standard deviation SDOBS is about 5 ppb (see Figure 5.1) for all ozone regimes. In the runs KF(O₃NO₂) and KF(O₃) the ratio RMSE/SDOBS is reduced from about 5 to 2 compared to the base run. There is no clear evidence that the assimilated ozone field benefit from the assimilation from NO₂. The large overestimation of the model simulation (BAS) at the urban and suburban is significantly diminished by KF(O₃) and KF(NO₂O₃), see Figure 5.3.

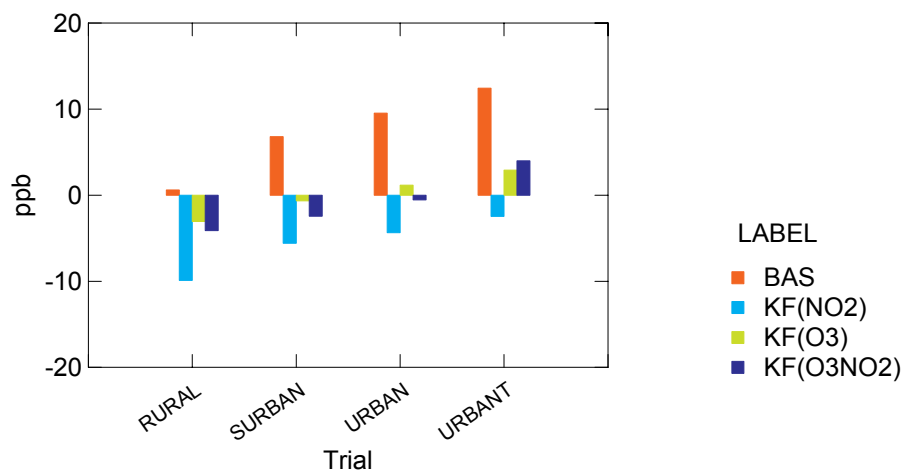


Figure 5.3 Bias (model – observation) of ozone observations for the base run (BAS), the KF assimilation of NO_2 -observations (KF(NO_2)), the KF assimilation of O_3 -observations (KF(O_3)) and the KF assimilation of O_3 - and NO_2 -observations (KF(O_3NO_2)) at rural, suburban, urban and polluted urban Stations

At rural station the assimilation lead to a small underestimation, which did not occur in the BAS run. The improvement of the RMSE in respect to the NO_2 -observations is not very pronounced (see Figure 5.4). All three assimilation runs, i.e. even KF(O_3), show some improvement for NO_2 .

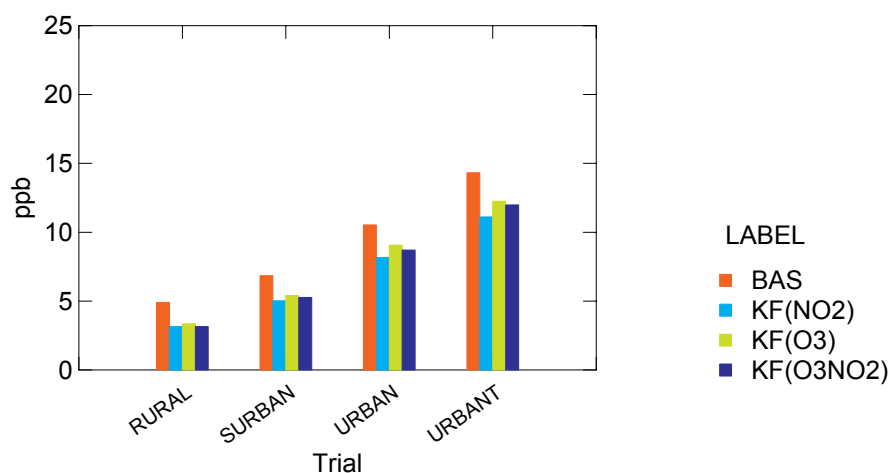


Figure 5.4 RMSE of NO_2 -observations for the base run (BAS), the KF assimilation of NO_2 -observations (KF(NO_2)), the KF assimilation of O_3 -observations (KF(O_3)) and the KF assimilation of O_3 - and NO_2 -observations (KF(O_3NO_2)) at rural, suburban, urban and polluted urban Stations

At all regimes the ratio between the RMSE and the standard variation of the observation error is rather closed to 1 in case of the assimilation (see Figure 5.5). The ratio is about 2 in case of the pure model simulation. This means that not much fur-

ther improvement can achieved by better techniques since the observation seem to be to “noisy“, i.e. not representative enough.

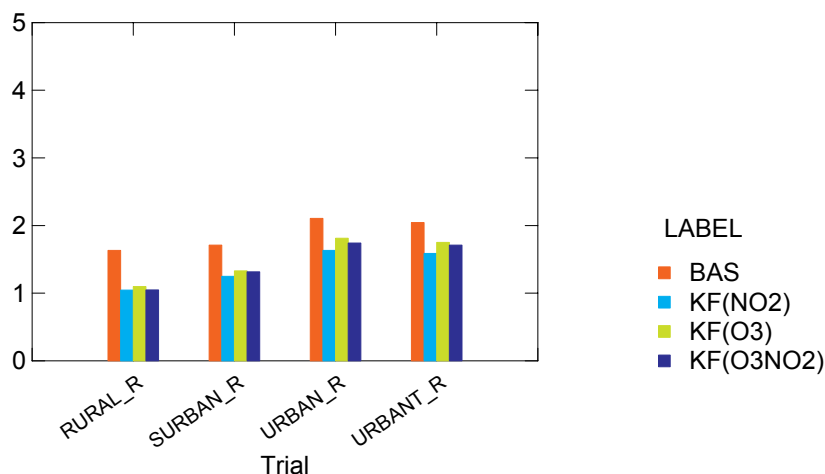


Figure 5.5 Ratio between RMSE and observation error standard deviation SDOBS of ozone for the base run (BAS), the KF assimilation of NO₂-observations (KF(NO₂)), the KF assimilation of O₃- observations (KF(O₃)) and the KF assimilation of O₃ and NO₂ – observations (KF(O₃NO₂)) at rural, suburban, urban and polluted urban Stations.

For ozone the decrease of the values by the assimilation is apparent in the whole model domain, c.f. Figure 5.6 and Figure 5.7. This means that the observation, which are restricted to Germany, have influenced the whole model domain. The homogeneous decrease is probably more a consequence of the long ranging spatial covariance by the given KF implementation (see discussion chapter 4.4) as a consequence of “transport” of the corrections in Germany by means of the model. Due to a lack of observations it is not possible to determine whether the decrease of the average ozone concentrations over the sea by data assimilation are realistic.

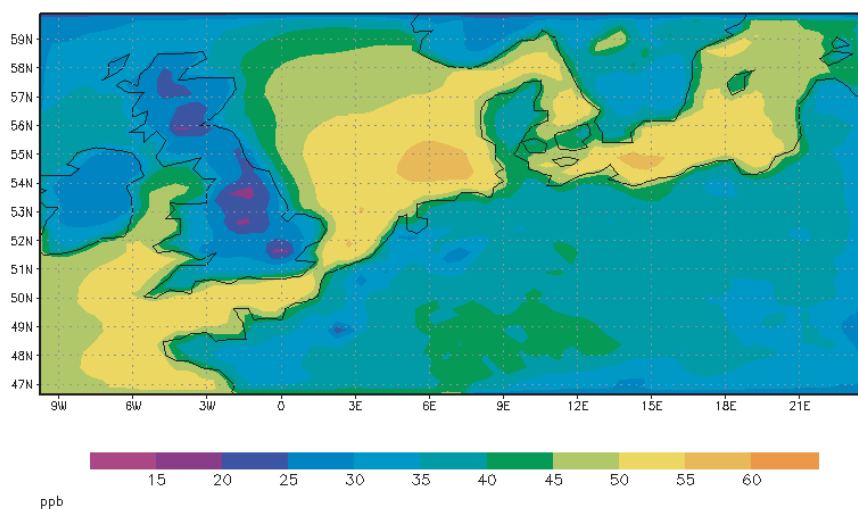


Figure 5.6 Average of O_3 - ground layer concentration for July 2001 from the base run, i.e. no assimilation

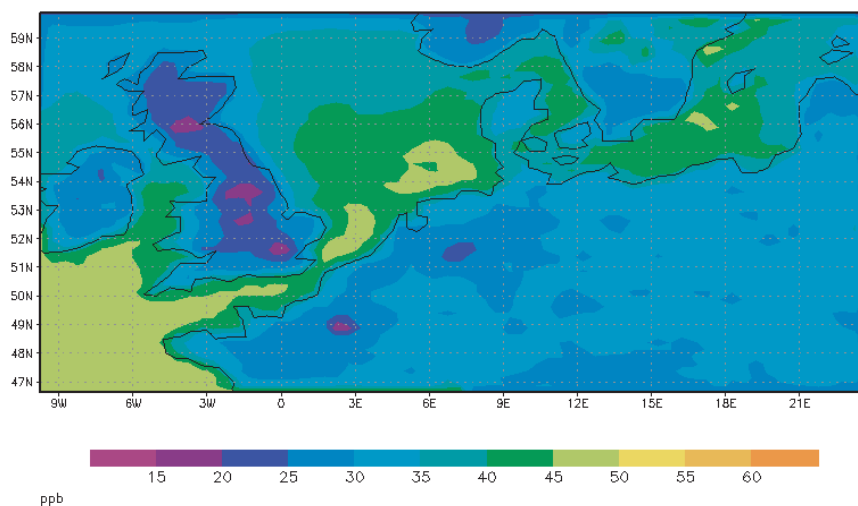


Figure 5.7 Average of O_3 - ground layer concentration for July 2001 from the KF assimilation of O_3 - observations in Germany

The areas of the highest NO_2 -pollution in the model domain are influenced by the assimilation, c.f. Figure 5.8 and Figure 5.9. The decrease of the very high model values in UK, Paris and the Ruhr area seems to be realistic.

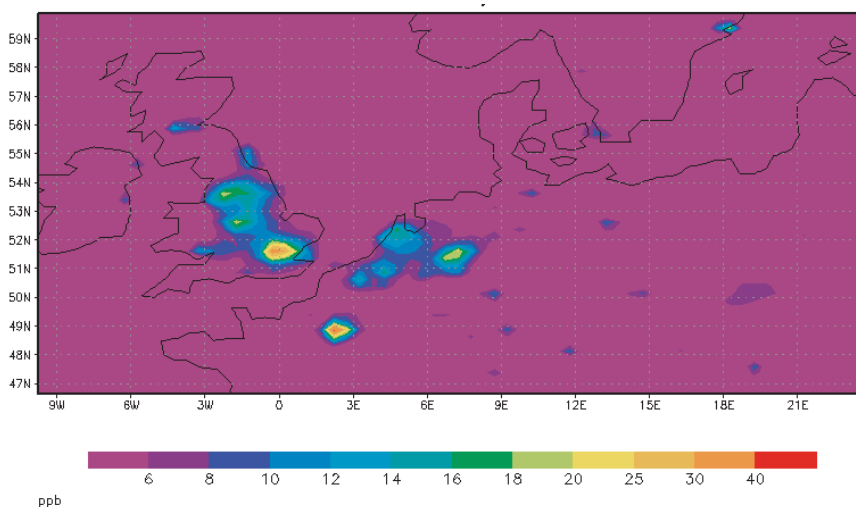


Figure 5.8 Average of NO_2 - ground layer concentration for July 2001 from the base run, i.e. no assimilation

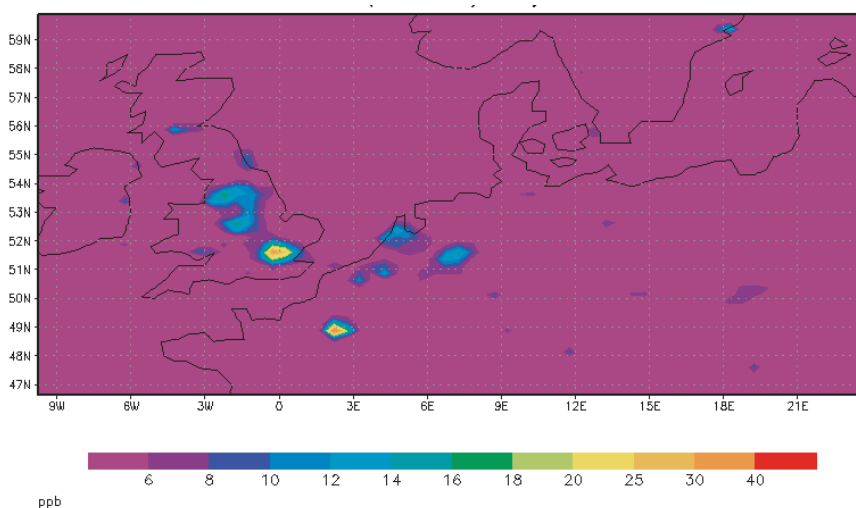


Figure 5.9 Average of NO_2 - ground layer concentration for July 2001 from the KF assimilation of O_3 and NO_2 - observations in Germany

5.4 The Kalman Filter versus Optimum Interpolation

In order to investigate the additional value of the KF relative to the OI scheme, in this Chapter the KF(O_3NO_2) simulation is compared to an OI(O_3NO_2) simulation, again for July 2001. A more detailed description of the differences between the schemes and their performance is given in Flemming and van Loon(2003) and in chapter 4.4.

In terms of RMSE and Bias the assimilation by OI is of almost equal quality as the KF assimilation.

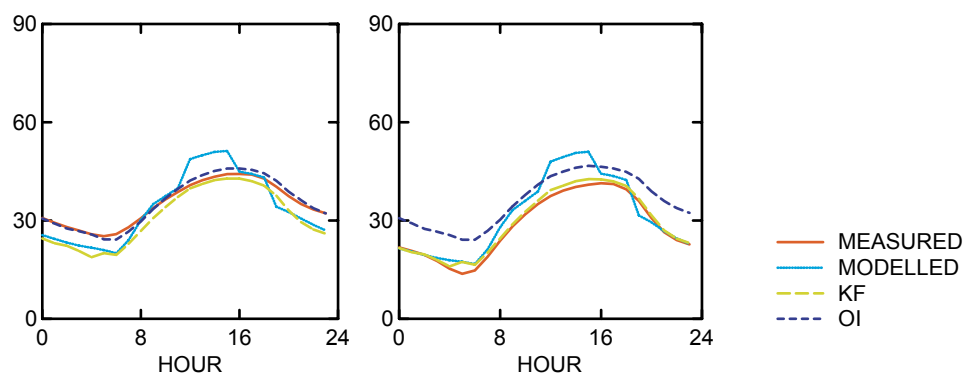


Figure 5.10 Mean daily cycles of the measured, modelled and assimilated (with OI and KF) ground level concentrations at rural (left) and urban (right) stations for Ozone.

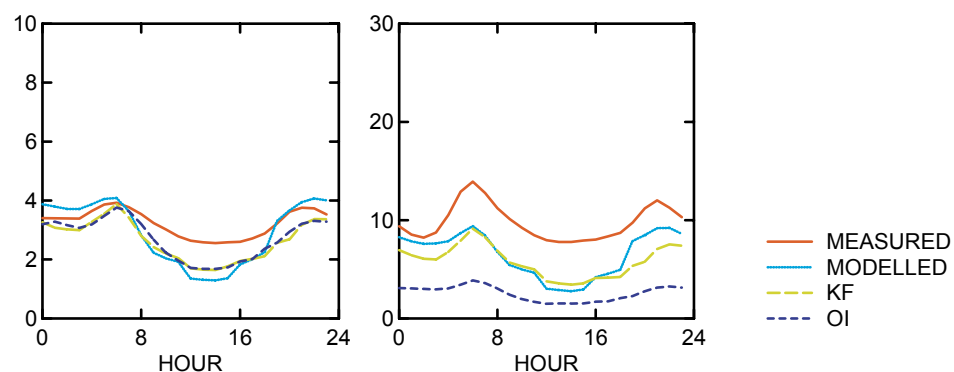


Figure 5.11 Mean daily cycles of the measured, modelled and assimilated (with OI and KF) ground level concentrations at rural (left) and urban (right) stations for NO₂.

Figure 5.10 and Figure 5.11 show the mean daily cycle of the measured, modelled and OI, respectively KF assimilated ground level concentrations at rural and urban stations for ozone (top) and NO₂.

The slightly better OI performance for the rural stations is mainly as a consequence of the bias correction based on the rural stations. KF performance for ozone is better in the urban regimes due to the better description of the inhomogeneity.

6. Conclusions and recommendations

In the beginning the project has severely suffered from the non-portability of the REM3/CALGRID code. This had mainly to do with binary input data in which various types of variables are mixed. It is therefore recommended to improve the I/O structure of the model. It should also be mentioned here that the model source is not very transparent, it is clearly a “research code” to which many people have contributed using different programming styles and using ad-hoc solutions etc. This caused that relatively much effort had to be put into connecting the REM3/CALGRID code to the KF routines. An overall revision of the model source code may be in place.

With respect to the results of the Kalman Filter as presented in this study the following conclusions are drawn:

- Application of Kalman Filtering to the REM3/CALGRID model using O₃ and/or NO₂ observations generally leads to improved agreement between simulated and observed concentrations.
- Covariances modelled as modelled by the KF and empirical covariances qualitatively agree. Due to the simple way the noise parameters are chosen, the spatial distributions are different.
- Results obtained with the KF are generally slightly better than with Optimum Interpolation. Given the simple noise parametrisation further performance can be expected with improving noise description.
- The Kalman Filter can help to detect typical deficiencies in the model parameters. The analysis suggest that a more turbulent exchange during the night and less during the day may improve the performance of the model.

Given the simple noise parametrisation, we consider the present results as very promising and further research is recommended into the following topics:

- More refined noise parametrisation
- Combination within the KF with empirical covariance modelling
- Reducing the computational costs of the KF system.
- Applying the systematic patterns of the analysed noise parameter of the vertical turbulent exchange coefficient to correct the current values in the model

7. References

- Bouttier, F. und Courtier, P., 1999, *Data assimilations concepts and methods*, in *Data assimilations concepts and methods*, eds. F. Bouttier und P. Courtier, Meteorological Training Course Lecture Series, ECMWF, Reading.
- Daley, R., 1991, *Atmospheric Data Analysis*, Cambridge University Press, Cambridge.
- Flemming, J., 2003, *Immissionsfelder aus Beobachtung, Modellierung und deren Kombination*, PhD-thesis FU-Berlin.
- Flemming, J., Stern, R. Reimer, E., 2002, *Data assimilation for CT-Modelling based on optimum Interpolation*, in *ITM Air pollution modelling and its applications XV*, eds. C. Borrego und G. Schayes, NATO CMS, Kluwer Academic / Plenum Publishers, New York.
- Flemming, J., 2003a, *Anwendung von Optimaler Interpolation und Kalman Filter Techniken für die Ausbreitungsrechnung mit dem chemischen Transportmodell REM/CALGRID*, Abschlußbericht zum Forschungs- und Entwicklungsvorhaben 298 41 252 des Umweltbundesamts "Modellierung und Prüfung von Strategien zur Verminderung der Belastung durch Ozon.
- Flemming, J., 2003b, *Klimatische Auswertung von modellierten und beobachteten Zeitreihen für Ozon, NO₂, NO SO₂ und Pm₁₀ in deutschland für die Jahre 1995-2001 und deren Charakterisierung nach Immissionsregimes mit einem Clusterverfahren*, Abschlußbericht zum Forschungs- und Entwicklungsvorhaben 298 41 252 des Umweltbundesamts "Modellierung und Prüfung von Strategien zur Verminderung der Belastung durch Ozon.
- Flemming, J., 2003c, *Immissionsfelder aus Beobachtung, Modellierung und deren Kombination*, PhD-thesis, <http://www.diss.fu-berlin.de/2003/71/>
- Flemming, J., van Loon, M. and Stern, R., 2003d, *Data assimilation for CT-Modelling based on Kalman Filter and optimum interpolation*, in *proceedings of 26th ITM on Air Pollution Modelling and its Application. May 26-30, 2003, Istanbul, Turkey.*
- Hollingsworth, A. and Lönnberg, P., 1986, *The statistical structure of short-range forecast errors as determined from radiosonde data, Part I: The wind field*, Tellus, 38:A, 111-136.
- Reimer, E. and Scherer, B., 1992, *An operational meteorological diagnostic system for regional air pollution analysis and long-term modelling*, in: *Air Pollution Modelling and its Applications IX.*, van Doop, H., ed., Plenum Press.

- Stern, R., Yarmatino, R. und Graff, A., 2003, *Dispersion modeling within the European community's air quality directives: long term modeling of O3, PM10 and NO2*, in proceedings of 26th ITM on Air Pollution Modelling and its Application. May 26-30, 2003, Istanbul, Turkey.
- Van Aardenne, J., Builtjes, P., Hordijk, L., Kroeze, C. and Pulles, M., 2002, *Using wind-direction-dependent differences between model calculations and field measurements as indicator for the inaccuracy of emission*, Atmospheric Environment, V. 36, p. 1195-1204.
- Van Loon, M. and Heemink, A.W., 1997, *Kalman Filtering for non linear atmospheric chemistry models: First experiences*, Technical Report MAS-R9711, CWI, Amsterdam.
- Van Loon, M., Builtjes, P.J.H. and Roemer, M.G.M., 1999, Reactive trace gas data assimilation – Final Report. BCRS publication USP-2 99-12.
- Van Loon, M., Builtjes, P. and de Leeuw, G., 2002, *PM2.5 concentrations over Europe combining satellite data and modelling*, in ITM Air pollution modelling and its applications XV, eds. C. Borrego und G. Schayes, NATO CMS, Kluwer Academic / Plenum Publishers, New York.
- Yamartino, B., 2003, *Refined 3-d Transport and Horizontal Diffusion for the REM/CALGRID Air Quality Model*, Abschlußbericht zum Forschungs- und Entwicklungsvorhaben 298 41 252 des Umweltbundesamts "Modellierung und Prüfung von Strategien zur Verminderung der Belastung durch Ozon.

8. Authentication

Name and address of the principal:

Freie Universität Berlin
Institut für Meteorologie

Names and functions of the cooperators:

M. van Loon
J. Flemming (Free University of Berlin)

Names and establishments to which part of the research was put out to contract:

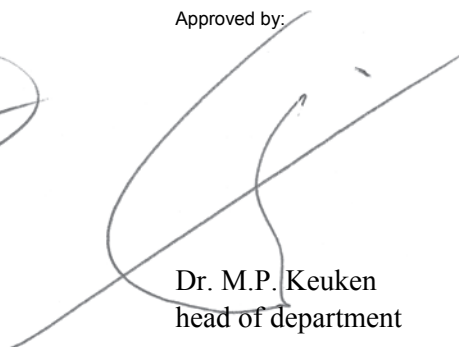
Date upon which, or period in which, the research took place:

Signature:



Dr.Ir. M. van Loon
projectleader

Approved by:



Dr. M.P. Keuken
head of department

Appendix A Technical description of the KF code

A.1 Description of the Kalman Filter routines as implemented around REM3/CALGRID

The original routines are all written in Fortran 90 (filenames have the extension .f90). For the implementation around REM3/CALGRID it was necessary to use Fortran 77 for one of the source files (kf_driver.f, extension .f instead of .f90), because in this file the actual connection with the model sources code is established. Recall that REM3/CALGRID is programmed in Fortran 77. The KF implementation consists of the following modules (all filenames associated with the KF start with kf_):

Source file name	Short description
kf_driver.f	Performs the connection of the KF with the model. It contains parts of the original source of REM3/CALGRID in order to bring the model in the form of Algorithm 2.
kf_state.f90	Describes the model state, in this case the concentration vector augmented by an array with output for a number of compounds at specific measurement locations and augmented with an array containing the noise parameters.
kf_output.f90	Performs the output of assimilated concentrations as well as KF specific parameters, like the estimated values of the noise parameters.
kf_reduce.f90	Performs the reduction of the covariance matrix (see theory), using a singular value decomposition (SVD). For the SVD routines from the publicly available numerical library LAPACK are used. This library is installed at FUB.
kf_meas.f90	Performs the reading and proper passing of the observations to the KF routines.
kf_noise.f90	Sets and resets the noise parameters. This routine needs to be edited for adding/removal of noise processes
kf_main.f90	The main program.

Below the source of kf_main.f90 is listed. Some explanation for some of the lines is given after the source listing.

```
1  program kf_RCG_main
2
3  ! using module driver gives access to all common block
4  ! variables, since they are all included in the
5  ! module driver
6
7  use driver
8  use state
9  use noise
10 use reduce
11 use measurements
12 use output_kf
13
14 ! the number of the time step
15 integer :: istep
16
17 ! first step or not
18 logical :: lfirst
19
20 ! the actual number of modes
21 integer :: nmodes
22
23 real :: noisefactor(nnoise)
24
25 ! set noise factors
26 do i=1,nnoise
27   noisefactor(i)=1.25
28   if(i == 4)noisefactor(i)=0.75
29 enddo
30
31 ! flag for very first time step
32 lfirst = .true.
33 !
34 ! set number of modes
35 nmodes = 0
36
37 xmast=0.0
38 ! init the program
39 print *, 'initializing the simulation..'
40 call init_model
41
42 ! init the Kalman state vector
43 call init_state(.true.)
44
45 ! init the measurements
46 print *, 'initializing the measurements..'
47 call init_meas(.true.)
48
49 print *, 'initialized the model simulation..'
50 print *, '#time steps to be done: ',irlg
51
52 ! read first day of measurements
53 print *, 'reading first day of measurements. njul = ',njul
54 call read_meas(njul,nyr)
55
56 ! open some output for fields of the modes and the covariances...
57 call open_output_kf(347,'kovar_grads'//trim(ver)//'.bin',nx*ny)
58 call open_output_kf(345,'modes_grads'//trim(ver)//'.bin',nx*ny)
59
60 ! the time integration loop
61 do istep = 1, irlg
62
63   ! save old grid for recalling zjump (Rem3/CALGRID source)
64   call save_htface_density_old
65
66   ! prepare timestep
67   call prepare_timestep(istep,1.0)
68
69   ! put initial values in x and xb
70   if (lfirst) then
71     call save_htface_density_old
72     call fill_state(x)
73     call fill_state(xb)
74     x(is_noise:is_noise+nnoise-1) = 1.0
75     xb(is_noise:is_noise+nnoise-1) = 1.0
76   endif
77
78   do imodes = 1,nmodes
79     ! the modes are deviations from the mean, add the mean for processing
80     modes(:,imodes) = x(:) + modes(:,imodes)
```

```

81     enddo
82
83     x(is_noise:is_noise+nnoise-1) = 1.0
84
85     ! now, do the time stepping for the state vectors
86     call fill_master2(xb(1:ilength1),xmast(ifulcon))
87     print *, 'updating base state'
88     call put_htfaceold
89     call perform_timestep(istep)
90     call fill_state(xb)
91
92     print*, 'updating state'
93     call fill_master2(x(1:ilength1) ,xmast(ifulcon))
94     call put_htfaceold
95     call perform_timestep(istep)
96     call fill_state(x)
97
98     do imodes=1, nmodes
99         print *, 'updating mode ',imodes
100         call fill_master2(modes(1:ilength1,imodes),xmast(ifulcon))
101         call put_htfaceold
102         call perform_timestep(istep)
103         call fill_state( modes(:,imodes) )
104         ! put noise parameter to one !!!!!
105         modes(is_noise:is_noise+nnoise-1, imodes) = 1.0
106     enddo
107
108     do imodes=nmodes+1, nnoise+nmodes
109         call fill_master2(x(1:ilength1), xmast(ifulcon))
110         print *, 'creating new noise mode ',imodes,imodes-nmodes
111         call setnoise(imodes-nmodes, noisefactor(imodes-nmodes))
112
113         if(imodes-nmodes.eq.4) then
114             ! initial value is the present state vector
115             print *, 'creating new noise mode for mixing height',imodes
116             ihr_metfub=ihr_metfub-1 ! read the same time step again
117
118             call put_densityold
119             call put_htfaceold
120             call prepare_timestep(istep,noisefactor(imodes-nmodes))
121             !no copying of xmast_save
122             xmast(ifulcon:iflcon+ilength1-1)=x(1:ilength1)
123         endif
124
125         call put_htfaceold
126         call perform_timestep(istep)
127
128         ! result must be in modes
129         call fill_state( modes(:,imodes))
130         ! reset the noise
131         call resetnoise(imodes-nmodes)
132         ! put noise to zero
133         ! fill the nonzero one
134         modes(is_noise:is_noise+nnoise-1, imodes) = 1.0
135         modes(is_noise+imodes-nmodes-1, imodes) = noisefactor(imodes-nmodes)
136     enddo
137
138
139
140     ! update the actual number of modes
141     nmodes = nmodes + nnoise
142
143     ! subtract the state vector from the modes
144     do imodes = 1,nmodes
145         modes(:,imodes) = modes(:,imodes) - x(:)
146     enddo
147
148     ! reduce the number of modes, if necessary
149     if (nmodes > maxmodes) then
150         print *, 'reducing rank of covariance matrix'
151         call reduce_rank( nmodes )
152     endif
153
154     ! finish the time step
155     ! this updates the time as well!
156     ! We need the present time, because its the end of the 1h averaging
157     ! interval, similar as in the measurement files
158     call finish_timestep
159     print *, 'finished time step',njul,nyr
160

```

Appendix A

```
161      ! output of modes amd state
162
163      call output_kf_feld_grads(345,modes(1:ilength1,1)) ! mode 1
164      call output_kf_feld_grads(345,modes(1:ilength1,2)) !
165      call output_kf_feld_grads(345,modes(1:ilength1,3)) !
166      call output_kf_feld_grads(345,x(1:ilength1)) ! 03 state
167      call output_kf_feld_grads(345,xb(1:ilength1)) ! bcground
168
169      lfirst = .false.
170
171      ! read new measurements if necessary
172      if (njul > njulold) then
173          print *, 'reading measurements for julian day ',njul
174          call read_meas(njul,nyr)
175      endif
176
177      !update the state vectors, x and xb, and the existing modes
178      if(nmodes == maxmodes) call measupdate( nmodes, nhr, njul)
179
180      call output_kf_feld_grads(345,x(1:ilength1)) ! 03 state
181
182      ! cjohn put averaged and assimilated state vector in xmaster for output
183      call fill_master_conv(x)
184
185      !output
186      call output(xmast(iconavg),xmast(idryflx),xmast(itopflx), &
187                xmast(ivd),xmast(iemt0t),istep)
188
189      lfirst = .false.
190
191  enddo
192
193      ! close the model
194      print *, 'closing the model'
195      call close_model
196
197  end program kf_RCG_main
```

Line	Explanation
7	The module driver contains all global variables from RCG
26-29	Noise factor are the factors by which some parameters are disturbed. Should be done by the user/modeller and should match the noise processes defined in <code>kf_noise.f90</code>
35	The number of modes is initialised at zero: no apriori information is taken into account. The system builds up its own covariance matrix
40	Initialisation of the model, using the appropriate RCG routines
43	The state vector of the Kalman Filter is initialised (allocated, based on RCG dimensions)
47	The files with observational data are opened. Station characteristics are read etc.
54	Measurements are read per day, for a given year and julian day
61	The loop over the number of hours that need to be simulated (<code>irlg</code>). This follows from the input to RCG.
64	The layer heights and the density need to be saved, since RCG overwrites them. However, the old values are needed for the modes as well.
67	Subroutine <code>prepare_timestep</code> does whatever is necessary for carrying out a time step with RCG.
72,73	<code>x</code> is state vector of the KF into which the observations are assimilated. <code>xb</code> is an additional state vector to which no assimilation is applied. In this way the effect of the assimilation can be seen immediately.
74,75	When propagating the state vector no noise should be taken into account (see theory), hence the noise values need to be 1.
86, 93, 100, 109	The subroutine <code>fill_master2</code> fills the master array of RCG with the proper concentration vectors. This is necessary, since this master array (<code>xmast</code>) is part of a common block and thus concentration arrays cannot be given to as an argument to the RCG time stepping routines without major changes in the source code of RCG.
88, 94, 101, 110	The time-stepping mechanism by RCG automatically update the layer heights (“ <code>htface</code> ”), so the old values may need to be put back.
98-106	The existing modes are updated (see theory). No noise needs to be set here.
108-137	New noise modes are created (see theory). Here noise needs to be set.
113-123	Noise parameter 4 (the last one in this case) applies to the mixing layer height. Since this affects other meteorological parameters as well, the meteo fields need to be made again. This is achieved by calling <code>prepare_timestep</code> .
141	New modes have been created, so the actual number of modes need to be updated.
144-147	The modes are defined as deviations from the mean (see theory) so the mean is subtracted.

Line	Explanation
149-152	If the number of modes exceeds the maximum number it is reduced to this maximum number by means of a singular value decomposition. Apart from a number of time steps in the beginning of the simulation, when the covariance matrix is building up, the rank reduction is performed every time step.
158	Everything is done with respect to the time propagation of the state vector and the modes, so the time step can be finished. This mainly consists of updating time variables.
172-175	If we entered a new day, new measurements need to be read.
178	The measurements for this specific date and hour are processed. This is only done if the covariance matrix has already been built up, i.e. when the number of modes is equal to maxmodes.

A.2 REM3/CALGRID code preparation for connection to the Kalman Filter routines

In order to be able to incorporate updates of the REM/CALGRID code in KF program a description is given how the original code has to be changed for use in the KF code framework. The objective is to use as much of the original REM/CALGRID code as possible.

Following the structure of Algorithm 2 (see section 3.2.1), the code of the model REM/CALGRID is split into four subroutines which are called by the main routine (kf_main.f90) of the KF program. The subroutines are put in to the module kf_driver.f. In the table below a specification is given of these routines and which parts of the original code they contain.

Subroutine in kf_driver.f	Contains which original source
init_model	main000.for (i.e. the call of routine setup) and the parts of comp_fub.for until the basic time integration loop 100
prepare_timestep	content of the basic time loop 100 in comp_fub.for until the sub time step loop 50. It includes meteo and emission processing
perform_timestep	complete time step loop 50 of comp_fub.f90, which calculates advection, diffusion and chemical conversion
finish_timestep	rest of comp_fub.for after loop 50

Remarks

- The basic time loop 100 over the hour of the run is integrated in the module `kf_main`.
- The only change in the sequence of the original REM/CALGRID code is a call of the subroutine `zjump2` just in front of the time loop 50, which is put into `perform_timestep`. `Zjump2` does the calculation of new grid height and the re-portion of the species mass according to the new grid, which has to be done for all modes. Due to the fact that `zjump2` is already called in `prepare_timestep` the old grid height have to be stored and put back in the model master state vector `xmast` before every call of `perform_timestep`. This is done by the subroutines `save_htfaceold` and `put htface_old`. They are part of the module `kf_state` and are call in the `KF_main.f90`. The call of `zjump2` in `prepare_timestep` cannot be omitted because the new grid is needed for the processing of the meteorological fields.
- The rest of `comp_fub.for` after loop 50 is put in the subroutine `finish_timestep`. The call of output has been removed to the `kf_main.f90` in order to insure that the analysed state vector is written to the output file.
- In order to split `comp_fub.for` in the 3 routines `prepare_timestep`, `perform_timestep` and `finish_timestep` one has to ensure that all implicit variables have been made explicit in the definition section of the module `kf_driver.f`. Next, one has to avoid that time variable `tin` does not accumulate by the several calls of `perform_timestep`. Time updates etc. have therefore been removed from `perform_timestep` and put into `prepare_timestep` and `finish_timestep`.

A.3 Description of output options REM/CALGRID KF

The general structure of the output file management in the KF version is the same as in the CTM stand-alone version. The species selected for output in the input control file (*.inp) are written to concentration output file (CONC). They are the averaged values from the analysed model state vector **x**. The output is done after the analysis step in `kf_main.f90`.

For additional diagnosis of the KF performance several new out put data files have been introduced (xxx refers to a run label which is taken from the name of the concentration output file `conc_xxx_*.*`):

- output of the analysed values of the noise parameters for every hour (**filter_noisexxx.txt**)
- output of the following quantities for every assimilated species at every station and hour of the run (e.g. **filter_NO2_xxx.txt**): measured value, assimilated value, modeled value without assimilation, modelled value before assimilation, model error variance, model error covariance with station first station in station list)

- grads files (**modes_grads.bin/ctl**) with the following actual : 1st - level concentration fields for every assimilated species and hour of the run: 3 most important modes, base model run without assimilation, model run with assimilation before and after the analysis step.
- grads files (**kovar_grads.bin/ctl**) with the following 1st - level covariance/variance fields for every assimilated species and hour of the run: variance, covariance between actual and averaged values at every grid point, covariance with grid point of first station in station list
- additional ascii files to report the size of the eigenvalues of the covariance matrix