Data assimilation in atmospheric chemistry models using Kalman filtering

Arjo Segers

Data assimilation in atmospheric chemistry models using Kalman filtering

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen

op dinsdag 2 april 2002 om 16:00 uur

door

Adrianus Johannes SEGERS

wiskundig ingenieur
geboren te Gouda.

Dit proefschrift is goedgekeurd door de promotoren:
Prof.dr.ir. A.W. Heemink
Prof.dr.ir. P.J.H. Builtjes

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. A.W. Heemink,	Technische Universiteit Delft, promotor
Prof.dr.ir. P.J.H. Builtjes,	Universiteit Utrecht, promotor
Prof.dr. R.M. Cooke,	Technische Universiteit Delft
Prof.dr. H. Kelder,	Technische Universiteit Eindhoven
Prof.dr. J.G. Verwer,	Universiteit van Amsterdam
Dr.ir. M. van Loon,	TNO-MEP, Apeldoorn
Dr.ir. M. Verlaan,	Technische Universiteit Delft

Published and distributed by: DUP Science

DUP Science is an imprint of
Delft University Press
P.O. Box 98
2600 MG Delft
The Netherlands
Telephone: +31 15 27 85 678
Telefax: +31 15 27 85 706
E-mail: DUP@Library.TUDelft.NL

ISBN 90-407-2286-2

Keywords: data assimilation, atmospheric chemistry, Kalman filter

Copyright © 2002 by Arjo Segers

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the publisher: Delft University Press.

Printed in The Netherlands

Contents

1 Introduction **1**
1.1 Air pollution: from local to global 1
1.2 Tropospheric ozone 3
  1.2.1 Problems related to ozone 3
  1.2.2 Measurements 3
  1.2.3 Ozone forecast 5
  1.2.4 Ozone models 6
1.3 Data assimilation 7
1.4 Motivation and overview 8

2 Tropospheric chemistry: the LOTOS model **11**
2.1 Introduction 11
2.2 Chemistry of tropospheric ozone 13
  2.2.1 Nitrogen oxides 13
  2.2.2 Volatile organic compounds 15
  2.2.3 NO_x- and VOC-limited regimes 16
  2.2.4 Methane and carbon monoxide 16
2.3 Vertical structure of the atmosphere 17
  2.3.1 Stratosphere/troposphere 17
  2.3.2 Free troposphere/boundary layer 18
  2.3.3 Mixing/stable/residual layer 19
2.4 The LOTOS model 20
  2.4.1 Model domain and grid 20
  2.4.2 Vertical exchange 21
  2.4.3 Vertical diffusion 22
  2.4.4 Emissions 22
  2.4.5 Chemistry 24
  2.4.6 Deposition and surface concentrations 25
  2.4.7 Advection and horizontal diffusion 26
  2.4.8 Example: budgets around Vreedepeel 27

3 Data Assimilation **31**
3.1 Introduction 31
3.2 Variational methods 32
3.3 Linear filters 34
  3.3.1 Direct insertion and blending 34
  3.3.2 Optimal Interpolation 35
  3.3.3 Kalman filter 37

3.4 Kalman filter versus variational methods 39
3.5 Stochastic model for the Kalman filter 41
3.6 The Kalman smoother 43

4 Application, part I: UK experience 47
4.1 Introduction 47
4.2 Experimental setup 49
4.3 Assimilation with simulated data 51
  4.3.1 Uncertainties in emissions 51
  4.3.2 Uncertain photolysis rates 55
  4.3.3 Uncertainties in deposition parameters 60
  4.3.4 Combination of emissions, photolysis rates, and deposition 62
4.4 Assimilation with data from observation network 64
  4.4.1 Uncertain NO_x and VOC emissions 65
  4.4.2 Uncertain emissions, photolysis, and deposition 68
  4.4.3 Forecast of ozone level 71
4.5 Summary and conclusions 72

5 Application, part II: European domain 75
5.1 Introduction 75
5.2 Domain and period 76
5.3 Stochastic model and filter 78
5.4 Assimilated ozone 79
5.5 Parameter estimation during assimilation 83
5.6 Forecast of the ozone maxima 85
5.7 Discussion and conclusions 87

6 Low-rank filters 89
6.1 Introduction 89
6.2 Kalman filter with full covariance matrix 91
6.3 Factorization of the covariance matrix 92
6.4 Kalman filter in square root form 94
6.5 Examples of factorized filters 95
  6.5.1 RRSQRT filter 96
  6.5.2 SEEK and SEIK filter 97
  6.5.3 ESSE: Error Subspace Statistical Estimation 98
6.6 Ensemble filter 99
6.7 POEnK Filter 101
6.8 General formulation of low-rank filter 102
6.9 Modification of reduction algorithm 106
  6.9.1 Unit-invariant reduction 106
  6.9.2 Amplification of important correlations 107
  6.9.3 Summary of transformations 108
  6.9.4 Experiments and results 108
6.10 Comparison of filter techniques 111

6.11 Summary and conclusions 116

7 Nonlinear filters **119**
7.1 Introduction 119
7.2 Nonlinear stochastic model and filter equations 121
7.3 Forecast methods for low-rank filters 122
  7.3.1 First order linearizations 124
  7.3.2 Second order linearizations 125
  7.3.3 Ensemble forecast 129
  7.3.4 Summary of forecast schemes 130
7.4 Application to LOTOS 130
7.5 Measuring nonlinearity 135
  7.5.1 Tracking the bias 135
  7.5.2 Experiments with nonlinearity measures 136
7.6 Summary and conclusions 139

8 Parallelization of low-rank filters **141**
8.1 Introduction 141
8.2 Parallel computing 142
8.3 Filter equations and covariance decomposition 145
8.4 Parallel filter: decomposition over the modes 146
  8.4.1 Adding distributed states 147
  8.4.2 Operations on the covariance square root 147
  8.4.3 Performance of the mode-decomposed filter 152
8.5 Parallelization of the LOTOS model 154
  8.5.1 Decomposition of the LOTOS domain 155
  8.5.2 Operator splitting 158
8.6 Parallel filter: decomposition over the domain 160
  8.6.1 Operations on domain-decomposed states 161
  8.6.2 Operations on covariance square root 161
  8.6.3 Performance of domain-decomposed filter 163
8.7 Comparison between parallelization strategies 164
8.8 Discussion 166
8.9 Conclusions 168

9 Discussion and conclusions **169**

A Gasphase reactions **173**

B Repeated scalar update **177**

C Taylor expansions **179**
C.1 Notations and conventions 179
C.2 First order linearizations 180
C.3 Second order accurate forecast 182
C.4 Ensemble forecast 183

C.5 Bias equation 184

D Minimal Exact Sampling **185**

E Symbols and notations **187**
E.1 Vectors and matrices 187
E.2 Mathematical accents 187
E.3 Definitions 187
E.4 Symbols 187

Acknowledgments **191**

Bibliography **193**

Summary **201**

Samenvatting **205**

Dankwoord **209**

Curriculum Vitae **211**

Chapter 1

Introduction

1.1 Air pollution: from local to global

The amount of attention nowadays paid to smog, the ozone layer, and greenhouse gases could easily suggest that *air pollution* is a late twentieth-century invention. However, concern about the quality of the air surrounding us has a long history. The importance of air for human beings was recognized far before 18^{th} century scientists like Rutherford and Lavosier discovered its chemical composition. Together with earth, water and fire, air belonged to the four elements Aristotle considered to be the four fundamental components of every existing material. Less philosophical, the importance of air for living was probably known by every human in history, since it is hardcoded in our brain: just put your head under water, and there will come a moment when you are desperately seeking for air. For sure, the quality of the air has been recognized as important too, since domestic burning places found in excavations are often combined with some form of ventilation. Although historical records are hard to find, the invention of the chimney was probably the first form of air pollution regularisation ever.

The relation between open fire and air pollution is easily made if smoke is released. The many small particles in smoke are clearly visible and tend to irritate the lungs and eyes, warning you about the quality of the air (unless the smoke contains nicotine which may suppress this natural reaction). Much of the mass released from a fire is invisible however. A wellknown example is carbon monoxide, released from primitive cooking or water-heating devices due to incomplete burning. Accidents with carbon monoxide contamination are nowadays quite rare in our country, but still a major problem in developing countries (*Encalada et al., 1998*). The solution to this form of air pollution is both simple and cheap: ventilation. Polluted air is mixed with large amounts of clean air to decrease the concentration of unhealthy components. Ventilation also decreases the unhealthy effect of an open fire by supplying oxygen, leading to cleaner combustion with less release of carbon monoxide and smoke particles.

Before the industrial revolution, any form of air pollution could be treated by mixing it with clean air, since clean air was available on an almost infinite scale. The amount of polluted air released from antropogene sources used to be negligible, apart from some cases of slash-and-burn agriculture. The effects of air pollution used to be limited to a small area close to the source. The development of industrial activities however, required huge quantities of energy for driving machines and metal production. Energy became first available from large-scale combustion of coal. After the invention of the gasoline engine at the end of the nineteenth century, the use of oil showed an exponential growth, also due to the growth of the population. The amount of fuel used in the pre-industrial time pales to insignificance besides the coal, oil and natural gas combusted nowadays, and the impact of the related air pollution has become a matter of concern. The constant release of waste gases from combustion and other human activities has led to measurable changes in the air quality, not only close to the source, but also in the whole city or even in the suburban or more remote areas.

The air in and around industrial and/or densely populated areas are nowadays characterized by relatively high concentrations of waste gases from fossil fuel combustion, mainly nitrogen oxides and hydrocarbons. The absolute amounts are very low: the maximum concentrations are still in the order of volume parts per billion. In comparison with 'clean' air, the concentrations of these trace gases are significant however. Under conditions of high temperatures and lack of mixing with clean air, the amount of waste gases in urbanized areas sometimes accumulate to such a level that the gases become visible to the human eye: a brown colored combination of smoke and fog, simply referred to as *smog*. Cities such as Los Angeles, Athens, and Mexico City suffer or have suffered in the past from an almost permanent smog, as a result of a dense population, warm climate, and unlucky geographical location. Although the color of smog is its most apparent characteristic (brown, caused by certain nitrogen oxides), the health risks of smog are mainly caused by invisible components. Waste gases are slowly degredated under impact of sunlight and radicals in the atmosphere. One of the degradation products is ozone, a highly reactive oxidant of which small amounts occure naturally in the atmosphere. Large concentrations of ozone are harmful to people's health. Lung tissue is especially at risk, so lung patients are warned to avoid urban areas during smog episodes. The increase of air pollution has doubled the amount of ozone throughout the troposphere (lowest part of the atmosphere) since the start of the industrial revolution (*Committee on Tropospheric Ozone, 1991*). The impact of air pollution is therefore not limited anymore to only the source area.

In the last decades a number of findings have lead to the insight that the current amount of emissions can have an impact on the entire globe. A highly publicized issue is the destruction of stratospheric ozone by chlorofluorocarbons (CFCs), with the Antarctic ozone hole as the most significant result. Where tropospheric ozone is mainly produced by human activities and regarded as unhealthy, the stratospheric ozone is produced naturally in large amounts and regarded as essential for to life as it absorbs dangerous radiation from the sun. Regularisation of CFC emissions has been quite successful, and the ozone depletion is believed to be stabilized. The concern about the ozone layer has been replaced by concern about the effects of emissions on the climate due to what is called the greenhouse effect. Large amounts of carbon dioxide emitted due to fossil fuel combustion tend to absorb thermal radiation from the earth, leading to warming of the atmosphere. Other and sometimes relatively more important greenhouse gases have been identified, such as CFCs, methane, but also tropospheric ozone. Reduction of tropospheric ozone by limiting the air pollution therefore decreases the greenhouse effect too.

The possible consequences of a climate change and especially the rate with which this might occur have increased the interest in everything which is related to the climate. One of the new issues in climate research is the impact of aerosols: systems of small particles or liquid droplets suspended in a gaseous phase. Volcanic eruptions led to the release of large amounts of aerosols in the atmosphere, and since extreme eruptions are known to have an impact on the climate (e.g. the explosion of the Krakatau in current Indonesia in 1883 and Mnt. Pinatubo on the Philippines in 1991), the interest in aerosols has increased. Aerosols are introduced directly into the atmosphere by the dispersal of solids, but also indirectly from chemical reactions. These reactions involve sulfur and nitrogen bonds emitted from anthropogenic sources; the interest in the global climate has therefore led to a renewed interest in local emissions and the air pollution related to it.

1.2 Tropospheric ozone

Of all interesting air pollution problems described above, this research focuses on the problem of tropospheric ozone on the scale of Europe. Ozone levels and presence of smog are closely linked: the ozone level is related to the health risk of smog, and is a good indication of the total pollution level. Tropospheric ozone is a key component in air pollution, and therefore discussed in detail in this section.

1.2.1 Problems related to ozone

The first observations that the accumulation of air pollution could led to dangerous ozone levels were made in the Los Angeles bassin in the 1940s. Damages to crops were shown to be caused by overexposure to polluted air. Research by Haagen-Smith and others established the important role of ozone in this. The formation of ozone from nitrogen oxides and organic compounds became widely studied in the following years, as well as the impact of ozone on the environment. Short-term exposure to ozone levels above 120 ppb were shown to have adverse effects on lung tissue and vegetation. The EU Ozone Directive (92/72/EEC) advises thresholds of 90 ppb (1 hour average) above which the public should be informed about the air quality, 180 ppb above which a real warning should be issued, and a threshold of 55 ppb for an 8 hour mean above which health protection measures should be taken. Plants may be damaged by long-term exposure to moderate ozone levels, so a lower threshold is used for vegetation. An often used guideline here is AOT 40c: the total amount of ozone exceeding a threshold of 40 ppb, accumulated over the growing season for crops (May-July) during daylight hours, should not exceed 3000 ppbh (UN-ECE/CLRTAP guideline). AOT levels indicate that damage is caused above certain ozone levels, and that this damage is not reduced by long periods of low ozone levels.

1.2.2 Measurements

To investigate whether guidelines in a certain area are exceeded, several networks measure the air quality in Europe. Most of these measurements concern the analysis of air samples at 1-3 m above the ground (figure 1.1). Ground-based measurements are the authorative source for air quality control, since these indicate the concentrations in the air inhaled by humans and in contact with crops. Concentrations of one or multiple components are determined automatically as hourly or half-hourly averages. Apart from the key component ozone, also concentrations of nitrogen oxides, carbon monoxide, sulfur oxides and hydrocarbons

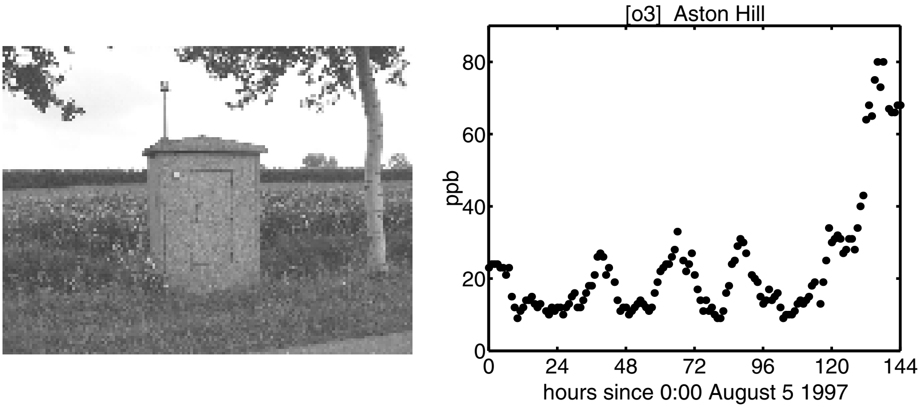


Figure 1.1: *Example of air quality measurement site in The Netherlands. The site consists of a small building with equipment and inlet at the top. The figure at the right shows an example of an ozone time series (measured at Aston Hill, Wales).*

are sometimes measured. Sites are placed at locations where the air samples taken are representative of a larger area, or where emissions from a specific source can be measured. Based on their representativeness, sites are classified as urban (inside a city), traffic (near highways), remote (not in a city or near a highway), or elevated. Measurement sites are often operational for decades, and therefore useful to analyze trends in air quality. Ozone was for example already measured in the nineteenth century, and comparison of these records with current ozone levels showed an increase of ozone levels with a factor three to four (*Graedel and Crutzen, 1993*). Another useful application is one that relates changes in the average concentrations to the emission rates. The emission rates are often computed on a yearly basis from, for example, the total fuel consumption, and trends in this consumption should be visible in the measurements.

Although ground-based measurements are the authorative source for judgment of the air quality, other types of measurements provide useful information too. Vertical profiles of ozone concentrations are available from lidar instruments and balloon soundings (figure 1.2). The measurements of balloon soundings extend to the middle stratosphere (20-30 km, until the balloon bursts). Balloon profiles are therefore useful to study the stratospheric ozone layer, but also provide information about the tropospheric level. Ozone soundings are launched from a limited number of sites at a frequency of 1-2 times a week; spatial and temporal coverage is therefore limited.

A larger spatial coverage could be obtained with satellite instruments. These instruments are able to provide information on areas where other measurements are sparse, for example over sea. With the growing interest in the stratospheric ozone layer related to the depletion by CFCs, a large number of instruments has been launched to measure the amounts of ozone. Most of these instruments measure total ozone columns, with limited information

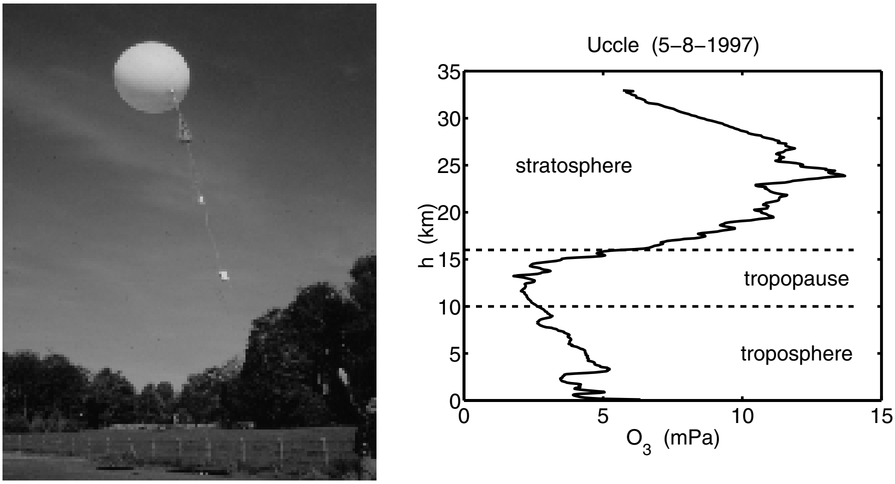


Figure 1.2: *Launch of a balloon sounding, and vertical profile measured with a sounding launched at Uccle (Belgium).*

about the vertical distribution. Especially the information about the troposphere is limited, since the bulk of ozone is located in the stratosphere and the satellite instrument is more or less blind where it concerns the ozone below (not to mention problems associated with clouds). Tropospheric ozone columns retrieved from the GOME instrument have been compared with soundings during the STROPDAS project (*Velders et al., 2001*), and were shown to be structurally biased (figure 1.3).

1.2.3 Ozone forecast

In order to warn the public about harmful ozone levels, most developed countries provide some form of ozone forecast. Similar to a weather forecast, the expected situation for the coming day(s) is provided on a daily basis. The interest in air quality is much smaller than the interest in the weather, however. The effort put in the smog forecast is therefore smaller than that put in the weather forecast, and methods used are quite simple. The ozone forecast system in the Netherlands is called PROZON and is based on a statistical approach (*Noordijk, 1994*). Given meteorological parameters measured for today, and forecasts of these parameters for a coming day, PROZON searches in a database of historical measurements for pairs of dates with same meteorological conditions. If a suitable number of matching pairs is found, the ratios between ozone maxima measured on these dates determine the forecast made for the coming day. PROZON thus assumes that the ozone levels today act the same as they used to do in the past; why they act the same is not considered. In spite of the simple method, statistical methods like PROZON work well, and their use is therefore common practice in smog forecasts.

If the ozone forecast is expected to reach a dangerous level, the government can take

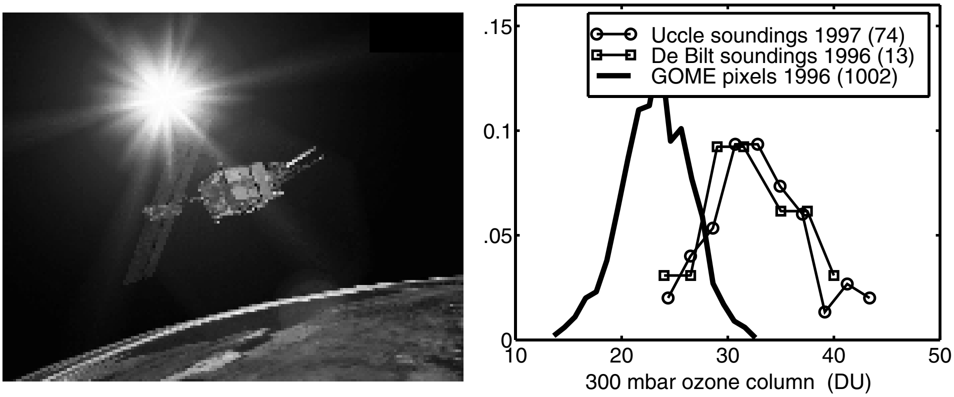


Figure 1.3: *Artist impression of* ERS2 *satellite carrying the* GOME *instrument (left; image ESA Visulab). Although the total ozone columns retrieved from* GOME *are quite accurate, the tropospheric information is limited and shows a large bias with balloon soundings (frequency distribution on the right).*

measures. For rather low critical levels, the public and especially people sensitive to air pollution are informed. For larger exceedence, the industry could be asked to decrease the emissions. A final measure could be the limitation of car traffic, but this restriction has never been taken yet in the Netherlands (in contrast to strongly polluted cities such as Athens, where during stringent smog episodes only a limited number of cars is allowed on the road). If the limitation of emissions is successful, the forecasted ozone level is not reached. This situation is completely different from weather forecasts, which predict a situation that cannot be influenced by human activity.

1.2.4 Ozone models

To study the underlaying mechanisms of ozone formation, a large variety of ozone models has been developed. Ozone models try to describe every physical or chemical mechanism in the atmosphere involved in the formation of ozone. If all mechanisms are understood correctly, the model should be able to predict the ozone level. This approach is therefore completely different from statistical approaches such as PROZON, which do not take into account any knowledge about ozone formation. Development of an ozone model requires knowledge about the structure of the atmosphere, the photochemical reactions involved in air pollution, emissions of pollutants, interaction with vegetation, etc., but also experience with numerical simulation techniques.

The spatial and temporal scales simulated by models differ widely. For studying gas phase reactions or for simulating the average pollution level in a city, a simple box model is often sufficient. Detailed study of spatial structures requires more-dimensional models, however. The area covered by these models range from the street between two buildings (a street canyon) to the scale of countries, continents, or the entire globe. Spatial detail decreases

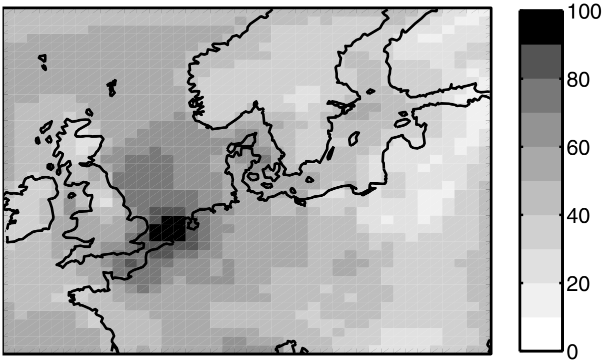


Figure 1.4: *Ozone pattern at ground level simulated by the* LOTOS *model. The figure shows the ozone concentration in ppb computed for August 10 1997, 15:00.*

with increasing extent; the smallest detail in a global model sometimes completely covers the domain of a local model. Time periods simulated by models are in general increasing with the extent too: where small models are used for episodes of a few days, the global models are used to simulate yearly variations or even climate changes over decades.

The ozone model used in this study is the LOTOS (LOng Term Ozone Simulation) model developed by the TNO institute in the Netherlands (*Builtjes, 1992*). LOTOS has been designed to simulate hourly ozone levels on a European scale (figure 1.4), for time periods of months up to several years. The resolution of the model is too coarse for the prediction of ozone levels within a city, but detailed enough to study ozone levels on a regional scale. The model has been used to study the impact of different emission scenarios on the ozone formation, and whether these exceed guidelines such as AOT40.

1.3 Data assimilation

In the previous discussion about tropospheric ozone, the topics *measurements* and *models* have been treated separately. In practice, both sources of information are closely linked however. The theory about chemistry and physics embedded in a model is often the result of research trying to explain what has been observed. A model is only accepted if it is able to explain what is measured now or in the past. Vice versa, models are used to simulate processes at locations or times for which no measurements are available. If the model is able to explain the measurements collected up to now, it is supposed to be valid everywhere, until new measurements prove the opposite. The final judgment is therefore based on the measurements; on the basis of these a model is accepted or rejected. Complete rejection of a model is rare, in practice, it is often sufficient to modify only certain parts of the model, since it has been shown to be accurate for other events before. Large three-dimensional models such as LOTOS are based on submodels for chemistry, emissions, deposition etc, and if one of these submodels is improved, the rest of the model is left unchanged.

A disadvantage of the described procedure is that a model is modified afterwards. The model simulates certain events, the result is compared with measurements, the model is modified, eventually rerun, compared again, etc. A better method would be to modify the model online. This is the idea behind *data assimilation*, a common name for a large variety of applications, methods, and techniques combining models and measurements. The term data assimilation is used in the field of geophysics, for a combination of measurements and large-scale atmospheric, hydraulic, and oceanographic models. The techniques used for data assimilation are often the same as those used in the field of system theory and control under the name 'filtering', or taken from the more general field of optimalization.

The first and still most important application of data assimilation is found in weather forecasting. If the current state of the atmosphere is known perfectly, the model forecasts for the coming days coincidence with the measurements quite well. For obtaining an accurate forecast one needs only to feed the model with the current state of the atmosphere, and this is achieved with data assimilation. Where measurements are available, these are used to describe the current state; the gaps are filled in by the model. If the model is started from this assimilated state, the result is a weather forecast based on all information available up to now: measured quantities and knowledge put in the model. If after a while the model and measurements deviate again, other measurements have become available, and a new initial state could be assimilated.

The successful application of data assimilation in meteorology has led to the introduction of similar techniques in related fields, including the field of air pollution. However, the targets of using data assimilation in air pollution are quite different than in meteorology, where assimilation is used to obtain the current state of the atmosphere to facilitate the weather forecast. For air pollution, the current chemical state is of minor importance, since accurate pollution forecasts can be made using simple statistical methods. Data assimilation in air pollution models is often related to parameter estimation. The most interesting air pollution parameter is probably emissions. Both scientists and policy makers show a large interest in who or what to blame for high pollution levels, and data assimilation techniques are therefore used to estimate emission rates or to identify sources; this application is sometimes referred to as *inverse modeling*.

1.4 Motivation and overview

This thesis describes the development of a data assimilation tool for a local area air pollution model, in particular LOTOS. The assimilation tool should be able to provide maps of tropospheric ozone using all available information in model and measurements. Since air pollution guidelines put a limit on the maximum ozone levels, at least the maxima should be estimated correctly. In addition, the assimilation procedure should provide insight in why the model and measurements differ from eachother. The measurements to be assimilated with LOTOS concern ground-based measurements only, which are available on a regular basis throughout the model domain. The limited vertical extent of the LOTOS model does not allow the use of vertical profiles from balloon soundings or lidars for assimilation. For the same reason, satellite measurements are not considered either; their use is also limited by their lack of resolution in the troposphere.

Two frequently used assimilation techniques which are able to satisfy the postulated requirements are variational methods and filter techniques, especially the Kalman filter. Variational methods (*Talagrand and Courtier, 1987*) are based on the minimalization of a cost or penalty function, quantifying the difference between model and measurements. The minimalization procedure requires an adjoint of the forward model, which is complicated for a chemistry model (*Elbern et al., 1997; Wang et al., 2001*). To avoid the development of an adjoint for LOTOS, we developed an assimilation tool that is based on a Kalman filter approach. Originally designed for guidance problems, the Kalman filter (*Kalman, 1960*) has a long history of merging (small) models and measurements in electrical engineering and control. The growing availability of cheap computing power during the last decade made the filter approach feasible for large geophysical models too. Approximations to the original filter are necessary however, since the computational burden is still too large. One particular class of approximations, the low-rank filter, has recently been introduced in a number of variations: ENKF (*Evensen, 1994*), PEKF (*Cohn and Todling, 1995*), RRSQRT (*Verlaan and Heemink, 1995*), SEEK/SEIK (*Verron et al., 1999*). Successful applications in combination with hydraulic and oceanographic models showed that the low-rank approximation is a suitable tool for data assimilation. Application to the transport of methane (*Zhang et al., 1999*) proved that the technique is suitable for air pollution problems too, encouraging the implementation with a chemistry model such as LOTOS.

The outline of this thesis could be split into three parts. The first parts contains an introduction to air pollution modeling and data assimilation. Chapter 2 describes the theory behind photo-chemical air pollution, and in particular how this has been implemented in the LOTOS model. Chapter 3 gives an overview of popular data assimilation techniques, with the emphasis on the Kalman filter. The second part describes the results of application of the Kalman filter to the LOTOS model. The experiments described in chapter 4 have a small setup, and examine the (im)possibilities of the assimilation technique with a chemistry model (*Heemink and Segers, 2000*). The domain of the LOTOS model is thereto limited to a test area covering England and Wales, and the time period is limited to five days. In chapter 5, the experience obtained with the small model is applied to the area of central Europe, for a time period of one month (*Segers et al., 2000c; van Loon et al., 2000*). The experiments focus on providing assimilated ozone fields. Attention is paid to parameter estimation too, where the filter tool is used to estimate model parameters such as emission rates and deposition velocities. In addition, the value of the filter tool for forecasts of the ozone maxima is examined too. The third part describes the technical 'details' of the filter used for the experiments. Chapter 6 describes the concept of the low-rank approximation of the Kalman filter (*Segers et al., 2000a*). Examples of this class of filters are discussed and compared. Chapter 7 discusses approximation techniques for dealing with the nonlinearities in the model (*Segers et al., 2000b*). The accuracy of different approximations is examined on a theoretical basis and tested in a practical application. Application of a Kalman filter to a model such as LOTOS is quite expensive, and was therefore performed by a parallel computer. Chapter 8 describes and compares two approaches for parallelization (*Segers* *and Heemink, 2002*). Finally, the main conclusions and recommendations drawn from this research are summarized in chapter 9.

Chapter 2

Tropospheric chemistry: the LOTOS model

A description of the key processes of air pollution modeling is given following the LOTOS *(LOng Term Ozone Simulation) model. The gas-phase chemistry involved in photochemical smog is described, with emphasis on the production of ozone. The vertical decomposition of the atmosphere plays an important role in smog formation, and is therefore discussed in detail too. The physics implemented in* LOTOS *are suitable for simulation of pollution events in the area of Europe.*

2.1 Introduction

After the first observations that photochemical air pollution has a damaging effects on vegetable crops (*Haagen-Smit et al., 1951*), the physical mechanisms behind this form of air pollution has been subject of extensive scientific research. The damage was soon shown to be caused by over-exposure to ozone, produced under certain meteorological conditions in presence of nitrogen oxides and volatile hydrocarbons. The basic mechanism behind photochemical air pollution are nowadays understood reasonable well, and have been implemented in a large variety of models.

The most simple models are box models for simulation of the gas-phase chemistry. Insight in the gas phase chemistry involved in air pollution has been obtained from experiments with smog-chambers. In these specialized reaction chambers, the conditions under which pollution is known to reach elevated levels are simulated: intense sun shine, relatively high temperature, and reduced mixing with clean air. After injection of typical urban emissions from car traffic or industrial activities, the concentrations of the reaction products are measured. From these studies, the most important reaction paths have been deduced, leading to long lists of reaction schemes and related parameters such as reaction constants and photolysis rates. The number of possible reactions in a smog chamber is almost infinite, but selection of the most important ones have lead to a number of limited reaction schemes describing the main processes. Examples of these schemes are ADOM, EMEP, RADM2, and CBM-IV, different from eachother in the components of the chemical state and the reactions included (see (*Kuhn et al., 1998*)). Boxmodels are based on numerical simulation of a reaction scheme, and thus differ from eachother with respect to the scheme and the numerical solver used to simulate the reactions. Box models are useful to investigate smog formation under different meteorological regimes or changing emission strengths. Intercomparison studies show that most box models are quite comparable in their calculation of ozone, but show large differences for higher organic compounds (*Kuhn et al., 1998*).

If a box model is to simple for an application, a number of box models could be connected to eachother. A model for the transport is required to simulate the exchange between the boxes. A simple configuration is to put a number of boxes onto eachother, to represent the different chemical regimes present in the vertical. The bottom box is subject to surface processes such as injection of emissions and uptake by vegetation, which leads to different chemical conditions if the exchange with the box above is limited. The vertical structure of the atmosphere becomes important here, and this structure is in fact quite complicated. Columns of box models are useful to simulate the diurnal cycle in the gas-phase chemistry, driven by sunrise and sun set, on time scales where horizontal transport is of minor importance. The vertical extent of a column is therefore limited, since at higher altitudes horizontal transport and mixing becomes rather strong.

To include horizontal transport, a number of 1-D columns could be combined to a 2-D array. This approach is for example used in the TNO-Isaksen model (*Roemer and van den Hout, 1992*), to simulate zonal averages in the global atmosphere. Due to the rotation of the earth, the longitudinal variations are much smaller than the latitudinal, and zonal averages are therefore suitable to characterize the global chemical composition. Zonal variations are induced by seasonal changes in solar angle and related temperature, and the distribution of land mass and human activities over the latitudes. Zonal average models are useful to investigate seasonal changes or long year trends in atmospheric composition. Transport is dominated by large circulations moving air upwards from near the equator towards the poles, and smaller reverse circulations at higher latitudes.

The final step in atmospheric modeling is a full 3-D model. In an Eulerian approach, large numbers of box models are connected in a 3-D array to cover a part or all off the globe. The chemical regimes in the boxes are determined by their position on the globe (solar angle, emissions), and meteorological parameters such as temperature and water vapor content. Transport between the boxes is modelled with fluxes through the boundaries computed from wind fields. In a Lagrangian approach, the boxes are not fixed on their position but move through the domain following the wind. The meteorological parameters for a 3-D model are often obtained from operational weather centers, and the simulations are therefore as close to reality as possible. Full 3-D models are rather expensive in computation time, since the model requires simulation of the chemistry in each grid box and advection of all components, in addition complicated by the interaction of advection and chemistry. To limit the costs, operational atmospheric chemistry models are therefore limited in either the chemistry or the domain. For accurate and detailed modeling on short time scales, limited-area models such as LOTOS, EURAD (*Elbern et al., 1997*), or MATCH (*Robertson et al., 1999*) use a complex chemical scheme on a fine grid, on a domain limited to an area of interest (for example central Europe). Climate modeling requires however computations over the complete globe, and are therefore necessarily defined with simplified chemistry on a coarse grid, such as the TM3 model (*Houweling, 2000*).

The LOTOS model used in this study is a typical example of a 3-D, Eulerian, limited-area, atmospheric chemistry model. The description of the model starts with the chemical mechanism applied in the grid boxes (§2.2). The vertical decomposition of the atmosphere is described in §2.3, with emphasis on the lowest 2-3 km of the troposphere where LOTOS is active. Finally, details of the implementation concerning grid definition and numerical schemes are given in §2.4.

2.2 Chemistry of tropospheric ozone

As a key component in air pollution, the chemical behavior of ozone has been subject of extensive study since the mid 1940's. In industrialized areas, the key processes in ozone formation involve reactions between nitrogen oxides and organic compounds, while carbon monoxide and methane play a role in remote area. This section describes the most important reaction mechanisms for the area of Europe. The actual reaction mechanism used is a version of CBM-IV, listed in appendix A. Reaction numbers (Rn) and constants k_n used in this section refer to the reactions the appendix.

2.2.1 Nitrogen oxides

Under the name *nitrogen oxides*, a large variety of gaseous components is known: molecules such as NO (nitric oxide), NO_2 (nitrogen dioxide), and N_2O_5, acids such as HNO_2 (nitrous acid), and radicals such as NO_3. The components NO and NO_2 (their sum often referred to as NO_x) play a role in smog formation; the other components are related to the nitrogen cycle and the formation of aerosols.

In the presence of sunlight, NO_2 is able to induce O_3 formation through a reaction with oxygen:

$$\text{(R1)} \qquad NO_2 + h\nu + O_2 \rightarrow O_3 + NO$$

where $h\nu$ denotes an ultra violet photon. Because the availability of oxygen in the atmosphere is almost infinite (in comparison with trace gases), reaction (R1) is bounded by the availability of NO_2 and sunlight only. Production of ozone due to photolysis of NO_2 is therefore related to the angle between the earths surface and the sun: the higher this angle (daytime, summer months), the more ozone is formed due to (R1). The combination of O_3 and NO formed in (R1) has a short lifetime due to the inverse reaction:

$$\text{(R3)} \qquad NO + O_3 \rightarrow NO_2 + O_2$$

Reaction (R3) is very fast: freshly emitted NO reacts with ozone on a time scale of about 2 minutes (*Kley et al., 1994*). Emission of NO will therefore decrease ozone levels and increase the amount of NO_2. The sum O_x of O_3 and NO_2 is however rather constant on short time scales. Figure 2.1 illustrates a simple chemical regime where (R1) and (R3) are the only reactions involved. The concentrations follow the *photo-stationary steady-state* condition:

$$J_{NO_2}\,[NO_2] = k_3\,[NO]\,[O_3] \tag{2.1}$$

where J_{NO_2} denotes the photolysis rate of NO_2 following (R1) and k_3 the reaction rate of (R3). Injection of extra NO to the system will increase the amount of nitrogen oxides,

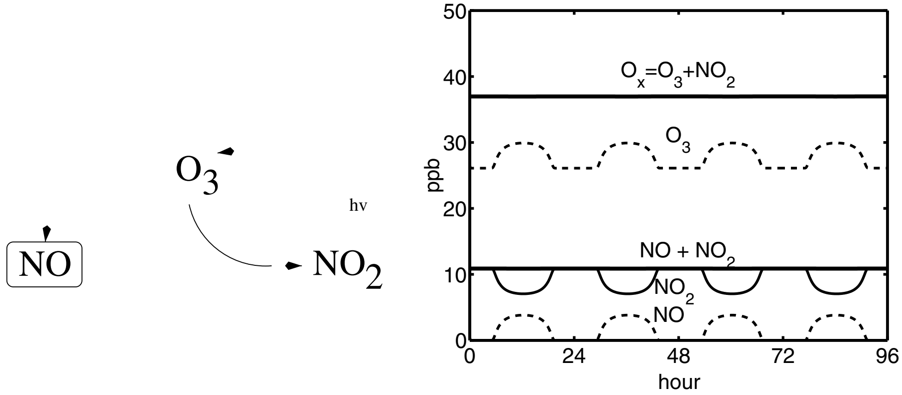


Figure 2.1: *Simplified representation of chemistry between* NO, NO_2, *and* O_3. *The panel on the right shows time series of these components and their sums simulated with a box model. During daytime,* NO *and* O_3 *are formed due to to photolysis of* NO_2*; after sunset, the production stops and* NO *and* O_3 *reach their initial values again.*

but will decrease the amount of ozone. Since 90% of the emissions of NO_x consist of NO, how could emission of nitrogen oxides then be blamed for causing air pollution? The only explanation is that reactions other than (R3) transfer NO into NO_2, without destroying ozone. In the next section it is state that this is the role of hydrocarbons. The very high reaction rate of (R3) ensures however, that the short term effect of NO emissions is always a decrease of ozone concentrations.

The most important source of NO_x in the troposphere is fossil fuel combustion. Since the atmosphere is filled with N_2 for about 80%, a partly oxidation of N_2 inside the hot and high pressure environment of a combustion engine can not be avoided. Emissions of NO_x are related to the use of fossil fuel, and their spatial distribution depends on economic and social parameters such as population densities, transport behavior, industrial activities, technological development, etc.

Nitrogen bonds are not only emitted to, but also removed from the air. The most import removal process is the production of nitric acid:

$$\text{(R20)} \quad NO_2 + OH \rightarrow HNO_3$$

The later is removed from the atmosphere through scavenging (wet deposition). Together with sulfhic acids, nitric acid is one of the major contributionairs to the so called *acid rain* observed in polluted areas. Apart from wet deposition, NO_x is also removed from the atmosphere through uptake by the vegetation (dry deposition).

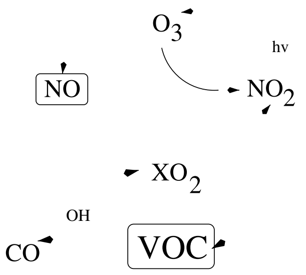


Figure 2.2: *Simplified representation of atmospheric chemistry with* NO_x *and* VOC*'s. Organic compound react with* OH*-radicals, leading to formation of peroxy radicals* XO_2 *and transformation of* NO *into* NO_2*; photolysis of* NO_2 *causes production of ozone. The chemistry is driven by ultra violet light and* OH *radicals, and initialized by availability of* NO *and* VOC.

2.2.2 Volatile organic compounds

A mix of various hydrocarbons present in the atmosphere is known under the name of *volatile organic compounds* (VOC's). The group of VOC's contains for example ethane, parafine, and formaldehyde. VOC's have typical concentrations of a few ppb, and have a rather short lifetime in the atmosphere. Methane is not considered as part of the VOC's because of its much longer lifetime (chemical less active), and related high concentrations of 1600–1800 ppb. VOC's are therefore also known under the name Non-Methane Hydro Carbons (NMHC).

Figure 2.2 shows a representation of the reactions involved in VOC's (*Liu et al., 1987*). The scheme is rather simplified, but indicates the net result: VOC's involve the formation of NO_2 out of NO, without destruction of ozone. Under impact of OH radicals, which are available in the atmosphere in very low concentrations, the organic compounds are transformed into other, less complex organic compounds. As intermediate products, various types of organic peroxy radicals are formed, in a general notation: XO_2. Peroxy radicals are able to transform NO into NO_2 (R68). The degradation of organic compounds continues until all carbon atoms appear as carbon monoxide (CO).

Ozone concentrations usually rise due to the degradation of VOC's, since NO_2 is formed out of NO without destruction of O_3. The reactions involved in VOC's do not always lead to higher NO_2 concentrations, since the degradation of VOC's also leads to higher OH concentrations and a related loss of NO_2 (R20). Complete degradation of a certain VOC into CO might provide x O_3 molecules, where x is called the stoichiometry factor for the particular VOC. Stoichiometry factors may range from zero and eight, depending on the type of VOC and the availability of NO_x. Simple reaction schemes use stoichiometric factors to model ozone formation, see for example (*van Loon, 1996*).

Similar as for nitrogen oxides, the most important emission source of VOC's is related to fossil fuel combustion. Complete burning of fuel would provide carbon dioxide and water vapour only, but the current state of technology has to allow a partly incomplete burning too. The products of incomplete combustion vary from simple carbon monoxide to complex aromatic structures, and are in general quite unpleasant for human health. Another important anthropogenic source of VOC's is their use as solvents. Some types of vegetation are known as important natural sources of VOC's; conifer forests release substantial amounts of isoprene for example.

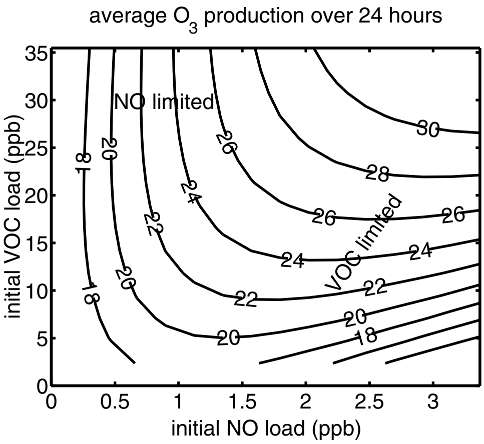


Figure 2.3: Average ozone production in ppb over 24 hours calculated with a box model, for different initial loads of NO *and* VOC. *The different carbon compounds are injected with a constant ratio, corresponding to urban emissions. In a* NO *limited regime, the ozone production is almost indifferent for the* VOC *load, and vice versa for a* VOC *limited regime. Note the trend to decreasing ozone concentrations for increasing initial concentrations of* NO.

2.2.3 NO_x- and VOC-limited regimes

The previous described production of ozone continues as long as precursors NO and VOC are available. The total ozone production is limited by the amount of one of them (figure 2.3). This property has lead to classification of areas as being either NO_x- or VOC-limited.

Rural areas are often NO_x-limited: emissions of nitrogen are low by lack of population, while natural releases of VOC's are rather strong. Small releases of NO will immediately contribute to a net production of ozone, until all of the available NO has reacted. The lifetime of additional NO in rural area's is therefore rather short, and is only visible through an increased ozone level.

Industrial areas with large emissions of NO are often subject to a VOC limited regime. In such a regime, the best option to limit smog production is to limit VOC emissions. Reduction of NO load might not reduce ozone formation, if it is not combined with VOC reduction. In fact, ozone reduction might be achieved by additional NO *production*, as observed for the so-called 'weekend smog' in Los Angeles. Traffic emissions limit ozone production on labor days, but with their absence in the weekend, ozone concentrations may exceed critical levels. This effect is only a short-term solution for smog-episodes; the additional load of nitrogen oxides will cause higher long term ozone levels in the presence of VOC. The best regularisation strategy for smog is therefore to reduce both NO and VOC emissions (*Builtjes, 1992*).

2.2.4 Methane and carbon monoxide

Where in industrialized areas the formation of ozone is strongly determined by the chemistry of NO_x and VOC, the ozone formation in remote area is more dependent on the chemistry of methane and carbon monoxide.

Methane (CH_4) is present in the atmosphere in rather high concentrations (1700-1800 ppb). Important sources of methane are rice fields and natural wetlands. In the presence of NO and ultra violet light, methane is oxidized into CO by a chain of reactions in which formaldehyde takes a central place (*Houweling, 2000*). A net result of the oxidation chain is the formation of NO_2 due to the reaction:

$$NO + HO_2 \rightarrow NO_2 + OH$$

Similar as for VOC's, this reaction provides a mechanism to transform NO into NO_2 outside the $NO/NO_2/O_3$ cycle, and therefore leads to an additional production of ozone.

Carbon monoxide is released into the atmosphere by industrial processes and biomass burning. Besides, it is the end product of the degradation of organic compounds and methane. Typical concentrations on the northern hemisphere range from 100 through 300 ppb. Under impact of OH radicals, carbon monoxide is oxidized into carbon dioxide (CO_2) under formation of HO_2, again leading to additional production of ozone. The impact of methane and carbon monoxide on ozon production is however limited if compared with the impact of NO_x and VOC, and therefore of minor importance in industrial areas.

2.3 Vertical structure of the atmosphere

Since the human view to the world is strongly based on visible perception, our first characterization of the atmosphere is probably determined by the existence of clouds. Given a picture of the sky, the amount, shape, color, but also the lack of clouds could give us an indication of where on earth the picture is taken. Clouds let us realize that the atmosphere is different over the globe, and since they are a good identification of the weather, we are often highly interested in their horizontal distribution. However, clouds should also let us realize that the atmosphere is different with the altitude: they always seem to appear at the same level.

Trying to find some vertical structure in the earths atmosphere seems to be similar to the work of a biologist, investigating the life on a apple's pare. An altitude of 50 km is the top for most atmospheric studies, which is less than a percent of the earths radius (about 6710 km). Since we use to live on the apple's pare however, it is still useful to distinguish some vertical structures. We will describe three divisions of the atmosphere, with vertical scales decreasing from 10 km to 100 m.

2.3.1 Stratosphere/troposphere

The most coarse division of the atmosphere is based on the different impact of solar radiation in slabs of air. Radiation with all kinds of wavelengths enters the atmosphere inducing photo-dissociation. The probability of a photon being absorbed increases with the number of available molecules, and since the air pressure increases during the way down, all radiation of a certain wavelength might be absorbed before it is able to reach the surface. For example, wavelengths with sufficient energy to dissociate oxygen are not able to penetrate the lowest 10 km of the atmosphere. The differences in photo-dissociation lead to a division of the atmosphere in *troposphere* (0-10 km), *stratosphere* (10-50 km), *ionosphere* (50-650 km), and *exosphere* (above 650 km). Atmospheric studies are often limited to tropo- and stratosphere only, since the processes in these layers have the most direct impact at ground level.

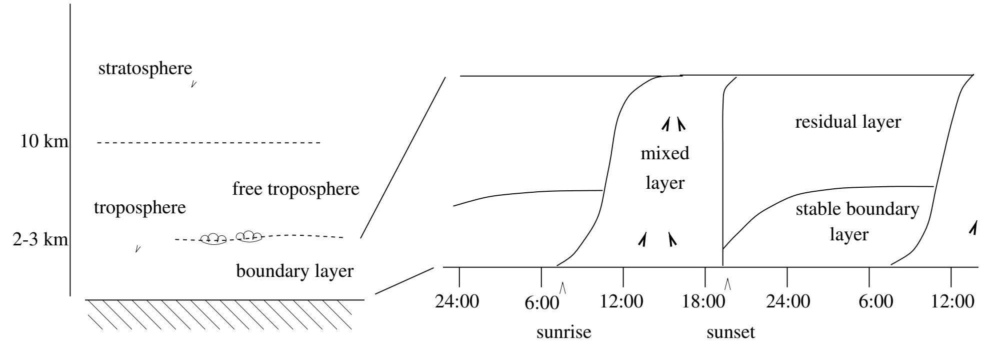


Figure 2.4: *Vertical division of the atmosphere based on impact of solar radiation (stratosphere/troposphere) and forcing from the surface (free atmosphere/boundary layer). On the right: development of mixing regimes in the boundary layer.*

The photo-dissociation of oxygen is the driving force of the atmospheric chemistry in the stratosphere. A reaction product of the photo-dissociation is ozone. The absolute amount of ozone reaches a maximum at 20–30 km: at higher altitudes, less ozone is formed due to a lack of oxygen, while at lower altitudes, the production is bounded by a lack of radiation. This *stratospheric ozone layer* is able to absorb ultra violet radiation with wavelengths harmful for living species on the earths surface. The strong depletion of stratospheric ozone observed at the end of the 20^{th} century gave rise to serious concern about the environment, especially when anthropogenic releases of halocarbons were found to be the driving force behind the depletion. Regularisation of halocarbon production and emissions have however stabilized the ozone loss. The concern about the ozone layer and good observability have made ozone the key component in stratospheric research.

Controversely to the stratosphere, photo-dissociation of oxygen is of minor importance for the chemistry of the troposphere. The chemistry of the troposphere is determined by what is released from the earths surface, and therefore depends on vegetation, land-use, and human activities. Since these parameters differ from place to place, the tropospheric chemistry shows large spatial differences. Since we breath tropospheric air, its quality has a direct impact on our health. The large emission of nitrogen oxides and volatile carbons characteristic for industrialized areas might lead to production of unhealthy amounts of ozone as described in section 2.2. High levels of tropospheric ozone are almost ever due to human activities, except for the rare case of stratospheric intrusions reaching the earths surface.

2.3.2 Free troposphere/boundary layer

The division of the atmosphere in boundary layer and free troposphere is based on different impact of the earths surface on slabs of air. The *boundary layer* is that part of the troposphere that is “directly influenced by the presence of the earth’s surface” (*Stull, 1988*). Air in the boundary layer is subject to forcing by friction (rotation of the earth), is heated and cooled from the surface (induced by radiation from the sun), retains water vapor and pollutants, etc. Parameters such as temperature show a diurnal cycle, related to sunrise and sunset. The depth of the boundary layer may range from 100 m to 3 km; at the top, clouds might appear.

The remainder of the troposphere is called the *free troposphere*. Time series of temperature hardly show a diurnal cycle here; the temporal scales over which variations occur are much longer for the free troposphere than for the boundary layer. Total air pressure and temperature are significant lower than in the boundary layer, a well-know fact for mountaineers and aircraft designers.

2.3.3 Mixing/stable/residual layer

The different characteristics of slabs of air observed at different hours of the day give rise to a refinement of the description of the boundary layer. The diurnal cycle of sun rise and sunset causes the boundary layer to be in different states of mixing during the day, leading to a classification in mixing, stable, and residual layer (see right part of figure 2.4).

At day time, the air in the boundary layer is heated from the surface and cooled from the top, leading to a state of convection driven turbulence. The slab of air which obtains this state rapidly grows after sunrise to reach a maximum in the early afternoon. Trace gases released from the surface become well mixed through the turbulent layer; the layer is therefore recognized as the *mixing layer*. The top of the mixing layer varies from day to day depending on cloud cover, air pressure, and the surface temperature. Typical maximum heights over Europe range from a few hundred meter over water to sometimes more than 2 km over land.

When after sunset the driving force behind the convective turbulence has disappeared, the air in the former mixing layer calms down. The lowest part of the boundary layer reaches an almost stable state; smoke plumes emitted into this *stable boundary layer* will hardly spread over the vertical but fan out in the horizontal. What remains of the boundary layer is initially filled with air similar to the former mixing layer and therefore called the *residual layer*. After the following sunrise, the stable and residual layer are merged to form a new mixing layer.

2.4 The LOTOS model

The tropospheric chemistry model used in this study is the LOTOS (LOng Term Ozone Simulation) model (*Builtjes, 1992*). LOTOS includes the concepts of boundary layer structure and tropospheric chemistry as described before, and computes hourly concentrations of the most important trace gases.

The LOTOS model is based on a discretization of the advection/diffusion equation:

$$\begin{aligned} \frac{\partial c_s}{\partial t} &= -\nabla \cdot (\mathbf{u}_h c_s) + \nabla \cdot (K_h \nabla c_s) + \frac{\partial}{\partial z}\left(K_z \frac{\partial c_s}{\partial z}\right) \\ &\quad + E_s + C(c_\star) - D(c_s) + V(c_s) \end{aligned} \tag{2.2}$$

where c_s is the concentration field of a trace gas s, $\mathbf{u}_h$ the horizontal velocity field, K_h and K_z the horizontal and vertical diffusion coefficients, and source/sink terms E, C, D, and V account for emissions, chemistry, deposition, and mean vertical exchange respectively. After discretization, the model takes the form:

$$\mathbf{c}[k+1] = \mathcal{L}(\ \mathbf{c}[k],\ t[k],\ t[k+1]\ ) \tag{2.3}$$

The concentration vector $\mathbf{c}[k]$ contains the concentrations of all considered components for each of the cells in the model grid, valid for time $t[k]$. The LOTOS operator $\mathcal{L}$ computes the concentrations at $t[k+1]$ given the concentrations and model data valid for $t[k]$. The default time interval between $t[k+1]$ and $t[k]$ is one hour. Accurate discretization of (2.2) into (2.3) is quite complicated since the processes on the right hand side involve many different time scales. Operator $\mathcal{L}$ is therefore implemented using a symmetric Strang-splitting technique (*Strang, 1968*), applying subprocesses to the concentration array in a symmetric order:

$$\begin{aligned} &\mathcal{L}(\ \mathbf{c}[k],\ t[k],\ t[k]+\Delta t\ ) \\ &\quad = \mathcal{L}_{mix}(\Delta t) \circ \mathcal{L}_{ade}(\Delta t/2) \circ \mathcal{L}_{dep}(\Delta t/2) \circ \mathcal{L}_{vdf}(\Delta t/2) \\ &\qquad \circ \mathcal{L}_{chem}(\Delta t) \circ \mathcal{L}_{vdf}(\Delta t/2) \circ \mathcal{L}_{dep}(\Delta t/2) \circ \mathcal{L}_{ade}(\Delta t/2)\ \mathbf{c}[k] \end{aligned} \tag{2.4}$$

with $\mathcal{L}_{mix}$, $\mathcal{L}_{ade}$, $\mathcal{L}_{dep}$, $\mathcal{L}_{vdf}$, and $\mathcal{L}_{chem}$ the operators for changing mixing height, advection/diffusion/emission, deposition, vertical diffusion, and chemistry respectively. The chemistry operator is applied only once, and performs an integration over a period Δt; all other processes are applied twice and perform integrations over $\Delta t/2$. If the maximum time step Δt or $\Delta t/2$ is less than one hour for a certain operator, the sequence (2.4) is repeated. Operator splitting involves an error since it decouples subprocesses which should actually interact with eachother, partly suppressed by the symmetric order of the split. The order of the operations is not necessarily the one in (2.4); see also §8.5.2. The different operators are described in detail in the next paragraphs.

2.4.1 Model domain and grid

The maximum domain of the LOTOS model covers Europe from the Atlantic Sea in the west to Russia in the east and from the Mediteranian Sea in the south to Scandinavia in the north (figure 2.5). In typical applications, the domain is however limited to smaller area's. The domain is divided in a regular grid with cell spacing of 1.0° lon × 0.5° lat (about 60×60 km at European latitudes).

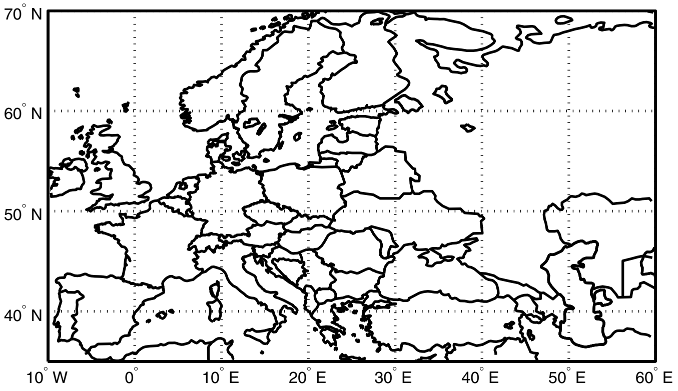


Figure 2.5: *Maximum horizontal domain of the* LOTOS *model. The domain is divided in* 70×70 *grid cells of* 1.0° *lon* $\times$ 0.5° *lat.*

2.4.2 Vertical exchange

Three layers of grid cells are placed onto eachother to describe the lowest two kilometers of the troposphere (figure 2.6). The lowest layer represents the mixing layer; the heights of the cells is time dependent and follows the rise and fall of the mixing height. During the night, the properties of a stable boundary layer are assigned to the first layer. Although not complete correct, it is convenient to call the first layer the 'mixing layer' even when it represents the stable situation. The mixing height is part of the meteorological input, and derived from measurements and models (*Seibert et al., 2000*).

Two reservoir layers of equal depth cover the mixing layer; the top of the second reservoir layer is fixed to 2000 m, or to a level high enough to have reservoir layers of 100 m thickness each. The contents of the reservoir layers is swallowed by the mixing layer during the rise of the mixing height. If no other physical processes act on the concentrations, the concentrations in a rising mixing layer are a weighted average from the concentrations in the stable boundary layer and the reservoir; this process is modelled in operator $\mathcal{L}_{mix}$ in (2.4). A fall of the mixing height does not influence the concentrations in the mixing layer, since both mass and volume decrease with the same rate.

The concentration value assigned to a layer represents the average concentration, supposed to be not to different from the actual profile. In polluted areas, this approach is sometimes too far simplified (*Roemer, 1996*). Observation of vertical profiles of NO and NO_2 showed for example a clear gradient of these components even within the mixed layer.

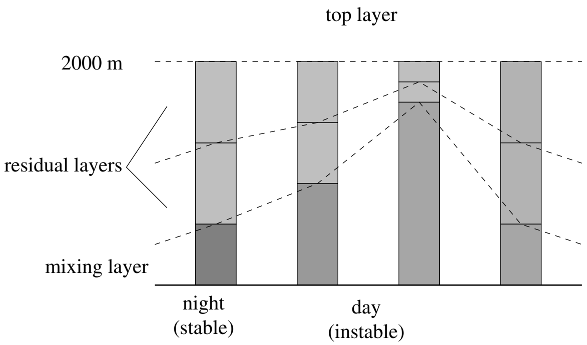


Figure 2.6: *Illustration of the layered grid in* LOTOS. *During the rise of the mixing height, the mixing layer swallows concentrations from the residual layers. With the fall of the mixing height, the residual layers obtain the concentrations of the formal mixing layer.*

The vertical resolution of the current model is therefore too low for accurate representation of NO_x measurements.

2.4.3 Vertical diffusion

In addition to the changing mixing heights and mean vertical flux induced by the horizontal wind, an additional exchange between the model layers is implemented in the form of vertical diffusion:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial z}\left(K_z \frac{\partial c_p}{\partial z}\right) \tag{2.5}$$

The diffusion constant K_z is set to 1.0 m^2/s for the boundary between the mixing and reservoir layers, and to 0.1 for the other boundaries. The upper boundary concentrations are given by the global 2-D TNO-Isaksen model (*Roemer and van den Hout, 1992*). This model computes zonal averages over all longitudes, and the output of the model might therefore differ significant from the local conditions over Europe, since the release of pollutants is much higher over here. Comparison of the TNO-Isaksen computations with ozone soundings from Uccle showed that for the ozone concentrations a correction factor should be applied (figure 2.7).

2.4.4 Emissions

Emissions of NO_x, VOC, CO, SO_x and CH_4 are injected into the model layers as part of the advection/diffusion operator $\mathcal{L}_{ade}$ from (2.4). Most emissions are injected in the lowest (mixing) layer, but releases through high chimneys are sometimes injected in the residual layers too, especially during the night.

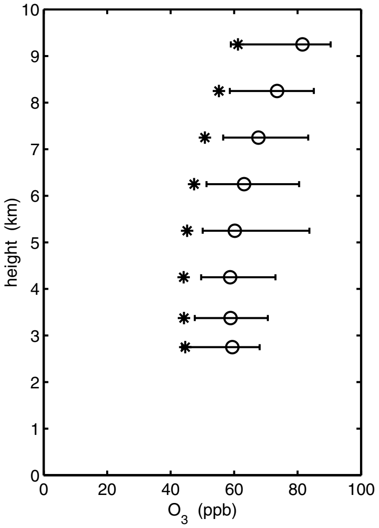


Figure 2.7: *Comparison between the TNO-Isaksen model used for upper boundary of the* LOTOS *model ($\star$), and balloon soundings from Uccle for august 1997 (errorbars). The TNO-Isaksen model under estimates the measured data; multiplication with a factor 1.3 (o) fits the model to the measurements.*

The emission database provides total emissions in terms of tons per year for each grid cell, based on inventories by local authorities or environmental agencies. Inventories for anthropogenic emissions are often combinations of bottom-up and top-down procedures. In a bottom-up procedure, emissions released by single point sources are added together to compute the total emission in a certain area. Such a procedure is only useful for large point sources such as powerplants. In a top-down procedure, estimates of total emissions for a large area are distributed over smaller areas. For example, the total emission of NO_x due to fuel combustion might be derived from sale figures from oil companies, and then assigned to small areas based on average traffic density. Inventories of biogenic emissions of VOC are derived by combining emission rates for certain types of vegetation with a landcover database.

Hourly emissions in LOTOS calculated from the yearly totals using profiles to account for temporal variations (a top-down procedure). The profiles consider the different emission rates in summer and winter, the day of the week, and the local time. Information about the temporal variations in emissions was collected in the previous PHOXA/LOTOS project GENEMIS (*Lenhart et al., 1995*). Conclusion was that "a considerable temporal variation of emissions from all major source sectors can be observed, not only from sector to sector, but also from country to country". The profiles used in LOTOS were shown to have an on average correct shape, but amplitudes are often to low (figure 2.8). Actual emissions might differ more than a factor 2 from the emissions modelled in LOTOS. Since LOTOS is compared with hourly measurements, much of the variations present in the measurements could be missed by the model.

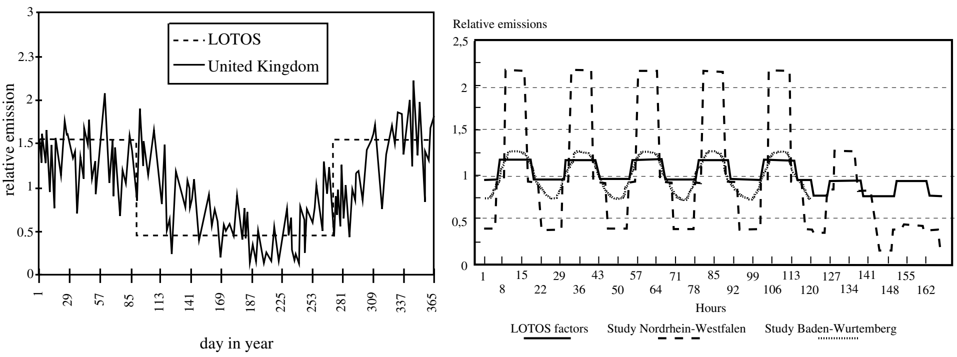


Figure 2.8: *Examples of differences between real emissions and time profiles used in* LOTOS *(Lenhart et al., 1995). Left panel shows relative daily small consumer* NO_x *emissions in the United Kingdom and* LOTOS *factors for 1990; right panel shows the hourly industrial fuel consumption during a week according to surveys in Nordrhein-Westfalen and Baden-Württenberg and the* LOTOS *hourly time-factors for industrial combustion.*
Sources: (Lenhart et al., 1995).

2.4.5 Chemistry

The chemistry model used in LOTOS is the *Carbon Bond Mechanism IV* (*Gery et al., 1989*). The chemical state is described in terms of the concentration of 26 components in total, see appendix A. Organic compounds are represented by mixtures of reactive groups rather than by single molecules, in order to limit both the chemical state and the number of reactions (about 60 in our implementation).

The impact of the reactions listed in the appendix is modeled in terms of a nonlinear differential equation for each of the components of the state. If for example the chemistry is limited to reactions (R1) and (R3) only, the ozone concentration should satisfy the equation:

$$\frac{d[O_3]}{dt} = k_1[NO_2] - k_3[NO][O_3] \tag{2.6}$$

where k_1 and k_3 denote the reaction rates of (R1) and (R3). The system of differential equations is solved using Gauss-Seidel iterations, for each single cell in the domain (operator $\mathcal{L}_{chem}$ in (2.4)). The number of iterations required for convergence might differ from cell to cell, since each of them has a different initial state. Besides, the reaction rates are different for each cell too, since these depend on time varying parameters such as temperature, water vapour concentration, solar angle and cloud cover. A time step of 15 minutes was found to be the maximum for accurate simulation of the chemistry.

2.4.6 Deposition and surface concentrations

An important removal process for pollutants is dry deposition on the surface. Vegetation is able to take up serious amounts of pollutants, leading to cleaner air but poisoned vegetation as well. As a typical example, the downward tendency observed for the growth of crops in the Los Angeles basin was found to be related with uptake of ozone. The constant uptake of trace gases induces a flux towards the ground. Air samples taken at regular measurement heights (2–3 meters above the ground) are therefore lower than the average value in the mixing layer. The effect of the deposition should be taken into account when comparing the LOTOS concentrations with measurements.

Since the LOTOS model does not divide the mixing layer into sub layers, the concentration of a chemical component in the mixing layer needs to be described with a profile. The profile $c_p(z)$ is assumed to satisfy a steady state diffusion equation:

$$\frac{\partial}{\partial z}\left(K_z(z)\frac{\partial c_p}{\partial z}\right) = 0 \quad , \quad K_z(z) = \frac{\kappa\, u_\star\, z}{\phi_l(z)} \tag{2.7}$$

That is, the flux $\Phi = K_z \partial c_p/\partial z$ is assumed to be constant with the height. The diffusion coefficient K_z depends on the stability of the boundary layer; its value follows from a parameterization of the vertical gradient of the wind speed:

$$\frac{\partial u}{\partial z} = \frac{u_\star}{\kappa}\frac{\phi_l(z)}{z} \quad , \quad \phi_l(z) = \begin{cases} (1-15z/l)^{-1/4} & , \; l<0 \quad \text{(unstable)} \\ 1+4.7z/l & , \; l>0 \quad \text{(stable)} \end{cases} \tag{2.8}$$

where κ is the Von Karman constant (≈ 0.35), friction velocity $u_\star$ is a scaling parameter, and the dimension less wind shear $\phi_l(z)$ depends on the value of the Monin–Obukhov length l. The Monin–Obukhov length is a measure for the turbulence, and is a function of the average wind speed at 10 meter height (part of the meteorological input), of the exposure class (function of the solar angle and cloud cover), and of the roughness length z_0, which can be obtained from a land-use database.

To solve equation (2.7), two boundary conditions are required. First, the profile is set to the mixing layers average c at a height h_{ref} of for example 50 m, where the impact of the deposition is of minor importance. This leads to the profile:

$$c_p(z) = c - \Phi\, R_a(z) \quad , \quad R_a(z) = \int_{s=z}^{h_{ref}} \frac{1}{K_z(s)}\,ds \tag{2.9}$$

R_a is called the *atmospheric resistance*, and can be computed exactly for the wind shears in eq. (2.8). The second boundary condition models the flux through the vegetation surface (and thus through all other horizontal surfaces too) with a first order resistance:

$$\Phi = \frac{c_p(0)}{R_c} \tag{2.10}$$

where $c_p(0)$ denotes the concentration at the vegetation surface. The *surface resistance* R_c determines how effective a chemical component is taken up by a certain type of vegetation, and is therefore part of the land-use database. Before a component is able to reach the vegetation surface, it has to pass the viscous sub layer between the surface and the roughness length z_0. The flux through the viscous sub layer is modelled as a first order resistance too:

$$\Phi = \frac{c_p(z_0) - c_p(0)}{R_b} \quad , \quad R_b = \frac{1}{2.2\,(u_\star)^{2/3}} \tag{2.11}$$

where R_b is the *viscous-sublayer resistance*. Elimination of $c_p(0)$ from (2.10) and (2.11) gives:

$$\Phi = \frac{1}{R_b + R_c}\, c_p(z_0) = \frac{1}{R_b + R_c}\,(\, c - \Phi\, R_a(z_0)\,) \tag{2.12}$$

which leads to the following equations for the flux and the deposition profile:

$$\Phi = v_d\, c \quad , \quad v_d = \frac{1}{R_a(z_0) + R_b + R_c} \tag{2.13a}$$

$$c_p(z) = c\,(\,1 - v_d\, R_a(z)\,) \tag{2.13b}$$

The equation for the *deposition velocity* v_d indicates that the deposition is in fact modeled as a serial connection of three resistances, describing the atmospheric, viscous and surface resistance (figure 2.9).

A deposition profile similar to (2.13) might be derived for each component of the state, using a different surface resistance R_c. However, the profile is based on a steady state assumption during a small time period, and this might not be valid if components react with eachother on a smaller time scale. An example is the reaction between O_3 and NO. Therefore, for O_3, NO, and NO_2, the deposition is applied to the more stable quantities NO_x and O_x, from which the original concentrations are recalculated afterwards by assumption of a photo-stationary-state.

The total mass in the mixing layer will decrease due to the deposition flux, and should satisfy the equation:

$$\frac{\partial(hc)}{\partial t} = -\,\Phi(t) = -\,v_d(t)\,c(t) \tag{2.14}$$

where h denotes the height of a grid cell. If during a time period Δt the meteorological conditions are rather unchanged, both the cell height and the deposition rate are more or less constant, and the solution of (2.14) is given by:

$$c(t + \Delta t) = c(t)\,\exp(-\Delta t\, v_d/h\,) \tag{2.15}$$

This relation is used in operator $\mathcal{L}_{dep}$ from (2.4).

2.4.7 Advection and horizontal diffusion

Horizontal advection is modeled by volume fluxes through the boundaries of the grid cells. The meteorological input provides horizontal wind fields for each of the three model layers. A vertical component of the wind is derived from the horizontal fields by the condition that the net volume flux equals the volume change prescribed by evolution of the mixing height.

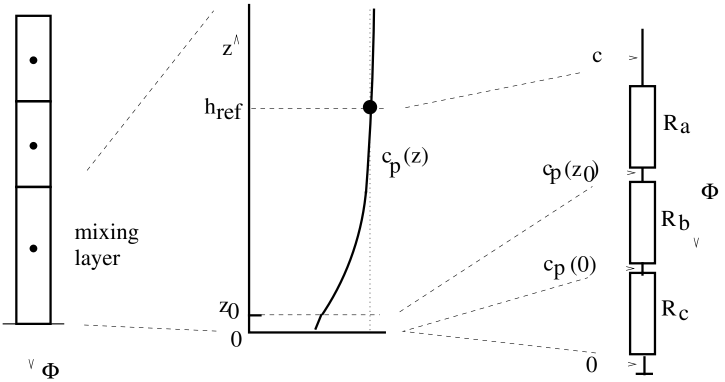


Figure 2.9: *Modeling of deposition flux and concentration profile. The deposition flux in the mixing layer is the result of a concentration gradient over three resistances for the atmosphere (R_a), the viscous sublayer (R_b), and the vegetation surface (R_c). Whenever a concentration near the ground is required, it should be computed from the deposition profile.*

The advection-diffusion operator $\mathcal{L}_{ade}$ in (2.4) is based on κ-discretization and a two stage Runge-Kutta method (*van Loon, 1996*). To compute the flux into a certain cell, the discretization uses a 9-point stencil (see also figure 8.5). The Courrant condition in 2-D provides an upper boundary for the time step:

$$\Delta t \leq \min_{i,j} \frac{1}{2} \Big/ \left(\frac{|u_{ij}|}{\Delta x_{ij}} + \frac{|v_{ij}|}{\Delta y_{ij}} \right) \tag{2.16}$$

where u and v denote the components of the wind vector and Δx and Δy the size of a grid cell. The Courrant condition relates the wind speed to the size of the grid cell. Within a single time step, a parcel of air may not be transported over more than one grid cell since the wind field might change significantly along the trajectory. Since the LOTOS cells are rather large (in order of 60 × 60 km), the Courrant condition is hardly ever violated, and the maximum time step is determined by the chemistry.

2.4.8 Example: budgets around Vreedepeel

To illustrate the impact of the different operations in the model, the changes in concentration in a single grid cell have been investigated in detail. Figure 2.10 shows the net sources and sinks of O_3, NO_x, and VOC due to different processes for the LOTOS cell around Vreedepeel (The Netherlands). Vreedepeel is located between large industrialized area's, and could be classified as sub-urban. The budgets have been computed as hourly averages during a LOTOS simulation over two weeks (first part of august 1997), to limit the impact of special events on the results.

The upper two panels in figure 2.10 show the sources and sinks for ozone in terms of absolute volume and concentration, to show the difference between netto fluxes and impact on concentrations. In terms of volumes, the main sources of ozone are chemical production (during the day), and import from higher altitudes due to raising mixing heights (in the early morning). Main sink is the loss of volume due to lower cell volumes after sunset. Deposition has only a minor impact in terms of total ozone volume. In terms of concentrations, the smaller cell volumes during the night lead however to a strong impact on the concentrations. The deposition leads to a strong ozone gradient from the mixing layer to the reservoir, inducing a downward diffusion flux during the night. With the rise of the mixing layer in the early morning, the decrease due to deposition is compensated for by accumulation of ozone from the higher model layers. Advection of ozone within the mixing layer could be neglected.

The main volume source of NO_x is emission of NO, with an overall higher rate during the day due to higher industrial activities and traffic densities. Main sink is chemical loss (during the day) and volume loss during the evening. In terms of concentrations, emission is again the main source. In spite of the lower emission rates, the impact on night time concentrations is still large due to the rather small cell volume. Since most of the emissions are injected in the bottom cell, a concentration gradient occurs over the cell ceiling. The gradient induces a diffusion flux to higher altitudes during the night, and overall lower concentrations during the rise of mixing layer via mixing with cleaner air. The VOC budgets show similar behaviour as those for NO_x, except that advection is now more important.

A conclusion of this simple budget study is that although a photo-oxidant model focuses on the chemistry, the aspect of vertical exchange is at least as important. The ground level concentrations of the key component ozone are strongly influenced by the rise and fall of the mixing layer: during the morning because of the mixing with clean air, and during the night because of the large impact of deposition in the stable boundary layer. The importance of all these processes should be reminded when the model is compared with measurements; biases between model and data could be caused by errors in many different parameters.

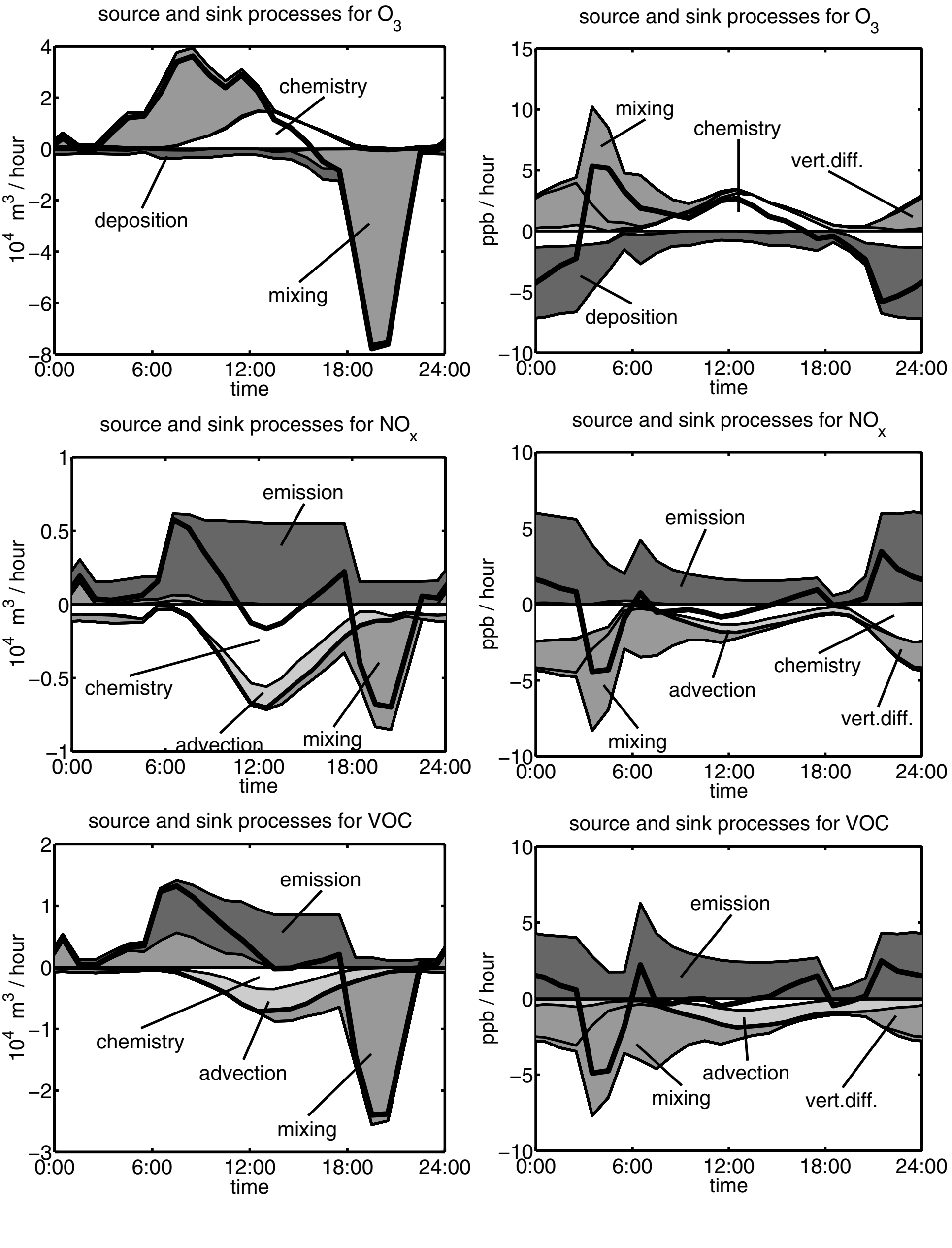


Figure 2.10: *Most important sink and source processes of* O_3, NO_x, *and* VOC *in terms of volumes and concentrations. The solid lines denote the net sink or source. Computed for the* LOTOS *cell around Vreedepeel (The Netherlands), for a model simulation over the first two weeks of august 1997.*

Chapter 3

Data Assimilation

In this chapter, some basic properties of data assimilation are introduced. The target to be achieved is discussed in terms of states, models, and data. Two common used classes of data assimilation techniques are described: linear filters and variational methods. The emphasis will be on the Kalman filter, since this technique will be used to assimilate data with the LOTOS *model. The general form of a stochastic model required by the Kalman filter is described in detail, as well as the stochastic model for a Kalman smoother.*

3.1 Introduction

Let for a physical process the state at a time $t[k]$ be described by a state vector $\mathbf{x}^t[k] \in \mathbb{R}^n$. If for example the process concerns the air quality in Europe, the elements of the state vector could be filled with gas-phase concentrations. The superscript 't' denotes that $\mathbf{x}^t$ is the *true* state; the exact value is probably unknown, but at least, it exists. To obtain insight in the true state, a model is developed. For a time dependent process, we assume for example that the state at a time $t[k+1]$ is a function of the state at $t[k]$ and other time dependent entities:

$$\mathbf{x}^f[k+1] \;=\; \mathbf{M}(\mathbf{x}^f[k], t[k]) \tag{3.1}$$

The superscript 'f' denotes that $\mathbf{x}^f$ is a *forecast* of the true state, in the best case a good approximation. For example, in the context of the LOTOS model (2.3), the state vector $\mathbf{x}^f$ is the concentration vector $\mathbf{c}$ and $\mathbf{M}$ denotes the model operator $\boldsymbol{\mathcal{L}}$. However, these entities are often only a part of the state and the model, and therefore the general notations $\mathbf{x}$ and $\mathbf{M}$ will be used.

The entities in the state are to be compared with data from an observational network, for example measurements of ozone. All available data for a time $t[k]$ is stored in a vector $\mathbf{y}^o[k] \in \mathbb{R}^r$. Apart from the observed data, there is also the 'true' data: the true values of the entities being measured, without measurement errors. The true data is supposed to be related with the true state according to a linear observation model:

$$\mathbf{y}^t[k] \;=\; \mathbf{H}'[k]\,\mathbf{x}^t[k] \tag{3.2}$$

Each measurement is supposed to be equal to a linear combinations of elements of the state. Often, there will be only one non-zero element in a row of $\mathbf{H}'$, equal to one. For simplicity we will assume that $\mathbf{H}'[k]$ is constant in time, that is, the number of data values is the same during each time step, and for each data item, the interpolation from the state is always the same too. A time dependent $\mathbf{H}'$ is not essentially different from a stationary one, and will only complicate the notations. The notation with a transposed matrix $\mathbf{H}'$ was chosen to maintain a consistent notation in case of a scalar observation; the matrix $\mathbf{H}'$ then reduces to a row vector $\mathbf{h}'$.

Through the observation operator $\mathbf{H}'$, a forecast of the observed data could be made from the forecast of the state:

$$\mathbf{y}^f[k] \;=\; \mathbf{H}'\mathbf{x}^f[k] \tag{3.3}$$

In practice there will be a difference between $\mathbf{y}^f$ and the actual observed value $\mathbf{y}^o$, often referred to as the *residual* or *innovation* vector (*Daley, 1991*):

$$\mathbf{d}^f[k] \;=\; \mathbf{y}^o[k] \;-\; \mathbf{y}^f[k] \;=\; \mathbf{y}^o[k] \;-\; \mathbf{H}'\mathbf{x}^f[k] \tag{3.4}$$

The ultimate target of a modeler is to have the residues as small as possible. Under the name of *data assimilation*, a variety of methods exist which all try to reach this target. The term 'data assimilation' refers to the fact that all methods try to merge model forecasts and measurements using the benefits of both sources of information. The final goal of an assimilation procedure is to obtain a time series of *assimilated* or *analyzed* states $\mathbf{x}^a[k]$ given model and measurements, with assimilated residues as small as possible:

$$\mathbf{x}^a[k] \;=\; \mathcal{A}(\,\mathbf{M}\,,\;\ldots\mathbf{x}^f[k]\ldots\,,\;\mathbf{H}'\,,\;\ldots\mathbf{y}^o[k]\ldots\,) \tag{3.5a}$$

$$\mathbf{d}^a[k] \;=\; \mathbf{y}^o[k] \;-\; \mathbf{H}'\mathbf{x}^a[k] \quad , \quad \|\mathbf{d}^a[k]\| \;\ll\; \left\|\mathbf{d}^f[k]\right\| \tag{3.5b}$$

Two families of data assimilation techniques are common used: variational methods and linear filters. General form and examples of both families will be discussed in the next sections.

3.2 Variational methods

Variational methods for data assimilation are based on minimization of a cost function. A common used approach in meteorology is to use the cost function to obtain an optimal initial state $\mathbf{x}[0]$ for a model forecast. The initial state should be not too different from a background state $\mathbf{x}^b$, and lead to model forecast $\mathbf{x}^f[1],\ldots,\mathbf{x}^f[K]$ as close to the data $\mathbf{y}^o[1],\ldots,\mathbf{y}^o[K]$ as possible. A suitable cost function to achieve this is the following (*Talagrand and Courtier, 1987*):

$$\begin{aligned}\mathcal{J}(\mathbf{x}[0]) \;=\;& \frac{1}{2}\left(\mathbf{x}[0]-\mathbf{x}^b\right)'(\mathbf{P}^b)^{-1}\left(\mathbf{x}[0]-\mathbf{x}^b\right)\\ &+\frac{1}{2}\sum_{k=1}^{K}\left(\mathbf{H}'\mathbf{x}^f[k]-\mathbf{y}^o[k]\right)'(\mathbf{R}[k])^{-1}\left(\mathbf{H}'\mathbf{x}^f[k]-\mathbf{y}^o[k]\right)\end{aligned} \tag{3.6}$$

Cost function (3.6) is the sum of quadratic terms, increasing if $\mathbf{x}[0]$ differs from the background state or if the model forecast $\mathbf{H}'\mathbf{x}^f[k]$ deviates from the observed data $\mathbf{y}^o[k]$. The weight of each of these differences in $\mathcal{J}$ is determined by the ratio between the background covariance $\mathbf{P}^b$ and the representation covariance $\mathbf{R}[k]$. The quadratic form of $\mathcal{J}$

is related to the multivariate Gaussian probability density. For a Gaussian distributed vector $\mathbf{x} \sim \mathcal{N}\left(\mathbf{x}^b, \mathbf{P}^b\right)$, the probability density p is given by:

$$p(\mathbf{x}) \;=\; \frac{1}{\sqrt{(2\pi)^n\,|\mathbf{P}^b|}}\,\exp\left(-\tfrac{1}{2}(\mathbf{x}-\mathbf{x}^b)'(\mathbf{P}^b)^{-1}(\mathbf{x}-\mathbf{x}^b)\right) \tag{3.7}$$

Minimization of the first term of $\mathcal{J}$ in (3.6) is equivalent to maximization of the probability that the background error $\mathbf{x}[0]-\mathbf{x}^b$ is a sample from $\mathcal{N}\left(\mathbf{o},\mathbf{P}^b\right)$; in this example, the maximum is reached for $\mathbf{x}[0]=\mathbf{x}^b$. Minimization of each of the other terms in $\mathcal{J}$ maximizes the probability that the representation error $\mathbf{H}'\mathbf{x}^f[k]-\mathbf{y}^o[k]$ is a sample of $\mathcal{N}(\mathbf{o},\mathbf{R}[k])$, reached for $\mathbf{H}'\mathbf{x}^f[k]=\mathbf{y}^o[k]$. If the background and representation errors are independent from eachother, which is almost ever assumed to be true, the joint probability of the separate events is maximized by minimization of $\mathcal{J}$.

If a suitable initial state $\mathbf{x}[0]$ has been obtained, the analyzed states are formed with a model forecast:

$$\mathbf{x}^a[0]=\mathbf{x}[0] \qquad , \qquad \mathbf{x}^a[k] \;=\; \mathbf{M}(\,\mathbf{x}^a[k-1]\,) \quad , \quad k=1,\ldots,K \tag{3.8}$$

The final analyzed state $\mathbf{x}^a[K]$ is optimized given data from spatial different locations and from different times in the interval $(t[0],t[K]]$. The variational approach is therefore often referred to as *4D-var*. Data from the period before $t[0]$ is used implicitly if it was used to form the background state $\mathbf{x}^b$. A common used approach is to set the background state to $\mathbf{x}^a[0]$, the last available analyzed state.

The minimization of the cost function is often based on quasi-Newton methods. These methods require computation of the gradient of the cost function, which is simplified by its quadratic form. The gradient is computed efficiently using the adjoint of the model $\mathbf{M}$. If the model is linear ($\mathbf{M}(\mathbf{x})=\mathbf{A}\mathbf{x}$), the adjoint is just the transpose $\mathbf{A}'$, otherwise, the adjoint is the transpose of the tangent linear model $\mathbf{A}=\partial\mathbf{M}/\partial\mathbf{x}$. For the large models used in atmospheric research, the linear operator $\mathbf{A}$ is hardly ever available in a matrix form, but is implicitly defined by a complex source code. The adjoint is therefore not simply computed as a transpose, but is implemented as an adjoint operator in source code form. The development of an adjoint used to be labor intensive work, but has been simplified significantly by automatic differentiation tools such as TAMC (*Giering and Kaminski, 1998*) or O∂YSSÉE (*Rostaing et al., 1993*).

After successful application of variational methods in operational weather forecast, 4D-var techniques have become popular data assimilation tools in air pollution modeling too. (*Elbern et al., 1997*) applied a 4D-var technique to the atmospheric chemistry model EURAD, to test the possibilities of data assimilation for online forecast of smog levels. A cost function similar to (3.6) was used to optimize the initial concentrations for the model. The variational approach is also suitable for offline parameter estimations. For this purpose, the cost function is extended with penalties for the parameters to be estimated, to obtain their optimal value during the assimilation interval. Examples of this approach in air pollution applications are described in (*Elbern et al., 2000*), where emission rates of $\mathrm{NO_x}$ and VOC were included in the cost function, and (*Houweling, 2000*), for estimation of global methane emissions.

3.3 Linear filters

Where the variational method is based on minimization of a cost function within a time interval, a *filter* analysis the state each time that data becomes available. In a *linear* filter, the analyzed state is a linear combination of the forecast state and the data elements following the equation:

$$\mathbf{x}^a[k] \; = \; \mathbf{x}^f[k] + \mathbf{K}[k]\,\mathbf{d}_k^f \qquad , \qquad \mathbf{d}^f[k] \; = \; \mathbf{y}^o[k] - \mathbf{H}'\mathbf{x}^f[k] \tag{3.9a}$$

$$\; = \; (\mathbf{I} - \mathbf{K}[k]\,\mathbf{H}')\,\mathbf{x}^f[k] + \mathbf{K}[k]\,\mathbf{y}^o[k] \tag{3.9b}$$

The first form of the analysis equation (3.9a) reflects that the analyzed state is adapted proportional to the residue; the second form (3.9b) reflects that the analyzed state is in between the original state and the measurements.

The *gain* matrix $\mathbf{K}[k]$ describes how elements of the state should be changed given a residue $\mathbf{d}^f[k]$. Each column of $\mathbf{K}$ acts as a point spread function, distributing the update towards one single measurement over all elements of the state vector. Different methods are in use to fill the gain matrix, from simple but cheap to sophisticated but expensive. Three methods will be described, which are in some sense extensions to eachother: direct insertion and/or blending, optimal interpolation, and the Kalman filter.

3.3.1 Direct insertion and blending

The method of direct insertion is based on replacement of state elements by data values. Such a procedure is only possible if there is a one-to-one mapping between observed entities and elements of the state, that is, each row of $\mathbf{H}'$ has exactly one non-zero element, equal to one. The gain matrix $\mathbf{K}$ for direct insertion is equal to a mapping from the measurement to the state vector, which is just the transpose of $\mathbf{H}'$:

$$\mathbf{K}^{DI} \; = \; \mathbf{H} \tag{3.10}$$

The columns of this gain are therefore discrete delta functions rather than point spread functions. Apart from the major advantage of a very simple implementation, direct insertion has almost only disadvantages. First, the method assumes that the measurements $\mathbf{y}^o$ are perfect and do not contain any measurement errors. If there is serious doubt about the quality or the representiveness of the measurements, one could choose to insert a weighted average of $\mathbf{y}^o$ and $\mathbf{H}'\mathbf{x}$. This approach is called *blending* (*Robinson et al., 1998*), and is formulated in terms of the gain matrix

$$\mathbf{K}^{BD} \; = \; \alpha\mathbf{H} \qquad , \qquad \alpha \in [0,1] \tag{3.11}$$

A second major disadvantage from which both direct insertion and blending suffer is the lack of smoothness in the analyzed state. An analyzed state will show peaks at positions where measurements are inserted, and this might lead to instabilities if the analyzed state is propagated by the model. This problem is less important if the number of measurements is large in comparison with the number of elements in the state.

3.3.2 Optimal Interpolation

The assimilation method of optimal or statistical interpolation uses a gain matrix based on an empirical covariance function or matrix. Basic assumption is that the difference between the model forecast and the true state has a known, Gaussian distribution:

$$\mathbf{x}^f[k] \;-\; \mathbf{x}^t[k] \;\sim\; \mathcal{N}\left(\mathbf{o},\mathbf{P}^f[k]\right) \tag{3.12}$$

A similar assumption is made for the difference between the observed data and its true value:

$$\mathbf{y}^o[k] \;-\; \mathbf{y}^t[k] \;\sim\; \mathcal{N}\left(\mathbf{o},\mathbf{R}[k]\right) \tag{3.13}$$

The idea of optimal interpolation (OI) is now to set the analyzed state to the conditional mean of the true state given the observations:

$$\mathbf{x}^a[k] \;=\; \mathrm{E}\left[\,\mathbf{x}^t[k] \mid \mathbf{y}^o[k]\,\right] \tag{3.14}$$

Application of Bayes theorem to Gaussian distributions shows that this could be achieved with a linear gain (*Anderson and Moore, 1979*):

$$\mathbf{x}^a[k] \;=\; \mathbf{x}^f[k] \;+\; \mathbf{K}^{OI}[k]\;\left(\mathbf{y}^o[k]-\mathbf{H}'\mathbf{x}^f[k]\right) \tag{3.15a}$$

$$\text{where}\qquad \mathbf{K}^{OI}[k] \;=\; \mathbf{P}^f[k]\,\mathbf{H}\;\left[\,\mathbf{H}'\mathbf{P}^f[k]\mathbf{H}+\mathbf{R}[k]\,\right]^{-1} \tag{3.15b}$$

Figure 3.1 illustrates the Bayes theorem in terms of probability densities. The gain matrix $\mathbf{K}^{OI}$ in (3.15b) is known under several names, such as *conditional mean gain* and *minimal variance* gain. The first name simply refers to the result of the Bayes theorem, while the second name refers to the property that the error in the analyzed state (3.15a) has the smallest variance of all possible analyzed states $\boldsymbol{\xi}$, if measured with the l_2-norm:

$$\mathrm{E}\left[\,\|\mathbf{x}^t-\mathbf{x}^a\|^2 \mid \mathbf{y}^o\,\right] \;\leq\; \mathrm{E}\left[\,\|\mathbf{x}^t-\boldsymbol{\xi}\|^2 \mid \mathbf{y}^o\,\right] \tag{3.16}$$

A problem is how to choose suitable covariance matrices **P** and **R**. The representation errors between $\mathbf{y}^o$ and $\mathbf{H}'\mathbf{x}$ are often supposed to be uncorrelated, leading to a diagonal matrix for **R**. The diagonal elements of **R** are just the squared standard deviations of the representation error, often set to a constant fraction of the data. For correlated measurements, one could always switch to uncorrelated measurements with the transformation $\tilde{\mathbf{y}} = \mathbf{R}^{-1/2}\mathbf{y}$. Definition of an appropriate forecast error covariance is more complicated, and in fact the main problem in every data assimilation problem. A simple choice for the forecast covariance would be to let $\mathbf{P}_k$ be diagonal too. Forecast errors in two different state elements are now completely uncorrelated. In case of the operator $\mathbf{H}'$ observing only a single state element per row, the net result of a diagonal $\mathbf{P}_k$ is a blending scheme (3.11), with for each observed element i a different $\alpha_i = p_{ii}/(p_{ii}+r_{ii})$. Uncorrelated forecast errors are not common practice, however. In geophysical models such as LOTOS, the patterns in the state vectors are smooth over large spatial distances. Within these distances, the state elements are correlated to eachother, and errors in their estimates are therefore likely to be correlated too. A suitable method to introduce correlations is the definition of a covariance function, defining the covariance between two arbitrary entities in the state vector, located at arbitrary positions.

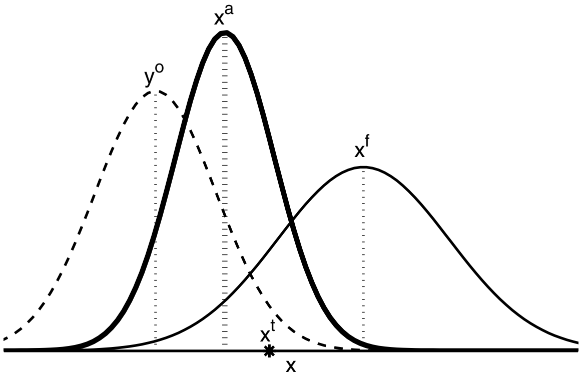


Figure 3.1: *Illustration of Bayes theorem for a scalar state x. The true value* x^t *is located somewhere on the axis. Two estimates of* x^t *are available: the observed data* y^o *with* $y^o - x^t \sim \mathcal{N}(o, R)$*, and the model forecast* x^f *with* $x^f - x^t \sim \mathcal{N}(o, P)$*. The minimum variance distribution for* x^t *is now the normal distribution with mean* $x^a = x^f + K(y^o - x^f)$ *and variance* $P^a = (1-K)P$*, for gain* $K = P/(P+R)$*.*

A valid covariance matrix is then formed by evaluating the covariance function on a finite grid. Useful introductions and applications of covariance functions are for example found in (*Daley, 1991*), or, for covariance functions on a sphere, in (*Gaspari and Cohn, 1999*). An often used approach in atmospheric applications is to define a covariance function with a separate treatment of horizontal and vertical correlations. The horizontal correlations are supposed to be isotropic (the same in each direction), although it is also possible to define a stronger correlation in the direction of the wind (*Rijshøjgaard and Källén, 1997*). Vertical correlations are often less strong due to the layered structure of the atmosphere. Once a general structure of a covariance function has been defined, unknown parameters such as correlation lengths and variances are obtained by fitting the function with measurements. Fitting is based on the distribution of the innovation vectors, which should match:

$$\mathbf{d}^f[k] \;=\; \mathbf{y}^o[k] - \mathbf{H}'\,\mathbf{x}^f[k] \;\sim\; \mathcal{N}\big(\,\mathbf{o}\,,\,\mathbf{H}'\,\mathbf{P}^f(\rho[k])\,\mathbf{H} \,+\, \mathbf{R}[k]\,\big) \tag{3.17}$$

The vector ρ contains unknown parameters of the covariance function. The probability that $\mathbf{d}^f[k]$ is a sample from the distribution at the right hand side of (3.17) is maximized over ρ. The parameters ρ could be estimated adaptively for each single analysis time. Dee (1995) stated that for this approach, the number of measurements should exceed the number of covariance parameters with at least a factor three. If this number is not available for each analysis time, or temporal variations are just small, a suitable ρ could be obtained from time series of residues, by minimization of the difference between their sample covariance and the covariance matrix at the right hand side of (3.17).

3.3.3 Kalman filter

The OI procedure described before has the disadvantage that for each assimilation time, the user needs to specify the forecast error covariance, independent of previous times. However, a large part of the current forecast error arises due to forecast errors made in the past. The Kalman filter (*Kalman, 1960*) can be seen as an extension of the OI scheme, accounting for the evolution of errors from previous times. Or, from a Kalman point of view, the OI method is just a simplification of the Kalman filter, neglecting the time evolution.

The target of the Kalman filter is to obtain a distribution for the true state in terms of a mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$, given the model and the measurements. The first step in a Kalman Filter is specification of an initial distribution for the true state. Similar as for the OI scheme, the distribution should be Gaussian:

$$\mathbf{x}^t[0] \;\sim\; \mathcal{N}\left(\hat{\mathbf{x}}^f[0], \mathbf{P}^f[0]\right) \tag{3.18}$$

The second step in the filter procedure is to specify the error between true state $\mathbf{x}^t[k{+}1]$ and the model forecast $\mathbf{A}[k]\,\mathbf{x}^t[k]$. To simplify coming formulae we use the linear form $\mathbf{Ax}$ for the model, instead of the general form $\mathbf{M}(\mathbf{x})$; the Kalman filter is in fact consistent for linear models only. The model error should be described in terms of a Gaussian distribution:

$$\mathbf{x}^t[k{+}1] \;-\; \mathbf{A}[k]\,\mathbf{x}^t[k] \;\sim\; \mathcal{N}(\mathbf{o}, \mathbf{Q}[k]) \tag{3.19}$$

or, equivalent:

$$\mathbf{x}^t[k{+}1] \;=\; \mathbf{A}[k]\,\mathbf{x}^t[k] \;+\; \boldsymbol{\eta}^t[k] \qquad , \qquad \boldsymbol{\eta}^t[k] \;\sim\; N(\mathbf{o}, \mathbf{Q}[k]) \tag{3.20}$$

The model error $\boldsymbol{\eta}^t$ is supposed to be independent of $\mathbf{x}^t$, and should cover all possible deviations of the model forecast from the true state. Typical errors included in $\boldsymbol{\eta}^t$ are unknown boundary conditions, uncertain model parameters, or just chaos. How to specify the model error correctly is the most difficult task in a filter procedure. For the moment we assume that a suitable definition is available, a more detailed description is given in §3.5. Given the stochastic model (3.19/3.20) and the initial condition (3.18), the Kalman filter is able to compute a probability density of the true state at any time in future. The stochastic model (3.20) completely defines the evolution of the distribution of the true state:

$$\begin{aligned}
\hat{\mathbf{x}}^f[k{+}1] &= \mathrm{E}\left[\,\mathbf{x}^t[k{+}1]\,\right] \\
&= \mathbf{A}[k]\,\hat{\mathbf{x}}^f[k] && (3.21\text{a}) \\
\mathbf{P}^f[k{+}1] &= \mathrm{E}\left[\,\left(\mathbf{x}^t[k{+}1] - \hat{\mathbf{x}}^f[k{+}1]\right)\left(\mathbf{x}^t[k{+}1] - \hat{\mathbf{x}}^f[k{+}1]\right)'\,\right] \\
&= \mathbf{A}[k]\,\mathbf{P}^f[k]\,\mathbf{A}[k]' \;+\; \mathbf{Q}[k] && (3.21\text{b})
\end{aligned}$$

since

$$\begin{aligned}
\mathbf{x}^t[k{+}1] - \hat{\mathbf{x}}^f[k{+}1] &= \mathbf{A}[k]\,\mathbf{x}^t[k] + \boldsymbol{\eta}[k] \;-\; \mathbf{A}[k]\,\hat{\mathbf{x}}^f[k] && (3.22\text{a}) \\
&= \mathbf{A}[k]\,\left(\mathbf{x}^t[k] - \hat{\mathbf{x}}^f[k]\right) \;+\; \boldsymbol{\eta}[k] && (3.22\text{b})
\end{aligned}$$

The third step in the filter is the analysis of data. If observations are available, the mean and covariance should be replaced by analyzed equivalents given the new information.

Thereto, a model should be specified for the representation error between observed data and true value, similar as for the OI scheme:

$$\mathbf{y}^o[k] \;-\; \mathbf{H}'\,\mathbf{x}^t[k] \;\sim\; \mathcal{N}(\mathbf{o},\mathbf{R}[k]) \tag{3.23}$$

or, equivalent:

$$\mathbf{y}^o[k] \;=\; \mathbf{H}'\,\mathbf{x}^t[k] \;+\; \mathbf{v}^t[k] \qquad , \qquad \mathbf{v}^t[k] \;\sim\; N(\mathbf{o},\mathbf{R}[k]) \tag{3.24}$$

If the mean is analyzed with a linear gain $\mathbf{K}$, an analysis equation for the covariance is simply derived too:

$$\begin{aligned}
\hat{\mathbf{x}}^a[k] \;&=\; \hat{\mathbf{x}}^f[k] \;+\; \mathbf{K}[k]\;(\mathbf{y}^o[k] - \mathbf{H}'\,\hat{\mathbf{x}}^f[k]) && \text{(3.25a)}\\
\mathbf{P}^a[k] \;&=\; \mathrm{E}\Big[\;(\mathbf{x}^t[k] - \hat{\mathbf{x}}^a[k])(\mathbf{x}^t[k] - \hat{\mathbf{x}}^a[k])'\;\Big] \\
&=\; (\mathbf{I} - \mathbf{K}[k]\mathbf{H})\;\mathbf{P}^f[k]\;(\mathbf{I} - \mathbf{K}[k]\mathbf{H})' \;+\; \mathbf{K}[k]\;\mathbf{R}[k]\;\mathbf{K}'[k] && \text{(3.25b)}
\end{aligned}$$

since

$$\begin{aligned}
\mathbf{x}^t[k] \;-\; \hat{\mathbf{x}}^a[k] \;&=\; \mathbf{x}^t[k] \;-\; \hat{\mathbf{x}}^f[k] \;-\; \mathbf{K}[k]\;(\mathbf{y}^o[k] - \mathbf{H}'\,\hat{\mathbf{x}}^f[k]) && \text{(3.26a)}\\
&=\; \mathbf{x}^t[k] \;-\; \hat{\mathbf{x}}^f[k] \;-\; \mathbf{K}[k]\;(\mathbf{H}'\mathbf{x}^t[k] + \mathbf{v}^t[k] - \mathbf{H}'\,\hat{\mathbf{x}}^f[k]) && \text{(3.26b)}\\
&=\; (\mathbf{I} - \mathbf{K}[k]\mathbf{H})(\mathbf{x}^t[k] - \hat{\mathbf{x}}^f[k]) \;+\; \mathbf{K}[k]\;\mathbf{v}^t[k] && \text{(3.26c)}
\end{aligned}$$

A common used choice for $\mathbf{K}$ is to use the minimal-variance or conditional-mean-gain, as used in the OI scheme too:

$$\mathbf{K}^{MV}[k] \;=\; \mathbf{P}^f[k]\;\mathbf{H}\;\big[\mathbf{H}'\;\mathbf{P}^f[k]\;\mathbf{H} \;+\; \mathbf{R}[k]\big]^{-1} \tag{3.27}$$

With this gain, equation (3.25b) for the analyzed covariance reduces to a simpler form:

$$\mathbf{P}^{a,MV} \;=\; \big(\mathbf{I} \;-\; \mathbf{K}^{MV}\;\mathbf{H}\big)\;\mathbf{P}^f \;=\; \mathbf{P}^f\;\Big(\mathbf{I} \;-\; \mathbf{H}\,\mathbf{K}^{MV\prime}\Big) \tag{3.28}$$

Why making a difference between a general and the minimal-variance-gain, if the later provides simpler equations and an analysis with minimal variance? The minimal-variance-gain (3.27) is the result of a pure algebraic minimization, based on the idea that the forecast and representation error covariances are exactly described by $\mathbf{P}^f$ and $\mathbf{R}$. If one of these is known to be inaccurate, however, a gain matrix different from (3.27) might be used if there are good arguments to do this. As an example, the common used low-rank approximations for the covariance matrix $\mathbf{P}^f$ (chapter 6) suffer from spurious correlations between elements which are in practice uncorrelated. To not let these correlations disturb the filtering process, the gain might be formed from a covariance matrix from which the spurious elements are removed (*Houtekamer and Mitchell, 2001*). Another example is the gain matrix used in the POENKF filter described in §6.7, which is constructed out of two different covariance matrices. In both cases, equation (3.25b) should be used to analyze the covariance matrix rather than (3.28). After (*Bucy and Joseph, 1968*), equation (3.25b) is sometimes called the Joseph form of the covariance analysis. The Joseph form is computational more expensive than the minimal variance form, but is less sensitive to roundoff errors in the gain.

The final result of the Kalman filter using the minimal variance gain is a time series of a mean and covariance of the true state, equal to the conditional mean and covariance given all available data from the past:

$$\hat{\mathbf{x}}^a[k] = \mathrm{E}\left[\, \mathbf{x}^t[k] \mid \mathbf{y}^o[k], \mathbf{y}^o[k-1], \ldots \right] \tag{3.29a}$$

$$\mathbf{P}^a[k] = \mathrm{E}\left[\, (\mathbf{x}^t[k] - \hat{\mathbf{x}}^a[k])(\mathbf{x}^t[k] - \hat{\mathbf{x}}^a[k])' \mid \mathbf{y}^o[k], \mathbf{y}^o[k-1], \ldots \right] \tag{3.29b}$$

The power behind the Kalman filter algorithm is that this is achieved with a sequential procedure. Once initialized, the filter is able to compute the result for $t[k+1]$ given entities from $t[k]$ only. The total Kalman filter procedure is in fact not very different from the OI procedure. The only difference is the origin of the forecast covariance. In OI, the user should specify this matrix for each time step, or simply assume that it is time independent. In the Kalman filter, the forecast covariance is specified only once at the initial time, and then propagated 'automatically' by the filter. Automatically is quoted here, since the problem of specification of the forecast covariance has been replaced by the problem of specification of the model error. Besides, the 'automatic' propagation is not cheap. The propagation of the covariance in eq. (3.21b) requires $2n$ evaluations of the model $\mathbf{A}$, which is expensive or even impossible if n becomes large. For air pollution models such as LOTOS, the size n of the state vector is $\mathcal{O}\left(10^4\right)$, and approximations to the original Kalman filter should be considered; an extensive discussion of approximate filters is left for chapter 6.

An important property of the Kalman filter is the conservation of Gaussian probabilities in case of a linear model. Both the forecast equations (3.21) and the analysis (3.25) preserve an initial Gaussian distribution for the true state, independent of the contents of the linear model $\mathbf{A}$ and the gain $\mathbf{K}$. This property is lost if the model is nonlinear, however. For a general nonlinear model $\mathbf{M}(\mathbf{x})$ instead of the linear form $\mathbf{Ax}$, the propagation (3.21) of mean and covariance could be approximated. nonlinear methods are discussed in detail in chapter 7.

3.4 Kalman filter versus variational methods

An important difference between 4D-var and Kalman filtering is the form of the final result (figure 3.2). 4D-var provides an assimilated result in the form of piece-wise model evaluations, with discontinuities at the assimilation intervals; the Kalman filter provides a result in terms of a mean and covariance. The filter mean is comparable with the 4D-var state, but differs at two points: the mean is discontinuous at each time step where data is available, and in between, the mean is only the result of a model evaluation if the model is linear. It is possible to show that for linear models and a quadratic cost function $\mathcal{J}$ as in (3.6), the 4D-var state at the end of the time interval is equal to the analyzed filter mean, if the same data and representation error covariance is used.

The availability of a mean and covariance for the true state is an important advantage of the Kalman filter. The covariance is a measure for the expected error in the state elements. A description of the quality is therefore included in the result. Time series of the covariance provide useful insight in how errors introduced in certain parts of the state evolve in time, and whether these become dominant or just fade away.

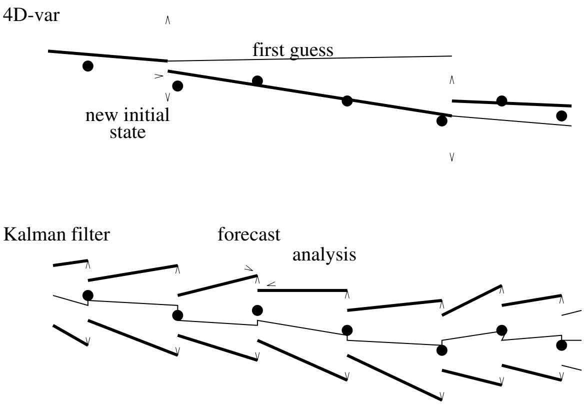


Figure 3.2: *Illustration of assimilation with 4D-var and Kalman filter. 4D-var provides an assimilation result in terms of a discontinuous model evaluation; the initial state at the start of a time interval is optimized to match with the measurements in the interval. The Kalman filter provides a result in terms of a mean and covariance, discontinuous at each time step that data is available.*

Both the Kalman filter and the variational methods are suitable to be used in online forecast applications. Nowadays operational weather forecast are based on the 4D-var approach, as a follow up of the OI (3D-var) techniques used before. There is a tendency to extend the assimilation procedure with Kalman filter techniques, however, for example to obtain useful background covariances for the cost function. Application of the Kalman filter to the large models in use for weather forecast has been facilitated by the introduction of low-rank approximations, discussed in chapter 6. For offline applications such as parameter estimations, the variational approach is often favored due to its clear insight in how parameters are optimized, by comparison of model forecasts based on certain parameter values with measurements. The Kalman filter could be used for parameter estimation too, by implementation of a Kalman *smoother*, described in §3.6. The theory behind the Kalman smoother is less clear than that for the variational method, however.

The effort to be put in the implementation is very different for the two approaches. For the 4D-var approach, a vast amount of time should be spent on building the adjoint code. This is a difficult and labour intensive work, although it has been facilitated by the introduction of automatic differentiation tools. Every minor change in the model could effect the adjoint, and maintenance of both codes should be matched carefully. Compared with 4D-var, a Kalman filter is in general quite simple to implement, since only the forward model is in use. The computational costs are hard to compare; for the Kalman filter these are dominated by the propagation of the covariance matrix, and for 4D-var by the number of iterations required for minimization of the cost function. Data assimilation with either 4D-var or the Kalman filter is expensive anyway, and effort has to be put in keeping the methods feasible.

For this research, the Kalman filter was chosen as the data assimilation tool for LOTOS.

Not needing to build an adjoint model was seen as a major advantage here, since an adjoint of the chemistry model is complicated. Other advantages of the Kalman filter which were considered are the availability of an analyzed covariance, describing the quality of the result, and the simple introduction of uncertainties in model parameters, described in the next section.

3.5 Stochastic model for the Kalman filter

A critical step in the assimilation procedure, present in both the Kalman filter and 4D-var, is the development of criteria for model errors, in the form of a stochastic model or a cost function. The user should quantify the error in model entities which are believed to be uncertain, such as initial states and model parameters. Also the representation errors for comparison with measurements need to be quantified. The description of the errors completely defines the result of the assimilation procedure, and is therefore more important than the method actually implemented. In spite of this importance, descriptions of data assimilation systems tend to describe 'technical' details about filter implementation and adjoint codes first, followed by a short description of the error model used (an approach followed in this chapter too). One reason for the lower attention paid to the error model is that for most applications hardly any knowledge about the errors is available. Insight is obtained itterative, by application of an error model with the chosen assimilation scheme, investigation of the results, changing the error model according to the results, etc.

The stochastic model (3.19) required for the Kalman filter is based on a specification of the error made by the deterministic model. That is, if the model is fed with the true initial state, what should be added to the model forecast to obtain the true state? For the LOTOS model $\boldsymbol{\mathcal{L}}$ defined in (2.3), the quest for the stochastic model is thus to define the contents of the vector $\boldsymbol{\eta}[k]$ such that:

$$\mathbf{c}^t[k+1] \;=\; \boldsymbol{\mathcal{L}}(\mathbf{c}^t[k],t[k]) \;+\; \boldsymbol{\eta}[k] \quad , \quad \boldsymbol{\eta}[k] \;\sim\; \mathcal{N}(\mathbf{o},\mathbf{Q}_{\mathcal{L}}) \tag{3.30}$$

The error term $\boldsymbol{\eta}[k]$ should quantify all possible difference between $\mathbf{c}[k+1]$ and $\boldsymbol{\mathcal{L}}(\mathbf{c}[k],t[k])$, present due to imperfectness of the chemistry/advection/diffusion/deposition operators, the finite grid, uncertainties in the injected emissions, interpolations in the meteorological input, errors in the landuse database, etc. Although the degree of freedom in $\boldsymbol{\eta}$ is in theory equal to n, the modelled degree is often smaller. If for example strong spatial correlations exist between the elements of the state, the bulk of the forecast error could be described in a limited number of modes. The stochastic model is therefore written in the form:

$$\mathbf{c}^t[k+1] \;=\; \boldsymbol{\mathcal{L}}(\mathbf{c}^t[k],t[k]) \;+\; \mathbf{G}_{\mathcal{L}}[k]\,\mathbf{w}[k] \quad , \quad \mathbf{w}[k] \;\sim\; \mathcal{N}(\mathbf{o},\mathbf{I}) \tag{3.31}$$

where $\mathbf{G}_{\mathcal{L}}\mathbf{G}_{\mathcal{L}}{}' = \mathbf{Q}_{\mathcal{L}}$ and $\mathbf{w}$ is white noise vector (uncorrelated in time) with a number of elements equal to the degree of freedom in $\boldsymbol{\eta}$. Stochastic models of this form are for example obtained from an analysis with empiric orthogonal functions (EOF's). In this approach, the columns of the matrix $\mathbf{G}_{\mathcal{L}}$ are filled with the dominant singular vectors of a sample covariance, computed over a large batch of model states. A stochastic model based on EOF's is the basic idea behind the SEIK filter (see §6.5.2). Another way to fill the matrix

$\mathbf{G}_{\mathcal{L}}$ is to use prior knowledge about uncertainties in the state. The concentrations of emitted components are for example highly uncertain, since spatial and temporal variations in the emissions are rather unknown (see §2.4.4). The uncertainty in emissions could be modelled according to:

$$\mathbf{e}^t[k] \;=\; \mathbf{e}[k] \;+\; \mathbf{G}_e\,\mathbf{w}[k] \quad , \qquad \mathbf{w}[k] \;\sim\; \mathcal{N}(\mathbf{o},\mathbf{I}) \tag{3.32a}$$

where $\mathbf{e}$ contains the deterministic emissions, $\mathbf{e}^t$ is the true value, and $\mathbf{G}_e$ distributes the noise $\mathbf{w}$ over the emission array. Emissions are simply injected into grid cells, and the effect of the uncertainty on the concentration array could be modelled with:

$$\mathbf{c}^t[k+1] \;=\; \boldsymbol{\mathcal{L}}(\mathbf{c}^t[k],t[k]) \;+\; \boldsymbol{\Gamma}_e\,\mathbf{G}_e\,\mathbf{w}[k] \tag{3.33}$$

where matrix $\boldsymbol{\Gamma}_e$ assigns the emissions to elements of the concentration array. If the emissions are subject to fast chemistry, the matrix $\boldsymbol{\Gamma}$ should account for changes in other than emitted components too. This could be achieved by the development of a special chemistry model, describing the impact of variations in emitted components. For the strong nonlinear chemistry, this is hard to achieve without knowledge of the prior chemical state, however. A better method is therefore to treat the emissions as a model input and to define these as stochastic, instead of adding the error caused by uncertain emissions afterwards:

$$\begin{aligned}\mathbf{c}^t[k+1] \;&:=\; \boldsymbol{\mathcal{L}}(\mathbf{c}^t[k],t[k],\mathbf{e}^t[k]) \\ &=\; \boldsymbol{\mathcal{L}}(\mathbf{c}^t[k],t[k],\mathbf{e}[k]+\mathbf{G}_e\mathbf{w}[k]) \quad , \qquad \mathbf{w}[k] \;\sim\; \mathcal{N}(\mathbf{o},\mathbf{I})\end{aligned} \tag{3.34}$$

This approach will be used in chapter 4 to build a stochastic model around the LOTOS model: uncertainties are assigned to a model parameter, and the stochastic parameter is used as input to the model.

A disadvantage of modeling uncertainties using a white noise input $\mathbf{w}$ is the introduction of rapid fluctuations, since $\mathbf{w}$ is uncorrelated in time by definition. To specify a more smoothed uncertainty, a colored noise process γ could be used instead of $\mathbf{w}$ (*Jazwinski, 1970*). In scalar form, a colored noise process satisfies the equation:

$$\gamma[k+1] \;=\; \alpha\gamma[k] \;+\; \sqrt{1-\alpha^2}\,\sigma\,w[k] \qquad , \qquad w[k] \;\sim\; \mathcal{N}(0,1) \tag{3.35}$$

If the time correlation parameter α is set to zero, the colored noise process is just equal to a white noise process with zero mean and variance σ^2. For $\alpha \in (0,1)$, a sample $\gamma[k]$ has a probability relative to α to be close to $\gamma[k-1]$ since the expected value is equal to $\alpha\gamma[k-1]$ and the variance is small (less than σ^2). The expected mean and variance over a long (infinite) time period are equal to zero and σ^2 respectively, however. For an α equal to one, $\gamma[k]$ becomes a constant value; to ensure that a large batch of samples of γ remains a variance of σ^2, the initial sample, e.g. $\gamma[0]$, should be random distributed with the desired statistics. The autocorrelation of the colored noise process is equal to $\mathrm{E}[\,\gamma[k+l]\,\gamma[k]\,] = \alpha^l$. Definition (3.35) is easily extended from a discrete time process $\gamma[k]$ to a continues time process $\gamma(t)$, with the discrete processes formed from samples of $\gamma(t)$ on t_k, t_{k+1}, etc. If the time step between two samples is not constant, the parameterization $\alpha = \exp(-|t_{k+1}-t_k|/\tau)$ could be used to obtain a stationairy autocorrelation function, and $\gamma(t)$ has become the common used *Ornstein-Uhlenbeck process* (*Jazwinski, 1970*). A generalization from scalar γ to a vector γ is easily made if the elements of the vector are independent of eachother. In that case, α and σ are replaced by diagonal matrices. If some kind of correlation is assumed between the elements of γ, full matrices $\mathbf{A}_\gamma$ and $\mathbf{G}_\gamma$ should be defined as the counterparts of α and $\sigma_\alpha = \sigma\sqrt{1-\alpha^2}$. To keep the notations simple, we will adopt the convention that elements of a vector γ are uncorrelated and all have the same time-correlation parameter α and variance σ^2, thus $\mathbf{A}_\gamma$ is equal to $\alpha\mathbf{I}$ and $\mathbf{G}_\gamma = \sigma_\alpha\mathbf{I}$.

3.6 The Kalman smoother

Given a stochastic model for dynamics and observations, the Kalman filter is able to compute the optimal estimate of the current state given all data from the past. Future measurements are not taken into account. For online forecast applications this is not a problem, since these are not available anyway. However, for offline applications such as parameter estimations, not taking into account data from after the analysis time is a serious disadvantage of a filter. Data from behind the analysis time is sometimes the only source of information about the value of a parameter at the analysis time. For example, emissions released at a certain moment are only visible in measurements after the time required to travel from source to receptor. The problem of not taking into account future data is less strong if parameters are estimated in a 4D-var context, since the cost function could be configured to estimate an initial value at the begin of a time interval given the data in the rest of the interval. With a special form of the stochastic model, the same result could be achieved in a Kalman context; the 'filter' is then called a *smoother*. For an extended description of smoothing in terms of Bayesian statistics we refer to (*Evensen and van Leeuwen, 2000*).

The general idea in a Kalman Smoother is to augment the state vector with the values of parameters in the past. For example, the state could exist of the concentration array of LOTOS at $t[k]$ augmented with a parameter vector $\boldsymbol{\lambda}_0$ valid for $t[0]$:

$$\mathbf{x}[k] \;=\; \begin{pmatrix} \mathbf{c}[k] \\ \boldsymbol{\lambda}_0 \end{pmatrix} \tag{3.36}$$

With a well designed stochastic model, the Kalman filter is able to compute the covariance of the true state:

$$\mathbf{P}^f[k] \;=\; \begin{pmatrix} \mathbf{P}_{cc}[k] & \mathbf{P}_{c\lambda_0}[k] \\ \mathbf{P}_{c\lambda_0}[k]' & \mathbf{P}_{\lambda_0\lambda_0} \end{pmatrix} \tag{3.37}$$

where $\mathbf{P}_{cc}$ and $\mathbf{P}_{\lambda_0\lambda_0}$ describe the covariances of $\mathbf{c}$ and $\boldsymbol{\lambda}_0$ respectively, and $\mathbf{P}_{c\lambda_0}$ the covariance between $\mathbf{c}$ and $\boldsymbol{\lambda}_0$. If the later is specified correctly, a measurement of a concentration provides information about the parameters at $t[0]$. Analysis of measured concentrations leads to a more accurate estimate of the concentration array $\mathbf{c}$, and thus, via the correlations in the covariance, also to a more accurate estimate of $\boldsymbol{\lambda}_0$. A suitable stochastic model to achieve this is the following:

$$\begin{pmatrix} \mathbf{c}[k+1] \\ \boldsymbol{\lambda}_0[k+1] \end{pmatrix} \;=\; \begin{pmatrix} \mathcal{L}(\mathbf{c}[k], \boldsymbol{\lambda}_0[k]) \\ \boldsymbol{\lambda}_0[k] \end{pmatrix} \quad , \quad \boldsymbol{\lambda}_0[0] \sim \mathcal{N}\left(\mathbf{o}, \mathbf{P}_{\lambda_0\lambda_0}\right) \tag{3.38a}$$

$$\approx \begin{pmatrix} \mathbf{A}_\mathcal{L} & \mathbf{B}_\mathcal{L} \\ \mathbf{O} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{c}[k] \\ \boldsymbol{\lambda}_0[k] \end{pmatrix} + \begin{pmatrix} \mathbf{O} \\ \delta_{0k}\mathbf{G}_{\lambda_0} \end{pmatrix} \mathbf{w}[k] \tag{3.38b}$$

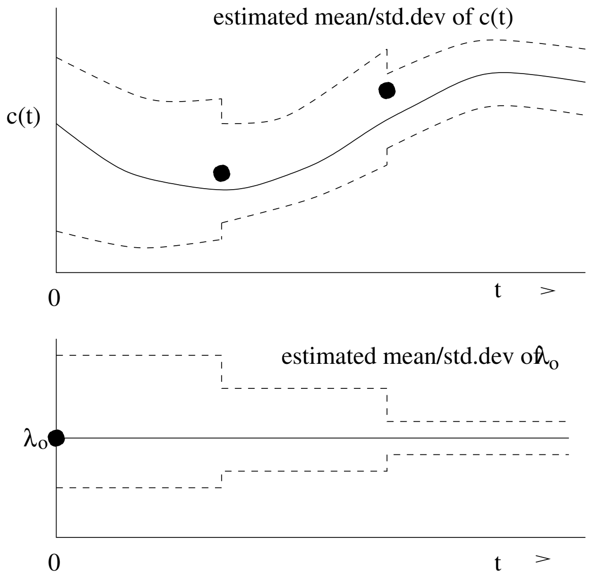


Figure 3.3: *Illustration of parameter estimation with a fixed point Kalman smoother. The constant parameter λ_0 with unknown value is correlated with a concentration $c(t)$, for which 2 measurements are available. The smoother provides estimates of parameter and concentration in terms of a mean and covariance; if the covariance between λ_0 and $c(t)$ is computed correctly, assimilation of the data narrows the estimation band towards the true value. The exact value of λ_0 will never be known since the measurements contain random errors.*

where $\boldsymbol{\lambda}_0[k]$ is an estimate of $\boldsymbol{\lambda}_0$ made at $t[k]$; matrices $\mathbf{A}_{\mathcal{L}}$ and $\mathbf{B}_{\mathcal{L}}$ denote linear approximations of $\boldsymbol{\mathcal{L}}$ with partial derivatives to $\mathbf{c}$ and $\boldsymbol{\lambda}_0$ respectively. The covariance $\mathbf{P}_{\lambda_0\lambda_0} = \mathbf{G}_{\lambda_0}\mathbf{G}_{\lambda_0}{}'$ is an initial guess for the covariance of the true value of $\boldsymbol{\lambda}_0$; for simplicity we assume a zero mean. The discrete Dirac function δ_{0k} is 1 for $k = 0$ and 0 elsewhere. The stochastic model (3.38) describes that our best estimate of $\boldsymbol{\lambda}_0$ at $t[k+1]$ is the same as the previous estimate, since no extra information has become available between $t[k]$ and $t[k+1]$. If a Kalman filter is applied to this model, the mean and covariance for $\boldsymbol{\lambda}_0$ are equal to the initial guesses $\mathbf{o}$ and $\mathbf{P}_{\lambda}$ until measurements of concentrations are assimilated; see the illustration in figure 3.3. A Kalman smoother in this configuration is called a fixed point smoother: data is assimilated to estimate entities at a single moment in the past.

The fixed point smoother could be extended to estimate parameters at multiple times, by augmenting the state with a time series of parameters. The covariance matrix should then describe the covariances between the concentration array and all previous values of the parameter. A suitable stochastic model is the following:

$$\begin{pmatrix} \mathbf{c}[k+1] \\ \boldsymbol{\lambda}_{k+1}[k+1] \\ \boldsymbol{\lambda}_{k}\ \ [k+1] \\ \boldsymbol{\lambda}_{k-1}[k+1] \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathcal{L}(\mathbf{c}[k], \boldsymbol{\lambda}_k[k]) \\ \mathbf{w}_{\lambda}[k] \\ \boldsymbol{\lambda}_k[k] \\ \boldsymbol{\lambda}_{k-1}[k] \\ \vdots \end{pmatrix} , \quad \mathbf{w}_{\lambda}[k] \sim \mathcal{N}\left(\mathbf{o}, \mathbf{P}_{\lambda_k\lambda_k}[k]\right) \tag{3.39}$$

or, with a colored noise model for $\boldsymbol{\lambda}[k]$:

$$\begin{array}{ccccccc} \begin{pmatrix} \mathbf{c}[k+1] \\ \boldsymbol{\lambda}_{k+1}[k+1] \\ \boldsymbol{\lambda}_{k}\ \ [k+1] \\ \boldsymbol{\lambda}_{k-1}[k+1] \\ \vdots \end{pmatrix} & = & \begin{pmatrix} \mathcal{L}(\mathbf{c}[k], \boldsymbol{\lambda}_k[k]) \\ \alpha\, \boldsymbol{\lambda}_k[k] \\ \boldsymbol{\lambda}_k[k] \\ \boldsymbol{\lambda}_{k-1}[k] \\ \vdots \end{pmatrix} & + & \begin{pmatrix} \mathbf{O} \\ \sigma_{\alpha}\mathbf{I} \\ \mathbf{O} \\ \mathbf{O} \\ \vdots \end{pmatrix} & \mathbf{w}[k] \\ \\ \mathbf{x}[k+1] & = & \mathbf{M}(\mathbf{x}[k]) & + & \mathbf{G} & \mathbf{w}[k] \end{array} \tag{3.40}$$

Application of the Kalman filter to this stochastic model provides estimations $\boldsymbol{\lambda}_k[k+l]$ of the parameters $\boldsymbol{\lambda}[k]$ given data up to $t[k+l]$, for lags $l = 0, 1, 2, \ldots$. The distribution of the lag-zero estimate $\boldsymbol{\lambda}_k[k]$ should be prescribed by the user. The estimates are likely to converge after a number of analyses, since in typical application the impact of a parameter fades to zero after a while; see illustration in figure 3.4. In (*Cohn et al., 1994*) this property is used to build a *fixed lag* smoother, where the state contains parameters over a fixed time interval only. The lag is comparable with the length of the assimilation interval in 4D-var. A difference between 4D-var and the fixed lag smoother is that the smoother provides estimates at each discrete time, while in most 4D-var applications, only estimates valid for an entire interval are made.

The quality of smoothed estimates is determined by how accurate the smoother computes the covariance matrix. Accurate computation is complicated by nonlinearities in the model, although this is not necessarily a problem if computation time is no constraint (see discussion about nonlinear methods in chapter 7). Another complication for the smoother is the use of approximate covariance matrices, if the full covariance matrix is too large to store. For example, the low-rank approximations used in this research (chapter 6) suffer from truncation of smaller correlations. In a low-rank Kalman smoother, more effort needs to be taken to remain at least the strongest temporal correlations. The number of parameters in the augmented state vector should therefore not be too large, to keep storage and preservation of the correlations feasible.

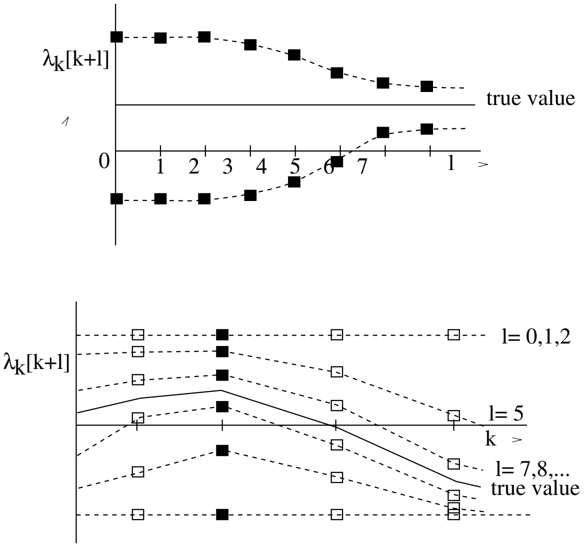


Figure 3.4: *Illustration of parameter estimation with a fixed lag Kalman smoother. Estimates of the mean and standard deviation of* $\lambda[k]$ *are available for different lags* $l = 0, 1, 2, \ldots$ *. The upper panel shows how these estimates would look like if the impact of a stochastic* $\lambda[k]$ *is visible in the measurements for the first time at* $t[k+2]$*, and fade away after* $t[k+7]$*. The lower panel shows a possible time series for the true value of* $\lambda[k]$ *(solid), and estimated mean plus and minus standard deviation (dashed). Up to lag 2, the estimation is equal to the first guess, here a zero mean and constant standard deviation. The quality of the estimate becomes more accurate for larger lag length. In this example, the estimate is not improved any more after lag 7.*

Chapter 4

Application, part I: UK experience

A data assimilation technique based on a Kalman filter has been applied in combination with the tropospheric chemistry model LOTOS. *Experiments with simulated ozone measurements show that the filter is able to account for uncertainties in emissions of* NO_x *and* VOC, *photolysis rates, and deposition velocity of ozone. The uncertainty in most of these parameters is reduced significant by the assimilation of ozone measurements only; accurate estimation of* VOC *emission requires measurements of other components too. A combination of uncertain parameters is necessary for application with measurements from an observational network.* [1]

4.1 Introduction

For application of a Kalman filter to the LOTOS model, a stochastic model should be specified for the model error. The stochastic model should describe the difference between model forecast and true state, if the forecast is made from the true initial state. Given this specification, the Kalman filter is able to assimilate measurements in the LOTOS model. The goal of the research described in this chapter is to identify an appropriate stochastic model.

Traditionally, the stochastic model describes the covariances between all elements of the state vector in terms of a covariance function, relating errors in one element of the state to errors in other elements. Early operational data assimilation schemes based on optimal interpolation used covariance functions to obtain optimal initial conditions for, for example, numerical weather prediction. Many covariance functions are based on simple parameterizations of standard deviation and spatial correlations. An approach often used in atmospheric applications is to separate horizontal and vertical correlations, with an isotropic correlation function in the horizontal (*Daley, 1991*). For use in 4D-var or Kalman like assimilation schemes, it is possible to extend the covariance function with temporal variations (*Eskes et al., 1999*). The unknown parameters in the covariance such as correlation lengths are obtained from fitting towards measurements (*Dee, 1995; Mitchell and Houtekamer, 2000*); a large number of spatially distributed measurements should be available to estimate the correlation parameters correctly. For the LOTOS model with ozone as the main element in the state, a natural choice for the stochastic model would be a specification of covariances

---

[1] Partly in *Modeling and prediction of environmental data in space and time using Kalman filtering* by A.W. Heemink and A.J. Segers. Submitted to *Stochastic Environmental Research and Risc Assessment.*

between ozone concentrations, since the available measurements concern ozone too. A covariance model for ozone concentrations only is not enough to specify the total model error, however. Ozone on its own has only a limited life time in the troposphere due to the strong deposition. Transport of ozone is therefore not able to explain smog episodes; also chemical production has to be taken into account. A useful covariance model should therefore include covariances between ozone and its precursors NO_x and VOC. Measurements of precursor components are however sparse (VOC) and/or hardly comparable with the LOTOS state (NO_x), such that covariances between these components are hard to estimate. The complex chemical relations with large spatial and temporal differences gives rise to doubt about the existence of a simple covariance function anyway.

Instead of using a stochastic model based on correlations between measurements, a stochastic model could be based on correlations in the model output too. Pham et al. (1998) applied this idea by using empiric orthogonal functions (EOF's) to describe the initial error for an oceanographic model; see also section 6.5.2 about the SEEK filter. The EOF's are obtained from long time series of model states. A similar approach based on principal oscillation patterns (POP's) was used in (*Hasselmann, 1988*). In both methods, spatial correlations are expressed in a limited number of modes, obtained from time series of model states, with optional temporal varying weights. This approach has the advantage that the user does not need to specify the correlations explicitly; instead, they have been defined implicitly in the model equations. The strong varying chemistry in LOTOS, with different regimes over small distances and time scales, complicates expression of all correlations in a small number of modes.

Instead of prescribing the covariance between all elements of the state, the stochastic model could be limited to a small part of the state only, if uncertainties in these elements are the major source of model errors. An example is the value of boundary conditions in transport problems. For water level prediction in the North Sea, (*Verlaan and Heemink, 1997*) defined a stochastic model for the error in the open boundary at the Atlantic Ocean. The error at the boundaries is propagated by the model through the complete domain, automatically building a time dependend covariance. The same approach could be used for specification of errors in other model parameters, not necessary included in the state. If the boundary values are set according to an input data set, for example obtained from a course resolution model, the stochastic model is in fact defined for the input data, which happens to be coupled directly with state elements. Without loose of generality, the stochastic model could be limited to other parameters than the boundary model too, as long as there is some relation with the state elements.

For an air pollution model, a natural choice for a stochastic model based on errors in parameters is to define uncertainties in emissions. Except that emissions are the driving force behind pollution events, they are also highly uncertain on spatial and temporal scale. Besides, policy makers show large interest in estimation of quantitative emissions strengths, in order to know who to blame for certain pollution events. Stochastic models based on uncertain emissions are therefore common use in air pollution applications. Stochastic varying emissions were for example used in (*Zhang et al., 1999*) for estimation of the methane budget emitted from Europe using a Kalman smoother. In a 4D-var context, stochastic emissions were used in (*Elbern et al., 2000*) for assimilation of measurements in a high resolution air quality model, and in (*Houweling, 2000*) for estimation of the methane budget from rice fields and wetlands. Since emissions are often injected in the bottom layers of these models, the stochastic model is in some sense still applied to the boundary conditions. The same would hold for the deposition model, if defined stochastic; deposition is just an outward flux through the bottom of the model. Other uncertain parameters are however clearly distinct from the elements in the state vector, such as photolysis and reaction rates.

With a stochastic model based on the definition of uncertain parameters, an assimilation procedure is able to produce an optimal estimate of the parameters given the measurements. In the Kalman filter context, parameter estimation is performed by simply augmenting the state vector with the uncertain parameters, leading to a Kalman smoother instead of a Kalman filter (see §3.6). Investigation of time series of estimated parameters might point to structural biases in the underlying system, for example an underestimation of certain emission categories. The results should be interpreted carefully, since the stochastic model might not cover all existing errors, and the uncertain parameters are blamed for errors which are not their fault. The opposite is true for a stochastic model based on a covariance function of the state, which is able to cover all errors in the model but hardly provides any information about their origin. For assimilation during online forecast this is no problem, since a proper initial state is more important for a short range forecast than insight in model dynamics. Analysis of time series of initial states might point to biases in the model on long term, leading to a better understanding of dynamics, but this is not a primary target. However, a stochastic model with uncertain parameters provides insight in the dynamics immediately. If this leads to a better forecast skill, this gives additional thrust in the result.

Since online forecast of ozone concentrations is no topic in this research, while model improvement is, the stochastic model for LOTOS will be based on uncertain parameters. A description of the LOTOS version used in this study is given in §4.2. Three groups of uncertain parameters are considered: for emissions, photolysis rates, and deposition. Stochastic models based on these groups of parameters or on combinations of them are examined during filter experiments with simulated data (§4.3). Finally in §4.4, the Kalman filter is applied in combination with ozone data from a measurement network.

4.2 Experimental setup

The LOTOS model used in this study is a spatial limited version of the model described in chapter 2. The model domains was limited to an area of 12 × 12 grid cells covering England and Wales (figure 4.1). This area was selected for its rather isolated location, which ensures that most air pollution phenomena arise due to local conditions. The bulk of the NO_x and VOC load are emitted from local sources; only long periods of eastern wind lead to a substantial inflow of pollutants from the continent. Another important property of the selected area is that a large number of ozone measurements from rural sites is available (p. 191).

A 6 day period from august 5 till august 10 1997 was selected as a test period. In the days before, a strong western wind has filled the area with clean air from the Atlantic; a high cloud cover has limited the production of ozone. From august 5, a period with advection from the east and/or the south occurred; clear sky conditions lead to high ozone levels during the last three days of the period. The initial state was taken from a three week model simulation using the maximum domain.

The purpose of this research is to test the impact of several uncertain parameters. Therefore, the stochastic model takes the general form (3.40) for variations in model parameters, smoothed up to lag 1:

$$\begin{pmatrix} \mathbf{c}[k+1] \\ \boldsymbol{\lambda}_{k+1}[k+1] \\ \boldsymbol{\lambda}_{k}\;\;[k+1] \end{pmatrix} = \begin{pmatrix} \mathcal{L}(\mathbf{c}[k], \boldsymbol{\lambda}_k[k], t[k]) \\ \alpha\,\boldsymbol{\lambda}_k[k] \\ \boldsymbol{\lambda}_k[k] \end{pmatrix} + \begin{pmatrix} \mathbf{O} \\ \sigma_\alpha \mathbf{I} \\ \mathbf{O} \end{pmatrix} \mathbf{w}[k] \tag{4.1}$$

$$\text{or} \quad \mathbf{x}[k+1] = \mathbf{M}(\,\mathbf{x}[k], t[k]\,) + \mathbf{G}\,\mathbf{w}[k]$$

In here, $\mathbf{c}$ denotes the concentration array, and $\mathcal{L}$ the LOTOS model described in §2.4. The white noise input $\mathbf{w}$ forces a colored noise process $\boldsymbol{\lambda}$, which operates on parameters in the model. The exact size of $\boldsymbol{\lambda}$ depends on which parameters in the model are considered to be stochastic. For each element of $\boldsymbol{\lambda}$, different values for time correlation parameter $\alpha = \exp(-1/\tau)$ and standard deviation $\sigma_\alpha = \sigma\sqrt{1-\alpha^2}$ might be used. Augmentation of the state with $\boldsymbol{\lambda}_k[k]$ and $\boldsymbol{\lambda}_{k-1}[k]$ ensures that the value of the parameters used during assimilation could be estimated. The model state is to be compared with observations $\mathbf{y}^o$ according to the equation:

$$\mathbf{y}^o[k] = \mathbf{H}'\mathbf{x}[k] + \mathbf{v}[k] \tag{4.2}$$

The observation operator $\mathbf{H}'$ assigns the ozone level in the surface layer of a grid cell to an observation; $\mathbf{v}$ is the representation error. The output of the Kalman filter is a mean and

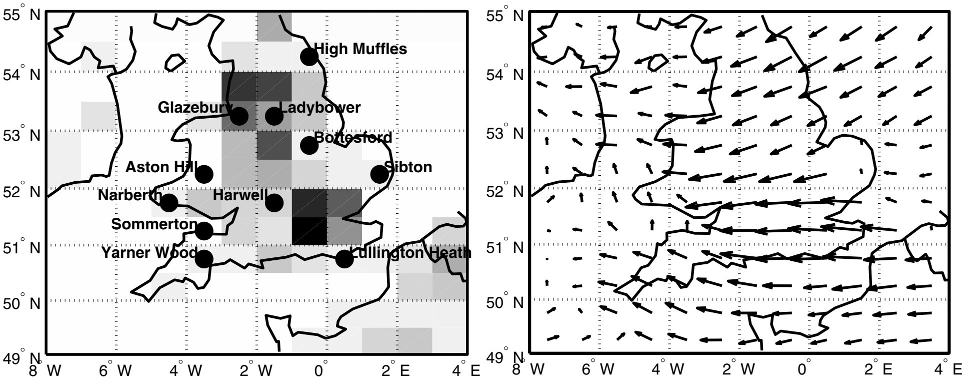


Figure 4.1: *Domain of the test region and location of the measurement sites (left). The pattern in the background shows the spatial distribution of emissions. The panel on the right shows the initial wind field for august 5, 1997, 0:00.*

covariance of the true state, given the previous observations:

$$\hat{\mathbf{x}}[k] \;=\; \mathrm{E}\left[\, \mathbf{x}^t[k] \mid \mathbf{y}^o[k], \mathbf{y}^o[k-1], \ldots \,\right] \tag{4.3a}$$

$$\mathbf{P}[k] \;=\; \mathrm{E}\left[\, (\mathbf{x}^t[k] - \hat{\mathbf{x}}[k])(\mathbf{x}^t[k] - \hat{\mathbf{x}}[k])' \mid \mathbf{y}^o[k], \mathbf{y}^o[k-1], \ldots \,\right] \tag{4.3b}$$

To deal with the expected nonlinear character of the general model **M**, all experiments are performed with an ensemble filter (see chapter 6 for a description). This type of filter is able to provide the correct solution to the filter problem up to any desired accuracy. The ensemble filter requires a number of $\mathcal{O}\left(10^2\right)$ independent evaluations of the model **M**, which is usually very expensive. For our experiments with small domain and short time period, the computation time is no constraint, however. Less expensive filters will be considered when a suitable definition of the stochastic model is found.

4.3 Assimilation with simulated data

This section describes the setup and result of filter experiments with simulated data. Measurements are drawn from simulations with a model in which certain model parameters contain random errors; if the filter is able to reconstruct these errors, it is in theory possible to detect the same kind of errors in the model given data from an observation network. The uncertain parameters considered here are emissions, photolysis rates, and deposition.

4.3.1 Uncertainties in emissions

Of all emissions present in the LOTOS model, the emissions of NO_x, VOC, and CO are the most important for the formation of summer smog. Sulfur oxides are related to winter smog, while methane is related to the background chemistry; uncertain emissions of these components are therefore not considered here.

NO_x emissions In a first experiment, the emissions of NO_x have been modelled uncertain. With $\bar{e}_{NO_x}$ the deterministic value of the NO_x emission in a grid cell, the stochastic emissions are modelled according to:

$$e_{NO_x}[k] \;=\; \max(\,0\,,\; \bar{e}_{NO_x}[k](1 \,+\, \lambda_{NO_x}[k])) \tag{4.4}$$

The standard deviation σ of λ is set to 30% and the time correlation parameter τ to 24 hours. All emissions are varied with the same factor; the size of the noise input w in (4.1) is therefore equal to one.

Figure 4.2 shows the variations in NO, NO_2, and O_3 due to this uncertainty, simulated for site Ladybower. Variations in NO during day time show a strong correlation with variations in the other two components; during the night, NO concentrations vanish immediately due to reaction with ozone. Since almost all NO_x emissions concern NO, variations in these emissions during the night can not be detected directly, but only through impact on other components such as NO_2 and ozone. Note the smoothing impact of the chemistry; where NO and NO_2 sometimes show rapid fluctuations, the variations in ozone are smooth.

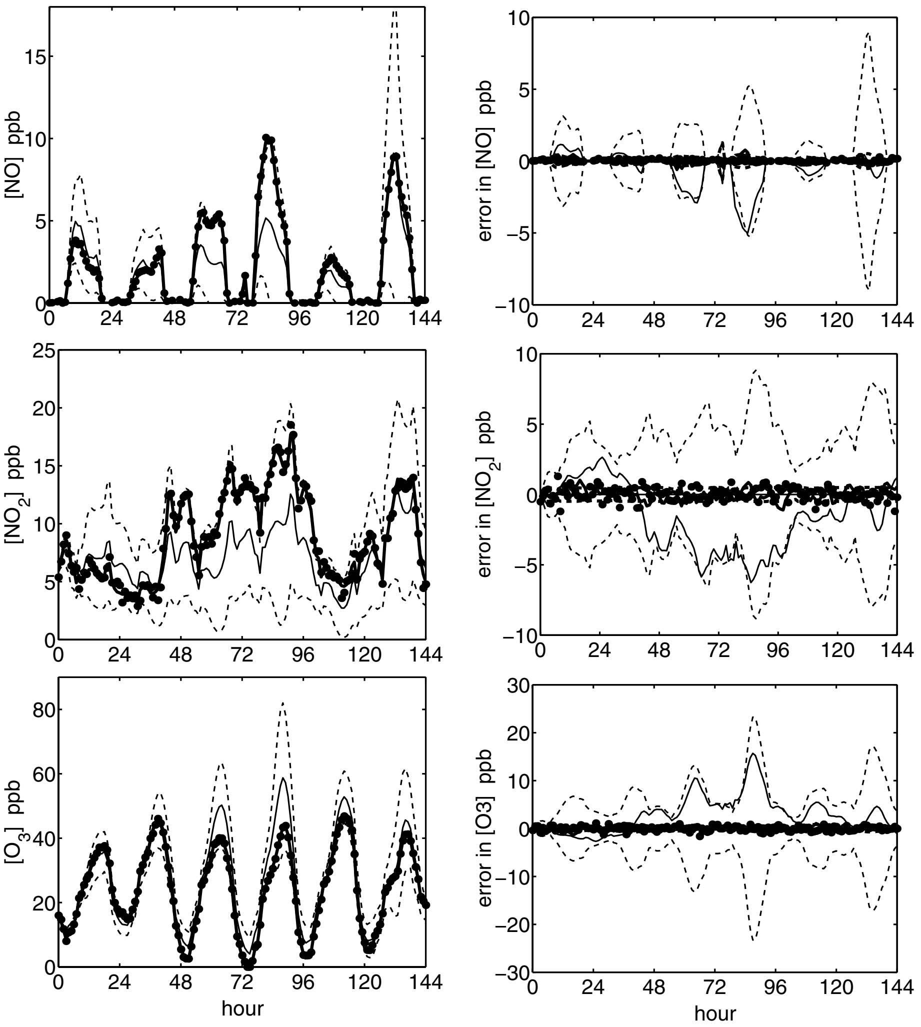


Figure 4.2: Selected results of assimilation experiment with uncertain NO *emissions in site Ladybower. Figures on the left show ground concentrations of* NO_2, NO, *and* O_3*: first-guess plus and minus one sigma (thin), simulated truth (thick), simulated measurements (dots); assimilation falls together with truth and is therefore omitted. Figures on the right show corresponding errors between model or filter and 'truth': first-guess and one-sigma bounds (thin), and similar after assimilation of the simulated* NO_2 *measurements (thick).*

For assimilation, a random 'truth' is generated with a model using random disturbed NO_x emissions (solid lines in figure 4.2). From the 'truth' run, measurements of NO_2 are simulated for sites Narberth and Ladybower, including a measurement error of 0.5 ppb. The filter is able to follow the NO_2 measurements perfectly. Errorbounds are decreased from 5 ppb before assimilation to about 1 ppb afterwards. Similar results are obtained for NO and O_3, indicating that the filter is able to determine the correlation between these components and NO_2. Due to the rather simple setup of the stochastic model (emissions disturbed with the same factor everywhere), the results are similar for other locations on the grid. Decreasing the errors in Ladybower and Narberth immediately leads to a similar decrease in other grid cells.

If the experiment is repeated using measurements of O_3 instead of NO_2, similar results are obtained. Errors after assimilation are slightly larger since the model errors are observed less direct. If the NO load is the only uncertainty in the model, assimilation of ozone measurements is therefore suitable to reconstruct the correct NO level.

VOC emissions In a second experiment, the emissions of VOC have been modelled as uncertain. A complication involved with uncertain VOC emissions is that measurements of hydrocarbons are much more seldom than for example NO_x measurements. Harwell is the only rural station in the domain where hydrocarbons are measured; other stations are located in city centers. Besides, it is difficult to relate measurements of hydrocarbons with the CBM-IV components in the LOTOS state. CBM-IV expresses hydrocarbons in reactive groups rather than molecules (see appendix A). Of the 24 organic compounds measured in Harwell, only ethene could be mapped directly to CBM-IV and vice versa. In addition, the aromatic components TOL and XYL could be compared with summed measurements of toluene and ethylbenzene, respectively m+p-xylene and o-xylene. Another strategy could be to divide the measured components into their reactive groups, and add these together. The obtained CBM-IV 'concentrations' are a minimum for the real values, since the measurements probably do not cover all hydrocarbons present in the atmosphere. Assimilation of hydrocarbon measurements is therefore no serious option, except for eventually ethene. Instead, the impact of VOC on other components (O_3 and NO_x) is considered.

The VOC emissions have been modelled stochastic similar as in eq. (4.4) for the NO_x emissions (standard deviation of 30%, decorrelation period $\tau = 24$ hour). Measurements of O_3 have been generated from a model run with random noise input ('truth'), including an error of 0.5 ppb. Figure 4.3 shows the result for the site Harwell. The total concentration of organic compounds is expressed in the number of carbon atoms per billion (ppbC), by adding concentrations of all carbon bonds in the state, weighted by the number of carbon atoms (table A.1). The total carbon load shows a sharp diurnal cycle. High emissions during the day lead to a rise of the carbon load; after sunset, organic compounds are lost to higher model layers. Emissions during the night are not able to compensate for the upwards flux, although these are injected into a thin and stable layer. The error in VOC concentrations due to noisy emissions therefore decreases during the night. The uncertain VOC emissions lead to uncertain ozone concentrations, but only during daytime; presence of sunlight is essential for the transformation of NO into NO_2 after degradation of VOC.

Provided with measurements of ozone, the filter is able to reduce the error in the VOC load during the day from $\sigma = 20$ to 10 ppbC. The largest decrease of the error takes place during sunrise, when the impact of the uncertain VOC load is noticed in the ozone con-

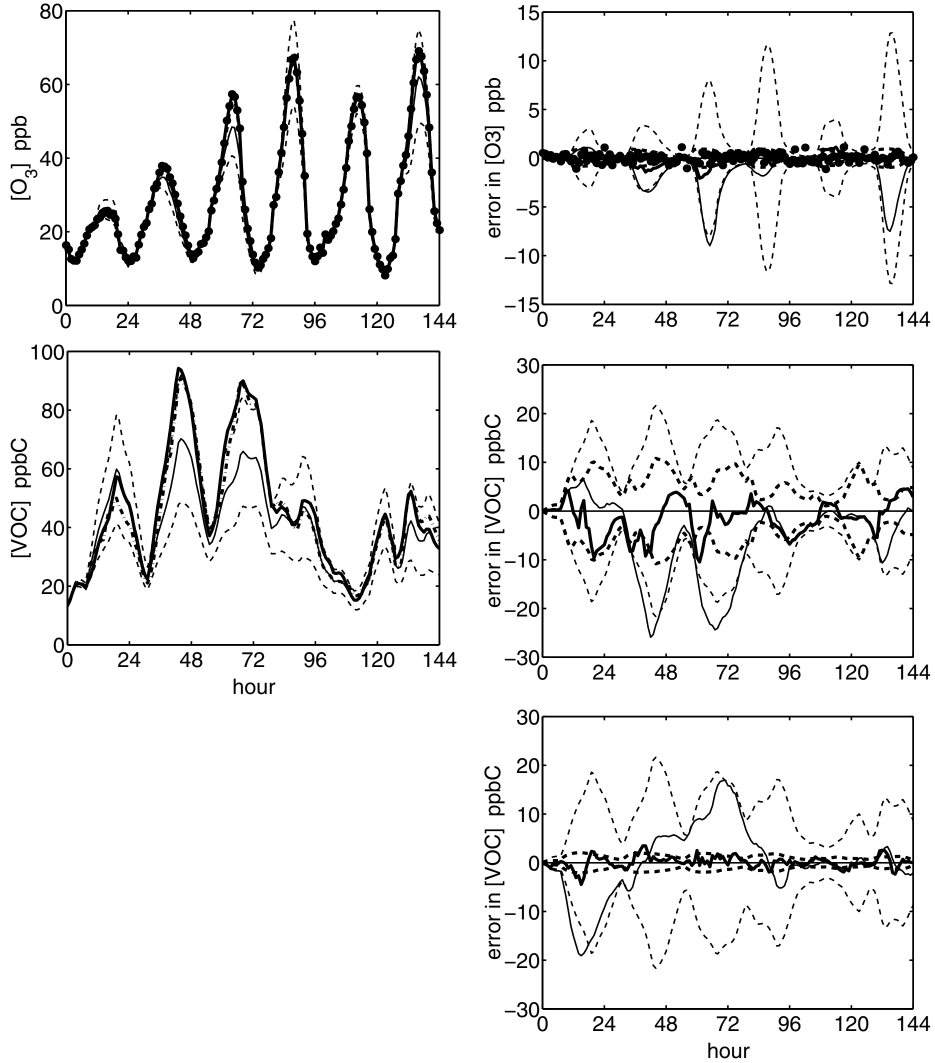


Figure 4.3: *Selected results for assimilation experiment with uncertain* VOC *emissions (time series for Harwell).*
Figures on the left show ground concentrations of VOC *and* O_3*: first-guess plus and minus one sigma (thin), simulated truth (thick, solid), simulated measurements of* O_3 *(dots), and assimilated mean (thick, dash/dot). Figures on the right show corresponding errors between model or filter and 'truth': first-guess and one-sigma bounds (thin), and similar after assimilation of ozone measurements (thick). The figure on the lower right shows the error in* VOC *concentration for a similar experiment but in addition assimilation of ethene concentrations too.*

centrations. Assimilation of ozone measurements only is not able to reconstruct the VOC load completely however, which was possible for the NO_x load when NO emissions were considered uncertain. Ozone and VOC's are not as tightly coupled as ozone and NO_x, and without additional information about the VOC composition, a better reconstruction is not possible. The lower right panel of figure 4.3 shows how the error in the VOC load might be reduced if measurements of ethene are assimilated (similar experimental setup, with a new random truth to simulate the measurements). The simple setup of the stochastic model ensures that errors in concentrations of carbon bonds are strongly correlated, and assimilation of ethene measurements leads to improved estimates of all other carbon bonds as well.

Emissions of NO_x, VOC, and CO During a third experiment, noise was included in the emissions of NO_x ($\sigma = 30\%$) and VOC (50%), and in addition, to the emissions of CO (30%). The uncertainty in VOC was increased since these emissions are in general believed to be more uncertain than that of the other components. The noise input $\mathbf{w}$ for the stochastic model is a three element vector now. The filter assimilated ozone measurements extracted from a random disturbed 'truth' for sites Harwell and Glazbury, including a random error with $\sigma = 1$ ppb. These sites are near the strongest emission sources, and changes in emission rates will be visible in the ozone measurements on a short time scale.

Figure 4.4 shows the simulated factors $1+\lambda_{nox}$, $1+\lambda_{voc}$, and $1+\lambda_{co}$ actually applied to the emissions, as well as their one-sigma bounds estimated with the lag-1 smoother. Due to the tight connection between ozone and NO_x, the filter is able to estimate the factors applied to the NO_x emissions almost perfectly. A small time lag is visible between the occurance of peaks in the truth-run and the estimation made by the filter. The filter is not able to detect emission changes immediately due to the delay between release from the source and change in ozone at the measurement site. The best estimations of λ_{nox} are obtained during the night, when the NO_x emissions are the only emissions which have a direct impact on the ozone concentrations. The standard deviation of λ_{nox} decreases from the initial 30% to less than 10%. During the day, the estimation of λ_{nox} is less accurate (up to 20%), since the ozone concentrations in the measurement sites are now also influenced by uncertain VOC concentrations.

The estimates of λ_{voc} show an opposite behavior. The most accurate estimations are made during the day (25–35%), while the nighttime emissions remain uncertain with 40-50%. Since the bulk of the emissions are released during daytime, an inaccurate estimation of the nighttime emissions is however of less importance. That sunlight is required for estimation of the VOC emissions is best seen for the first day. During the night, the emission factor is estimated with zero mean and increasing standard deviation; after the first significant photolysis, the estimation shifts towards the 'true' value immediately.

The lower panel in figure 4.4 shows that the impact of CO emissions on the measured ozone concentrations is minor in comparison with the impact of NO_x and VOC. The estimates of λ_{co} remain uncertain with about 30%, and is not decreased by the assimilation.

4.3.2 Uncertain photolysis rates

Since the chemistry is the driving force in an air pollution model, errors in chemical parameters will have a large impact on computed concentrations. An example is the value of

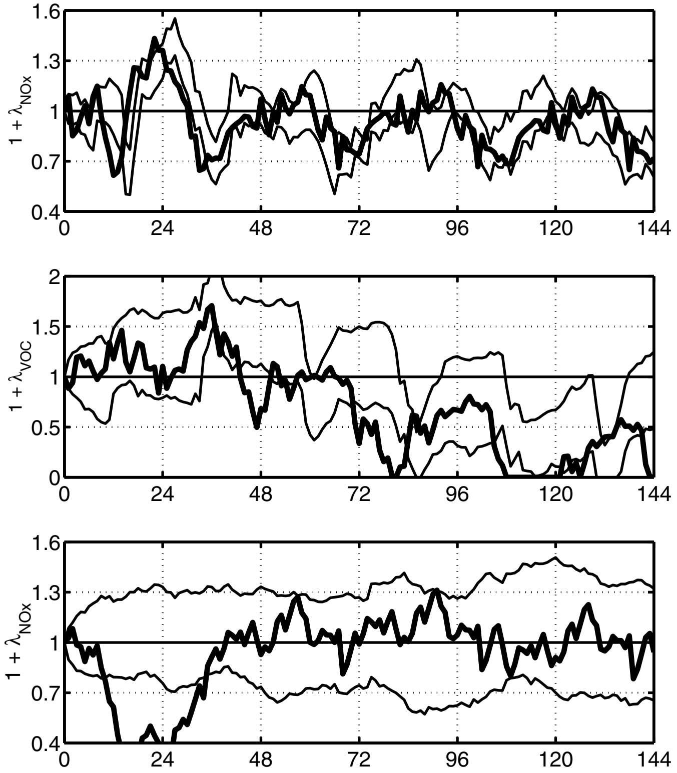


Figure 4.4: *Stochastic factors applied to total emissions of* NO_x, VOC, *and* CO, *during assimilation experiment with simulated ozone measurements: simulated truth (thick), and mean plus or minus one sigma after assimilation (thin).*

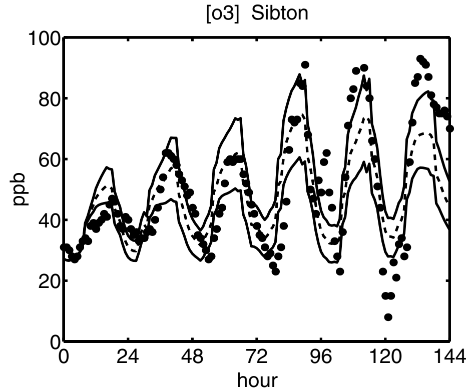


Figure 4.5: *Impact of uncertain photolysis rates in Sibton: deterministic model (dashed), measurements from observation network (dots), and* 2σ *bounds from filter without assimilation (solid). The specified uncertainty in the photolysis rates of* NO_2 *and* O_3 *is able to explain a large part of the difference between model and measurements, except for irregular high ozone levels during night 4-5, and the low values during night 5-6.*

photolysis rates. (*Thompson and Stewart, 1991*) studied the effect of uncertain photodissociation and reaction rates in a tropospheric model using a Monte Carlo method. Due to the uncertainties, ozone concentration were found to have standard deviations up to 16% for urban and clean continental conditions (mid latitude). The primary photo dissociations of NO_2 into NO and O_3, and of O_3 into O_2 and $O(^1D)$ were found to have the largest impact on the chemical state in the troposphere. Concentrations of almost all important trace gases show strong correlations with the source or reaction products of these two photolysis reactions. In particular, both reactions have a large impact on ozone concentrations since they lead to ozone formation and destruction respectively. Other important reactions were found to be those between NO and O_3 (R3 in appendix A) and between OH and NO_2 (R20).

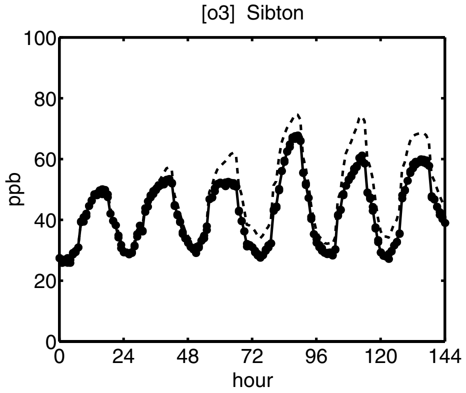
Uncertainties in photolysis and reaction rates arise due to unprecise knowledge of parameters such as absorption cross sections, quantum yields, solar fluxes and activation energy. A detailed description of the possible errors in these parameters is beyond the scope of this research; similar to (*Thompson and Stewart, 1991*), we will therefore assign overall standard deviations to the reaction rates, following a lognormal distribution. The list of uncertain reaction rates is limited to photolysis rates of NO_2 (R1) and O_3 (R8). According to table 2 in (*Thompson and Stewart, 1991*), a standard deviation of 30% should be assigned to J_{NO_2}. Reaction (R8) used is LOTOS is a combination of photolysis of ozone into O and $O(^1D)$ with standard deviations of 10% and 40% respectively; we will use an overall value of 30%. The photolysis rates used in LOTOS are now computed from:

$$J[s] \;=\; \overline{J[s]}\,\exp(\lambda_{J[s]})\,\phi_{sol}\,\phi_{cld} \qquad , \qquad s=\{NO_2,O_3\} \tag{4.5}$$

where $\overline{J[s]}$ denotes the deterministic value, ϕ_{sol} and ϕ_{cld} denote correction factors for solar angle and cloud cover, and λ denotes a sample of a colored noise process with zero mean and standard deviations $\sigma = 0.3$. Similar as for uncertain emissions, a strong time correlation of $\alpha = \exp(-1/24)$ is assumed. The impact of the uncertain photolysis rates modelled in this way on the ozone concentrations is illustrated in figure 4.5 for site Sibton.

An advantage of the formulation of the stochastic model in terms of a lognormal distribution is that samples are always positive, such that truncation of unwanted (negative) values could be omitted. A disadvantage is that the nonlinearity of the state state space model (4.1)

Figure 4.6: *Ozone concentrations in Sibton during assimilation experiment with uncertain photolysis rates: deterministic model (dashed), measurements simulated with random disturbed model run (dots), and assimilated mean (solid).*



increases, which puts a larger demand on filter resources. Note that a large spatial uncertainty is related with the cloud cover; shape, thickness and the water vapor load all have impact on photolysis rates in the boundary layer and are hard to quantify. We will neglect the uncertainties in cloud cover, and consider uncertainties in overall photolysis rates only.

An assimilation experiment with simulated data has been performed with the uncertain reaction rates, similar to the experiments with uncertain emissions. A set of measurements was simulated for sites Yarner Wood and Sibton from a model run with random disturbed photolysis rates of NO_2 and O_3, including a measurement error of 0.5 ppb. Figure 4.6 shows resulting time series for the ozone concentrations in Sibton. The random generator produced photolysis rates of NO_2 and O_3 which are both on average smaller than the deterministic values. The net effect is a decreased ozone level since the photolysis of NO_2 is faster than that of ozone. The filter is able to follow the simulated measurements perfectly, which is no surprise since the error description is perfect.

It is more interesting to investigate whether the filter is able to distinguish between errors in ozone concentrations due to errors in J_{NO_2} and due to errors in J_{O_3}. Figure 4.7 shows the time series of the random disturbed photolysis rates and the estimates made by the filter, with and without assimilation of measurements. The results show that the filter is able to produce reliable estimates of both photolysis rates. The standard deviation of the estimated photolysis rates decreases from initial 30% to 10% for J_{NO_2} and to 15% for J_{O_3}. Smaller sigma bounds might be obtained if measurements from more sites are assimilated, although some uncertainty will remain due to the measurement errors. The more reliable estimate of J_{NO_2} shows that the measured ozone concentrations are more sensitive to changes in the photolysis of NO_2 than to changes in photolysis of ozone itself. From the almost equal 2σ bounds just after sunrise for filters with and without assimilation it is concluded that a substantial amount of sunlight is required, before errors in photolysis rates affect the measured ozone concentrations. The filter reacts with a strong adjustment of the estimated photolysis rate, leading to irregular shaped 2σ bounds during the morning.

Note that the filter does not estimate the rate coefficients directly, but rather the stochastic processes λ_J in (4.5), included in the state vector. While the photolysis rates vanish during the night, the stochastic processes do not, although they do not have any physical interpretation during the night. The statistics of λ_J therefore slowly return to their first guess values

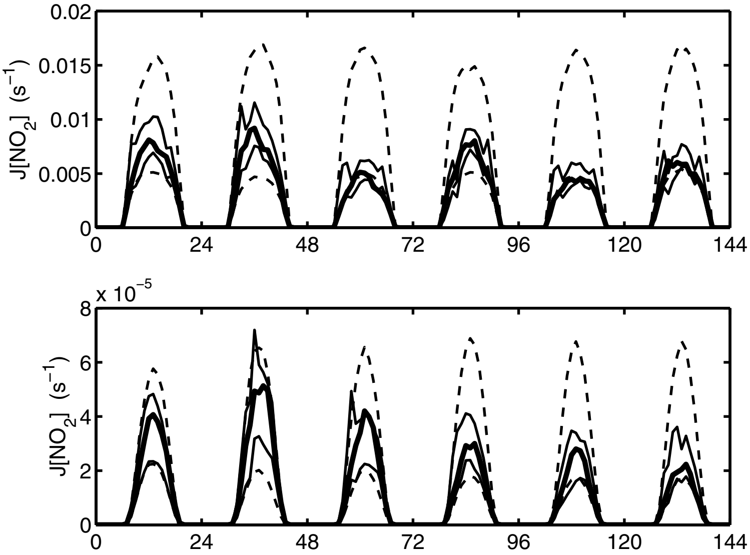


Figure 4.7: *Estimates of photolysis rates during assimilation experiment with simulated data: random disturbed photolysis rates (solid, thick), two-sigma bounds due to specified uncertainties (dashed), two-sigma bounds after assimilation of simulated measurements (solid). The shown photolysis rates include the adjustment for solar angle.*

(zero mean, standard deviation of 30%), such that the estimates of the photolysis rates become almost as uncertain as they used to be without assimilation. Only a small part of the information about the rate coefficients survives the night due to the strong time correlation included in the stochastic model. For application to ozone data from a network, a better solution would be to freeze the estimates of the driving stochastic processes during the night, or to start at sunrise with diurnal averages of the previous day.

An advantage of modeling photolysis rates as uncertain parameters is the rather large area where their impact is present. Changes in photolysis rates act on ozone production in the complete domain, while uncertain emissions only affect downwind area's. Besides, the photolysis also affects the concentrations in higher model layers. A combination of uncertain emissions and uncertain reaction rates is therefore a promising approach for an assimilation procedure. While the number of measurements sites under large impact of emissions is limited, almost all sites are influenced by photolysis.

4.3.3 Uncertainties in deposition parameters

Where chemical production is the main source of ozone, deposition is the main loss. Investigation of the impact of uncertain deposition parameters is therefore necessary to obtain a useful stochastic extension to the LOTOS model.

In the context of LOTOS, the deposition acts directly on the concentrations in the mixing layer following (2.15). In addition, the parameterization of the deposition is also used to form the deposition profile (2.13) from which ground level concentrations are computed. Errors in the deposition rates will therefore be visible in the model output immediately. For assimilation of ozone measurements, the deposition rate of ozone is a perfect source of uncertainty. Almost every difference between model and measurements might be corrected by choosing an appropriate amount of deposition or a lack of deposition. The only exception is the case of a large underestimation by the model even if the deposition is omitted.

To investigate the impact of uncertain deposition, four different deposition parameters have been considered stochastic: the atmospheric and viscous-sublayer resistance, which depends on the surface structure and surface wind:

$$R_t[k] \;=\; \overline{(R_a(z_0)\;+\;R_b[k])}(1+\lambda_{R_t}[k]) \tag{4.6}$$

and three surface resistances for components $s = \mathrm{O}_3$, NO, and NO_2:

$$R_{c,s}[k] \;=\; \overline{(R_{c,s})}(1+\lambda_{R_{c,s}}[k]) \tag{4.7}$$

The bars mark deterministic quantities, and each $\lambda[k]$ denotes a colored noise process with zero mean, standard deviation $\sigma = 0.3$ and time correlation parameter $\tau = 24$. The deposition parameters in (4.6) and (4.7) form the basic input for the deposition model, and are based on landcover and uptake of trace gases by vegetation. Since many of these parameters are quite unknown, the assumed standard deviation of 30% might be a conservative assumption. The parameters have been multiplied with the same factor for each grid cell covering land (deposition of ozone is almost zero over sea anyway).

To investigate the impact of uncertain deposition on concentrations in the model, the stochastic model has been propagated over the test period without assimilation of measurements. Due to the uncertain deposition, the ozone concentrations over land obtained

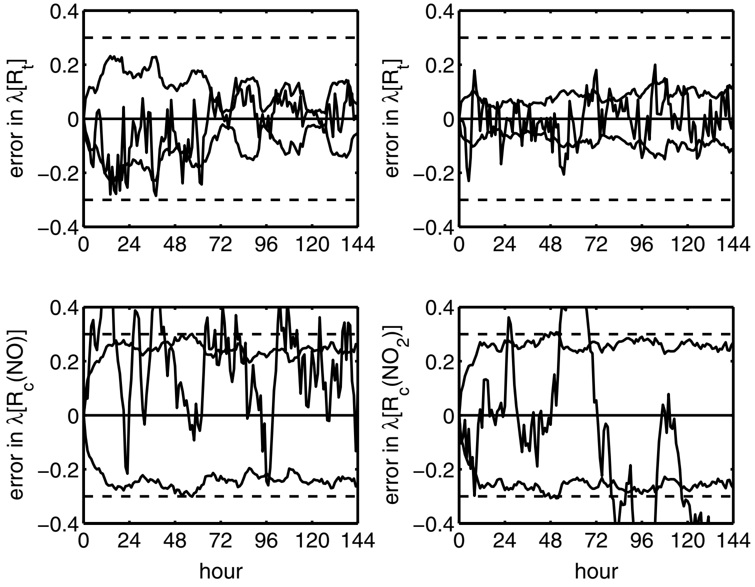


Figure 4.8: *Assimilation with uncertain deposition parameters and simulated measurements: actual errors in estimates of deposition parameters (filter mean minus simulated truth), and estimated one-sigma bounds.*

standard deviations of 5-7 ppb. The variations in ozone are rather constant in time, although a small diurnal cycle is visible. The variability increases during day time since deposition is proportional with the concentrations.

An assimilation experiment has been performed using simulated measurements in 4 sites, drawn from a random disturbed model run and a random measurement error of 0.5 ppb. Similar as for the experiments with uncertain emissions and photolysis rates, the assimilated ozone concentrations perfectly follow the (simulated) measurements. Figure 4.8 shows the errors in the estimates of the deposition parameters after assimilation. The filter is able to reduce the error bounds significantly for the atmospheric and viscous-sublayer resistance R_t and the surface resistance of ozone R_{c,O_3} from initial 0.3 to about 0.15 after assimilation. The value of 0.15 is in good agreement with the actual errors between assimilated mean and simulated truth. The estimates of R_t are most accurate during the night, up to a standard deviation of 0.10 . The later could be explained from the higher values of the atmospheric resistance R_a during the night (stable conditions); variations with a random factor such as used in (4.6) will then lead to a stronger response in the deposition. The filter is not able to provide accurate estimates of the surface resistances for NO and NO_2; the impact of these parameters on ozone concentrations is almost negligible in comparison with the other parameters.

Now that only the parameters R_t and R_{c,O_3} seem to be relevant in a filter procedure, the stochastic model might be simplified to a stochastic variation of the deposition velocity of ozone only:

$$v_{d,o3} = \frac{1}{R_{topo} + R_c(\mathrm{O}_3)}(1 + \lambda_{v_{d,o3}}) \tag{4.8}$$

An assimilation experiment with this choice for the stochastic deposition ($\sigma = 0.3, \tau = 24.0$,simulated measurements) showed that the filter is perfectly able to detect variations in a random disturbed $v_{d,o3}$.

4.3.4 Combination of emissions, photolysis rates, and deposition

The experiments with simulated data showed that the filter technique is able account for errors in emissions, photolysis rates, and deposition parameters, if one of these parameters contain stochastic variations. In practice, one would like to define stochastic variations in all these parameters at the same time, since they are likely to contribute to errors between model and measurements all together. Therefore, a filter experiment has been carried out with a combination of several types of stochastic parameters: emissions of $\mathrm{NO_x}$ and VOC (standard deviations of 30% and 50% respectively), photolysis rates of $\mathrm{NO_2}$ and $\mathrm{O_3}$ (30%), and the deposition velocity of ozone (30%). The stochastic models are implemented similar as described before, except that now the stochastic variations in the photolysis rates are frozen. Measurements have been simulated in 5 measurements sites from a random disturbed model, including a measurement error of 0.5 ppb. The chosen locations are either under direct impact of the emission sources (Harwell, Bottesford, and Glazebury) or more remote (Yarner Wood and Sibton).

Figure 4.9 shows the errors in the estimates of the stochastic processes λ driving the selected model parameters, as well as the 2σ bounds with and without assimilation. The error bounds are decreased for all selected parameters, indicating that assimilation of ground measurements is able to distinguish between the different errors given the available measurements. The standard deviations provided by the filter are reliable since the actual errors are almost everywhere within the 2σ bounds.

The results show that fluctuations in the deposition rate of ozone might be reconstructed very accurate, which is not surprising given the tight connection between deposition and the surface measurements. The filter produces estimates of $\lambda_{v_{d,o3}}$ with 2σ-bounds of $5-15\%$; the actual value of the parameter is within these bounds at almost every hour. The most accurate estimates are obtained during the night, when the deposition is almost the only process acting on ozone.

The value of the $\mathrm{NO_x}$ emissions is estimated up to $2\sigma = 30\%$; during the night, the estimate is more accurate due to the direct impact of NO on ozone. The reverse holds for the VOC emissions: estimation up to 50% during daytime (from initial 100%), but decreasing accuracy during the night. Note the rather large errors during the night between day 3 and 4, which are not discovered by the filter until next sun rise. The uncertainties after assimilation are still quite large, indicating that reconstruction of uncertain VOC from ground measurements of ozone is difficult. If the model is expected to be perfect except for the emissions, ozone measurements might be able to account for uncertainties in VOC emissions; other-

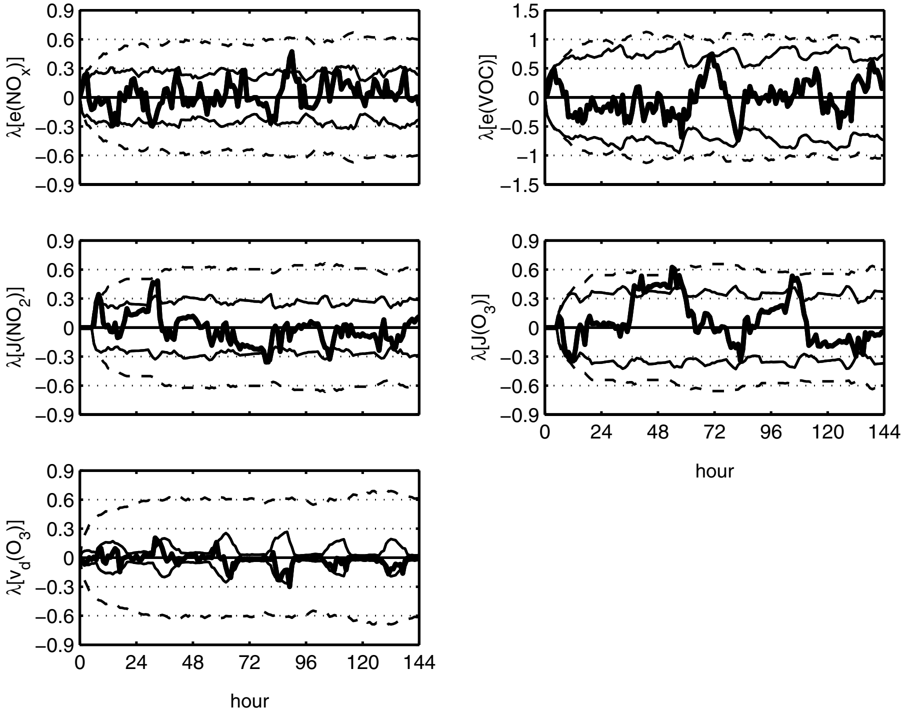


Figure 4.9: *Errors in stochastic factors during assimilation experiment with uncertain emissions, photolysis rates, and deposition velocity (simulated measurements). Figures show the errors in* λ *after assimilation (mean minus simulated truth; thick lines) as well as the* 2σ*-bounds (thin); dashed lines denote* 2σ*-bounds before assimilation.*

wise measurements of hydrocarbons are required to obtain a more accurate estimate of the VOC load.

For the photolysis rates, the most accurate estimate is obtained for $J[\mathrm{NO}_2]$ rather than for $J[O_3]$, from initial 60% to 30% to 40% respectively. Photolysis of NO_2 seems to have a larger impact on the ozone measurements than photolysis of ozone itself, which is explained from the lower rate of the later process. The most accurate results are obtained around noon, when the photolysis reach their highest rates. Note that although the stochastic processes driving the photolysis rates are frozen during the night, the small fluctuations in the errors suggest that the filter still adapts the photolysis factors. This effect is a result of the use of a finite ensemble in the filter technique, which leads to undesired correlations in the covariance matrix, for example between the values of λ used for the photolysis rates and the values used for the deposition. These correlations are hard to avoid (only with infinite ensemble size), and are acceptable since they do not influence the actual photolysis.

Concluding, a set of uncertain emissions, photolysis rates, and deposition velocity is useful for a stochastic model around LOTOS. Uncertainties in these parameters contribute to uncertainties in ozone concentrations in both polluted and more remote area's of the domain, and their values might be estimated from the assimilation of ozone measurements.

4.4 Assimilation with data from observation network

The experiments with simulated data showed that the filter technique is able to correct several types of model errors, if the errors are specified correctly. These experiments therefore only showed that assimilation of ozone data might be possible, if the chosen error specifications are indeed the main error sources. One might not expect that differences between model and data are decreased if they arise due to other errors. Therefore, before any ozone data is assimilated, a comparison between model and data has to be made to decide what might be achieved with the assimilation.

The LOTOS simulations have been compared with observations available for the selected domain and period (see p. 191). A first analysis of the time series shows that they are often in quite good agreement. During the first days of the selected period, the ozone concentrations are rather low, especially in the southern part of the domain. Investigation of the meteorological data shows this could be explained from a rather large cloud cover. Later on, the cloud cover decreases, and measured ozone levels start to show high peaks during the day, with maximum values of 100 ppb. The model simulates the occurance of lower concentrations in the beginning and the high peaks later on correctly, but is not able to reproduce the height of the peaks. Some of the peaks are underestimated with more than 30 ppb. Part of this misfit might be explained from the coarse resolution of the model, which tends to spread local high concentrations over a larger area. However, the height of the peaks are not reproduced for almost all cells at the same time, suggesting a systematic underestimation of the ozone production. Since the focus of the model is on simulation of smog episodes, it is at least this underestimation which should be compensated for by an assimilation procedure.

The variability in the measurements is large: the measured values sometimes differ with more than 10 ppb from one hour to another, while the long term mean over several hours is almost constant. One explanation is the occurance of measurement errors. Based on experience with calibration and experiments with identical measurement devices located near eachother, the overall measurement errors are estimated to be 5–10% of the measured value (*Tilmes and Zimmermann, 1998*). Another source of variability is the impact of local weather conditions on the ozone concentrations. For example, a single hour of sunshine in a period with clouded sky might lead to an occasional higher ozone level. Due to the large natural variability in the measured data, an assimilation procedure might not be expected to follow each single measurement. A standard deviation of the error between model and measurement of 5-10 ppb is acceptable.

During the selected period, the wind is directed to west or north west, apart from days four and five when the wind is directed to north or north east. The modeling of the eastern boundary, where polluted air from the continent flows into the domain, could therefore be an important source of errors. The simulated values in Lullington Heath and Sibton (located on the southern and eastern coast) are in good agreement with the measurements, however, which makes an error in the boundary values unlikely.

4.4.1 Uncertain NO_x and VOC emissions

In a first experiment, the total emissions of NO_x and VOC were considered to be uncertain, with standard deviations of 30% and 50% respectively, and a decorrelation period of $\tau = 24$h. Similar as for the experiments with simulated data, the emissions have been multiplied with single factors in the complete area covering England and Wales. The ozone measurements from Harwell, Bottesford, and Glazebury were used for assimilation, since the ozone levels at these sites are under direct influence of the largest emission sources. Site Ladybower could have been used instead of Bottesford and Glazebury, since it is actually surrounded by strong emitting gridcells. However, the site is located at a rather high altitude (420 m), and timeseries of the ozone observations show that during the night, the station observes the reservoir layer rather than the mixing layer. The daytime values measured at Ladybower show a strong correlation with the values measured at surrounding stations, while the nighttime values are fixed at a relative high value. NO_x emissions injected into the mixing layer during the night would not be visible in Ladybower, and therefore the site is only used for validation.

Figure 4.10 shows selected results of the assimilation experiments. The time series of the ozone concentrations in Glazebury (upper left panel) show that the filter is able to decrease the residue between model and measurements. The high ozone levels during daytime which were missed by the model are now reached within the assumed measurement error. Similar results were obtained for the other assimilated stations. The spatial distribution of the adjustment at day 4, 15:00, shows that the underestimation of ozone is corrected in a plume released from the largest emission sources (see distribution of emissions in figure 4.1). At the specific hour, increased ozone levels are correlated with decreased NO_x levels (especially less NO to destroy O_3), and increased VOC levels. The filter tends to decrease the NO_x levels during the complete period. Comparison with NO_x measurements from Ladybower show that this is in agreement with reality during daytime (figure 4.10, lower left panel). The nighttime values are decreased too far, however. The measurements of NO_x often show large fluctuations within a small period; therefore, the figure was plotted using

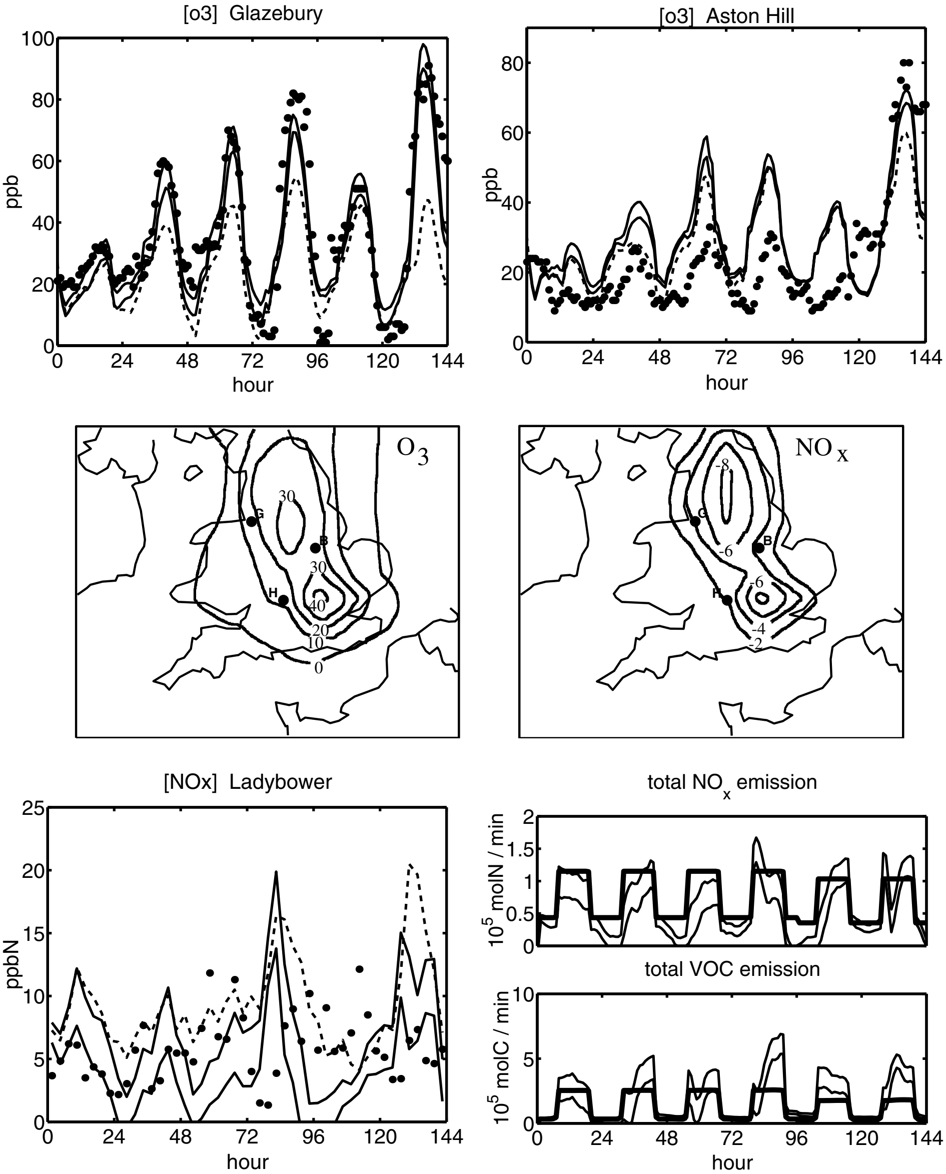


Figure 4.10: Selected results for assimilation experiment with data from an observation network, and stochastic model with uncertain emissions only. ***Upper***: *time series of ozone in Glazebury (assimilated) and Aston Hill (diagnosed): measurements (dots), model (dashed), assimilated mean*$\pm 2\sigma$ *(solid).* ***Middle***: *spatial adjustments (assimilated mean minus model) at day 4, 15:00.* ***Lower left***: *3 hour average of* $\mathrm{NO_x}$ *in Ladybower.* ***Lower right***: *estimates of total emission in England/Wales: deterministic values (thick), and* 2σ *bounds after assimilation (thin).*

three hour averages. If NO_x measurements are to be used during an assimilation procedure, a large representation error should be assigned.

Outside the plume, the adjustments are minor, although the local emissions are adjusted with the same values for all grid cells. The ozone concentrations in Aston Hill for example are hardly influenced by the assimilation during the first five days (figure 4.10, upper right panel). Thick cloud cover limited ozone production during the first days, and a wind blowing from the south prevents the inflow of emissions later on. During day 6 however, the sky was clear and the wind directed from the east, and the filter produced correct ozone concentrations due to the assimilation in Harwell.

Similar as for the experiments with simulated data, an estimation of the emissions has been extracted from the λ's in the state. The lower right panel in figure 4.10 shows the total amount of NO_x and VOC emitted in the area England/Wales. Due to the modeling with time profiles, the default emissions (E) show a periodic and blockshaped pattern of high or low emission rates following day/nighttime and week/weekend (day 5/6). The values used in the filter are equal to $E(1+\lambda)$. The results in figure 4.10 show that the adjustments made to the NO_x emissions are rather small. During nighttime, the emissions are decreased to obtain higher ozone values in the assimilated sites. During daytime, the emissions start at a lower level following the trend from the night before, to increase to a peak emission just before sunset. Note the behavior in the morning of day six: the nighttime emissions have been slightly increased, which causes a large additional emission when the emission profile changes to daytime rate. This higher rate is soon regarded as a mistake, and the emission rate returns to a lower level. If the peak is regarded as a mistake, the emissions follow a rather smoothed profile on day six, which shows much resemblance with the profiles on day 2, 3, and 5. The total amount of emitted NO_x has hardly changed; from default 7.5 to 6.0–8.6 10^4 ppbN/min after assimilation.

The average VOC emissions used by the filter are overall higher. Increased emissions during daytime are used to increase the ozone level. As a consequence, this leads to increased emissions during the night too, since the emission changes are strongly correlated in time. Without the time correlation, the night time emissions would probably be left unchanged, and should be followed by even larger adjustments during the day. The adjustments are already rather strong; although the emissions were given a large degree of freedom (standard deviation of 50%), the filter sometimes used emissions up to 3 times the default value. The average emissions change from default 1.3 to 1.8–2.7 10^5 ppbC/min after assimilation. One should not conclude from this result that *the* VOC emissions should be increased with say 150%, since the chosen stochastic model does not include spatial variations nor variations in VOC composition. In theory, the emission model could be complete correct apart from the modeling of one single, but for ozone production very important component which requires an additional release, while all other emissions are correct.

Concluding, the filter approach is able to improve ozone simulations given uncertainties in the emissions of NO_x and VOC. Including the correction factors for the emissions in the state provides useful insight in how the filters corrects differences between model and measurements. The values of the actual emissions estimated in this way should be interpreted carefully however. A careful conclusion would be that at least the timeprofiles used for the

NO_x emissions are subject to uncertainties, and that VOC emissions are under estimated.

4.4.2 Uncertain emissions, photolysis, and deposition

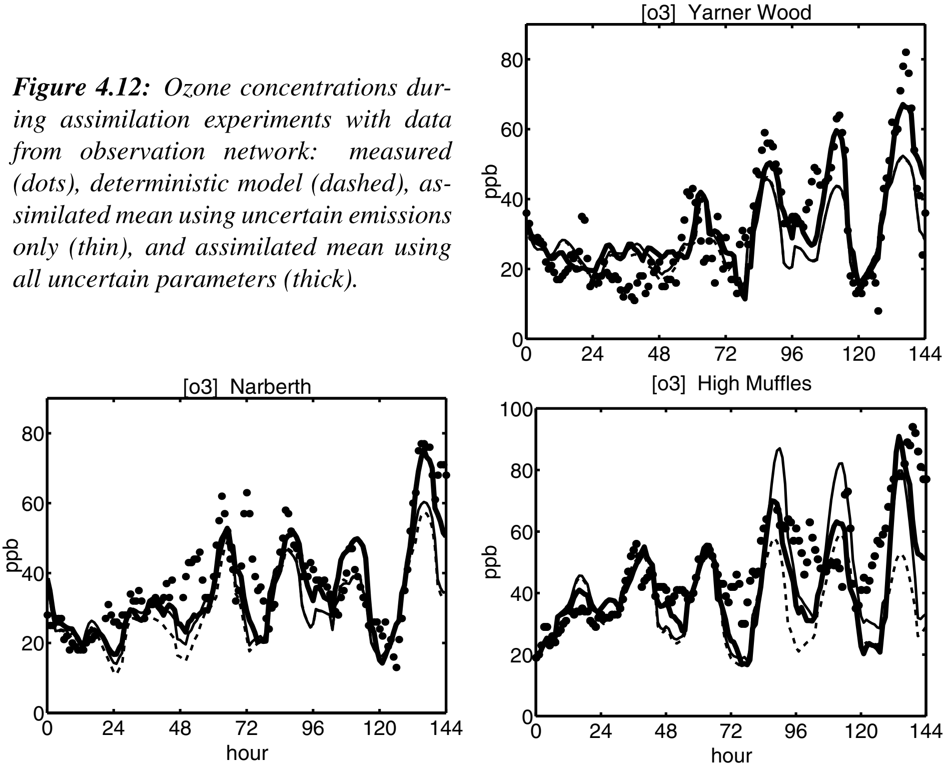
In a second assimilation experiment with ozone data, the stochastic model was extended with uncertainties for photolysis rates of NO_2 and O_3, and the deposition velocity of ozone. Since these parameters also affect ozone concentrations in locations outside the emission plume, the list of assimilated sites was extended with Yarner Wood and Sibton.

Figure 4.12 shows the ozone time series in Yarner Wood (assimilated) and Narberth and High Muffles (diagnosed). Where assimilation with uncertain emissions only did hardly affect the ozone concentrations in these sites, including photolysis rates and deposition improves the results significantly. The high ozone peak at day 6 is computed by the filter for all sites, although only the measurements from Yarner Wood are assimilated. Similar, the underestimation during the night between days 4 and 5 is corrected. These results suggest that a part of the difference between model and measurements in the three sites could be explained from the same model errors in photolysis or deposition. Note the peak in the measurements in Narberth around hour 72; the only explanation for this peak during the night is an unmodeled inflow of high ozone concentration from the sea (if an error in the measurement equipment is omitted). Similar situations occurred in Sommerton during a few nights. Since these small scale effects are not covered by the stochastic model, the assimilation procedure is not able to improve the results here.

On first sight, the results for the sites around the emission sources are hardly different from the experiment with uncertainties in the emissions only, except for some small scale improvements during the night. The high peaks underestimated by the model are still corrected by the filter. The results for High Muffles (figure 4.12, lower right) show that a part of the differences around the source areas should be explained from uncertainties other than the emissions, however. The adjustment of emissions only leads to an overestimation during day 3; with the extended stochastic model however, the computed ozone is in much better agreement with the measurements. The night time concentrations computed for High Muffles remain as terrible as they used to be, since the model is not able to produce accurate simulations here anyway.

Figure 4.13 shows the deterministic values of the selected parameters, as well as their estimates after assimilation. From the differences between estimation with and without uncertain photolysis rates/deposition is is possible to identify how other parameters than emissions become blamed for differences between model and measurements. For example, the very low NO_x emissions during the fourth night used in the emission-only filter are replaced by decreased deposition. Both settings lead to higher ozone levels, but the later choice should be trusted more since it is in agreement with a larger set of measurements. Similar, the increase of VOC emissions during the last three days is replaced by an increased photolysis of NO_2 and a decreased photolysis and deposition of ozone, both leading to higher ozone levels. Extension of the stochastic model with photolysis rates and deposition is therefore necessary to explain differences between model and measurements.

Figure 4.12: *Ozone concentrations during assimilation experiments with data from observation network: measured (dots), deterministic model (dashed), assimilated mean using uncertain emissions only (thin), and assimilated mean using all uncertain parameters (thick).*



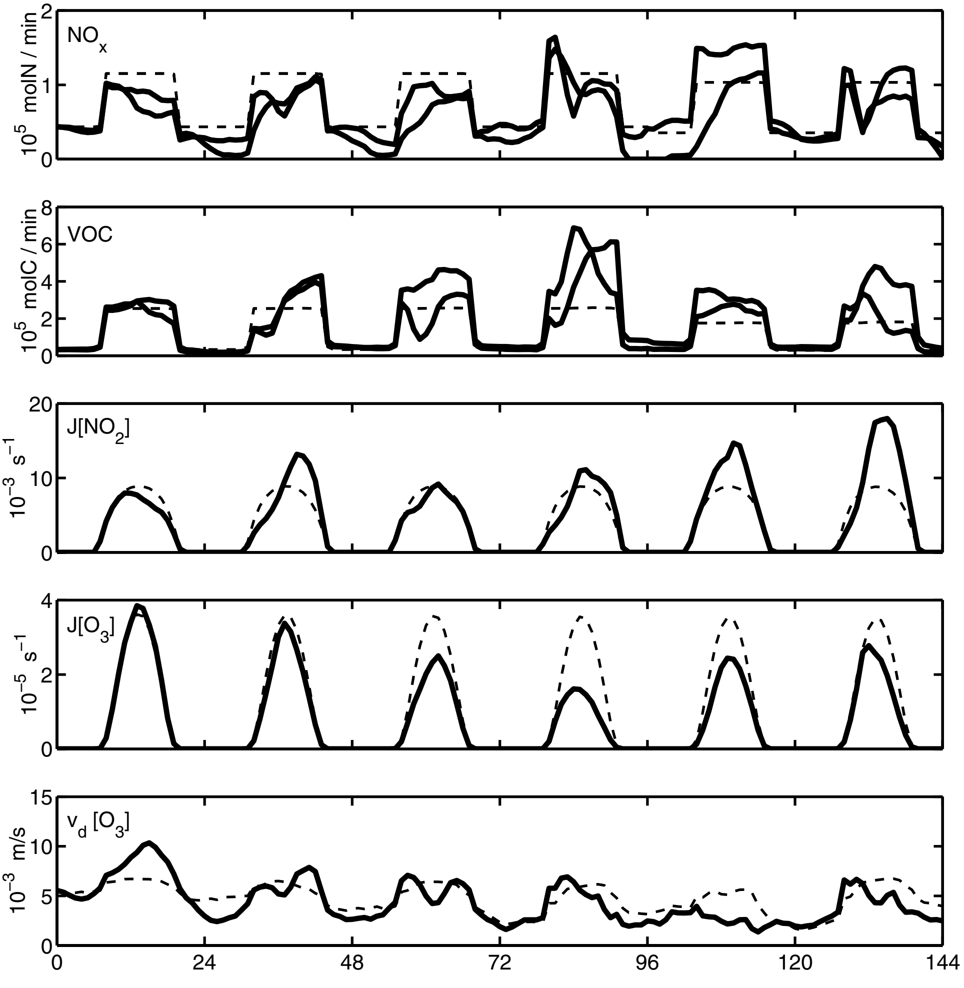


Figure 4.13: *Values of stochastic parameters during assimilation experiment with data from observation network: deterministic (dashed), assimilated mean using stochastic model with uncertain emissions only (thin), and similar with all uncertain parameters (thick).*

4.4.3 Forecast of ozone level

In a third experiment, the filter technique was tested for its value in an ozone forecast system. The previous described experiments showed that the mean state obtained with the filter is a more accurate approximation of the true state than that computed with a deterministic model run. If a model run is started with an assimilated mean as initial state, it is therefore expected to be in better agreement with the measurements. This property could be used in an online forecast system. The filter should provide an optimal initial condition for a deterministic forecast run; if later on new measurements have become available, the filter continues the assimilation and provides a new optimal initial state.

Apart from the initial state, the quality of an ozone forecast also depends on the quality of the meteorological input. The forecast skill of meteorological data is limited to 3-5 days, and one could therefore not expect an ozone forecast to be accurate over more than a few days. A forecast over one or two days is in practice suitable, since ozone forecast are usually provided in order to warn the public for unhealthy conditions during the coming day. The offline experiments described here used analyzed meteorological data, which hardly differs from forecast data during the first few days but has a higher quality later on.

The forecast skill of the filter has been examined for the state at day 4, 15:00, when the ozone concentrations have reached their maximum levels. During two forecast runs, the stochastic model (4.1) including noise in emissions, photolysis rates and deposition was propagated starting from the assimilated mean. In the first run, the stochastic model was driven by noise inputs $\mathbf{w}[k]$ equal to zero and decorrelation parameter $\alpha = \exp(-1/\tau)$ for $\tau = 24$ hr. With this setup, the values of the disturbed emissions etc decayed from their mean value at 15:00 to their deterministic value with a rate of α^k, where k is the number of hours past since 15:00 . In a second run, the disturbed parameters were fixed to their value 15:00, equivalent to $\alpha = 1$ and $\mathbf{w} = \mathbf{o}$. This setup reflects the idea that errors in model parameters are persistent, such that the settings for the afternoon of day 3 are the best settings for simulation of the ozone maximum in future too.

Figure 4.14 shows the deterministic, assimilated, and forecasted ozone concentrations for site Glazebury. The results show that a forecast with zero noise input rapidly converges to a deterministic model run. After 24 hours, the forecast is still in good agreement with the measurements (as the deterministic model run is); the high ozone level after 48 hours is missed for 60%, however. The forecast is still better than the model simulation, partly because of the better initial condition, and partly because 10% of the adjustments to the model parameters are still present. If emissions and other uncertain parameters are fixed to their last obtained value, the forecast skill improves significantly. The forecast of the maximum ozone level are close to the (optimal) assimilated value in that case. This result suggests that the optimal settings for the model parameters to simulate the ozone peak at day 6 are close to the adjustment required at day 4.

If not the model but the complete filter is used to provide a forecast (without assimilation of measurements), one is able to provide an expected quality of the forecast. The filter provides a forecast of the true state in terms of a mean $\hat{\mathbf{x}}^f$ and covariance $\mathbf{P}^f$, and together with the representation error covariance $\mathbf{R}$, this leads to an expected distribution of the observations:

$$\mathbf{y}^o \;=\; \mathbf{H}'\mathbf{x}^t \;+\; \mathbf{v} \;\sim\; \mathcal{N}\left(\mathbf{H}'\hat{\mathbf{x}}^f, \mathbf{H}'\mathbf{P}^f\mathbf{H} \;+\; \mathbf{R}\right) \tag{4.9}$$

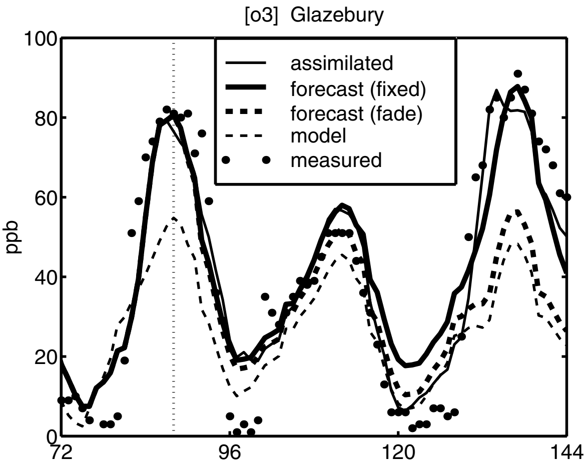


Figure 4.14: *Simulations and forecasts of the ozone concentration in Glazebury. The forecasts starts on august 8, 15:00 (vertical dotted line) from analyzed concentrations. Uncertain parameters in the stochastic model are either fixed to their value at the end of the assimilation, or fade to zero with decorrelation parameter* $\tau = 24$ *hr.*

The distribution describes confidence intervals in which elements of $\mathbf{y}^o$ are expected to be with some probability. The 95% confidence interval (2σ-bounds) for the 15:00 forecasts at day 6 in Glazebury obtained in this way are $[21, 85]$ ppb for the fading forecast (low to mid range ozone level), and $[51, 110]$ ppb for fixed (mid range to high ozone level). Both intervals include the actual measured value of 80 ppb and are therefore both reliable, although for fading forecast the measured value is close to the upper boundary of the interval. The 2σ bounds have grown substantially during the two days after the latest assimilation, indicating that the forecast includes much uncertainty. Providing σ-bounds with the forecast is expensive, since the filter should be run without assimilation of measurements but this is almost as expensive as running with assimilation. However, the σ-bounds provide useful insight in the quality of the forecast, and should be computed if the computation time is no restriction.

4.5 Summary and conclusions

In this research, the LOTOS model has been extended with stochastic variations in model parameters. With the stochastic variations, a Kalman filter is able to assimilate measurements with LOTOS simulations. Experiments with simulated data in a small domain (UK) showed that assimilation of measurements is able to compensate for uncertainties in several model parameters.

Uncertainties in NO_x emissions are compensated for by assimilation of NO_x and/or ozone measurements. The tight chemical connection between these components ensures that errors in modelled emissions are visible in ozone too. The possibility of using ozone measurements for estimation of NO_x is important here, since assimilation of NO_x measurements is hardly possible. The lack of NO_x measurements for rural sites, and worse representation of the vertical mixing of NO_x in LOTOS limit the comparison of model and NO_x data. Similar, errors in VOC emissions can not be corrected from assimilation of carbon measurements, since their number is sparse too and comparison with CBM-IV components is complicated. Ozone measurements provide useful information about VOC's through chemical coupling with NO_x, however. A stochastic model with uncertain NO_x and VOC emissions was shown to be useful for assimilation of ozone measurements; uncertain CO emissions are of minor use. Uncertain emissions only are not able to explain all differences between LOTOS and measurements, since the spatial impact of emissions is limited to the area's around the strongest emission sources.

Uncertainties in photolysis rates of O_3 and NO_2 were shown to have a large impact on computed ozone concentrations, and therefore suitable for the stochastic model too. Assimilation of ozone measurements is suitable to distinguish between errors in both photolysis rates. The impact of $J[NO_2]$ on the ozone level is stronger than the impact of $J[O_3]$ due to the larger value of the first.

Of the deposition parameters tested for the stochastic model, only those acting direct on ozone concentrations were found to be useful during assimilation of ozone measurements. Definition of the deposition velocity of ozone as the only uncertain deposition parameter is therefore the best option for the stochastic model. Uncertain deposition velocities are the most important error source for the ozone level during the night; the other uncertainties considered (emissions, photolysis) require day-light to become visible in ozone concentrations (except for titration with NO).

A combination of uncertain NO_x and VOC emissions, photolysis rates of NO_2 and O_3, and deposition velocity of O_3 was found to be useful for a stochastic model. Uncertainties in all these parameters are necessary to explain differences between simulated ozone and data from a measurement network. Assimilation of ozone measurements is able to decrease the uncertainties in these parameters. Especially the uncertainties in photolysis rates and deposition are decreased, since these act directly on ozone in remote as well as sub-urban sites. Assimilation of measurements with a simple Kalman smoother provides estimates of the uncertain parameters in terms of a mean and standard deviation. The time series of these estimates show which parameters are blamed for the differences between model and measurements. The time series from the experiments performed here are to short to draw conclusions about structural errors in the LOTOS input. Forecast experiments showed that the estimates of the model parameters improve the forecast skill of the model significantly. Fixation of the uncertain parameters to their latest estimated mean value leads to the best forecast skill for afternoon ozone maxima, which gives trust in the stochastic model.

Chapter 5

Application, part II: European domain

The Kalman filter around the LOTOS *model developed in chapter 4 and tested for a small domain is applied to a larger region covering west and central Europe and for a longer period. An error of 10–15 ppb between computed and measured ozone concentrations is left after assimilation. Estimates of the uncertain parameters obtained during the assimilation point to the existence of systematic biases in the model during the selected period. If these biases are taken into account, the filter is able to provide useful initial states for forecasts of ozone maxima.*

5.1 Introduction

The Kalman filter developed in chapter 4 was shown to be suitable for assimilation hourly ozone measurements in LOTOS, for a domain covering England and Wales over a five day time period (august 5 to 10, 1997). Although both domain and time period are rather small, many interesting characteristics were included: the domain covers industrialized and rural area's as well as sea, and during the simulation period a smog episode was build up. The results with a Kalman filter based on uncertain emissions of NO_x and VOC, photolysis rates of O_3 and NO_2, and deposition velocity of O_3 were satisfactory in all these circumstances, encouraging experiments for a large domain and longer period.

An issue which has not been discussed so far, but which becomes important for a large scale experiment, is the computation time of the filter. For the small scale experiments in chapter 4, the computation time was no constraint since it was limited anyway; the most expensive experiments took 3–4 hours on a work station. It was possible to configure the filter to produce the best possible solution for the stated problem, regardless of the computational costs. Therefore, an ensemble filter was used with an overhead of ensemble members (100–150), which provides the exact solution of the filter problem regardless of the stochastic model and complicating features such as nonlinearities. Experiments described in chapter 6 and chapter 7 showed that the ensemble filter is not the most efficient choice in terms of computation time, however. Therefore, for the large experiments described here a RRSQRT formulation of the Kalman filter is used, which was shown to provide similar results as could be obtained with the ensemble filter but for lower costs. The total costs of the filter are still impressive: for a model domain of 40×40 cells, similar stochastic model as used for the small domain, and a simulation period of one month, the total computation time is estimated on 5-6 days on a work station. The filter has therefore been implemented on a parallel computer; details of the parallelization are described in chapter 8.

Apart from the computational aspects, the filter procedure used for the large domain is also different with respect to the stochastic model. The spatial variability in model parameters is more important for a large domain, and the number of stochastic varying parameters has therefore been increased. The covariance of the measurement or representation error has been made time dependent, since comparison of LOTOS simulations with the measurements showed structural biases between model and measurements, especially during the night. A procedure is included to adapt the representation error automatically before assimilation.

The experiments on the large domain start with a description of the domain (§5.2) and stochastic model (§5.3), including the adaptive representation error. In following sections, results are described concerning ozone concentrations (§5.4), parameter estimation (§5.5), and forecast skill of the assimilation (§5.6).

5.2 Domain and period

The domain of the model was limited to an area of 40 × 40 grid cells (figure 5.1) A reasonable number of measurements is available for this area which might be represented by LOTOS simulations; elevated sites are excluded, as well as urban and traffic sites. The classification of certain sites might differ according to different sources; site Harwell for example is classified as 'rural' by the UK *National Environmental Technology Center*, while it obtained classification 'urban' according to (*Tilmes and Zimmermann, 1998*). By doubt, the site was accepted if LOTOS simulations are in good agreement with the measurements. The measurements from 23 sites were selected for use in the assimilation procedure, while from 18 sites the data was used to diagnose the assimilation result. Three of the diagnosed sites were excluded from the assimilation since they actually observe the reservoir layer; 4 sites in the northern part of the domain were used to diagnose the impact of transport and the adjustments to the photolysis rates which act on all grid cells.

August 1997 was selected as time periode for the assimilation. Except for the first 4 and last three days, august 1997 is characterized by overall high ozone concentrations. Exceptional high ozone levels (above 100 ppb) were measured in the UK on august 10 and august 19, and in central Europe at august 14 and 21/22.

Figure 5.2 shows the diurnal statistics of the difference between LOTOS and the measurements for the sites to be assimilated. The largest errors occur during the night (average underestimation of 8 ppb). Also the concentrations after sunrise are not simulated correctly; the rise of the concentrations occurs much faster in the model than in the measurements. For the later, a large number of explanations might be given, from which poor description of the rise of the mixing height is the most likely (the meteorological input is refreshed every three hours only). Around 15:00, when the ozone peak occurs, the model is on average in good agreement with the measurements.

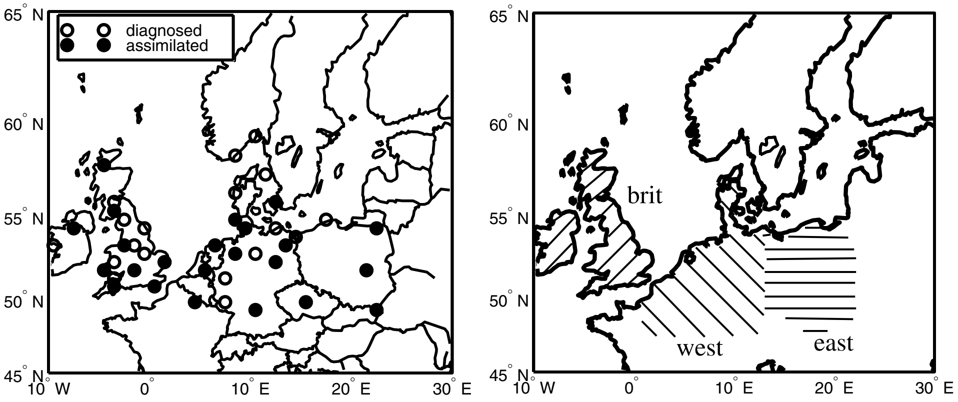


Figure 5.1: *Sites from the* EMEP *network used on the* 40×40 *grid, and areas used in the stochastic model.*

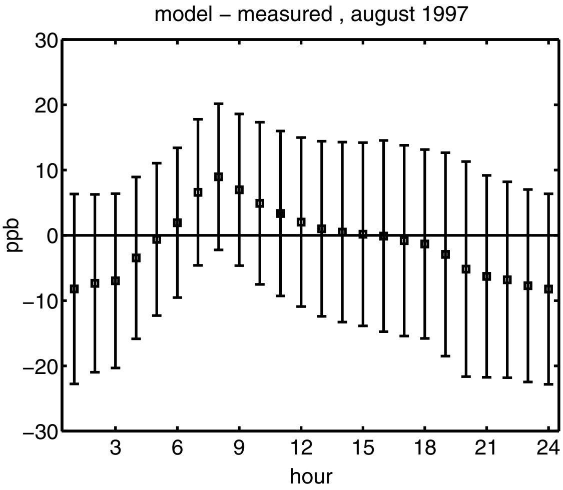


Figure 5.2: *Mean and standard deviation of the difference between model and measurements as a function of the hour of the day, for the sites selected for assimilation.*

5.3 Stochastic model and filter

The stochastic model around LOTOS used in this research contains a combination of uncertain emission, photolysis rates, and deposition velocity, shown to be useful in chapter 4. The experiments on the small domain described in chapter 4 showed that the filter technique is able to account for errors in these parameters. The standard deviations of the uncertain parameters are set to 40% for the NO_x emissions, 50% for VOC, and 30% for the photolysis rates and the deposition of ozone. The time correlation parameter was set to $\tau = 12$ hour to let day and night time variation be more or less independent from eachother. Emissions and deposition velocity are defined stochastically varying in three regions (figure 5.1) covering the British islands, west, and central Europe respectively. These areas are rather large to ensure that each contains a reasonable number of measurement sites, and the problem of estimation of uncertain parameters does not become ill-posed.

How much of the difference between model and measurements might be explained from these uncertainties? Although the stochastic model provides a reasonable degree of freedom, it can not explain every residue between model and measurements. For example, the spatial degree of freedom is limited to variations in three different area only (emissions and deposition) or does not contain any spatial variation at all (photolysis). The filter will therefore tend to produce a situation which is optimal over the assimilated sites on average. The remaining residue should be explained from the representation error between the chosen stochastic model and the measurements. The diurnal variation observed for the residue (figure 5.2) suggests that the representation error is time dependent, or at least varying with the hour of the day. Therefore, the representation error covariance is determined adaptively following the observed residues, instead of being prescribed on forehand. Such a procedure is an example of an *adaptive filter*. In the usual setup for an adaptive filter (*Dee, 1995; Mitchell and Houtekamer, 2000; Ménard et al., 1999*), the unknown parameters in a parameterization of the forecast error covariance matrix are tuned with the observed residues. In here we will use the same procedure to tune the representation error covariance, with the forecast error left unchanged.

For the technique of adaptively choosing the representation error we follow the procedure described by (*Ménard et al., 1999*). Let $\mathbf{d} = \mathbf{H}'\mathbf{x}^f - \mathbf{y}^o$ be the residue of the forecast at a time $t[k]$. In the context of the Kalman filter, the residue vector is supposed to have zero mean and covariance $\boldsymbol{\Gamma}(\rho) = \mathbf{H}'\mathbf{P}^f\mathbf{H} + \mathbf{R}(\rho)$, where $\mathbf{P}^f$ is the forecast error covariance and $\mathbf{R}(\rho)$ the representation error covariance. The value of the representation error covariance depends on a parameter ρ. The representation errors are chosen to be uncorrelated with standard deviation equal to ρ, thus $\mathbf{R}(\rho) = \mathbf{I}\rho^2$. The unknown variance ρ^2 is set to a value such that the probability of $\mathbf{d}$ being a sample out of $\mathcal{N}(\mathbf{o}, \boldsymbol{\Gamma}(\rho))$ reaches a maximum:

$$\max_{\rho}\ p(\mathbf{d};\rho) \;=\; \frac{1}{\sqrt{(2\pi)^r \det(\boldsymbol{\Gamma}(\rho))}} \exp\left\{-\frac{1}{2}\mathbf{d}'\boldsymbol{\Gamma}(\rho)^{-1}\mathbf{d}\right\} \tag{5.1}$$

The output of the filter is now not only the mean state and covariance matrix, but also the tuned variance of the representation error. The representation variance describes how a measurement might differ from an the average concentration in a large parcel of air, if this average is computed by the LOTOS model with certain parameters modelled stochastic.

The Kalman filter used to assimilate the measurements and the stochastic model is described in detail in chapter 6. The filter is based on a low-rank parameterization of the covariance matrix with about 60 modes; a RRSQRT formulation is used to solve the filter equations. For treatment of the nonlinearities in the model, the filter is configured to use the forecast step of the SEIK filter, described in chapter 7. Computational costs are dominated by evaluation of the LOTOS model, which has to be performed at least one time for each mode of the covariance matrix. A single work station is not able to perform this task in a reasonable time (apart from storage problems). Therefore, the filter was run on 8 processors in parallel using a domain decomposition approach; see chapter 8 for the parallelization.

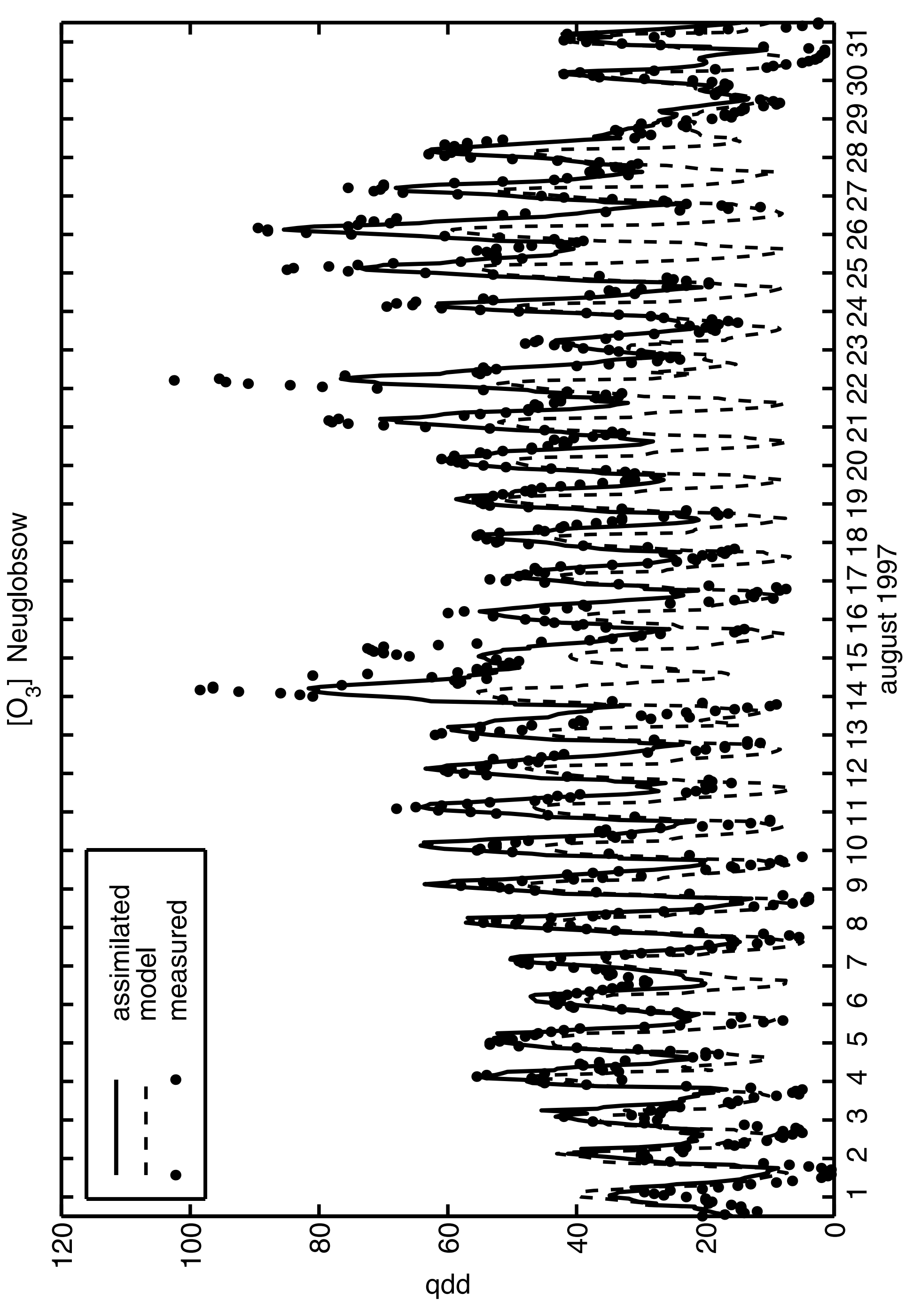
5.4 Assimilated ozone

The previous described stochastic model and Kalman filter have been used to assimilate ozone measurements in LOTOS. Figure 5.3 shows an example of the ozone concentrations during the assimilation for site Neuglobsow (eastern Germany, included in the assimilation). Many features present for the time series in Neuglobsow are also found for other sites. From the time series it is clear that without assimilation, the model tends to produce long term averages. The simulated ozone concentrations show a very regular, sinusoidal pattern, with minima and maxima almost constant during the complete month. The measurements roughly show a similar sinusoidal pattern but with much more variation in the minima and maxima. The model is not able to simulate these short term variations correctly, except for some occasions when the afternoon ozone maximum is obvious lower than the day before (august 15, 23, and 29). The lower ozone levels are both visible in data and model simulation, and investigation of the model input showed that these features are related with cloud cover and lower mixing heights. At the other days, the maxima are systematically underestimated by the model. This cannot be explained from the fact that the model simulates average concentrations in large grid cells rather than the concentration at the measurement site, since the underestimation of the maxima is noticed at surrounding sites too. The lack of variation in the simulated ozone has therefore to be explained from the lack of variation in the input data for for example the emissions and the deposition model.

As the assimilated time series for Neuglobsow show, the assimilation procedure is able to produce ozone concentrations which are in much better agreement with the variations in the measurements. Most of the extrema in the data are covered by the assimilation, except for, unfortunately, the two highest measured ozone peaks at august 15 and 22. The build up of an ozone episode at these days is too fast to be followed by the model; results are better if the increase is more gradual (august 23–26).

As an example of the spatial characteristics of the assimilation, maps of the simulated, assimilated, and measured ozone maximum at august 26 are plot in figure 5.4. The measurements were interpolated on the grid through simple Kriging (see for example (*Zhang, 1996*)). The assimilation is able to reproduce the shape and level of the smog episode in central Europe, where the model did not simulate the highest level at all. The differences between model simulation and filter also extends to the alpine region and the Baltic sea. Although the model shows serious biases with the measurements for those areas, the assimilation leads to a small but significant improvement of the simulations. This result could

Figure 5.3: *Ozone concentrations in Neuglobsow.*



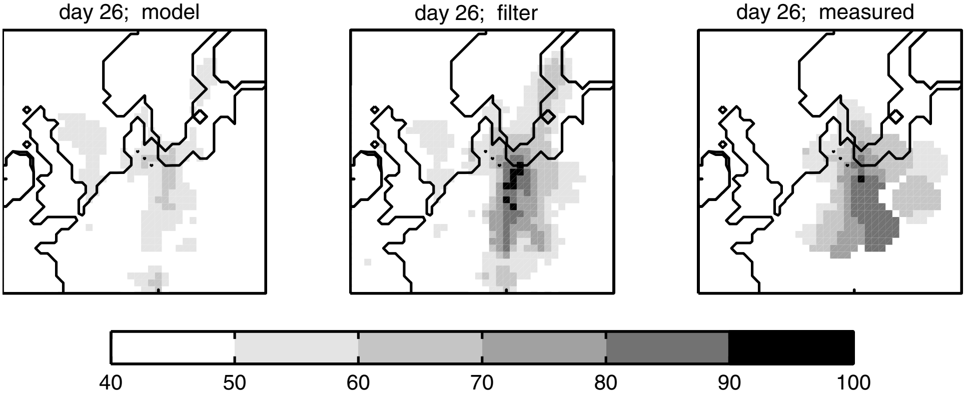
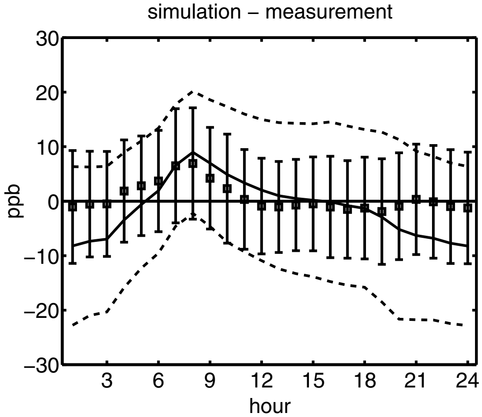


Figure 5.4: *Maps of ozone maximum at august 26 according to model simulation, filter, and measurements. The measurements were interpolated on the grid through simple Kriging; no data was plotted at more than 200 km of a measurement site.*

be explained from transport of improved concentrations from central Europe, and/or the adjustments to photolysis rates, which impact is present at all grid cells.

Figure 5.5 shows the average deviation of the remaining residue between assimilated mean and measurements, as a function of the hour of the day. In comparison with the errors before assimilation (lines, same as figure 5.2), the residues have become unbiased during almost all hours of the day. The only bias still present is the increase of ozone after sunset which is still to fast. The stochastic model does not account for the most likely origin of this bias (inaccurate rise of the mixing layer), and should be extended to cover this model error to. The residues are decreased for almost all hours of the day, especially during the afternoon. During these important hours (the ozone maximum is reached here), the standard

Figure 5.5: *Mean and one sigma-bands of the error between simulation and measurements as a function of the hour of the day for the assimilated sites: before assimilation (lines, same as in figure 5.2) and afterwards (error bars).*



deviation of the residues has decreased from 15 ppb to less than 10 ppb.

The improvement of the simulation is also shown from the statistics of the standard deviation ρ of the representation errors, adapted during the filter process. The filter was run first without assimilation of measurements, leading to values for ρ between 8 and 20 ppb (mean±std.dev.). Figure 5.6 shows that after assimilation, a standard deviation of 7–15 has to be accepted for the representation error. The best representation is obtained for the afternoon, when the standard deviation is decreased from 8–16 without assimilation to 7–12 ppb with assimilation. The highest values for ρ were found for the evening and nighttime hours between 20:00 and 7:00. After assimilation, the simulation of the ozone built up during sunrise has became slightly worse than it was before. The deterministic model already started to build up ozone too early, and with the filter in a mode producing a higher afternoon maximum, the build up has became even stronger. This is a result from the assimilation in the days before, since the filter is blind for future measurements; the model parameters stored in the state vector direct the model towards increased ozone production.

To judge the overall errors left after the assimilation, the root mean square errors of the simulated ozone is computed for for each individual site according to:

$$\text{RMS} = \sqrt{1/\nu \sum_{i=1}^{\nu} (c_i - y_i^o)^2} \tag{5.2}$$

where c denotes an ozone concentration simulated with the model or the filter, y^o a measurement, and ν denotes the number of available measurement/simulation pairs for a site and during certain hours of the day. The results are plot in figure 5.7, for the simulations during the night (21:00–6:00) and day time (9:00-18:00). The figure shows that the assimilation reduces the night-time RMS errors to less than 15 ppb for the bulk of the sites. The outliers concern sites which actually observe the residual layer during the night or suffer from irregular inflow from sea side, and had therefore been rejected for assimilation on forehand. The representation gets slightly worse at four sites, but the errors remain less than 15 ppb. The daytime RMS errors are reduced to less than 12 ppb for most of the sites, except for 3 sites which suffer from irregular inflow from sea. The assimilation has improved the

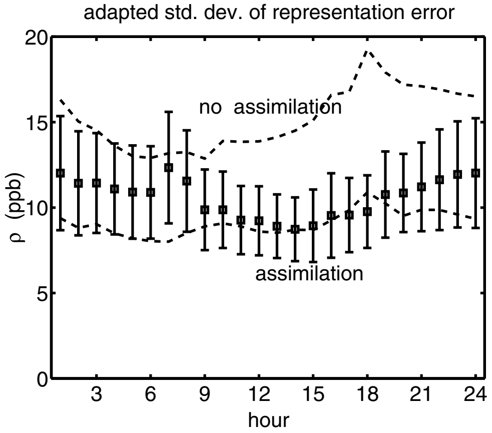


Figure 5.6: *Statistics of the adapted standard deviation* ρ *of the representation error as a function of hour-of-the-day before assimilation (dashed) and after assimilation (error bars). The smallest values for* ρ *are obtained for the afternoon hours, while the maxima are obtained around midnight and sunrise.*

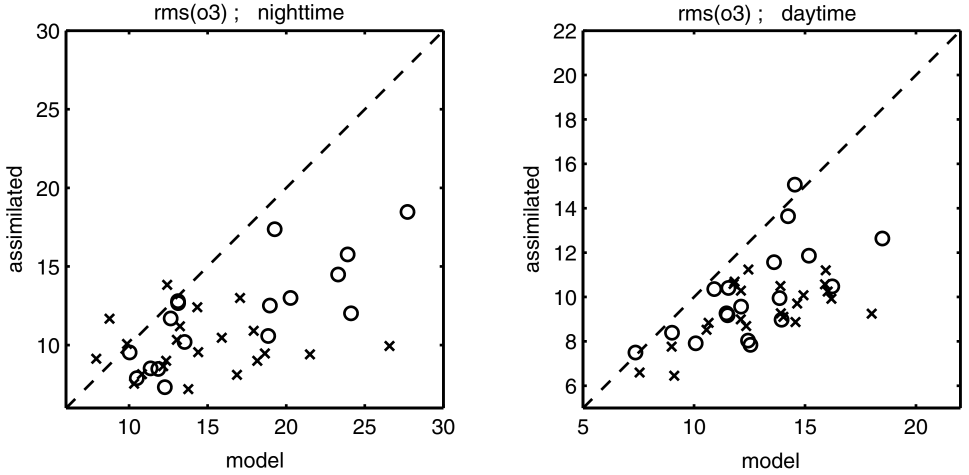


Figure 5.7: *Root-mean-square errors between model and measurements and between assimilated mean and measurements during nighttime hours (21:00–6:00) and daytime (9:00–18:00), for assimilated (×) and diagnosed (○) sites.*

representation of the measurements for both the assimilated and the diagnosed sites, which indicate that for these sites the residuals can be explained from similar model uncertainties.

5.5 Parameter estimation during assimilation

After evaluating the simulated ozone concentrations after assimilation, this section describes how the model parameters have been changed by the filter. That is, how did the filter use the degree of freedom in the stochastic model ?

The filter provides estimates of the stochastic model parameters in terms of a mean and covariance, for each hour in the assimilation period. Figure 5.8 shows the statistics of the mean values of the parameters in the stochastic model as a function of the hour of the day. The σ-bounds give an impression of the actual emissions, depositions and photolysis rates used during the filtering process.

The figure shows that the emission rates used in the filter have more or less the same distribution as those used in the stochastic model, appart for the NO_x emissions in western Europe which are on average 25% higher. The deposition velocities are also strongly biased from their deterministic values: increased during day time at the British islands, and decreased on continental Europe. The model tends to underestimate the ozone concentrations at some of the continental sites and the filter therefore decreases the removal here. As a result, the deposition in the easter part of the domain has decreased with about 30%. For the British sites however, in a significant number of the more remote sites the model shows a small over estimation, leading to an increased deposition of about 10%. Underestimation of

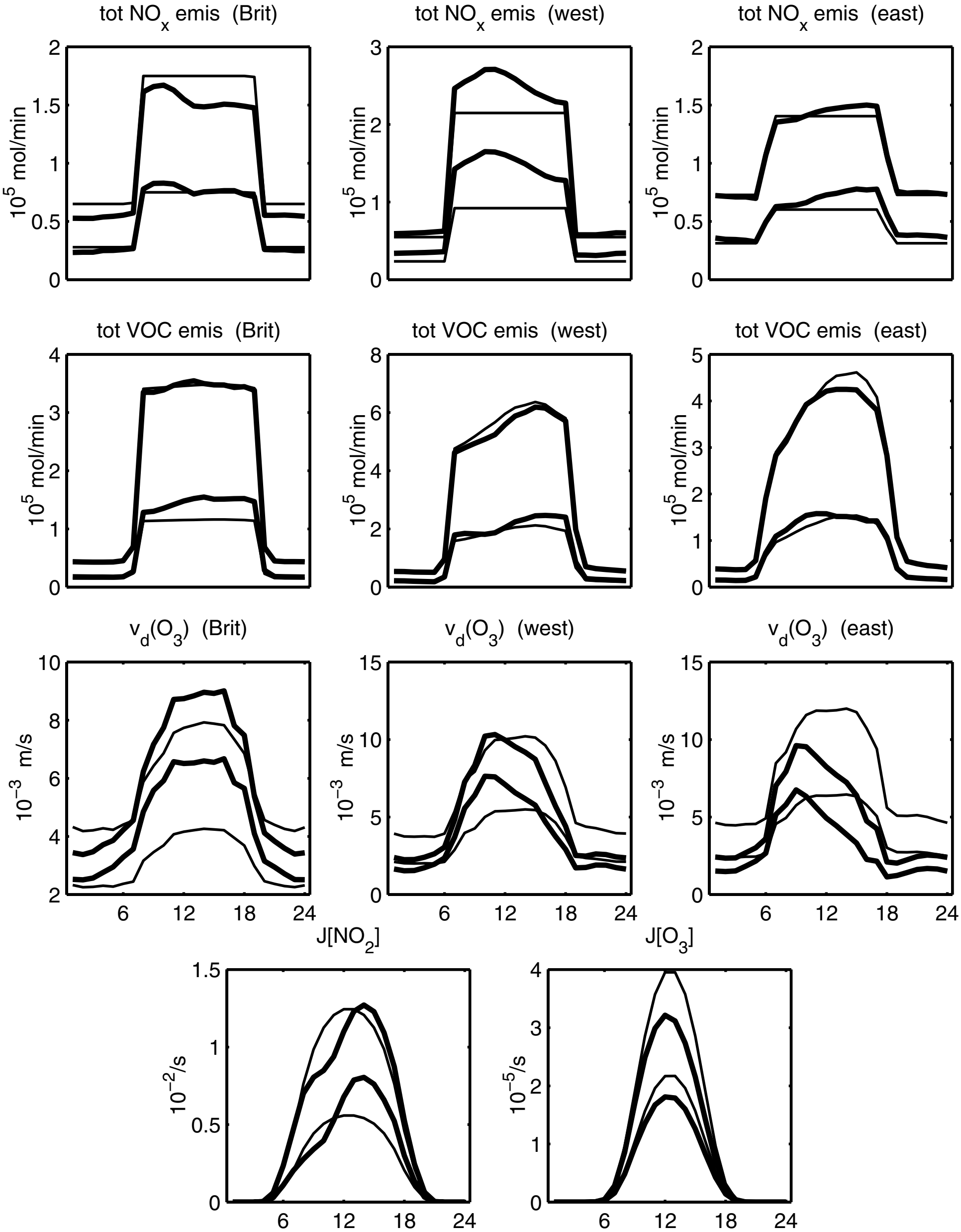


Figure 5.8: *Mean ± standard deviation of the mean value of stochastic modelled parameters as a function of the hour of the day: before assimilation (thin lines) and after assimilation (thick). Total emissions are summed over all grid cells in a certain area; the deposition and photolysis rates are averages.*

high ozone levels is for the British sites only present around industrial areas, and the filter reacts with a small increase of VOC emissions and decreased titration with NO.

The photolysis rates show on average a strong bias too: $J[O_3]$ is decreased with about 20% during all hours, while $J[NO_2]$ is decreased during the early morning and increased during the afternoon with about the same value. If the stochastic model is correct, these results suggest that in the model the photolysis rate of ozone should be decreased and that a day profile should be included for the the photolysis rates of NO_2. A physical explanation for such a profile is the effect that clouds decrease the ratio $J[\mathrm{NO}_2]/J[\mathrm{O}_3]$ (*Matthijsen, 1995*). This effect could lead to a lower $J[\mathrm{NO}_2]$ in the morning since cloud coverage is higher at this time. A cloud dependency for the photolysis-ratio is not included in the model, and the filter might have partly compensated this lack. The change in the photolysis rate found in figure 5.8) is too strong to be be explained completely from the described mechanism, however.

Do the observed biases point to biases in the model parameterizations? They indicate at least that our stochastic model is not correct since we made the assumption of the uncertainties being unbiased. Only for the VOC emissions and some of the NO_x emissions this seems to be correct, although the situation might change if measurements of nitrogen and organic compounds are included in the assimilation. For the other parameters it is not sure whether the deterministic model is really biased or not, since other sources of uncertainty are not be included in the stochastic model, and we might have better results because of the wrong reason. The chosen degree of freedom allows reasonable variations in the model parameters, and due the use of an adaptively chosen representation error, the values of the parameters fed to the model are not to far outside this degree of freedom. An interesting experiment is to see whether the estimates of model parameters obtained with the filter are able to provide a proper simulation, if they are just inserted in the model. The stochastic model was therefore run over the assimilation period using model parameters set to with lag-one smoothed parameters in the assimilated mean. The resulting ozone simulations are comparable with the assimilation results: the RMS errors as drawn in figure 5.7 increased with less than 2 ppb. The largest differences were left when large ozone maxima were completely missed by the model, and the filter required a degree of freedom not provided by the stochastic model.

5.6 Forecast of the ozone maxima

With the assimilation period of one month, it is now possible to analyze the forecast skill of the filter in more detail. Similar as for the forecast experiments on the small grid described in §4.4.3, the analyzed mean at 15:00 was used as an initial state for a run with the stochastic model. Each of the initial states was propagated over 130 hours, such that a forecast of the ozone maximum could be made up to 5 days ahead.

Figure 5.9 shows an example of simulated and forecasted ozone maxima for the site Neuglobsow (see also figure 5.3). The parameters in the stochastic model were set to their deterministic value or fixed to the last value obtained with the filter; fading the parameters from assimilated towards deterministic values (as used in §4.4.3) provides results in between these extreme situations. The results in figure 5.9 show that fixation of the parameters pro-

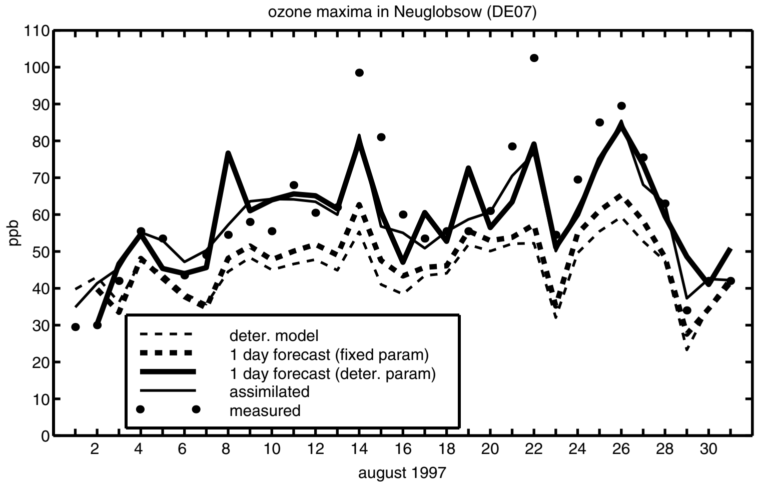


Figure 5.9: *Ozone maxima in Neuglobsow during august 1997 during assimilation and forecast experiments. (see figure 5.3 for complete time series).*

vides a forecast that is close to the assimilated mean on almost every day. Exceptions are the forecasts for august 8 and 19 which exceed the assimilated (and measured) value with more than 20 ppb; the forcast suffered here from exceptional low values for the deposition rate. If the parameters in the stochastic model are not fixed but reset to their deterministic values, the one-day forecast is already close to the values simulated with the deterministic model. This results shows that an improved initial concentration only does hardly lead to a better forecast. Similar results were found in (*Elbern and Schmidt, 2001*), where a 4D-var method was used to obtain initial concentration fields for EURAD model. In their experiments, significantly better forecasts were obtained in the 6 to 12 hour range, including prediction of afternoon ozone peak; afterwards, an improvement is still visible but quickly fading. For the current generation of ozone models, an estimation procedure of model parameters needs to be included in a forecast system.

Figure 5.10 shows the RMS errors of the forecast as a function of the forecast period. In addition, the RMS errors obtained with the deterministic model and the filter have been computed too. With parameters set to the deterministic value, the forecast error converges within three days to the error obtained with the deterministic LOTOS model. With fixed parameters, the rms error is about 3 ppb less during all days. Even for a five days forecast, the fixed-parameter forecast is still an improvement in comparison with the model. The fixed-forecast RMS does not necessarily converge to the model RMS for longer forecast intervals, since the parameters set during assimilation have changed the model.

Note that the offline 'forecast' system used here will always perform better than a similar operational system, since it is based on analyzed meteorological data. The meteorological forecast for the coming two days is not very different from the data analyzed afterwards. The five days forecast might however contain serious deviations, and this will influence the

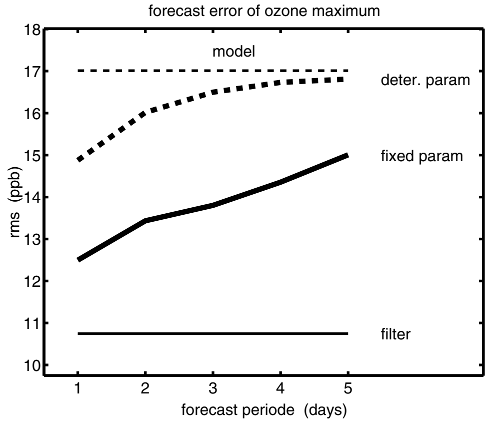


Figure 5.10: *Root-mean-square error between measured and simulated ozone maximum as a function of the forecast period; the rms is taken over all sites and all hours for which both measurements as well as five different forecasts are available.*

forecasts of the ozone maxima.

5.7 Discussion and conclusions

In this research, a large scale Kalman filter has been applied to assimilate ozone measurements in the LOTOS model. Hourly ozone measurements from 41 sites in the north-western part of Europe were found to be comparable with the LOTOS simulation. This number is in fact rather low, since the total number of air quality sites in the area exceeds the number of 200 (*Tilmes, 1999*). Sites in city centers and elevated sites are hardly comparable with LOTOS, however, due to the coarse grid and poor vertical resolution. The remaining sites provide useful information about the quality of the model. Timeseries of measurements and model simulations show for example that the model systematically under estimates the night-time ozone concentrations.

Many of the differences between model and measurements were found to be covered by a stochastic model with uncertain emissions (NO_x and VOC), photolysis rates (NO_2 and O_3) and deposition velocity (O_3). The remaining differences are due to uncertainties in other model parameters (for example the height of the mixing layer), a spatial variability that is larger than modelled here, or just the lack of representation of a measurement site by the LOTOS model. An adaptive procedure has therefore been included in the filter to estimate the optimal value of the representation error during each hour of the assimilation. The results after assimilation show that a standard deviation of 7–15 ppb should be accepted for the representation error. The best representation is achieved for the afternoon ozone maximum (7–12 ppb). The adaptive procedure prevents the filter from adjusting LOTOS parameters beyond the limits defined in the stochastic model. Some form of adaptive tuning of the stochastic model should be included in every filter procedure, to obtain covariances consistent with the actual observed residues. Besides, analysis of the tuning parameters provides useful information about periods where model and measurements diverge.

Estimations of the uncertain model parameters during the assimilation show that especially the uncertainties in deposition are used to explain the difference between LOTOS and measurements. The impact of emissions is only visible in the measurement sites close to the largest emission areas. The same holds for the photolysis rates, whose impact is strongest in the emission plumes. Deposition rates are therefore a key parameter for ozone at ground level in large parts of the domain. Ground based measurements in rural areas might not be suitable to draw conclusions about emissions from urbanized areas. If the filter around the LOTOS model is to be used explicitly for emission estimations, the horizontal resolution should be increased for better represention of measurements within urbanized area's such that these can be assimilated. Another option is to assimilate satellite measurements which measure total columns rather than surface values. Experiments with the LOTOS clone EUROS during the STROPDAS project (*Velders et al., 2001*) showed however that the quality of tropospheric ozone columns from satellite instruments is too low at the moment (see also figure 1.3). With improvement of the tropospheric columns, the vertical extent of the models should be improved too to cover at least the total troposphere.

The assimilation procedure is able to improve the quality of an ozone forecast significantly. The rms error in the one-day forecast of the ozone maximum decreases with 25% in comparison with a deterministic forecast. To achieve this, the uncertain parameters in the stochastic model should be fixed to the values estimated by the filter for the afternoon. This results shows that errors in the model parameters are rather persistent, since parameter settings suitable to simulate the ozone maxima at a certain day are suitable for following days too. An improvement of the forecast skill is still visible after five days. If the uncertain parameters are set to their deterministic values, and the only improvement from the assimilation is a better initial condition, the forecast skill is much smaller and almost negligible after two days. Therefore, a stochastic model for LOTOS or comparable models should be based on uncertain parameters, since improvement of the initial concentrations only is not sufficient to improve a forecast.

Chapter 6

Low-rank filters

The filter around the LOTOS *model applied in chapters 4 and 5 takes the form of a low-rank filter. These kind of filters form a broadly used class of approximate Kalman filters, suitable for data assimilation in models with large state vectors. The background, algorithm, and (dis)advantages of several forms of low-rank filters are discussed and compared:* RRSQRT, SEEK/SEIK, ESSE, ENKF, *and* POENKF. *The performance of these filters in combination with the* LOTOS *model has been tested during experiments with simulated data, with the best results obtained for a* RRSQRT *filter incorporating the forecast step of the* SEIK *filter.*

6.1 Introduction

Application of the Kalman filter in guidance and electrical engineering have lead to decades of experience in merging measurements with numerical models. With the growing interest of assimilation of measurements in numerical weather, ocean, and climate models, the Kalman filter has therefore been proposed as a natural tool for solving assimilation problems (*Ghil et al., 1981*). Direct application of the traditional Kalman filter techniques to these kind of models is hampered by the required computation power, however, growing quadratic with the number of elements in the model state. For geophysical models, the later is often in order of hundred thousands, and implementation of the Kalman filter is therefore only possible for massive parallel computers (*Lyster et al., 1997*). Even if the computational resources could solve the problem in theory, practical considerations about computation time and financial costs will often reject the option of a traditional Kalman filter.

Since the quadratic growing costs of the Kalman filter are related to storage and propagation of the covariance matrix, many solutions have been proposed for limiting the costs associated with this matrix. In (*Heemink, 1988*), the covariance matrix was almost completely avoided by using a Kalman filter with steady state gain for filtering a 2D shallow water model. This approach is based on the assumption that the covariance matrix hardly changes after some initialization period, and can be computed and stored off-line. Another method to limit the costs of the covariance matrix is to reduce its effective size. Model reduction techniques might be used to limit the dimension of the error model. Examples of this approach are found in (*Heemink and Kloosterhuis, 1990*) and (*Fukumori and Melanotte-Rizzoli, 1995*), where the model errors are defined on a coarser grid than the model state. Without reducing the size of the covariance matrix, the costs of the covariance propagation could be reduced by using a simplified propagation. In (*Cohn and Todling, 1995*), a filter was implemented with the error propagation based on the leading singular vectors of the tangent linear model. Although this method leads to an efficient propagation, the storage of the covariance matrix is still a problem if the dimension of the state increases.

Since devellopers of geophysical models tend to increase the size of the state vector with the years (higher resolutions, additional physics), storage of a complete covariance matrix has become almost impossible. Therefore, approximate filters have been proposed, to limit both the storage as well as the propagation costs. In (*Parrish and Cohn, 1985*), the covariance matrix was approximated by a band matrix, based on the assumption that spatial correlations vanish at large distances. This method suffers from an increasing bandwidth which is hard to avoid; simple truncation of off-diagonal elements has the danger of loosing the positive-definiteness of the covariance. In for example (*Eskes et al., 1999*), a parameterization of the covariance matrix is used with separate treatment of spatial correlations and standard deviations, for assimilation of ozone columns in a 2D global model. The standard deviations form a field on the model grid, propagated in time by the default advection; the spatial correlations are extracted from measurement data.

In this research, the method of low-rank approximations of the covariance matrix will be discussed. The low-rank approximation is based on the observation that in many practical filter problems, the covariance matrix is dominated by a limited set of modes, typical ten to hundred. The Partial Eigendecomposition Kalman Filter (PEKF) proposed in (*Cohn and Todling, 1995*) uses a parameterization of the covariance based on the leading eigenvectors, computed from a full covariance function with a Lanczos algorithm. A similar approach based on a singular value decomposition was introduced for the Reduced Rank SQuare RooT (RRSQRT) filter (*Verlaan and Heemink, 1995*). The RRSQRT approach does not rely on a user defined covariance function, but builds a covariance matrix from zero. Expression of the model error in terms of empirical orthogonal functions lead to the formulation of two other low-rank filters: the Singular Evolutive Extended/Interpolated Kalman (SEEK/SEIK) filter (*Pham et al., 1998; Verron et al., 1999*) and the Error Subspace Statistical Estimation (ESSE) framework (*Lermusiaux and Robinson, 1999b*). A natural advantage of the low-rank filters is that the covariance matrix never needs to be evaluated in full form, although each element is computed easily if necessary. Besides, the low-rank parameterizations ensure that the covariance matrix is always positive definite.

All of the mentioned low-rank filters explore the fact that the eigenvectors or modes of the covariance matrix have an ensemble interpretation. Each mode has the shape of a state vector, and together these state vectors form an ensemble of deviations around the mean state. The reverse approach, building a covariance from state vectors, forms the basic idea behind the Ensemble Kalman Filter (ENKF) (*Evensen, 1994*). An ensemble of model states is used as an estimation of the true state; whenever the filter requires statistics such as mean and covariance, these are obtained from the sample statistics of the ensemble. This approach is therefore completely different from other low-rank filters, where the basic formulation starts with an approximation of the covariance matrix, which turns out to have the interpretation of an ensemble afterwards. The simple formulation has lead to application of the ENKF in many oceanographic and meteorological studies (*Evensen and van Leeuwen, 1996; Evensen, 1997; Keppenne, 2000; Houtekamer and Mitchell, 1998*). As a result, features of the ENKF have been incorporated in other filter schemes, especially to treat nonlinearities.

The sharp division between the different approaches has therefore disappeared.

The capability of handling models with large state vectors make the low-rank filters suitable to assimilate measurements in the LOTOS model. The background and implementational details of some popular low-rank filters are therefore studied in detail. For completeness, the discussion is started with the full rank formulation (§6.2). Some general properties of the low-rank formulation are discussed in §6.3 and §6.4. The methods based on low-rank factorizations of the covariance matrix (RRSQRT, SEEK/SEIK, and ESSE) are described in §6.5. The ensemble method is discussed in §6.6 as a separate case. Recently, a combination of factorization and ensemble methods has been proposed to combine benefits of both methods: the Partially Orthogonal Ensemble Kalman Filter (*Heemink et al., 2001*); the POENKF formulation is described in §6.7. A summary of the implementations of all low-rank filters is given in §6.8. A critical step in the methods based on factorization, the truncation of the covariance matrix up to some rank, is discussed afterwards in detail in §6.9. The performance of the different filter techniques has been tested for the LOTOS model in experiments with simulated data; setup and results are discussed in §6.10.

6.2 Kalman filter with full covariance matrix

Let for implementation of a Kalman filter around LOTOS the evolution of the state and observation of measurements be described with the stochastic system:

$$\mathbf{x}^t[k+1] = \mathbf{A}[k]\,\mathbf{x}^t[k] + \boldsymbol{\eta}[k] \tag{6.1a}$$

$$\mathbf{y}^o[k] = \mathbf{H}[k]'\mathbf{x}^t[k] + \mathbf{v}[k] \tag{6.1b}$$

with $\mathbf{x}^t[k] \in I\!R^n$ the true state vector at time $t[k]$, $\mathbf{A}[k]$ a deterministic model, $\boldsymbol{\eta}[k] \in I\!R^n$ a Gaussian distributed model error (zero mean, covariance $\mathbf{Q}$), and $\mathbf{y}^o[k] \in I\!R^r$ a vector of observations with $\mathbf{v}[k]$ the representation error (Gaussian with zero mean and covariance $\mathbf{R}$). Indices 't', 'o', and later on 'f' and 'a' refer to true, observed, forecasted and analyzed entities respectively, as introduced in chapter 3. The notation with a linear operator $\mathbf{A}$ is chosen in order not to complicate the formula, although the stochastic model $\mathbf{M}(\mathbf{x})$ used in chapter 4 and 5 is in fact nonlinear in $\mathbf{x}$. The treatment of nonlinearities is discussed in detail in chapter 7; for the moment, the linear interpretation is suitable and more clear. The time indices for $\mathbf{A}$ and $\mathbf{H}'$ will be dropped in coming equations, assuming that the time is implied by the state where the operators act on.

The goal of the filter operations is to obtain the mean $\hat{\mathbf{x}}^a$ and covariance $\mathbf{P}^a$ for the probability density of the true state. The filter equations for this system have been derived in

§3.3.3 and are summarized by:

forecast:

$$\hat{\mathbf{x}}^f[k+1] = \mathbf{A}\,\hat{\mathbf{x}}^a[k] \tag{6.2a}$$

$$\mathbf{P}^f[k+1] = \mathbf{A}\,\mathbf{P}^a[k]\,\mathbf{A} + \mathbf{Q}[k] \tag{6.2b}$$

analysis:

$$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f + \mathbf{K}\,(\mathbf{y}^o - \mathbf{H}'\hat{\mathbf{x}}^f) \tag{6.2c}$$

$$\mathbf{P}^a = \begin{cases} (\mathbf{I}-\mathbf{K}^{MV}\mathbf{H}')\mathbf{P}^f \quad , \quad \mathbf{K}^{MV} = \mathbf{P}^f\mathbf{H}'(\mathbf{H}'\mathbf{P}^f\mathbf{H}+\mathbf{R})^{-1} \\ (\mathbf{I}-\mathbf{K}\mathbf{H}')\mathbf{P}^f(\mathbf{I}-\mathbf{K}\mathbf{H}')'+\mathbf{K}\mathbf{R}\mathbf{K}' \quad , \quad \text{arbitrary gain } \mathbf{K} \end{cases} \tag{6.2d}$$

In combination with a large model, the propagation of the covariance matrix in (6.2b) is the most expensive part in the full rank filter. The dynamical model is called $2n$ times to perform the operation $\mathbf{A}(\mathbf{A}\mathbf{P})'$. Even with the aid of parallel computing this is hardly feasible (*Lyster et al., 1997*). The only practical method to implement a full rank filter is to use simplifications for the model **A**. A very strong simplification of the model is for example to replace it by the identity, such that the covariance only grows through the introduction of dynamic noise in **Q**. Cohn and Todling (1995) performed a singular value decomposition of the dynamical operator **A**, under assumption that the growth of the error covariance is dominated by a few rapidly growing singular modes of the model.

Note that 'full rank' does not imply 'completely filled': the covariance matrix might be sparse. This is in fact a necessary requirement for implementation of a full rank filter, since storage of a $n \times n$ matrix is practically impossible for large n. If correlations between grid points are supposed to vanish with increasing distance, the covariance matrix becomes sparse and only the non-zero elements have to be stored. A problem with this method is that the sparseness is only maintained during propagation if the dynamical model describes advection only. If the model is diffusive, new off-line elements are introduced. In (*Parrish and Cohn, 1985*), this problem is solved through truncation of covariance elements below some threshold, but obtained difficulties with preservation of positive definiteness.

Limiting both the number of model evaluations as well as the storage requirements is better achieved by reducing the rank of the covariance matrix. In the next sections we will describe a number of filter algorithms based on this method.

6.3 Factorization of the covariance matrix

A covariance matrix is positive definite. This property of a covariance matrix is easily lost when numerical operations are performed, for example due to truncation errors. To avoid this problem, (*Bierman, 1977*) proposed to rewrite the equations for the Kalman filter using the factorization $\mathbf{P} = \mathbf{S}\mathbf{S}'$. Numerical inaccuracies made in computation and storage of the matrix **S** will never affect the property of positive definiteness of **P**. Inaccuracies will even be reduced since the condition number of **S** is only the square root of the condition number of **P**.

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

$$\underset{n\times n}{\mathbf{P}} = \underset{n\times m}{\mathbf{S}} \; \underset{m\times n}{\mathbf{S}'} \tag{6.3}$$

Matrix **P** is a valid covariance matrix independent of the shape and contents of the matrix **S**. Even for a square root formed by a singe column, the product $\mathbf{SS}'$ is still a valid covariance matrix. The rank of **P** is equal to the rank of **S**, and thus less than or equal to m. An arbitrary element of the covariance matrix **P** is equal to the inner product of two rows of **S**:

$$p_{ij} = \mathbf{S}_{(i,:)} \, \mathbf{S}_{(j,:)}{}' = \sum_{k=1}^{m} s_{ik} s_{jk} \tag{6.4}$$

With this formula it is a simple exercise to show that the complete covariance matrix is formed from a sum of 'outer' products of columns of **S**:

$$\mathbf{P} = \sum_{k=1}^{m} \mathbf{S}_{(:,k)} \mathbf{S}_{(:,k)}{}' = \sum_{k=1}^{m} \left(\mathbf{s}_k \mathbf{s}_k{}'\right) = \sum_{k=1}^{m} \mathbf{P}_k \tag{6.5}$$

where $\mathbf{P}_k = \mathbf{s}_k \mathbf{s}_k{}'$ denotes the rank-one covariance matrix formed from column $\mathbf{s}_k$. A full rank **P** might be formed from a sum of at least n matrices $\mathbf{P}_k$ (symmetric, rank-one). If the matrices $\mathbf{P}_k$ are ordered descending according to some norm, a factorization in square roots is equivalent to approximation of **P** by a truncated series:

$$\mathbf{P} = \sum_{k=1}^{\infty} \mathbf{P}_k \approx \sum_{k=1}^{m} \mathbf{P}_k = \mathbf{SS}' \tag{6.6}$$

An eigenvalue decomposition is in fact based on this approach. Rank-one matrices $\mathbf{P}_k$ are obtained from the eigenvectors and are weighted with the corresponding eigenvalue:

$$\mathbf{P} = \mathbf{L}\boldsymbol{\Lambda}\mathbf{L}' = \sum_{k=1}^{n} \lambda_k \left(\mathbf{l}_k \mathbf{l}_k{}'\right) \tag{6.7}$$

where the diagonal matrix $\boldsymbol{\Lambda}$ contains the eigenvalues in descending order and **L** the corresponding eigenvectors. The covariance square root represented in this way is filled with vectors $\sqrt{\lambda_k}\mathbf{l}_k$. The amplitude of the elements in $\sqrt{\lambda_k}\mathbf{l}_k$ decrease with k; including more eigenvalue/vectors introduces more detail.

Relations (6.6) and (6.7) show that adding two low-rank covariance matrices to eachother is equivalent with combination of the two square roots:

$$\mathbf{P}^A + \mathbf{P}^B = \sum_{k=1}^{m_A} \left(\mathbf{s}_k^A \mathbf{s}_k^{A\prime}\right) + \sum_{k=1}^{m_B} \left(\mathbf{s}_k^B \mathbf{s}_k^{B\prime}\right) = \left[\, \mathbf{S}^A \,,\, \mathbf{S}^B \,\right] \left[\, \mathbf{S}^A \,,\, \mathbf{S}^B \,\right]' \tag{6.8}$$

where $[.,.]$ denotes that the new matrix is formed from the columns of the original matrices. If all columns in square roots $\mathbf{S}^A$ and $\mathbf{S}^B$ are linear independent, the rank of $\mathbf{P}^A + \mathbf{P}^B$ is equal to the sum of the ranks of the square roots.

A suitable interpretation of the columns of $\mathbf{S}$ is that it represents a base for how the true state might differ from the mean state. If we accept that the mean $\hat{\mathbf{x}}$ and covariance $\mathbf{S}\mathbf{S}'$ approximates the distribution of the true state, we also accept that $\hat{\mathbf{x}}+\mathbf{S}\mathbf{w}$ is a possible realization for each $\mathbf{w} \sim \mathcal{N}(\mathbf{o},\mathbf{I})$. The low-rank filters described in this chapter use this property to create special ensembles of states to be propagated by the model operator. In the next section it is shown that the minimal variance gain $\mathbf{K}^{MV}$ in (6.2d) is within the subspace spanned by $\mathbf{S}$. The columns of $\mathbf{S}$ define how a forecast of the true state might be analyzed towards the the measurements, and are therefore also referred to as the 'modes' of the filter.

Instead of extracting an ensemble from the columns of the covariance square root, an ensemble of states could be used to build a suitable square root too. The sample covariance of the ensemble $\{\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_m\}$ with sample mean $\overline{\boldsymbol{\xi}}$ is given by:

$$\mathbf{P} \;=\; \frac{1}{m-1}\sum_{j=1}^{m}\left(\boldsymbol{\xi}_j-\overline{\boldsymbol{\xi}}\right)\left(\boldsymbol{\xi}_j-\overline{\boldsymbol{\xi}}\right)' \;=\; \left[\cdots\frac{\boldsymbol{\xi}_j-\overline{\boldsymbol{\xi}}}{\sqrt{m-1}}\cdots\right]\left[\cdots\frac{\boldsymbol{\xi}_j-\overline{\boldsymbol{\xi}}}{\sqrt{m-1}}\cdots\right]' \tag{6.9}$$

The rank of the square root obtained in this way is equal to $m-1$, since the m ensemble members define a subspace of dimension $m-1$.

6.4 Kalman filter in square root form

The Kalman filter equations from §6.2 are easily rewritten in terms of factorized covariance matrices. Apart from the previous described factorization $\mathbf{P}=\mathbf{S}\mathbf{S}'$ for the covariance of the true state, we also introduce factorizations $\mathbf{Q}=\mathbf{T}\mathbf{T}$ and $\mathbf{R}=\mathbf{U}\mathbf{U}'$ for the covariance of the forecast and representation error respectively. Further, a matrix $\boldsymbol{\Psi}'=\mathbf{H}'\mathbf{S}$ is introduced for the mapping of the forecast covariance square root to the observation space.

After (6.2a-6.2b), the forecast of mean and covariance become:

$$\begin{array}{rrcl}
 & \hat{\mathbf{x}}^{f}[k{+}1] & = & \mathbf{A}\,\hat{\mathbf{x}}^{a}[k] \qquad (6.10a)\\
 & (\mathbf{S}^{f}\mathbf{S}^{f\prime})[k{+}1] & = & \mathbf{A}\,(\mathbf{S}^{a}\mathbf{S}^{a\prime})[k]\,\mathbf{A} \;+\; \mathbf{T}\mathbf{T}'[k]\\
\text{or} & \mathbf{S}^{f}[k{+}1] & = & [\,\mathbf{A}\mathbf{S}^{a}[k]\;,\;\mathbf{T}[k]\,] \qquad (6.10b)
\end{array}$$

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

The equations for the analysis of the covariance square root are derived from eq. (6.2d). Analysis with minimal variance gain or arbitrary gain are discussed separately. The analysis equations for a minimal variance gain reduce to:

$$\mathbf{K} = \mathbf{S}^f \boldsymbol{\Psi}(\boldsymbol{\Psi}'\boldsymbol{\Psi} + \mathbf{R})^{-1} \tag{6.11a}$$

$$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f + \mathbf{K}\,(\mathbf{y}^o - \mathbf{H}'\hat{\mathbf{x}}^f) \tag{6.11b}$$

$$\mathbf{S}^a\mathbf{S}^{a\prime} = (\mathbf{I} - \mathbf{K}\mathbf{H}')\,\mathbf{S}^f\mathbf{S}^{f\prime} = \mathbf{S}^f\left[\mathbf{I} - \boldsymbol{\Psi}(\boldsymbol{\Psi}'\boldsymbol{\Psi} + \mathbf{R})^{-1}\boldsymbol{\Psi}'\right]\mathbf{S}^{f\prime} \tag{6.11c}$$

$$\text{or} \qquad \mathbf{S}^a = \mathbf{S}^f\left[\mathbf{I} - \boldsymbol{\Psi}(\boldsymbol{\Psi}'\boldsymbol{\Psi} + \mathbf{R})^{-1}\boldsymbol{\Psi}'\right]^{1/2} \tag{6.11d}$$

Equations (6.11) requires three major operations: solving $\boldsymbol{\Theta}$ from the system $(\boldsymbol{\Psi}'\boldsymbol{\Psi} + \mathbf{R})\boldsymbol{\Theta} = \boldsymbol{\Psi}'$, factorization of $\mathbf{I} - \boldsymbol{\Psi}\boldsymbol{\Theta}$ into $\mathbf{B}\mathbf{B}'$, and the transformation $\mathbf{S}^f\mathbf{B}$. If the number of measurements is limited to 10–100, the cost of the first two operations is minor, and the total coasts of the analysis are completely determined by the transformation of the square root. It is possible to solve the analysis equations (6.11) without solving matrix systems and factorizations, by treating the available measurements as a sequence of uncorrelated scalar measurements. The algorithm for this *repeated scalar update* is described in appendix B. The repeated scalar update is simple to implement since no matrix systems need to be solved, although the later is not necessary a problem since many fast and accurate software libraries are available. For very large numbers of measurements (much more than the number of modes), a repeated scalar update becomes less efficient than the matrix method.

For an analysis with an arbitrary gain matrix, the analysis of the covariance is transformed into:

$$\mathbf{S}^a\mathbf{S}^{a\prime} = (\mathbf{I} - \mathbf{K}\mathbf{H}')\mathbf{S}^f\mathbf{S}^{f\prime}(\mathbf{I} - \mathbf{K}\mathbf{H}')' + \mathbf{K}\mathbf{U}\mathbf{U}'\mathbf{K}' \tag{6.12a}$$

$$\mathbf{S}^a = \left[\,(\mathbf{I} - \mathbf{K}\mathbf{H}')\mathbf{S}^f\,,\ \mathbf{K}\mathbf{U}\,\right] = \left[\,\mathbf{S}^f - \mathbf{K}\boldsymbol{\Psi}'\,,\ \mathbf{K}\mathbf{U}\,\right] \tag{6.12b}$$

Similar as for the forecast error during the forecast stage, the representation error leads to the introduction of new directions in the covariance during the analysis. The new directions reflect that the observation vector used during the analysis contains random errors, and could have had a different value leading to a different analysis with equal probability. The memory requirements for analysis (6.12) could exceed the available capacity, if the gain matrix $\mathbf{K}$ is stored as a separate entity. A more efficient approach is to append the gain matrix to the existing $\mathbf{S}^f$, to replace the first m columns with $\mathbf{S}^f - \mathbf{K}\boldsymbol{\Psi}'$, and finally to replace the last columns with $\mathbf{K}\mathbf{U}$. If the gain matrix is the result of an analysis with local support (*Houtekamer and Mitchell, 2001*), the number of non-zero elements in the gain is limited. Instead of using a gain matrix with state vector shaped columns, a special implementation should be considered now, with the columns defined on a spatial limited grid. With this implementation, it is possible to analyze large numbers of measurements with a factorized filter.

6.5 Examples of factorized filters

A number of filter technique is in use which all exploit the concept of a factorized covariance matrix. The filters are based on the concept that, although the degree of freedom in the state is very large, the errors in the state are described very well by a limited number of directions, typically less than 100. Whether these directions are called singular vectors, modes, or EOF's, the basic equations in all filter implementations remain the same.

6.5.1 RRSQRT filter

The *Reduced Rank SQuare RooT* (RRSQRT) filter was developed for assimilation of water level measurements in a shallow water model. The RRSQRT filter has been applied successfully to mainly hydro-dynamical models (*Verlaan, 1998; Voorrips et al., 1999; Cañizares, 1999*).

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

$$\hat{\mathbf{x}}^f[k{+}1] = \mathbf{A}\,\hat{\mathbf{x}}^a[k] \tag{6.13a}$$

$$\mathbf{S}^f[k{+}1] = [\,\mathbf{A}\,\mathbf{S}^a[k]\;,\;\mathbf{T}[k]\,] \tag{6.13b}$$

$$\boldsymbol{\Psi} = \mathbf{H}'\mathbf{S}^f[k{+}1] \tag{6.13c}$$

$$\mathbf{K} = \mathbf{S}^f[k{+}1]\,\boldsymbol{\Psi}\,[\,\boldsymbol{\Psi}'\boldsymbol{\Psi} + \mathbf{R}[k{+}1]\,]^{-1} \tag{6.13d}$$

$$\hat{\mathbf{x}}^a[k{+}1] = \hat{\mathbf{x}}^f \;+\; \mathbf{K}(\mathbf{y}^o[k{+}1] - \mathbf{H}'\hat{\mathbf{x}}^f[k{+}1]) \tag{6.13e}$$

$$\mathbf{S}^a[k{+}1] = \mathbf{S}^f[k{+}1]\left[\mathbf{I} \;-\; \boldsymbol{\Psi}(\boldsymbol{\Psi}'\boldsymbol{\Psi} + \mathbf{R}[k{+}1])^{-1}\boldsymbol{\Psi}'\right]^{1/2} \tag{6.13f}$$

$$\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}' = \mathbf{S}^a[k{+}1]'\mathbf{S}^a[k{+}1] \tag{6.13g}$$

$$\tilde{\mathbf{S}}^a[k{+}1] = \mathbf{S}^a[k{+}1]\tilde{\mathbf{V}} \tag{6.13h}$$

The algorithm is initialized with an empty covariance square root; new columns are added every time step due to the introduction of system noise (6.13b). As a consequence, the filter will have to spent some time on building an appropriate covariance matrix. If for example the system noise is specified to be in the boundary conditions, the filter has to perform a number time integrations before this uncertainty is propagated through the domain. For each of the m modes stored in $\mathbf{S}$, the forecast of the covariance requires one evaluation of the model $\mathbf{A}$. The analysis steps (6.13d)–(6.13f) are usually implemented in the form of a sequential update for scalar measurements (appendix B), since for the applications where the filter has been implemented the number of measurements is limited.

An important part of the RRSQRT algorithm is the reduction of the covariance square root (6.13g–6.13h). With the introduction of system noise in eq. (6.13b), the number of modes has grown from m to $m+q$, where q is the number of columns in $\mathbf{T}$ (rank of $\mathbf{Q}$). The reduction step reduces the size to m again. Matrix $\tilde{\mathbf{V}}$ contains the eigenvectors of $\mathbf{S}^{a\prime}\mathbf{S}^a$ corresponding with the largest m eigenvalues. The new matrix $\mathbf{S}^a\tilde{\mathbf{V}}$ is an approximation of $\mathbf{S}$ maintaining the largest singular vectors; see §6.9 for the details of the reduction. If the square root $\mathbf{S}$ is never reduced, or reduced to a $n \times n$ matrix if the number of columns exceeds n, then it is straightforward to show that the RRSQRT-filter is equal to the linear Kalman filter. Otherwise, some of the correlation structure stored in the covariance matrix will be lost after reduction. The number of modes required for an accurate estimation of a covariance grows when the stochastic model is more unstable, when the rank of $\mathbf{Q}$ is large, or when the observations are distributed sparse in time. In practice, the maximum number of modes which can be used, is limited by available computing power. If the maximum number of modes is not able to express the covariance correctly, the stochastic model and thus the filter problem should be simplified.

In term of computational costs, the most expensive part of the RRSQRT filter is formed by the propagation of the modes (6.13b), when for each mode the model should be called once. The reduction should therefore reduce the number of modes as far as possible, while still preserving the most important structures in the covariance matrix (see section 6.9). The costs of the reduction are limited if matrix multiplication $\mathbf{S}\tilde{\mathbf{V}}$ is combined with other multiplications with $\mathbf{S}$ (see §6.8).

6.5.2 SEEK and SEIK filter

The *Singular Evolutive Extended Kalman* (SEEK) filter (*Pham et al., 1998*) and the *Singular Evolutive Interpolated Kalman* (SEIK) filter (*Pham, 1996*) are two versions of a factorized Kalman filter based on empiric orthogonal functions (EOF's). The filter has been applied in combination with oceanographic models, see for example (*Verron et al., 1999*). The SEIK filter is different from the SEEK filter in the use of finite difference approximation during the forecast stage rather than a tangent linear model. Since this research a TLM is not considered, the SEIK formulation will be discussed here.

The basic idea of the SEEK/SEIK filter is to make corrections only in a base spanned by a limited number of EOF's. EOF's are just the eigenvectors $\mathbf{l}$ of a sample covariance matrix $\mathbf{P} = \mathbf{L}\boldsymbol{\Lambda}\mathbf{L}'$, computed over a large number of state vectors obtained with the deterministic model. A covariance square root is formed with $\mathbf{S} = \mathbf{L}\boldsymbol{\Lambda}^{1/2}$. The algorithm of the SEIK filter is summarized with the following equations:

$$\begin{aligned}
\hat{\mathbf{x}}^f[k+1] &= \overline{\mathbf{A}(\hat{\mathbf{x}}^a[k] + \mathbf{L}[k]\boldsymbol{\Omega})} && (6.14a)\\
\mathbf{L}[k+1] &= \left(\mathbf{A}(\hat{\mathbf{x}}^a[k] + \mathbf{L}[k]\boldsymbol{\Omega}) \;-\; \hat{\mathbf{x}}^f[k+1]\right)\boldsymbol{\Omega}^{-1} && (6.14b)\\
\boldsymbol{\Psi} &= \mathbf{H}'\mathbf{L}[k+1] && (6.14c)\\
\boldsymbol{\Pi} &= (\mathbf{L}[k+1]'\mathbf{L}[k+1])^{-1}\mathbf{L}'[k+1] && (6.14d)\\
(\boldsymbol{\Lambda}[k+1])^{-1} &= \left[\boldsymbol{\Lambda}[k] + \boldsymbol{\Pi}\mathbf{Q}[k]\boldsymbol{\Pi}'\right]^{-1} + \boldsymbol{\Psi}\mathbf{R}[k+1]^{-1}\boldsymbol{\Psi}' && (6.14e)\\
\mathbf{K} &= \mathbf{L}[k+1]\;\boldsymbol{\Lambda}[k+1]\;\boldsymbol{\Psi}\;\mathbf{R}[k+1]^{-1} && (6.14f)\\
\hat{\mathbf{x}}^a[k+1] &= \hat{\mathbf{x}}^f[k+1] \;+\; \mathbf{K}(\mathbf{y}^o[k+1] - \mathbf{H}'\hat{\mathbf{x}}^f[k+1]) && (6.14g)
\end{aligned}$$

For propagation of the covariance matrix, the SEIK filter forms an ensemble of states from the columns of $\mathbf{L}$, in simplified notation: $\hat{\mathbf{x}}^a + \mathbf{L}\boldsymbol{\Omega}$. The $m \times (m+1)$ matrix $\boldsymbol{\Omega}$ is chosen such that the ensemble has sample mean $\hat{\mathbf{x}}^a$ and covariance $\mathbf{L}\boldsymbol{\Lambda}\mathbf{L}'$; see chapter 7 for details. The forecast of the mean (6.14a) and the covariance factorization (6.14b) are equivalent to the sample statistics of the propagated ensemble. The SEIK filter is therefore different from the regular Kalman filter, because it does not make a separate forecast for the new mean. The use of a sample mean shows more agreement with the ensemble filter (section 6.6), especially since there is also some randomness included in the computed $\boldsymbol{\Omega}$. In the SEEK variant of the filter, the mean and covariance base $\mathbf{L}$ are propagated separately by a tangent linear model $\mathbf{A} = \partial\mathbf{M}/\partial\mathbf{x}$, and this version of the filter is therefore comparable with the Extended Kalman Filter.

After the philosophy that corrections are only made within a base spanned by the columns of $\mathbf{L}$, the SEIK filter performs the analysis of measurements in terms of an update of the matrix $\boldsymbol{\Lambda}$ (eq. (6.14e), actually an update of $\boldsymbol{\Lambda}^{-1}$); the base $\mathbf{L}$ remains the same. System noise is also introduced in the base spanned by $\mathbf{L}$; a projection of $\mathbf{Q}$ onto $\mathbf{L}$ is introduced during the analysis step (6.14e). This procedure is justified by the assumption that the EOF's in $\mathbf{L}$ point in directions amplified by the dynamical model. The benefit of corrections made in other directions will be marginal. The base in which the filter operations are applied is initial orthogonal, but this property will soon be lost if it is propagated by a nonlinear model. Depending on the dynamics applied on the base vectors, some of them will tend to diverge while groups of other vectors tend to converge to a single direction. To avoid numerical instabilities, the vectors will have to be periodically renormalized (*Pham, 1997*), or re-orthogonalized (*Verron et al., 1999*). If a re-orthogonalization is applied after each filter step, the original idea of adjustments within a fixed base is lost, and the algorithm becomes almost identical to the RRSQRT filter.

6.5.3 ESSE: Error Subspace Statistical Estimation

Error Subspace Statistical Estimation is a mathematical framework introduced in (*Lermusiaux and Robinson, 1999a; Lermusiaux and Robinson, 1999b*) for assimilation techniques based on a low-rank covariance matrix. Investigation of the goals and constraints for ocean-atmosphere data assimilation lead to the definition of the filter problem in terms of a low-rank, evolving, and flexible sized *error subspace* (ES) that spans and tracks the scales and processes where the dominant errors occur. Derivations in the framework are based on general formulations, but the current applications end up with error subspaces in terms of singular vectors of the covariance matrix and EOF's. The error subspace is therefore comparable with the square root $\mathbf{S}$ in the RRSQRT filter and the base $\mathbf{L}$ in the SEEK/SEIK filter.

The ESSE system described in (*Lermusiaux and Robinson, 1999a*) covers a large number of different applications: filtering, forecast, smoothing, parameter estimation, etc. Almost all popular techniques for low-rank filtering are available somewhere in the system, to be selected depending on the application. In the basic formulation, the error covariance matrix is approximated by a factorization $\mathbf{P} \approx \mathbf{E}\boldsymbol{\Pi}\mathbf{E}'$ where $\mathbf{E}$ is a base for the error subspace and $\boldsymbol{\Pi}$ contains the eigenvalues according to some norm. The rank of the error subspace is assumed to be flexible and might be reduced to a suitable size using singular value decompositions. Forecasts of the covariance are based on propagation of an ensemble of states, chosen random or more structured out of the statistical distribution defined by the mean and the error subspace. The analysis equations follow either the derivations (6.11d) and (6.12) for the covariance square root, or the ensemble analysis described in section 6.6.

The framework of the ESSE provides a useful overview of techniques available for low-rank filters. Which of the techniques is used in practice depends on the application and considerations about computational costs.

6.6 Ensemble filter

Where the RRSQRT, SEEK/SEIK and ESSE approaches are based on factorization of the covariance matrix, the Ensemble Kalman Filter (ENKF) is based on convergence of large numbers. Both approaches lead to a low-rank approximation of the covariance matrix. The ensemble filter was introduced in (*Evensen, 1994*) for assimilation of data in oceanographic models.

The basic idea behind the ensemble filter is to express the probability function of the state in an ensemble of possible states $\{\xi_1,\ldots,\xi_N\}$. Each ensemble member is assumed to be a single sample out of the distribution of the true state. All ensemble members have the same weight, and operations are performed on each single ensemble member rather than on the complete ensemble itself. Whenever necessary, statistical moments are approximated with sample statistics:

$$\hat{\mathbf{x}} \approx \frac{1}{m}\sum_{j=1}^{m}\boldsymbol{\xi}_j \qquad , \qquad \mathbf{P} \approx \frac{1}{m-1}\sum_{j=1}^{m}(\boldsymbol{\xi}_j-\hat{\mathbf{x}})(\boldsymbol{\xi}_j-\hat{\mathbf{x}})' \qquad , \qquad \ldots \tag{6.15}$$

The sample statistics will always converge to the true values with increasing ensemble size. Convergence is rather slow (order $1/\sqrt{m}$), however, and this the only serious disadvantage of the ensemble filter. Evensen (1996) stated that for practical ensemble sizes of $\mathcal{O}(100)$, the errors in the filter will be dominated by statistical noise, not by closure assumption or unbounded error variations growth as have been observed for the EKF. To remove a part of the statistical noise, (*Houtekamer and Mitchell, 1998*) used a cutoff radius after which correlations are ignored, whenever these are extracted from the ensemble.

An important difference between the pair $(\hat{\mathbf{x}},\mathbf{P})$ of the Kalman or factorized filter and the ensemble statistics (6.15) is that the later are much more connected with eachother. In the traditional Kalman filters, $\hat{\mathbf{x}}$ and $\mathbf{P}$ are processed more or less independent from eachother. The mean $\hat{\mathbf{x}}$ is analyzed using a gain matrix computed from $\mathbf{P}$, but $\mathbf{P}$ is never affected by $\hat{\mathbf{x}}$; the covariance and gain could even be computed off-line.

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

$$\mathbf{P} \;=\; \sum_{k=1}^{m}\mathbf{e}_k\mathbf{e}_k{}' \;=\; \mathbf{E}\mathbf{E}' \qquad , \quad \mathbf{e}_k \;=\; \frac{\boldsymbol{\xi}_k-\bar{\boldsymbol{\xi}}}{\sqrt{m-1}} \tag{6.16}$$

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

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

$$\boldsymbol{\xi}_k^f[k+1] \;=\; \mathbf{M}(\boldsymbol{\xi}_k^a[k]) \;+\; \boldsymbol{\eta}_k[k] \qquad , \qquad \boldsymbol{\eta}_k[k] \sim \mathcal{N}(\mathbf{o},\mathbf{Q}[k]) \tag{6.17}$$

where a sample of the system noise is obtained from a random generator. The ensemble forecast is the same for **M** being linear or non-linear. Whenever measurements are available, each of the ensemble members is analyzed with a linear gain:

$$\xi_j^a[k{+}1] \;=\; \xi_j^f[k{+}1] \;+\; \mathbf{K}(\mathbf{y}^o[k{+}1] + \mathbf{v}_j - \mathbf{H}'\xi^f[k{+}1]) \quad , \quad \mathbf{v}_j \sim \mathcal{N}(\mathbf{o}, \mathbf{R}[k{+}1]) \tag{6.18}$$

The vectors $\mathbf{v}_j$ denote samples of the representation error, drawn from a random generator. With $\mathbf{P}^e$ and $\mathbf{R}^e$ the sample covariances of the vectors ξ_j and $\mathbf{v}_j$ respectively, this analysis scheme leads to an analyzed mean and covariance given by (a bar denotes an ensemble mean):

$$\hat{\mathbf{x}}^a \;=\; \overline{\xi_j^a} \;=\; \overline{\xi_j^f} \;+\; \mathbf{K}(\mathbf{y}^o + \overline{\mathbf{v}_j} - \mathbf{H}'\overline{\xi_j^f}) \tag{6.19a}$$

$$\mathbf{P}^{e,a} \;=\; \overline{(\xi_j^a - \overline{\xi_j^a})(\xi_j^a - \overline{\xi_j^a})'} \tag{6.19b}$$

$$\begin{aligned} &= [\mathbf{I} - \mathbf{K}\mathbf{H}]\,\mathbf{P}^{e,f}\,[\mathbf{I} - \mathbf{K}\mathbf{H}]' \;+\; \mathbf{K}^e\mathbf{R}^e\mathbf{K}' \\ &\quad + \mathcal{O}\left(\overline{(\mathbf{v}_j - \overline{\mathbf{v}_j})(\mathbf{v}_j - \overline{\mathbf{v}_j})} - \mathbf{R}\right) \;+\; \mathcal{O}\left(\overline{(\xi_j^a - \overline{\xi_j^a})(\mathbf{v}_j - \overline{\mathbf{v}_j})}\right) \end{aligned} \tag{6.19c}$$

The last two terms converge to zero with order $1/\sqrt{m}$. If these terms are omitted, the analysis scheme produces what is expected from (6.2d) for analysis of covariance $\mathbf{P}^e$ with an arbitrary gain matrix **K**. The ensemble analysis (6.18) is independent of the gain matrix used. Under the assumption that the probability densities of both state and measurements are close to Gaussian, a gain matrix for the ensemble filter might be formed using the ensemble covariance:

$$\mathbf{K}^e = \mathbf{P}^e\mathbf{H}\;\left[\mathbf{H}'\mathbf{P}^e\mathbf{H} + \mathbf{R}\right]^{-1} \tag{6.20}$$

The ensemble filter is different from the other low-rank filters in not using any kind of orthogonalization. There is no need for rank reduction, since the ensemble size does not grow due to the introduction of system noise or analysis with a general gain matrix. Ensemble members are almost independent from eachother. The only point where an ensemble member might notice the existence of the other members is during the analysis stage, if the gain matrix is computed from ensemble statistics as in (6.20). This is not necessary, however, since a gain might be defined completely independent from the ensemble too. An advantage of the filter acting on ensemble members rather than a covariance matrix is the clear insight in how the filter performs, since tracing values in individual members is less complicated than tracing parts of a covariance matrix.

Note 1. In the original implementation of the ensemble filter (*Evensen, 1994*), the ensemble members were analyzed without random samples $\mathbf{v}_j$ of the representation error in (6.18). The true covariance $\mathrm{E}\left[\,(\mathbf{x}^t - \hat{\mathbf{x}})(\mathbf{x}^t - \hat{\mathbf{x}})'\,\right]$ of the ensemble mean was shown to be analyzed correctly, if the ensemble happens to represent the true covariance exactly. In (*Burgers et al., 1998*) it was noticed that this procedure would lead to an analyzed ensemble covariance $\mathbf{P}^{e,a}$ in (6.19) without an analog for the term $\mathbf{K}^e\mathbf{R}^e\mathbf{K}^{e\prime}$, however. This term is lost because each ensemble member is analyzed with the same observation vector $\mathbf{y}^o$, while this is only one specific realization from a broad range of possible observations. Without the random representation errors, the spread in the ensemble members has become to small, and will hamper the computation of covariances at future time steps. Including random representation errors does not influence the true covariance of the ensemble mean.

Note 2. It is possible to rewrite the analysis scheme of the ensemble filter in terms of adding additional members to the ensemble, analog to equation (6.12) for the factorized filter. The $\tilde{m}$ new members are formed from the forecast ensemble mean, analyzed with $\tilde{m}$ different random representation errors; the first m members of the new ensemble are formed from analysis with the average representation error:

$$\left\{\ldots \boldsymbol{\xi}_j^f + \mathbf{K}(\mathbf{y}^o + \overline{\mathbf{v}_l} - \mathbf{H}\boldsymbol{\xi}^f)\ldots, \ldots \overline{\boldsymbol{\xi}_j^f} + \mathbf{K}(\mathbf{y}^o + \mathbf{v}_l - \mathbf{H}\overline{\boldsymbol{\xi}_j^f})\ldots\right\}$$
$$j = 1, \ldots, m \quad , \quad l = 1, \ldots \tag{6.21a}$$

The new ensemble has sample mean and covariance equal to (6.19). The practical value of eq. (6.21a) is limited; it reminds that including random observation errors (note 1) introduces new directions in the ensemble. A continuous growing ensemble is not practical, however, and a mechanism for reducing the ensemble should be included.

6.7 POEnK Filter

A new direction in implementation of low-rank filters is the use of two filters next to eachother. The combination should compensate for errors made in one or both of the individual filters.

In (*Houtekamer and Mitchell, 1998*), a *Double Ensemble Kalman Filter* (DENKF) was proposed in order to prevent 'inbreeding' of the Ensemble Filter: in the analysis step, the ensemble is updated with a gain calculated from the ensemble itself. This situation might lead to an underestimation of the covariance. In the DENKF, this effect was limited by using two ensembles, and analyzing each of them with a gain matrix calculated from the other one. In (*van Leeuwen, 1998*) the issue of inbreeding has been discussed in more detail. It was shown that the DENKF also suffered from this effect, but on a smaller scale. An important part of the inbreeding was shown to arise from neglecting higher order moments in the gain matrix. It was argued that using a single ensemble of double size instead of two small ensembles next to eachother will lead to an overall better performance for the same computational costs, since covariances are calculated more accurate.

The *Partially Orthogonal Ensemble Kalman Filter* (POENKF) proposed in (*Heemink et al., 2001*) runs a RRSQRT filter next to an ENKF. The basic idea is to let the RRSQRT part compute the bulk of the covariance structure, described in the first modes. The ENKF part should account for the truncation error, by introducing directions in the covariance matrix that have been lost during the reduction. This procedure incorporates the advantages of both filter types, and accounts for their major disadvantages. Ensemble filters suffer from a lack of convergence; many ensembles are required before sample mean and correlations are stable. An ensemble filter is able to estimate and maintain any correlation introduced by the stochastic model, however. The reverse holds for the RRSQRT filter: a few modes are sufficient to describe the main part of the covariance structure, but some of the correlation structure is lost during the reduction.

The gain matrix used in the POENK filter is computed with a covariance matrix $\mathbf{P}^{poen}$ formed from the covariances in the two underlying filters. The bulk of $\mathbf{P}^{poen}$ is obtained from covariance $\mathbf{P}^{rr}$ of the RRSQRT part, and the remainder from a projection of the ensemble covariance on the orthogonal complement of $\mathbf{P}^{rr}$:

$$\mathbf{P}^{poen} = \mathbf{P}^{rr} + \mathbf{P}^{en\perp rr} \tag{6.22a}$$

$$\mathbf{K}^{poen} = \mathbf{P}^{poen}\,\mathbf{H}\,(\,\mathbf{H}'\,\mathbf{P}^{poen}\,\mathbf{H} + \mathbf{R}\,)^{-1} \tag{6.22b}$$

Gain (6.22b) is used to analyze both $\hat{\mathbf{x}}^f$ and $\mathbf{S}^f$ of the RRSQRT part, and the ensemble members in the ENKF part. Since the gain has not the form of the minimal variance gain, the general analysis scheme (6.12) should be used for the RRSQRT part. The new gain matrix acts as a variance reductor for the ensemble, since the ensemble mean is less sensitive to fluctuations due to small ensemble sizes (*Heemink and Segers, 2000*). The gain matrix $\mathbf{K}^{poen}$ is efficiently computed using the square root $\mathbf{S}^{poen}$ of $\mathbf{P}^{poen}$. Expressed in the square root $\mathbf{S}$ from the RRSQRT part and the 'ensemble square root' $\mathbf{E}$ defined in (6.16), the square root $\mathbf{S}^{poen}$ is computed from:

$$\boldsymbol{\Pi}^{\|} = \mathbf{S}(\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}' \tag{6.23a}$$

$$\mathbf{E}^{\|} = \boldsymbol{\Pi}^{\|}\,\mathbf{E} \tag{6.23b}$$

$$\mathbf{E}^{\perp} = \mathbf{E} - \mathbf{E}^{\|} \tag{6.23c}$$

$$\mathbf{S}^{poen} = \left[\,\mathbf{S}^{rr}\,,\,\mathbf{E}^{\perp}\,\right] \tag{6.23d}$$

where $\boldsymbol{\Pi}^{\|}$ is the projection matrix on the subspace spanned by the columns of $\mathbf{S}$. Thus, the covariance square root of the POENK filter is obtained from adding a number of columns to $\mathbf{S}^{rr}$ equal to the ensemble size. Storage of $\mathbf{S}^{poen}$ as a separate variable is not very efficient due to the duplication of the memory requirements, however. A more efficient approach is to treat $\mathbf{E}^{\|}$ as a separate variable, and then to store the gain computed from $\mathbf{S}^{rr}$, $\mathbf{E}$, and $\mathbf{E}^{\|}$ temporarily behind $\mathbf{S}^{rr}$ following the remarks at the end of section 6.4.

6.8 General formulation of low-rank filter

Although the previously described filter algorithms are sometimes quite different in philosophy and detail, the actual implementations turn out to be very similar. In this section an overview is given of the basic matrix/vector operations performed during a filter step. All operations are defined in terms of the mean/covariance-square-root pair $(\hat{\mathbf{x}},\mathbf{S})$, which could be interpreted as an ensemble too.

initialization

The algorithm starts with an initial pair $(\hat{\mathbf{x}}^a[k],\mathbf{S}^a[k])$ which is the best estimate for mean and covariance of the true state at $t[k]$.

formation of forecast ensemble

An ensemble of state vectors is formed and stored in a matrix:

$$[\ldots,\boldsymbol{\xi}_j[k],\ldots] \;=\; \hat{\mathbf{x}}^a[k] \;+\; \mathbf{S}^a[k]\,\boldsymbol{\Omega} \tag{6.24}$$

The matrix $\boldsymbol{\Omega}$ might take many shapes (square, rectangular) and contain special parts such as zero columns or diagonals. A detailed discussion of the contents of $\boldsymbol{\Omega}$ is left for chapter 7 about nonlinear methods; see especially table 7.2. The columns of matrix $\boldsymbol{\Omega}$ define linear combinations of columns of $\mathbf{S}^a[k]$, such that adding the linear combination to $\hat{\mathbf{x}}^a$ provides a specific example of a state vector.

propagation of ensemble

Each ensemble member is propagated by the model:

$$\boldsymbol{\xi}_j[k+1] \;=\; \mathbf{A}[k]\,\boldsymbol{\xi}_j[k] \;+\; \boldsymbol{\eta}_j[k] \tag{6.25}$$

The noise vectors $\boldsymbol{\eta}_j$ are samples of system noise: zero or unity for RRSQRT, always zero for SEIK, random for ESSE and ENKF. See again chapter 7.

reconstruction of mean/covariance-square-root

A new mean/covariance-square-root pair $(\hat{\mathbf{x}}^f[k+1],\mathbf{S}^f[k+1])$ is formed from the propagated ensemble, for example:

$$\hat{\mathbf{x}}^f[k+1] \;=\; \overline{\boldsymbol{\xi}_j[k+1]} \tag{6.26a}$$

$$\mathbf{S}^f[k+1] \;=\; \big(\,[\ldots,\boldsymbol{\xi}[k+1],\ldots] \;-\; \hat{\mathbf{x}}^f[k+1]\,\big)\,\boldsymbol{\Omega}^{-1} \tag{6.26b}$$

Matrix $\boldsymbol{\Omega}^{-1}$ is a general inversion operator, depending on the particular choice made for $\boldsymbol{\Omega}$ in (6.24).

analysis

The equations for analysis with minimal variance gain take the form:

$$\hat{\mathbf{x}}^a[k+1] \;=\; \hat{\mathbf{x}}^f[k+1] \;+\; \mathbf{S}^f[k+1]\,\mathbf{a} \tag{6.27a}$$

$$\mathbf{S}^a[k+1] \;=\; \mathbf{S}^f[k+1]\,\mathbf{B} \tag{6.27b}$$

See table 6.1 for the exact form of **a** and **B** if $(\hat{\mathbf{x}}^f,\mathbf{S}^f)$ represents a factorization or an ensemble, and for analysis with arbitrary gain.

rank reduction

Whenever necessary, the rank of the covariance matrix is reduced by approximation of **S** with the largest singular values:

$$\mathbf{S}'\mathbf{S} \;=\; \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}' \tag{6.28a}$$

$$\tilde{\mathbf{S}} \;=\; \mathbf{S}\tilde{\mathbf{V}} \tag{6.28b}$$

In here is $\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}'$ the eigenvalue decomposition of $\mathbf{S}'\mathbf{S}$, and contains $\tilde{\mathbf{V}}$ the eigenvectors corresponding to the largest eigenvalues. See section 6.9 for details of the reduction mechanism.

<table>
<tr><th></th><th>factorized filter</th><th>ensemble filter</th></tr>
<tr><td rowspan="4"></td><td></td><td>$\mathbf{v}_j \sim N(\mathbf{o},\mathbf{R})\ ,\ {\scriptstyle j=1,\dots,m}$</td></tr>
<tr><td>$\overline{\mathbf{v}} = \mathbf{o}$</td><td>$\overline{\mathbf{v}} = \overline{\mathbf{v}_j}$</td></tr>
<tr><td>$\mathbf{z}_j = \mathbf{o}$</td><td>$\mathbf{z}_j = (\mathbf{v}_j - \overline{\mathbf{v}})/\sqrt{m-1}$</td></tr>
<tr><td colspan="2">$\mathbf{d} = \mathbf{y}^o + \overline{\mathbf{v}} - \mathbf{H}'\hat{\mathbf{x}}^f$</td></tr>
<tr><td rowspan="5">minimal variance gain gain</td><td colspan="2">$\boldsymbol{\Theta} = (\boldsymbol{\Psi}'\boldsymbol{\Psi} + \mathbf{R})^{-1}\boldsymbol{\Psi}$</td></tr>
<tr><td colspan="2">$\mathbf{a} = \boldsymbol{\Theta}\mathbf{d}$</td></tr>
<tr><td>$\mathbf{B}\mathbf{B}' = \mathbf{I}\ -\ \boldsymbol{\Theta}\boldsymbol{\Psi}'$</td><td>$\mathbf{B} = \mathbf{I}\ +\ \boldsymbol{\Theta}(\mathbf{Z}' - \boldsymbol{\Psi}')$</td></tr>
<tr><td colspan="2">$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f\ +\ \mathbf{S}^f\mathbf{a}$</td></tr>
<tr><td colspan="2">$\mathbf{S}^a = \mathbf{S}^f\mathbf{B}$</td></tr>
<tr><td rowspan="3">arbitrary gain $\mathbf{K}$</td><td colspan="2">$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f + \mathbf{K}\mathbf{d}$</td></tr>
<tr><td colspan="2">$\mathbf{D} = \mathbf{Z}' - \boldsymbol{\Psi}'$</td></tr>
<tr><td>$\mathbf{S}^a = \left[\mathbf{S}^f + \mathbf{K}\mathbf{D}\ ,\ \mathbf{K}\mathbf{U}\right]$</td><td>$\mathbf{S}^a = \left[\mathbf{S}^f + \mathbf{K}\mathbf{D}\right]$</td></tr>
</table>

Table 6.1: *Summary of analysis equations of form (6.27) in use for low-rank filters. The pair* $(\hat{\mathbf{x}},\mathbf{S})$ *denotes either a mean/covariance-square-root for a factorized filter or the sample mean/deviations for an ensemble filter; these are analyzed with a minimal variance or an arbitrary gain. The analysis equations for the* SEEK/SEIK *filter in (6.14) are originally expressed in terms of a matrix* $\boldsymbol{\Lambda}$*, but are equivalent to analysis of a factorized filter with a minimal variance gain, and are therefore not discussed as a special case.*

The computational most expensive operations are the propagation of the ensemble in (6.25), the various transformations of the covariance square-root in (6.24), (6.26b), (6.27b), and (6.28b), and the eigenvalue decomposition in (6.28a). All other operations are of minor costs, although the operations to form the analysis matrix **B** might become expensive if many measurements are to be analyzed (order 10^3).

The transformations of **S** are not all of similar costs; the matrices $\boldsymbol{\Omega}$ for the forecast are often sparse for example. The analysis with optimal gain and the rank reduction require transformation with a full matrix, however. Transformation of the $n \times m$ matrix **S** with the $m \times \bar{m}$ matrix **B** requires $2nm\bar{m}$ flops (p. 187) and the costs of this operation increase therefore quadratic with the number of columns in **S**. There is no need to apply each transformation at once, however. That is, the covariance square root could be defined as the pair $(\mathbf{S}, \mathbf{B})$ rather than a stand alone **S**, such that filter operations are applied **B** rather than **S**. This idea is for example used in the SEEK/SEIK filter, where all operations are performed on $\boldsymbol{\Lambda}$, an equivalent of $\mathbf{BB}'$. The only moment that the transformation really needs to be carried out is during the formation of the forecast ensemble. Collecting the transformations for analysis, reduction, and forecast leads to the following operation replacing (6.24):

$$[\ldots, \boldsymbol{\xi}_j[k], \ldots] \;=\; \hat{\mathbf{x}}^a[k] \;+\; \mathbf{S}^a[k]\,(\mathbf{B}\tilde{\mathbf{V}}\boldsymbol{\Omega}) \tag{6.29}$$

The costs of the eigenvalue decomposition (6.28a) are associated with computation of $\mathbf{S}'\mathbf{S}$ (nm^2 flops) and the actual eigenvalue decomposition ($\mathcal{O}\left(m^3\right)$; for example the symmetric QR algorithm (*Golub and van Loan, 1996, §8.3*) requires about $4/3\; m^3$ flops). Thus, the total reduction sequence of $\mathbf{S}'\mathbf{S}$, $\mathbf{V}\boldsymbol{\Lambda}\mathbf{V} = \mathbf{S}'\mathbf{S}'$, and $\mathbf{S}\tilde{\mathbf{V}}$ requires about $nm^2 + \mathcal{O}\left(m^3\right) + 2nm\tilde{m}$ flops. In typical applications, the number of elements in the state is far beyond the number of modes ($n \gg m$), and the costs of a reduction are dominated by computation of $\mathbf{S}'\mathbf{S}$ and the transformation, both $\mathcal{O}\left(nm^2\right)$ flops. If the transformation $\mathbf{S}\tilde{\mathbf{V}}$ is combined with other transformations through (6.29), the only additional costs of a reduction is the computation of $\mathbf{S}'\mathbf{S}$. Thus, the reduction step becomes an expensive part of the filter if the number of flops required for the dot products in $\mathbf{S}'\mathbf{S}$ and/or the linear combinations in $\mathbf{S}\tilde{\mathbf{V}}$ are not negligible in comparison with the operation $\mathbf{Ax}$. An example of this situation is described in (*Cañizares, 1999*), where a RRSQRT and ensemble filter are applied in combination with a simple hydro-dynamical model. The reduction part of the RRSQRT filter turned out to cost about $0.1\; T_{eval}\; m^2$ flops, where T_{eval} denotes the computation time for a single model integration $\mathbf{Ax}$. The total costs of the filter are now about $T_{eval}(m + 0.1m^2)$. The reduction part dominates the filter; for 100 modes in the covariance matrix, the RRSQRT filter has become 10 times as expensive as a comparable ensemble filter. For the chemistry models applied in this research, this situation does not occur; computation of a single concentration in a single grid cell one time step ahead requires $\mathcal{O}\left(10^3\right)$ flops rather than $\mathcal{O}\left(10\right)$, and the model propagation is much more expensive than the linear algebra operations applied in the filter.

6.9 Modification of reduction algorithm

The reduction algorithm present in the RRSQRT filter and POENKF, and re-orthogonalization in the SEEK/SEIK or ESSE filter are here discussed in detail [1].

The target of the reduction algorithm is to replace the $n \times m$ covariance square root $\mathbf{S}$ by a matrix $\tilde{\mathbf{S}}$ with less columns, while the structure of the covariance matrix $\mathbf{P} = \mathbf{S}\mathbf{S}'$ is maintained as much as possible in $\tilde{\mathbf{P}} = \tilde{\mathbf{S}}\tilde{\mathbf{S}}'$. This is achieved by building $\tilde{\mathbf{P}}$ from the largest eigenvectors of $\mathbf{P}$. With $\mathbf{V}\Lambda\mathbf{V}'$ the eigenvalue decomposition of $\mathbf{S}'\mathbf{S}$ (stored in descending order), the eigenvalue decomposition of the original covariance matrix is given by:

$$\mathbf{P} \,=\, \mathbf{S}\mathbf{S}' \,=\, \left(\mathbf{S}\mathbf{V}\Lambda^{-1/2}\right)\Lambda\left(\mathbf{S}\mathbf{V}\Lambda^{-1/2}\right)' \tag{6.30}$$

That the eigenvalue decomposition of $\mathbf{P}$ is given by (6.30) is shown by multiplication with the eigenvector-matrix $(\mathbf{S}\mathbf{V}\Lambda^{-1/2})$ and using that $\mathbf{V}$ is orthogonal, since $\mathbf{S}'\mathbf{S}$ is symmetric. Following eq. (6.7), an approximation by the largest $\tilde{m}$ eigenvalues can be written as a truncated series:

$$\mathbf{P} \,=\, \sum_{j=1}^{\tilde{m}} \lambda_i \left(\frac{(\mathbf{S}\mathbf{v}_j)}{\sqrt{\lambda_j}} \, \frac{(\mathbf{S}\mathbf{v}_j)'}{\sqrt{\lambda_j}} \right) \,=\, \left(\mathbf{S}\tilde{\mathbf{V}}\right)\left(\mathbf{S}\tilde{\mathbf{V}}\right)' \,=\, \tilde{\mathbf{S}}\,\tilde{\mathbf{S}}' \tag{6.31}$$

where $\mathbf{v_j}$ denotes the j-th column of $\mathbf{V}$, and $\tilde{\mathbf{V}}$ the matrix with the first $\tilde{m}$ columns. Each of the $\tilde{m}$ matrices in parentheses is a rank-one covariance matrix on its own, with unit weight when measured with the l_2-norm, since the eigenvalue decomposition of $\mathbf{S}'\mathbf{S}$ gives $\left\|\mathbf{S}\mathbf{v}_j\right\|_2^{\,2} = \lambda_j$. The $m - \tilde{m}$ rank-one covariance matrices with the smallest corresponding weights λ_i are neglected when the rank of $\mathbf{P}$ is reduced to $\tilde{m}$.

For successful application of the reduction or re-orthogonalization, the amount of reduction should be balanced between as much as possible to limit the costs of future model integrations, and as minimal as possible for limited loss of covariance structure. The reduction algorithm has been modified to limit the loss of structure and maximize the reduction by focusing on some specific characteristics of a filter around an atmospheric chemistry model.

6.9.1 Unit-invariant reduction

For a filter around an atmospheric chemistry model such as LOTOS, the state $\mathbf{x}$ consists in general of the concentrations of several chemical components in each of the grid cells. Each of those components is expressed in a typical unit, for example ppb, kg/m^3, or mol/l. Which unit is used is rather arbitrary, and is often subject of change. If an application requires that concentrations should be expressed in a different unit, one simply multiplies each element of the state with an appropriate factor.

A problem of the existing large variety in units, is that the reduction is not invariant for a change of units. In mathematical terms, a change of units is a transformation of the state

---

[1] Revised from *A modified rrsqrt-filter for assimilating data in atmospheric chemistry models* by A.J. Segers and A.W. Heemink and M. Verlaan and M. van Loon. *Environmental Modeling and Software*, 15(6–7):663–671,2000.

from $\mathbf{x}$ to $\boldsymbol{\Pi}\mathbf{x}$, where $\boldsymbol{\Pi}$ is a diagonal matrix. The transformed covariance matrix is equal to $\boldsymbol{\Pi}\mathbf{P}\boldsymbol{\Pi}'$, whose eigenvalues will be different from the eigenvalues of $\mathbf{P}$ (unless $\boldsymbol{\Pi}$ is unitary, which is the same as identity if $\boldsymbol{\Pi}$ is diagonal). Unfortunately, the covariance square root according to these eigenvalues. The algorithm has now been modified, to make it invariant for the units in which concentrations are expressed.

Define $\sigma(\mathbf{P})$ to be the square root of the diagonal matrix with the same main diagonal as $\mathbf{P}$. It is straight forward to show that for each covariance matrix $\mathbf{P}$, the eigenvalues of the matrix

$$\sigma(\mathbf{P})^{-1}\,\mathbf{P}\,\sigma(\mathbf{P})^{-1} \tag{6.32}$$

are invariant for a state transformation with a diagonal matrix. If the reduction is applied to (6.32) instead of $\mathbf{P}$, the result is invariant for the units chosen. If expressed in elements of the covariance square root $\mathbf{S}$, the double transformation of $\mathbf{P} = \mathbf{S}\mathbf{S}'$ with $\sigma(\mathbf{P})^{-1}$ is equal to division of each square root element s_{ij} by $\sum_k s_{ik}^2$. In the rare case that all elements in a row are equal to zero, the elements remain zero; a row filled with zeros would not have impact in the filter anyway.

With the proposed transformation, one does in fact not reduce the *covariance* matrix $\mathbf{P}$, but the *correlation* matrix $\boldsymbol{\Gamma}(\mathbf{x},\mathbf{x})$ with elements:

$$\gamma_{ij}(\mathbf{x},\mathbf{x}) \;=\; \frac{\mathrm{E}[\,(x_i-\mathrm{E}[\,x_i\,])(x_j-\mathrm{E}[\,x_j\,])\,]}{\mathrm{E}[\,(x_i-\mathrm{E}[\,x_i\,])^2\,]^{1/2}\,\mathrm{E}[\,(x_j-\mathrm{E}[\,x_j\,])^2\,]^{1/2}} \tag{6.33}$$

The reduction algorithm collects the largest elements of a matrix in the first modes, and will now collect the largest correlations rather than the largest covariances. In fact, this is a more natural thing to do in the framework of a Kalman filter. The analysis step updates all elements of the state given an observation, because all elements are correlated with the observation through the matrix $\mathbf{P}$. The correlations described within $\mathbf{P}$ are therefore just the structure one likes to maintain.

6.9.2 Amplification of important correlations

Although (6.32) makes the reduction invariant for a change in units, the eigenvalues of the correlation matrix (transformed covariance matrix) can still be influenced if the state is additionally transformed with a (diagonal) matrix with unit-less elements. This fact can be exploited in favor of the filter performance. The idea is to construct an additional transformation such that the elements in the state are weighted according to their relevance for the filter. For example, in a system with an atmospheric chemistry model and observations of ozone, the correlations between ozone and nitrogen oxides near the observation sites are probably more important for the filter than correlations between methane and carbon monoxide in remote area. At the moment that a reduction is applied, one cannot say which correlations will become important for the filter in future (this is a general disadvantage of a filter, which is blind for the future by definition). However, one can always give an indication of which correlations will *probably* become important.

Therefore, let

$$\bar{y} \;=\; \bar{\mathbf{h}}'\mathbf{x} \;+\; \bar{v} \tag{6.34}$$

denote a scalar observation of a state $\mathbf{x}$, where $\bar{v}$ denotes a representation error with variance $\bar{r}^2$. Observation (6.34) should observe those elements in the state which probably become important in future. The (unit-less) correlation of an element in the state with $\bar{y}$ is easily calculated and expressed in elements of the covariance square root:

$$\begin{aligned}\gamma_i(\mathbf{x},\bar{y}) &= \frac{\mathrm{E}[\,(x_i-\mathrm{E}[\,x_i\,])(\bar{y}-\mathrm{E}[\,\bar{y}\,])\,]}{\mathrm{E}[\,(x_i-\mathrm{E}[\,x_i\,])^2\,]^{1/2}\;\mathrm{E}[\,(\bar{y}-\mathrm{E}[\,\bar{y}\,])^2\,]^{1/2}} \\ &= \frac{\sum_k s_{ik}\,(\bar{\psi})_k}{\sqrt{\sum_k\, s_{ik}{}^2}\;\sqrt{\bar{\psi}'\bar{\psi}+\bar{r}^2}} \end{aligned} \tag{6.35a}$$

where the row vector $\bar{\psi}'$ denotes $\bar{\mathbf{h}}'\mathbf{S}$, $(.)_k$ the k-th element of a vector, and s_{ik} a single element of $\mathbf{S}$. Let $\mathbf{C}(\mathbf{x},\bar{y})$ denote the diagonal matrix with diagonal elements $\gamma_i(\mathbf{x},\bar{y})$. A state transformation with $\mathbf{C}(\mathbf{x},\bar{y})$ will amplify those elements which are highly correlated with an element which is assumed to be important for the filter. Therefore, the corresponding correlations in the covariance matrix are amplified too, and the reduction algorithm will collect them in the first modes.

6.9.3 Summary of transformations

The extension of the reduction algorithm with transformations adds three extra steps to the reduction algorithm:

1 computation of transformation matrices $\boldsymbol{\sigma}(\mathbf{P})$ for invariance to a change in units, and optional $\mathbf{C}(\mathbf{x},\bar{y})$ for additional amplification of correlations;

2 transformation: $\mathbf{S}^{tr} \;=\; \left(\mathbf{C}(\mathbf{x},\bar{y})\;\boldsymbol{\sigma}(\mathbf{S}\mathbf{S}')^{-1}\right)\;\mathbf{S}$

regular reduction: $\tilde{\mathbf{S}}^{tr} \;=\; \mathbf{S}^{tr}\;\tilde{\mathbf{V}}^{tr}$

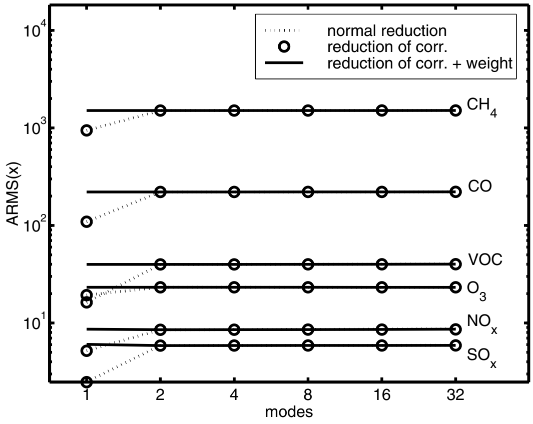
3 inverse transformation: $\tilde{\mathbf{S}} \;=\; \left(\mathbf{C}(\mathbf{x},\bar{y})\;\boldsymbol{\sigma}(\mathbf{S}\mathbf{S}')^{-1}\right)^{-1}\;\tilde{\mathbf{S}}^{tr}$

The additional operations are not very expensive in terms of computation time. Both $\boldsymbol{\sigma}(\mathbf{P})$ and $\mathbf{C}(\mathbf{x},\bar{y})$ consist of the main diagonal only: the diagonal elements for $\boldsymbol{\sigma}(\mathbf{P})$ are simply computed from the rows of $\mathbf{S}$ (see also §8.4.2 and §8.6.2), and the diagonal of $\mathbf{C}(\mathbf{x},\bar{y})$ can be calculated very fast if the number of non-zero elements in $\bar{\mathbf{h}}'$ is small. The major part of the additional computation time is spent on the actual transformations of $\mathbf{S}$ and $\tilde{\mathbf{S}}$; in comparison with the calculation of $\mathbf{S}\tilde{\mathbf{V}}$, this is only a minor part of the total costs of the reduction algorithm, however.

6.9.4 Experiments and results

The impact of modifications to the reduction algorithm have been tested for a RRSQRT filter around the LOTOS model. The experimental setup was similar to the one described in §4.3.1: a horizontal grid of 12 × 12 cells, uncertain emissions of $\mathrm{NO_x}$ and VOC, and CO, and simulated ground measurements in 5 different sites. The filter was first applied with the default reduction, then using a reduction of the correlation matrix rather than the

Figure 6.1: *Average* RMS *concentrations of filter mean during filter period of 24 hours, as a function of the number of modes. The filter used either a default reduction step, or a reduction of the correlation matrix, or a reduction of the correlation matrix with in addition amplification of ozone correlations.*



covariance matrix (transformation (6.32)), and finally using an additional weight as in eq. (6.35). Because the filter assimilated ozone measurements, the correlations with ozone were assumed to be the most important; therefore, the observation function $\bar{\mathbf{h}}$ from eq. (6.34) was set to observe the average ozone concentration over the whole grid. Each hour, the number of modes was reduced to either 1, 2, 4, 8, 16 or 32 modes; the modifications to the reduction mechanism are expected to lead to changed convergence for growing numbers of modes.

For each filter-run, the average root-mean-square of the mean state over 24-hour was calculated according to:

$$\mathrm{ARMS}(\hat{c}_s) = \frac{1}{T}\sum_{k=1}^{T}\sqrt{\frac{1}{n_{cell}}\sum_{x,y,z}\hat{c}_s(x,y,z,t_k)^2} \tag{6.36}$$

where $\hat{c}_s$ denotes the computed mean concentration of a component s, or a sum of concentrations if s denotes a group of components (NO_x or VOC). Figure 6.1 shows the ARMS values for the six main chemical components of the state. For all components, the convergence of the mean concentration is very fast: expressing the covariance matrix in only 2 modes is enough for convergence. This low amount of modes can be explained from the rather simple setup of the experiment: the noise input had only 5 elements, and each of them act on the emissions without any spatial differences. The results show that there is no significant impact on the mean state when reducing the correlation matrix instead of the covariance matrix. If additional weight is assigned to correlations with ozone, reduction to only one mode seems to be enough for convergence, however.

The origin for the impact of the 'weighted' reduction is found in the convergence of the covariance matrix. The average RMS of a components standard deviation has been calculated similar to (6.36), with mean $\hat{c}_s$ replaced by standard deviation $\sigma\{c_s\}$. The value of $\mathrm{ARMS}(\sigma\{c_s\})$ quantifies the amount of covariance structure preserved by the reduction.

The results in figure 6.2 show that when the reduction is applied to the correlation matrix rather than to the covariance matrix, the structures preserved in the first modes are no longer influenced by a components mean value. Components with accidental small mean values profit from the new reduction, because correlations in which they are involved are

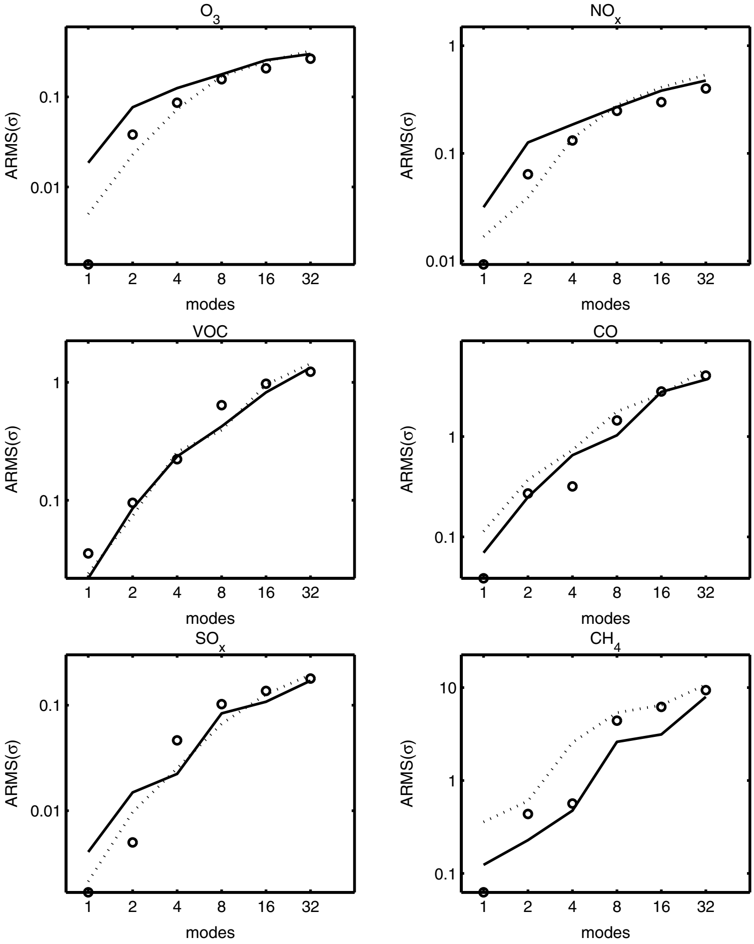


Figure 6.2: *Amount of covariance structure preserved by the reduction, as a function of the number of modes to which is reduced. Dotted lines (···) denote default reduction, bullets (o) denote reduction of the correlation matrix, and a solid line (—) denotes additional amplification of ozone correlations.*

not smaller than other correlations, while their covariances are. Especially VOC (a group of 9 most small valued components) seems to profit from the new reduction; the amount of covariance structure preserved in the first mode increases, while the amount decreases for all other groups.

If additional weight is assigned to correlations involving ozone, the amount of preserved ozone variances increases significant. With reduction to only one mode, about a factor 10 as much is preserved in comparison with reduction without additional weight. This explains the impact of the weighted reduction on the convergence of the mean state: the filter assimilates ozone measurements, and now that variances in ozone are collected more efficient, a single mode is enough for convergence. The results show that also the variances of nitrogen oxides are preserved much better in the first modes, due to their tight chemical correlation with ozone. A component which hardly correlates with ozone is methane, and the effect of the weighted reduction is clear: methane variances are preserved much worse than before, and only reach the unweighted level if the covariance matrix is expressed in a large amount of modes.

6.10 Comparison of filter techniques

The performance of three types of low-rank filters (RRSQRT, ENKF, and POENKF) was tested during a filter experiment with simulated data. The SEEK/SEIK filters were not considered explicitly, since the stochastic model for these methods is based on EOF analysis of the state. The stochastic model used in this research is based on uncertain parameters; for such a stochastic model, the SEEK/SEIK approach hardly differs from a RRSQRT filter. The forecast step of the SEIK is incorporated in the RRSQRT filter, however; in chapter 7 it is shown that this forecast provides most accurate results for lowest costs. The reduction step in RRSQRT and POENKF is implemented following §6.9.

The model area was limited to 24×24 grid cells covering the British island and north-west Europe (figure 6.3). Three densely populated and industrialized area were selected from which 40% of the NO_x and 30% of the VOC emissions are released. In each of these three area, the emissions of NO_x and VOC have been defined stochastic with standard deviations of 50% and time correlation parameter of 12 hours. A three days time periode from august 5–7 1997 was selected with in general eastern wind. Due to these uncertain emissions, three plumes of uncertain concentrations arise downwind from the emission areas (figure 6.3).

A set of simulated 'true' concentration patterns was produced from a run with the stochastic model using random noise input. The difference between the deterministic and 'true' ozone concentrations at august 7, 15:00, is plotted in the right panel of figure 6.4. The differences show that the deterministic model underestimates the ozone concentrations over the Irish Sea (up to -24 ppb) and the North Sea between England and The Netherlands (up to -8.5 ppb), where due to the lack of deposition a difference in ozone concentrations is maintained much longer than over land. Ozone concentrations in the plume from London are overestimated, with a maximum of 19 ppb.

Five measurement sites were selected to filter the uncertain emission flow (figure 6.3). For each site, a set of ozone measurements was generated including a simulated random error with standard deviation of 0.5 ppb. Given the locations of the sites, a filter is ex-

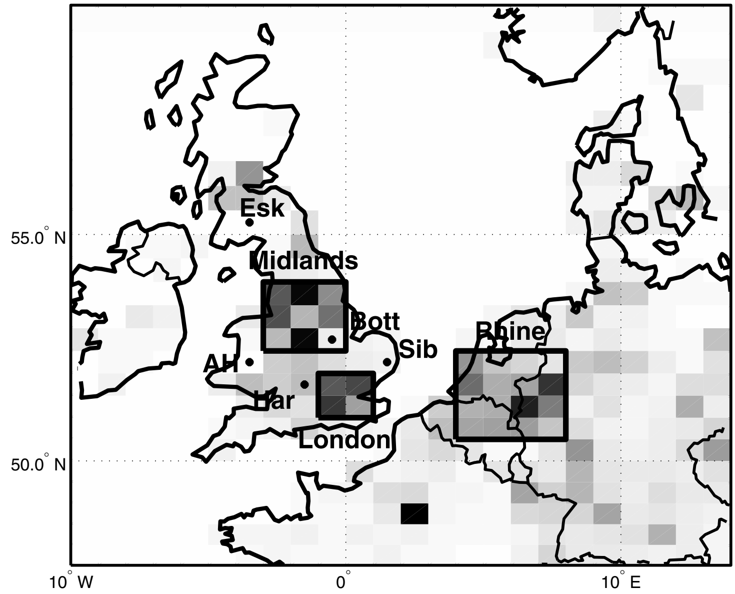


Figure 6.3: *Model domain for experiments with different low-rank filters. The solid rectangles surround the industrialized area's with the largest emissions. As an example of the spatial distribution of the emissions, the strength of the total* $\mathrm{NO_x}$ *emissions is displayed in the background. Measurement sites are Harwell (Har), Aston Hill (AH), Bottesford (Bott), Eskdalemuir (Esk), and Sibton (Sib).*

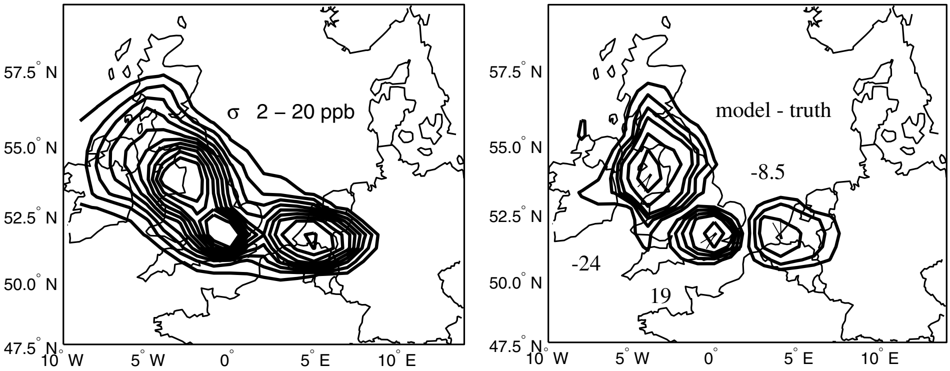


Figure 6.4: *Left: standard deviation in ozone due to uncertain emissions at day 3, 15:00 (august 7); contour lines from 2–20 ppb with interval of 2 ppb. Right: deterministic model minus simulated truth at same hour; contour lines at* $\pm 2.5, 5.0, 7.5, 10.0, 15.0, 20.0$.

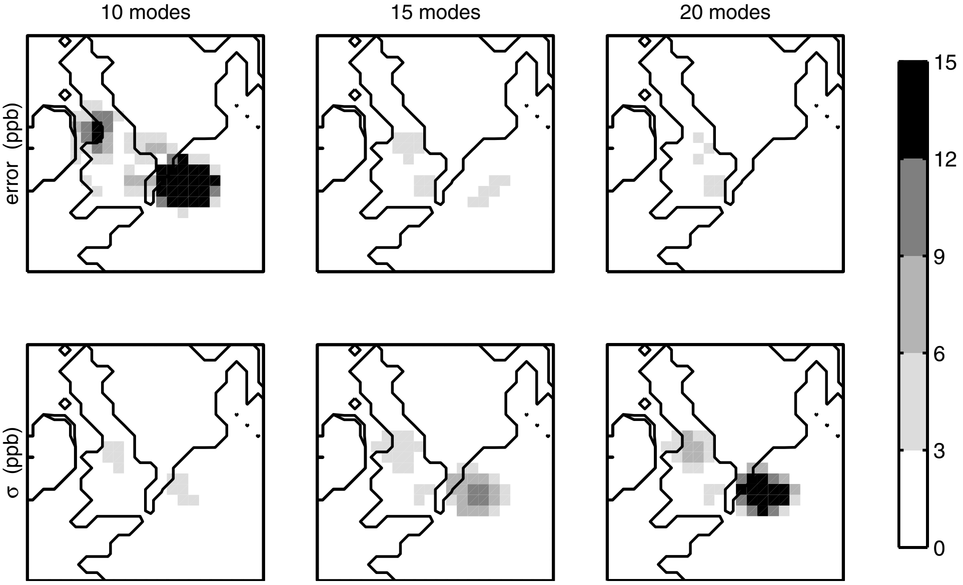


Figure 6.5: *Error and estimated standard deviation in ozone at august 7, 15:00, for the* RRSQRT *filter with 10, 15, or 20 modes. Upper panels: filter mean minus truth (absolute). Lower panels: standard deviation of the error according to the covariance matrix computed by the filter.*

pected to reconstruct the true emission most accurate for the area downwind from London, since two sites (Harwell and Aston Hill) are located in the plume. The impact of uncertain emissions from the Midlands is measured less direct, through advection to Eskdalmuir and diffusion to the site Bottesford (this site is located in the emission area, but the corresponding cell hardly releases emissions; most emissions are released downwind from Bottesford). The uncertainties in the Rhine plume are only visible in Sibton, and the filter problem for this plume is therefore underestimated (two uncertain emissions filtered with measurements from one site); a filter should be able to estimate the standard deviation of the error correctly, however.

The system of uncertain emissions and simulated measurements has been filtered with the three types of low-rank filters for different settings: the ENKF with $m_e = 10, 20, 30, 40$, or 60 ensemble members, the RRSQRT filter with reduction to $m_r = 2, 4, 6, 8, 10, 15, 20, 30$, and 50 modes, and the POENKF with $m_r = 10, 20$, or 30 modes in the RRSQRT part and $m_e = 10, 20$, or 30 ensemble members in the ENKF part. To investigate the impact of the random numbers used in ENKF and POENKF, each of the experiments using one of these filters was repeated four times.

Comparison of filtered time series and spatial patterns with the ’true’ values showed that for the RRSQRT filter at least 20 modes are required for accurate approximation of the covariance matrix (figure 6.5). If 10 or less modes are used, the filter is not able to uncorrelated the errors in the Rhine plume from other errors, and as a result, the concentrations over The Netherlands and the Irish sea are completely messed up. The almost negligible standard deviations suggest that the filter result is accurate, however. For 15 modes, the filter assigns a substantial variance to the concentrations in the Rhine plume, and the error is decreased to less than a few ppb in all grid cells. For 20 modes, the computed standard deviation for the Rhine plume is almost comparable with the standard deviations without assimilation, which is close to the true value since the errors in this plume are hardly filtered. A similar convergence was noticed for the ENKF assimilations. For small ensemble sizes (10–30 members) the concentration patterns are sometimes completely wrong. An ensemble size of at least 40 members was necessary to provide accurate and reproducible results.

To compare the performance of the three different filters with eachother, the root mean square errors were computed over the ozone concentrations at ground level for day 3, 15:00 (filter mean minus simulated truth). The RMS errors are plotted in figure 6.6 versus the number of required model evaluations (m_e for ENKF, $1+m_r$ for RRSQRT, and $1+m_r+m_e$ for POENKF), which is a suitable measure for the total computation time.

The slow convergence of the ENKF filter is illustrated by the large spread in the corresponding RMS errors. For 10 or 20 ensemble members, almost all computed ozone patterns are worse than the first guess run; for 30 members, there is still a significant probability on less accurate results. As already noticed for the concentration patterns in figure 6.5, the number of modes used in the RRSQRT algorithm should exceed a critical level too, before the filter converges. Although the error seems to converge if the number of modes is increased from 2 to 6, the results get worse when their number is increased further to 15 modes. At least 30 modes are required to obtain a stable filter during all hours of the assimilation periode. These results show that the convergence of the RRSQRT filter is much faster than the convergence of the ensemble filter; while the ensemble filter requires at least 40 model evaluations, the reduced rank filter can do with 10 less.

For the POENK filters it is possible to compute RMS-errors for two different means: the mean of the RRSQRT part and the mean of the ENKF part. The final result of a POENKF is always the result of one of the underlying parts; in (*Heemink et al., 2001*) the mean and covariance of the RRSQRT part is used. It is possible to define a 'combined' result in terms of a (weighted) average between RRSQRT part and ENKF part, but this option is not considered here. Instead, the errors in both filter means are computed and compared with their stand alone counter parts. Comparing the results of the stand alone RRSQRT filter with the RRSQRT part of the POENK filter shows that the introduction of random ensembles in the gain matrix is able to stabilize the filter, if the number of modes is too small for convergence. While 10 modes for a single RRSQRT filter is too small for a stable filter (RMS error exceeds the value obtained with the deterministic model), the POENKF variant with 10 modes always provides more accurate results. However, there is still a large probability on results less accurate than the first guess. As soon as the RRSQRT filter has converged (30 modes), including random ensembles in the gain matrix disturbs the results (figure 6.6, lower right panel). Only a coincidental lucky set of random numbers might be able to produce more accurate results, which did not occure in the experiments performed here.

Comparison of stand alone ENKF with the ensemble parts in POENKF shows that introduction of modes in the gain could indeed act as a variance reductor. For a fixed ensemble size, the spread in the RMS errors becomes much smaller if the gain incorporates the modes of a

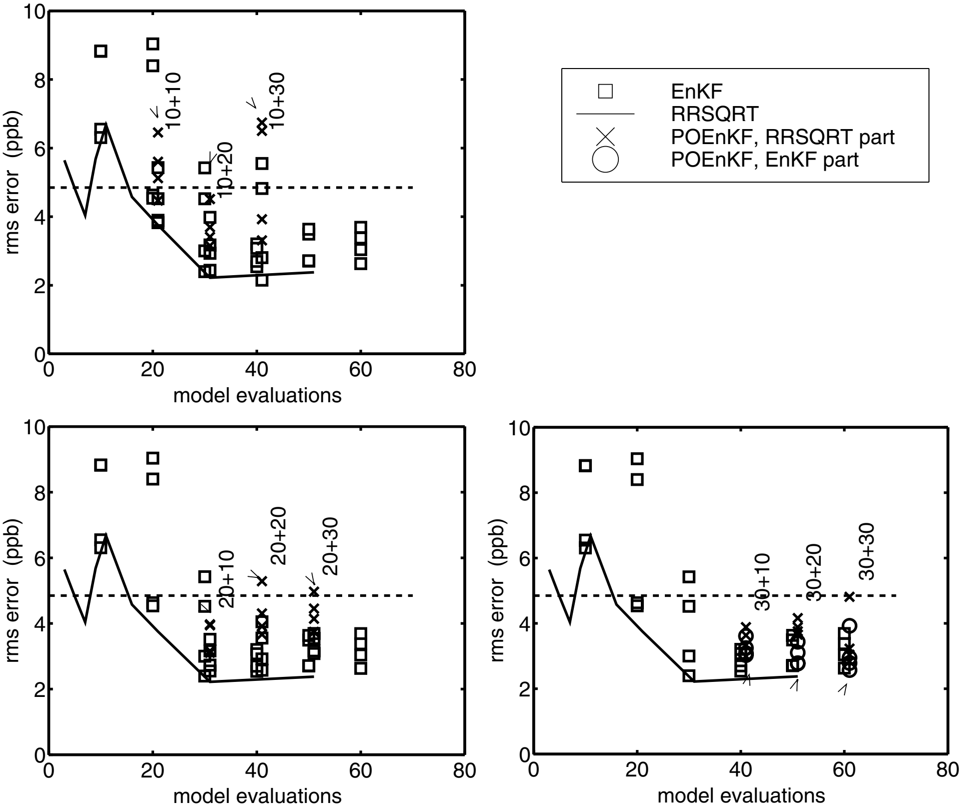


Figure 6.6: *RMS error in ozone concentrations at day 3, 15:00 (filter mean minus simulated truth). The arrows point from the results obtained with a RRSQRT filter to results obtained with a POENKF variant using a similar RRSQRT part; the number of modes and ensemble members is displayed near the heads. The dashed line denotes the error left with the deterministic model.*

RRSQRT filter. The only exception is the variant with 10 modes and 30 ensemble members, when an inaccurate RRSQRT filter sometimes leads to an instable filter.

If the RMS errors of all experiments are compared, the RRSQRT algorithm seems to be the most efficient choice for this particular application. The filter provides an accurate and constant result at a level of required model evaluations where the other algorithms still suffer from random fluctuations. Even for small numbers of modes, the results are more accurate than what could be achieved with an ENKF approach with comparable ensemble size. The POENKF filter is only able to produce more accurate results when compared with the stand alone versions of the underlying filters. For this application, the additional model evaluations spent on a second filter could be spent more efficient on improving the performance of one of the underlying filters.

Similar results are found if the filter types are judged on estimation of other components than ozone, or on the estimates of the (co)variances rather than the absolute errors. The tight connection between different components in the state ensures that either all are accurate or none. The variances were found to converge with the same number of modes/ensemble members as the absolute error; underestimation of the variances immediately leads to large errors in the filter.

6.11 Summary and conclusions

In this chapter, the background, implementation, costs and performance of some common used low-rank filters have been compared.

low-rank filters are either based on factorization of the covariance matrix (RRSQRT, SEIK, and ESSE filter), or approximation of statistics from a finite ensemble (ENKF). A new direction in filter implementation is the use of two filters next to eachother of the same form (DENKF) or hybrid (POENKF). The factorization approach is often based on the linear Kalman filter which has been extended towards nonlinear models; the ensemble technique is a reformulation of the filter problem in a statistical approach.

In spite of the different philosophies, all low-rank filters turn out to have a similar implementation. Evolution of mean and covariance is in each of the filters performed by propagation of an ensemble of state vectors by the model; how the ensemble is formed depends on the filter approach and is discussed in detail in chapter 7. The propagation of the forecast ensemble is the most expensive part of the filter. Four different approaches exist for the analysis of measurements, based on whether the gain will lead to a minimal variance or not, and whether the filter is based on the factorization or the ensemble approach. The forms with a minimum variance gain are in practice most often used, and differ hardly from eachother in computational costs.

The main data structure in all filters is the covariance square root: a large low-rank matrix, with state vectors stored in the columns. The covariance square root needs to be transformed at least one time during each time step, which is an expensive operation. Combination of all transformations in a single operation leads to an efficient filter, however, in which the forecast and the transformation are the major costs. In addition to forecast and analysis, the filters based on factorization require a singular value decomposition or re-orthogonalization of the covariance square root. Two state transformations have been proposed to make the decomposition less sensitive to changes in the definition of the state vector, and to collect for the filter important information in the first modes. The largest impact of the transformations was found in an improved convergence of the filters covariance matrix.

Three different low-rank filters have been implemented around the LOTOS model: based on factorization (RRSQRT, incorporating the forecast scheme of the SEIK filter), ensemble statistics (ENKF), or on a hybrid approach (POENKF, combining a RRSQRT and ENKF filter). All three methods were found to be suitable to assimilate ozone measurements in a LOTOS model with stochastic varying emissions. The ensemble filter suffers from statistical noise due to the use of a random number generator; the results still show a large spread where a RRSQRT filter with comparable costs already converged. As a consequence, also the POENKF filter suffers from the statistical noise in its ENKF part. Due to the fast convergence and accurate results reached with the RRSQRT filter, the benefit of additional random directions in the gain of the POENKF was limited. For comparable costs, the RRSQRT filter produces stable and more accurate results than ENKF or POENKF. The approach of a RRSQRT filter combined with the forecast step of the SEIK filter is therefore the most efficient choice for the filter around LOTOS.

Chapter 7

Nonlinear filters

Four different methods for treating nonlinearities in a Kalman filter problem have been compared and applied to the atmospheric chemistry model LOTOS. *The type of nonlinear dynamics present in such a model complicates an accurate forecast of the state of the system. The different nonlinear forecast methods are either based on linearizations or ensemble statistics. A filter based on minimal exact sampling is shown to produce accurate and stable results with minimal costs. Ensemble statistics are able to produce even more accurate results, but with the cost of at least a double amount of computation time.* [1]

7.1 Introduction

Photo oxidant models are typical examples of nonlinear models. In general, all chemical reactions in an air pollution model are weak to strongly nonlinear; an extensive overview is given in (*Lin et al., 1988*). Other operations such as advection, diffusion, but also deposition are more or less linear.

The problem of dealing with a nonlinear model in a Kalman filter is related to proper evolution of the probability density between successive analysis of measurements. A Gaussian distributed state propagated by a linear model remains Gaussian distributed, and since it is also maintained by the analysis equations, the linear filter is a closed operation between Gaussian distributions. This property is however lost with the smallest nonlinearity in the model, leading to distributions which are only Gaussian in approximation. The Gaussian assumption is essential for a Kalman filter, since it forms the base of the analysis equation; efficient analysis schemes for other distributions do not exist (yet). For practical applications, a Gaussian assumption during the analysis is often accurate enough, but to keep this accuracy, the probability density should be propagated as accurate as possible, taking into account the nonlinearities.

Application of Kalman filtering techniques to nonlinear models has been investigated by many authors. The Extended Kalman Filter (EKF) was designed as an extension of the linear Kalman Filter to weakly nonlinear models (*Jazwinski, 1970*). The EKF uses a new linear model, build from partial derivatives of the nonlinear model, and to this approximation, the

---

[1] Revised from *Nonlinear Kalman Filters for Atmospheric Chemistry Models* by A.J. Segers, A.W. Heemink, M. Verlaan and M. van Loon, In *Inverse Methods in Global Biogeochemical Cycles*, Kasibhatla et al., editors, 2000.

standard Kalman Filter is applied. If second order partial derivatives are available too, a second order accurate filter might be used which incorporates extra terms in the filter equations (*Jazwinski, 1970; Henriksen, 1980*). An overview of theory and practice of the extended filters is given in (*Miller et al., 1994*), where the EKF and other techniques are compared and applied to the small but illustrative double well and Lorenz models. In (*Evensen, 1992; Evensen, 1993*), the EKF is applied to a larger multi layer ocean model. It was shown that the truncation of higher order partial derivatives might result in instability of the covariance evolution, (an unrealistic growth of the error covariance not bounded by analysis of measurements). Instability of the filter occurs if the nonlinearities dominate the model evolution in the time period between successive assimilation of measurements. In many applications however, the linearization of the model equations works surprisingly well, suggesting that the long term evolution is close to linear. In (*Khattatov et al., 1999*) for example, it was shown from experiments with a box model that linearization of the chemistry was sufficient to assimilate satellite observation of trace gases in a trajectory model.

The linear model used in an EKF often takes the form of a tangent linear model (TLM), which provides the perturbations in output variables given perturbations of the model input. If the output variables are of the same form as the input, the TLM is just the Jacobian matrix of the model around the initial state. Where development of a TLM used to be complicated, the work has been simplified by the availability of automatic differentiation tools (*Giering and Kaminski, 1998; Rostaing et al., 1993*). Development of a TLM next to the default model remains rather expensive, however. To overcome implementation of Jacobian matrices or a TLM, approximations to the EKF based on finite differences have been proposed. For large models, considerations about storage and computation of the covariance matrix lead to introduction of low-rank filters, discussed in chapter 6. In these approximate filter, the covariance matrix is parameterized around a limited number of state vectors. For this parameterization, a finite difference approximation of the Extended Kalman filter is a logical step. Finite difference approximations have been proposed for the RRSQRT filter (*Verlaan and Heemink, 1995*) and the SEEK filter (*Pham et al., 1998*), which was then renamed to SEIK filter (*Verron et al., 1999*). The existing finite difference schemes are accurate up to first or second order partial derivatives, truncating higher order nonlinearities.

The observed instability of the covariance evolution in the EKF (*Evensen, 1992; Gauthier et al., 1993*) lead to a reformulation of the filter equations in terms of ensemble or Monte Carlo realizations. With the introduction of the Ensemble Kalman Filter (*Evensen, 1994; Burgers et al., 1998*), filter technique was in some way rebuild from scratch, since statistical moments such as mean and covariance are not the main data structures to compute and evolve anymore. Instead, the ENKF stores and evolves an ensemble of model realizations, from which statistical moments are extracted if necessary. The ENKF is ultimately effective in dealing with nonlinear models, without assumptions about vanishing higher order derivatives. The accuracy of the ensemble method increases with the ensemble size, with the advantage that any desired accuracy might be reached if the ensemble is large enough. The minimum ensemble size might be rather large, and this is the only serious drawback of the method. Thanks to its simple and robust formulation, the ensemble filter has become very popular in geophysical applications such as ocean circulation (*Evensen and van Leeuwen, 1996*), tidal flow (*Cañizares, 1999*), and weather forecast (*Houtekamer and Mitchell, 1998; Keppenne, 2000*).

In this research, the different techniques of nonlinear filtering have been compared and tested for a filter around the LOTOS model. (chapter 2). The general form of the stochastic model and the filter equations are introduced in §7.2. Four different forecast methods are discussed in §7.3: first and second order linearization of the EKF, a method based on minimal exact sampling adapted from the SEIK filter, and the forecast of the Ensemble Kalman filter. Since no tangent linear model is available for LOTOS, implementation of an EKF in the original form is not considered. For each of the forecast methods, the theoretical background is discussed, as well as the implementation in a low-rank filter as discussed in chapter 6. Whenever possible, the accuracy of a method is discussed using Taylor expansions. The performance of the different methods has been tested during filter experiments with the LOTOS model (§7.4). A new development in nonlinear filter applications is quantification of the nonlinearity in terms of a single number (*Verlaan and Heemink, 2001*). The definition and properties of this nonlinearity number has been examined during the experiments, and the results are discussed in §7.5.

7.2 Nonlinear stochastic model and filter equations

The model/observation pair used in this chapter is a generalization of the linear form (6.1) introduced in chapter 6:

$$\mathbf{x}^t[k+1] = \mathbf{M}\left(t[k], \mathbf{x}^t[k]\right) + \boldsymbol{\eta}[k] \tag{7.1a}$$

$$\mathbf{y}^o[k] = \mathbf{H}'[k]\,\mathbf{x}^t[k] + \mathbf{v}[k] \tag{7.1b}$$

In here, $\mathbf{M}$ is a dynamic model acting nonlinear on the state $\mathbf{x}$, $\boldsymbol{\eta}$ is the dynamic noise (zero mean, covariance $\mathbf{Q}$), $\mathbf{y}^o$ is a vector with observations, $\mathbf{H}'$ is the linear observation operator, and $\mathbf{v}$ denotes a random observation error (zero mean, covariance $\mathbf{R}$). The time indices for $\mathbf{M}$ and $\mathbf{H}'$ will be skipped in the rest of this chapter whenever possible; the time for which an operator is valid is implied by its arguments. The Kalman filter is able to compute a mean and covariance of the true state, given our knowledge of physical laws put in $\mathbf{M}$ and the observations in $\mathbf{y}^o$ (see also chapter 3):

$$\hat{\mathbf{x}}^a[k] = \mathrm{E}\left[\, \mathbf{x}^t[k] \mid \mathbf{y}^o[k], \mathbf{y}^o[k-1], \ldots \,\right] \tag{7.2a}$$

$$\mathbf{P}^a[k] = \mathrm{E}\left[\, (\mathbf{x}^t[k] - \hat{\mathbf{x}}^a[k])(\mathbf{x}^t[k] - \hat{\mathbf{x}}^a[k])' \mid \mathbf{y}^o[k], \mathbf{y}^o[k-1], \ldots \,\right] \tag{7.2b}$$

The analyzed mean and covariance are computed in a sequence of forecast and analysis stages. If for a time $t[k]$ an analyzed mean and covariance pair $\{\hat{\mathbf{x}}^a, \mathbf{P}^a\}$ is available, the forecast gives a prediction of these entities using the model and the noise input; in a general form:

$$\left\{\hat{\mathbf{x}}^f[k+1], \mathbf{P}^f[k+1]\right\} = \mathcal{F}(\, \mathbf{M},\ \hat{\mathbf{x}}^a[k],\ \mathbf{P}^a[k],\ \mathbf{Q}[k]\,) \tag{7.3}$$

In case of a linear model $\mathbf{M}(\mathbf{x}) = \mathbf{A}\mathbf{x}$, the forecast $\mathcal{F}$ takes the form of the linear Kalman forecast:

$$\hat{\mathbf{x}}^f[k+1] = \mathbf{A}\,\hat{\mathbf{x}}^a[k] \tag{7.4a}$$

$$\mathbf{P}^f[k+1] = \mathbf{A}\,\mathbf{P}^a[k]\,\mathbf{A}' + \mathbf{Q}[k] \tag{7.4b}$$

This chapter will discuss the implementation of (7.3) in case of a nonlinear $\mathbf{M}$. nonlinearity of $\mathbf{M}$ does not influence the analysis equations, which remain equal to the default Kalman analysis:

$$\mathbf{K}[k{+}1] \;=\; \mathbf{P}^f[k{+}1]\,\mathbf{H}\,(\,\mathbf{H}'\,\mathbf{P}^f[k{+}1]\,\mathbf{H} \,+\, \mathbf{R}[k{+}1]\,)^{-1} \tag{7.5a}$$

$$\hat{\mathbf{x}}^a[k{+}1] \;=\; \hat{\mathbf{x}}^f[k{+}1] \,+\, \mathbf{K}[k{+}1]\,(\,\mathbf{y}^o[k{+}1] \,-\, \mathbf{H}'\,\hat{\mathbf{x}}^f[k{+}1]\,) \tag{7.5b}$$

$$\mathbf{P}^a[k{+}1] \;=\; (\,\mathbf{I} \,-\, \mathbf{K}[k{+}1]\,\mathbf{H}'\,)\,\mathbf{P}^f[k{+}1] \tag{7.5c}$$

Although $\mathbf{M}$ does not appear in the analysis equations, the nonlinearity has an indirect influence. The analysis is based on the assumption that the probability density of the state is Gaussian, and thus completely defined by a mean and covariance. For a problem description with nonlinear dynamics, this is only true in approximation. nonlinear dynamics indicate that opposite deviations around a mean are not of equal importance, and this violates with a (symmetric) Gaussian distribution. Even in case of a linear model, a pure Gaussian distribution is not always correct, for example if the state space is limited to positive entities. The Gaussian assumption is the only practical method available, however, and in practice often good enough.

7.3 Forecast methods for low-rank filters

The Kalman filter used in this research is based on a low-rank approximation of the covariance matrix. An extensive discussion of the backgrounds and formulation of this kind of approximate filter has been given in chapter 6. The filter equations used in this chapter are based on the general low-rank filter formulation as described in §6.8, including a reduction mechanism when necessary.

The low-rank approximation is based on factorization of the covariance matrices $\mathbf{P}$ and $\mathbf{R}$ in low-rank square roots: $\mathbf{P} = \mathbf{S}\mathbf{S}'$ and $\mathbf{Q} = \mathbf{T}\mathbf{T}'$. The number of columns in $\mathbf{S}$ or $\mathbf{T}$ is in order 10-100, and are often referred to as the modes of the filter. The general form of the forecast in terms of the square root factors becomes:

$$\left\{\hat{\mathbf{x}}^f[k{+}1], \mathbf{S}^f[k{+}1]\right\} \;=\; \mathcal{F}(\,\mathbf{M},\, \hat{\mathbf{x}}^a[k],\, \mathbf{S}^a[k],\, \mathbf{T}[k]\,) \tag{7.6}$$

For each of the reduced rank filters discussed in chapter 6 (ENKF, RRSQRT, SEIK) one or more forecasts methods have been proposed. Although these techniques are based on sometimes very different concepts, the implementations turn out to be quite the same. All methods propagate an ensemble of state vectors in order to make a forecast of the mean and covariance. For the ensemble filter, this is the basic concept; for the other filters, this is a solution to overcome the nonlinearity problem. In a general notation, each forecast consists of three stages (see also figure 7.1):

1. Formation of the ensemble: given an analyzed mean and covariance square root, an ensemble of state vectors is formed:

   $$\{\boldsymbol{\xi}_1[k], \boldsymbol{\xi}_2[k], \ldots\} \;=\; \mathrm{Ens}(\hat{\mathbf{x}}^a[k], \mathbf{S}^a[k]) \tag{7.7}$$

   A same number of noise vectors $\boldsymbol{\eta}_j$ is formed too.

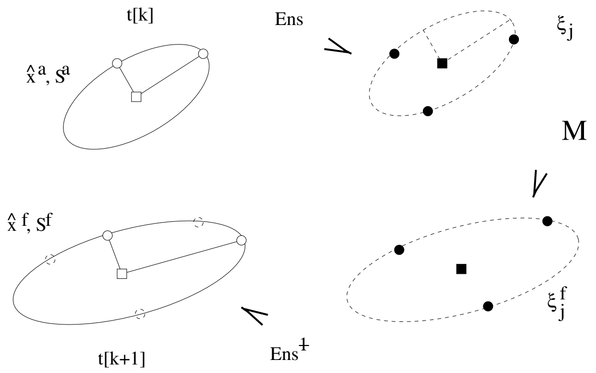


Figure 7.1: *Illustration of forecast of mean and covariance square root in a low-rank filter. The mean and the columns of the square root define an area in the state space where the true state is expected to be with large probability. To propagate this area in time, an ensemble of states is formed at* $t[k]$ *from the mean and the modes; these are propagated by the model, and serve as building blocks for the new mean and modes at* $t[k+1]$.

2. Propagation: each ensemble member is propagated by the model, forced by the corresponding noise vector:

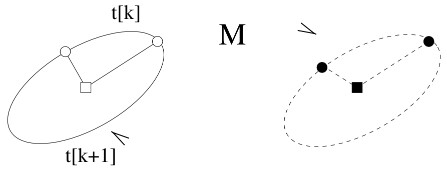
$$\boldsymbol{\xi}_j[k+1] \;=\; \mathbf{M}(\boldsymbol{\xi}_j[k]) \;+\; \boldsymbol{\eta}_j[k] \qquad , \qquad j = 1,2,\ldots \tag{7.8}$$

3. Finally, a new mean and covariance square root are reconstructed from the propagated ensemble; how this is done depends on how the ensemble was formed, and is therefore in general the inverse of (7.7):

$$\left\{\hat{\mathbf{x}}^f[k+1], \mathbf{S}^f[k+1]\right\} \;=\; \text{Ens}^{-1}(\boldsymbol{\xi}_1[k+1], \boldsymbol{\xi}_2[k+1], \ldots) \tag{7.9}$$

The actual form of the forecast ensembles is now discussed in detail for the different methods.

7.3.1 First order linearizations



The forecast method based on first order linearizations is an approximation of the Extended Kalman filter (EKF), and has been proposed for the RRSQRT filter (section 6.5.1). The EKF was introduced as an extension of the original Kalman filter (KF) for dynamical models which are weakly nonlinear (*Jazwinski, 1970*) The idea is to apply the KF to a linear system which approximates the nonlinear one. If the dynamics are not too complicated, a linearization of the underlying model in terms of Jacobian matrices could be used:

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{M}(\mathbf{x}[k]) + \boldsymbol{\eta}[k] \\ &= \mathbf{M}(\mathbf{x}_0[k] + \boldsymbol{\delta}\mathbf{x}[k]) + \boldsymbol{\eta}[k] \approx \left.\frac{\partial \mathbf{M}}{\partial \mathbf{x}}\right|_{\mathbf{x}_0} \boldsymbol{\delta}\mathbf{x}[k] + \boldsymbol{\eta}[k] \end{aligned} \tag{7.10}$$

For complicated dynamics, calculation of the Jacobian matrices is not feasible, and one has to use a numerical approximation of the linearizations:

$$\left.\frac{\partial \mathbf{M}}{\partial \mathbf{x}}\right|_{\mathbf{x}_0} \boldsymbol{\delta}\mathbf{x}[k] \approx \frac{\mathbf{M}(\mathbf{x}_0 + \varepsilon\,\boldsymbol{\delta}\mathbf{x}) - \mathbf{M}(\mathbf{x}_0)}{\varepsilon} \tag{7.11}$$

The EXT1 forecast proposed for the RRSQRT filter is a direct implementation of the forecast equations (6.10) for a low-rank filter, with the linear model replaced by such finite difference approximations:

$$\hat{\mathbf{x}}^{f,ext1}[k+1] = \mathbf{M}(\,\hat{\mathbf{x}}^a[k]\,) \tag{7.12a}$$

$$\begin{aligned} \mathbf{S}^{f,ext1}[k+1] &\approx \left[\; \left.\frac{\partial \mathbf{M}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}^a[k]} \mathbf{S}^a[k] \;,\; \mathbf{T}[k] \;\right] \\ &\approx \left[\cdots \frac{\mathbf{M}\Big(\,\hat{\mathbf{x}}^a[k] + \varepsilon\,\mathbf{s}^a_j[k]\Big) - \mathbf{M}(\,\hat{\mathbf{x}}^a[k]\,)}{\varepsilon} \cdots \;,\; \cdots \mathbf{t}_l[k] \cdots \right] \\ & \qquad\qquad j = 1,\ldots,m \quad , \quad l = 1,\ldots,q \end{aligned} \tag{7.12b}$$

The modes $\mathbf{s}_j$ and $\mathbf{t}_l$ are interpreted as deviations from the mean state; if these deviations are not too large and the nonlinearities are not too strong, the approximations are accurate. In view of eq. (7.7-7.8), the ensembles of states used during the forecast are given by $(j = 0, .., m+q)$:

$$\begin{array}{lccccccccc} \boldsymbol{\xi}_j \in \{ & \hat{\mathbf{x}}^a & , & \hat{\mathbf{x}}^a + \varepsilon\mathbf{s}^a_1 & , \ldots , & \hat{\mathbf{x}}^a + \varepsilon\mathbf{s}^a_m & , & \hat{\mathbf{x}}^a & , \ldots , & \hat{\mathbf{x}}^a \; \} \\ \boldsymbol{\eta}_j \in \{ & \mathbf{o} & , & \mathbf{o} & , \ldots , & \mathbf{o} & , & \mathbf{t}_1 & , \ldots , & \mathbf{t}_q \; \} \end{array} \tag{7.13}$$

Similar as in the original Kalman filter, the mean and covariance (square root) are propagated more or less independent from eachother. The new mean is just a propagation of the previous one; new modes are formed from propagation of the previous modes (m in total) or by introduction of model noise (q). Introduction of model noise is separated from the propagation of the modes: the m 'mode-modes' represent uncertain knowledge of the true state due to an uncertain initial condition, while the q 'noise-modes' represent uncertainties in the model. The number of modes in the covariance square root has grown during the forecast from m to $m+q$; to prevent the filter from growing out of computational limits, a filter using this forecast scheme should include a reduction mechanism.

The scale-factor ε is set to a value such that the ensemble members are suitable states to be propagated by the model. Besides, it can be used to obtain additional accuracy. A simple rule to set the scale-factor is the following:

> *If q modes are correlated with a mode $\mathbf{s}_j$ (including $\mathbf{s}_j$ itself), then a scale-factor $\varepsilon = \sqrt{q}$ should be used to form a state vector $\hat{\mathbf{x}} + \varepsilon \mathbf{s}_j$.*

This rule is explained from the values a state vector could take given a mean/covariance pair. The covariance bounds the amplitude of the variation in each possible direction in terms of a variance. For a covariance matrix in square root form, the variance in a certain direction is the sum of the variances of the modes into this direction. If q modes contribute equally to a variance σ^2, then each of them has an average amplitude of $\sigma/\sqrt{q}$. With the ε set according to the proposed rule, the model is called with input states which deviate from their mean with an average amplitude of σ. Following the epsilon-rule, an appropriate choice for the scale-factors for the EXT1 forecast is to set $\varepsilon = 1.0$ if the modes are orthogonal. This is for example true if the rank of covariance square root is reduced as in the RRSQRT filter, or formed from EOF's as in the SEIK filter.

In appendix C, the accuracy of the first order linearizations has been studied using Taylor expansions. As shown by the derivations in §C.2, the forecast of the state and covariance are accurate up to terms containing first order partial derivatives of $\mathbf{M}$. If second order partial derivatives are zero or small, the error made during the forecast will not be very large.

7.3.2 Second order linearizations

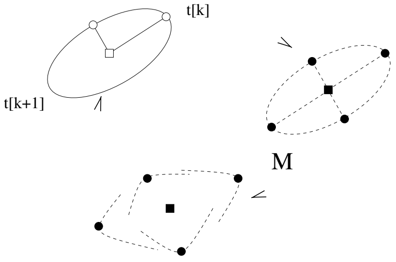
Where the EXT1 forecast is an approximation of the Extended Kalman filter (without Jacobian matrices), a second order accurate forecast is an approximation of the truncated second order filter (*Jazwinski, 1970*), without computation of Hessian matrices. A second order filter does not contain any truncation errors in the forecast if the model $\mathbf{M}(\mathbf{x})$ does not contain third or higher order nonlinear terms.

A second order forecast is almost similar to the first order EXT1 forecast with respect to the formation of ensembles of states and noise vectors. However, the ensemble of state vectors is not formed from the columns of $\mathbf{S}$ on a one-to-one basis such as in (7.13), but using a new set of columns. The new columns should specify the same covariances but have a zero mean in addition:

$$\sum_{j=1}^{\bar{m}} \bar{\mathbf{s}}_j \bar{\mathbf{s}}_j' = \mathbf{S}\mathbf{S}' \quad , \quad \sum_{j=1}^{\bar{m}} \bar{\mathbf{s}}_j = \mathbf{o} \tag{7.14}$$

Two methods will be described to form the new covariance square roots: the second order extended update proposed for the RRSQRT filter, and the minimal exact sample used in the SEIK filter.

Second order extended forecast



The *second order extended forecast* (EXT2) is a straight forward method to construct a set of modes matching the requirements for a second order forecast. Given an arbitrary covariance matrix, specified by the columns $\mathbf{s}_1,\ldots,\mathbf{s}_m$ of its square root $\mathbf{S}$, it is straight forward to show that the set

$$\bar{\mathbf{s}}_j \;\in\; \left\{+\tfrac{1}{\sqrt{2}}\mathbf{s}_1, -\tfrac{1}{\sqrt{2}}\mathbf{s}_1, \ldots, +\tfrac{1}{\sqrt{2}}\mathbf{s}_m, -\tfrac{1}{\sqrt{2}}\mathbf{s}_m\right\} \tag{7.15}$$

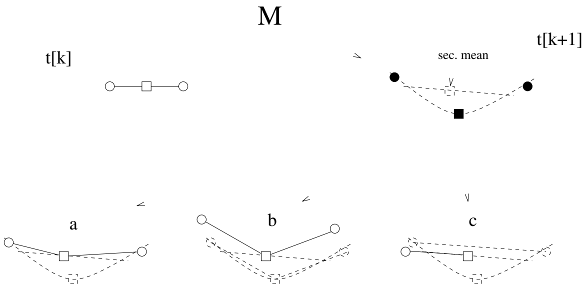
satisfies (7.14). The number of modes has been doubled from m to $\bar{m} = 2m$. Ensembles of this form were proposed in (*Julier et al., 1995*) for application in robot guidance, and in (*Verlaan and Heemink, 1996*) for a RRSQRT filter around a shallow water model. The doubled set of modes is used to form input states for the model in a way similar to the first order forecast:

$$\begin{array}{lllllllllllll} \boldsymbol{\xi}_j & \in & \{ & \hat{\mathbf{x}}^a & , & \hat{\mathbf{x}}^a + \varepsilon\bar{\mathbf{s}}^a_1 & , \ldots , & \hat{\mathbf{x}} + \varepsilon\bar{\mathbf{s}}^a_{\bar{m}} & , & \hat{\mathbf{x}}^a & , \ldots , & \hat{\mathbf{x}}^a & \} \\ \boldsymbol{\eta}_j & \in & \{ & \mathbf{o} & , & \mathbf{o} & , \ldots , & \mathbf{o} & , & \mathbf{t}_1 & , \ldots , & \mathbf{t}_q & \} \end{array} \tag{7.16}$$

The propagated ensemble is used to form a new mean and covariance square root. Where in the first order forecast the new mean is just the propagation of the previous mean, the second order forecast contains an extra term:

$$\hat{\mathbf{x}}^{f,ext2}[k+1] \;=\; \boldsymbol{\xi}_0[k+1] \;+\; \sum_{j=1}^{\bar{m}} \frac{\boldsymbol{\xi}_j[k+1] - \boldsymbol{\xi}_0[k+1]}{\varepsilon^2} \tag{7.17}$$

The extra term ensures that (7.17) is the correct forecast of the mean if the model $\mathbf{M}(\mathbf{x})$ contains only second order nonlinear terms; see appendix C.3 for a prove with Taylor expansions. For computation of the new modes at t_{k+1}, three different methods have been suggested, summarized in table 7.1. Each of the methods use (7.17) for the forecast of the mean. method 'a' (*Julier et al., 1995*) computes the new modes using deviations from the second order forecast of the mean, similar to the sample covariance in an ensemble method. Therefore, the method implies a scale-factor $\varepsilon = \sqrt{2m}$ since otherwise (7.17) is not equal to the sample mean. However, the arguments behind the epsilon-rule at page 125 suggest that such a scale-factor is undesirable if the modes of the covariance matrix are orthogonal, which is true after a reduction as defined in §6.8 and might be true for the general covariance factorization used in (*Julier et al., 1995*). To avoid these problems, method 'b' (*Verlaan and Heemink, 1996*) computes the modes with deviations from the central forecast of the mean. The second order accuracy is now obtained for any scale-factor. The epsilon-rule at page 125 suggests that $\varepsilon = \sqrt{2}$ is a suitable choice if the original modes are orthogonal, since each direction in the new ensemble is specified by two ensemble members. Finally, method 'c' was proposed by Voorrips et al. (1998) to limit the number of new formed modes, and thus to reduce the costs of the reduction mechanism. The difference between the propagations of



a	$\mathbf{s}_j^{f,ext2} = \frac{\boldsymbol{\xi}_j - \hat{\mathbf{x}}}{\varepsilon_j}$	$j=1,\ldots,2m$	(*Julier et al., 1995*)
b	$\mathbf{s}_j^{f,ext2} = \frac{\boldsymbol{\xi}_j - \boldsymbol{\xi}_0}{\varepsilon_j}$	$j=1,\ldots,2m$	(*Verlaan and Heemink, 1996*)
c	$\mathbf{s}_j^{f,ext2} = \frac{\boldsymbol{\xi}_{2j-1} - \boldsymbol{\xi}_{2j}}{\varepsilon_{2j-1}+\varepsilon_{2j}}$	$j=1,\ldots,m$	(*Voorrips et al., 1999*)

Table 7.1: *Illustration of different techniques to compute new modes in the* EXT2 *forecast. The propagated mean and modes define a quadratic surface; the second order mean (7.17) is in the convex subspace enclosed by the surface. The new modes are computed from the difference between propagated modes and second order mean (a), the difference between propagated modes and central forecast of the mean (b), or from the cord between propagations of two opposite modes (c). The modes are numbered according to ensemble (7.16).*

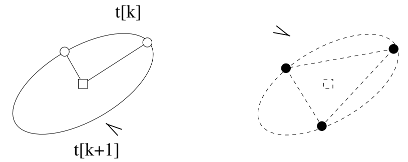
two opposite ensemble members is used to form a single new mode. Taylor expansions in appendix C.3 show that this method leads to a small under estimation of the true covariance, and therefore to an increased danger of filter divergence. Since the model evaluations form the majority of the costs of the filter, reducing the costs of the reduction has no priority in our applications, and method 'b' will be used for computation of the modes.

Application of the second order forecast using the suggested sets of modes requires $1+2m$ model evaluations: two for each of the original modes and one for the mean. An extended second order forecast is therefore quite expensive. In comparison with the EXT1-forecast, the second order extended forecast produces more accurate results with the cost of doubled computation time and memory use. A rank reduction is absolutely necessary, since the number of modes has at least been doubled after the forecast.

Minimal Exact Sampling (MES)

An efficient and elegant algorithm for construction of a set of modes matching (7.14) was suggested in (*Pham, 1996*) for use in the SEIK filter. Where the previous described EXT2 forecast requires at least a double amount of model evaluation, the method of Minimal Exact Sampling (MES) requires the same number of model evaluations as the first order forecast

EXT1, but gives a second order accurate result.



The idea behind a MES is to replace an existing $n \times m$ covariance square root $\mathbf{S}$ by the $n \times \bar{m}$ square root $\bar{\mathbf{S}} = \mathbf{S}\boldsymbol{\Omega}$, where the $m \times \bar{m}$ matrix $\boldsymbol{\Omega}$ is chosen such that requirements (7.14) are matched. For the columns $\boldsymbol{\omega}_j$ this leads to the following requirements:

$$\sum_{j=1}^{\bar{m}} \mathbf{S}\boldsymbol{\omega}_j = \mathbf{o} \quad \Rightarrow \quad \sum_{j=1}^{\bar{m}} \boldsymbol{\omega}_j = \mathbf{o} \tag{7.18a}$$

$$\sum_{j=1}^{\bar{m}} (\mathbf{S}\boldsymbol{\omega}_j)(\mathbf{S}\boldsymbol{\omega}_j)' = \mathbf{S}\mathbf{S}' \quad \Rightarrow \quad \sum_{j=1}^{\bar{m}} \boldsymbol{\omega}_j \boldsymbol{\omega}_j' = \boldsymbol{\Omega}\boldsymbol{\Omega}' = \mathbf{I} \tag{7.18b}$$

A specific example is the set of modes (7.15) used for the second order extended forecast, which is the result of an operation $\bar{\mathbf{S}} = \mathbf{S}\boldsymbol{\Omega}$ with in each column of $\boldsymbol{\Omega}$ only one non-zero element, equal to plus or minus $1/\sqrt{2}$. The algorithm proposed in (*Pham, 1996*) is able to produce a suitable $\boldsymbol{\Omega}$ for any $\bar{m} \geq m+1$. The algorithm is given in appendix D. If the algorithm is used to form an $\boldsymbol{\Omega}$ with a minimal $\bar{m} = m+1$, the *Minimal Exact Sample* (MES) drawn from a mean $\hat{\mathbf{x}}^a$ and a m-column covariance square root $\mathbf{S}^a$ is defined as the set:

$$\boldsymbol{\xi}_j \in \{\hat{\mathbf{x}}^a + \sqrt{\bar{m}}\mathbf{S}^a\boldsymbol{\omega}_1, \ldots, \hat{\mathbf{x}}^a + \sqrt{\bar{m}}\mathbf{S}^a\boldsymbol{\omega}_{\bar{m}}\} \tag{7.19}$$

The algorithm listed in appendix D includes a random generator, such that the set (7.19) is still more or less a random sample. Each of the new sample members is build from contributions of all original modes; the choice $\varepsilon = \sqrt{\bar{m}}$ for the scale-factors is therefore in agreement with the epsilon rule from page 125.

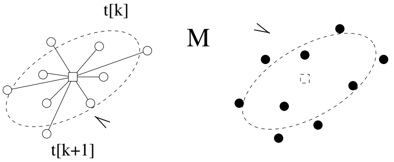
The forecast of the SEIK filter (*Pham, 1996*) consists of formation of a MES from a mean state and a covariance square root defined on a base of EOF's. After propagation of the MES by the model, the new mean and covariance square root specified by the sample mean and inverse of (7.19):

$$\hat{\mathbf{x}}^{f,seik} = \frac{1}{\bar{m}} \sum_{j=1}^{\bar{m}} \boldsymbol{\xi}_j^f \tag{7.20a}$$

$$\mathbf{S}^{f,seik} = \tfrac{1}{\sqrt{\bar{m}}} \left[\boldsymbol{\xi}_1^f - \hat{\mathbf{x}}^{f,seik}, \ldots, \boldsymbol{\xi}_{\bar{m}}^f - \hat{\mathbf{x}}^{f,seik}\right] \boldsymbol{\Omega}' \tag{7.20b}$$

The transformation with $\boldsymbol{\Omega}'$ ensures that the number of columns in $\mathbf{S}^f$ does not grow due to the forecast. If this transformation is omitted, the next forecast step should not include a forward transformation with a new $\boldsymbol{\Omega}$. The introduction of dynamic noise is included in the analysis equations of the SEIK filter; see section 6.5.2. In the context of general low-rank filter as defined in section 6.8, a forecast based on a MES is similar to equations (7.19) and (7.20) for propagation of mean and covariance square root; in addition, the columns of the matrix $\mathbf{T}$ are appended to $\mathbf{S}$. The number of columns in the covariance square root will grow through appending of $\mathbf{T}$. Transformation with $\boldsymbol{\Omega}'$ as used in (7.20b) to avoid the growth of the number of modes could be omitted, since this number will grow anyway.

7.3.3 Ensemble forecast



The *ensemble forecast* is the driving force behind the Ensemble Kalman Filter (*Evensen, 1997*); see section 6.6 for a detailed description. An ensemble of m states is propagated by the dynamics, where the noise input for the model is taken from a random generator. Whenever a mean or covariance are required, these are set to the sample statistics of the ensemble, and used as an estimate of the true statistics. The ENKF differs here from filters evoluted from the linear Kalman filter (such as the RRSQRT and SEIK filter), where the probability density of the state is expressed in terms of an explicit computed mean and covariance. For treatment of a nonlinear model, these filters form an ensemble of state vectors from the mean and covariance, and the propagated ensemble is used to reformulated the mean and covariance. In the ENKF however, the ensemble is not an intermediate state in the filter but defines the probability in all stages.

For an arbitrary ensemble $\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m\}$ of state vectors defining the probability density, the ensemble forecast is simply a propagation by the model of each member, forced by random noise input:

$$\boldsymbol{\xi}_j[k+1] = \mathbf{M}(\boldsymbol{\xi}_j[k]) + \boldsymbol{\eta}_j[k] \qquad , \qquad \boldsymbol{\eta}_j[k] \sim \mathcal{N}(\mathbf{o}, \mathbf{Q}[k]) \tag{7.21}$$

The ensemble members $\boldsymbol{\xi}_j$ usually define the probability density in an ENKF, but might be formed from a given mean/covariance pair too:

$$\boldsymbol{\xi}_j[k] \sim \mathcal{N}(\hat{\mathbf{x}}[k], \mathbf{P}[k]) \tag{7.22}$$

or, in square root form:

$$\boldsymbol{\xi}_j[k] = \hat{\mathbf{x}}[k] + \mathbf{S}[k]\,\mathbf{w}_j \qquad , \qquad \mathbf{w}_j \sim \mathcal{N}(\mathbf{o}, \mathbf{I}) \tag{7.23}$$

The number of random ensemble members drawn in this way is infinite, and might be different from the number of columns in $\mathbf{S}$.

The advantage of the ensemble forecast is that the sample statistics of the propagated ensemble converge to the true statistics for growing ensemble size, even in case of nonlinear dynamics. It is always possible to choose an ensemble large enough to let the error in the computed statistics be less than some desired accuracy. However, the convergence of the ensemble forecast is slow (of order $1/\sqrt{m}$), requiring a large ensemble size and many model evaluations. For small ensembles, statistical noise dominates the filter result; in practice, the number of ensembles should be in order 10^2.

The most important difference between the ensemble forecast and the forecasts discussed up to now is the rather brute force strategy. Model noise is introduced with a random generator; if the number of ensemble members is large enough, the 'true' sample is probably included. The EXT1, EXT2, and MES forecasts try to limit the number of samples as much as possible, by using ensemble members with special properties. The implementation of a specialized forecast is therefore more complicated, while an ensemble forecast is simple, easy to understand, and very robust. The only serious drawback of the ensemble forecast is the large number of random samples necessary for convergence and removal of statistical noise. Random drawn vectors will always show spurious correlations, which can only be removed by increasing the ensemble size.

7.3.4 Summary of forecast schemes

The previous discussed forecast strategies are summarized in terms of a general reduced rank filter as described in §6.8. Let the probability density of the state be defined by the mean/covariance square root pair $(\hat{\mathbf{x}}^a, \mathbf{S}^a)$; in case of an ensemble filter, these specify the ensemble following (7.23). The first step in each of the forecasts is to form a forecast ensemble, in a general notation:

$$[\ldots, \boldsymbol{\xi}_j[k], \ldots] \;=\; \hat{\mathbf{x}}^a[k] \;+\; \varepsilon\, \mathbf{S}^a[k]\; \boldsymbol{\Omega}_S \tag{7.24a}$$

$$[\ldots, \boldsymbol{\eta}_j[k], \ldots] \;=\; \mathbf{T}[k]\; \boldsymbol{\Omega}_T \tag{7.24b}$$

The notation chosen here reflects that each ensemble member is formed from a linear combination of columns of $\mathbf{S}$ or $\mathbf{T}$; for all members together, this is the same as multiplication with matrices $\boldsymbol{\Omega}_S$ and $\boldsymbol{\Omega}_T$. Forecast techniques differ from eachother with respect to the size and shape of the matrices $\boldsymbol{\Omega}_S$ and $\boldsymbol{\Omega}_T$, and the value of the scale-factors ε. Table 7.2 gives an overview of these entities for the forecast methods discussed.

The second step is propagation of the forecast ensemble from $t[k]$ to $t[k+1]$, which is similar for all methods:

$$\boldsymbol{\xi}_j[k+1] \;=\; \mathbf{M}\left(\boldsymbol{\xi}_j[k+1]\right) \;+\; \boldsymbol{\eta}_j[k+1] \qquad , \qquad j = 1, \ldots \tag{7.25}$$

Finally, the new mean and covariance square root are formed from the propagated ensemble. The new mean is either the central forecast, the second order forecast from eq. (7.17), or the sample mean over all propagated ensemble members. The new modes are formed from deviations between the propagated ensemble members and either the central forecast (for the forecasts based on the EKF) or the new mean (for the forecasts based on samples).

7.4 Application to LOTOS

The performance of the different nonlinear methods has been tested for the experimental setup described in detail in §4.4.2. The domain of the LOTOS model is bounded to the southern part of the UK; the stochastic model includes uncertainties in emissions of NO_x and VOC, photolysis rates, and deposition velocity of ozone. The filter assimilates hourly ozone measurements from five different sites, during a 5 day period in august 1997. Figure 7.2 shows an example of the filter result for site Glazebury.

In a large number of experiments, the filter problem has been solved using first order linearization (EXT1), second order linearizations (EXT2), a minimal exact sample (MES), or an ensemble technique (ENS). All except the ensemble filters required reduction of the covariance square root; the number of modes is reduced to either 10, 20, 30, 40, or 50 modes after each analysis. Ensemble sizes for the ensemble filter were set to similar numbers. If a method uses random numbers (MES and ENS), the experiment was repeated 4 times to study the impact of the random generator. The computational costs of a method are determined

	centr. forc.	size $\boldsymbol{\Omega}_\star$ $m \times \ldots$	$\boldsymbol{\Omega}_S$ $\boldsymbol{\Omega}_T$	ε	mean:	modes around:
EXT1	x	$m{+}q$	$[\mathbf{I},\mathbf{O}]$ $[\mathbf{O},\mathbf{I}]$	1	centr. forc.	centr. forc.
EXT2	x	$2m{+}q$	$[\mathbf{I},-\mathbf{I},\mathbf{O}]/\sqrt{2}$ $[\mathbf{O},\mathbf{O},\mathbf{I}]$	$\sqrt{2}$	(7.17)	centr. forc.
MES		$m{+}1{+}q$	$[\boldsymbol{\Omega}^{mes},\mathbf{O}]$ $[\mathbf{O},\mathbf{I}]$	$\sqrt{m+1}$	sample mean	sample mean
ENS		m	$\mathbf{I}$ $\mathcal{N}(\mathbf{o},\mathbf{I})$	$\sqrt{m}$	sample mean	sample mean

Table 7.2: *Summary of differences between forecast algorithms: whether a central forecast of the mean is included or not, size of the transformation matrices $\boldsymbol{\Omega}_S$ and $\boldsymbol{\Omega}_T$ (= size of the forecast ensemble), contents of the transformation matrices (see appendix D for matrices $\boldsymbol{\Omega}^{mes}$), the value of scale-factor ε, and how the new mean and the new modes are formed.*

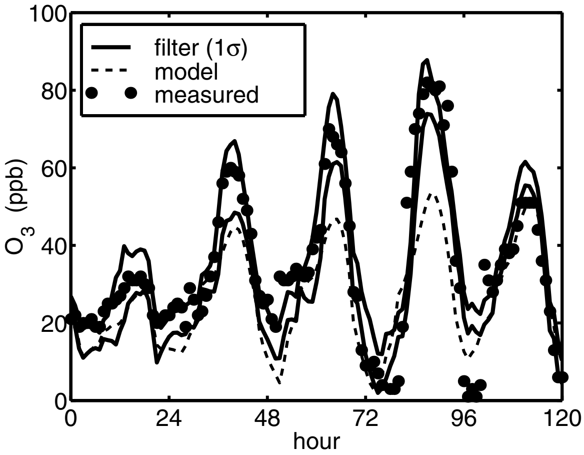


Figure 7.2: *Example of ozone time series during the experiments with different forecast schemes (site Glazebury).*

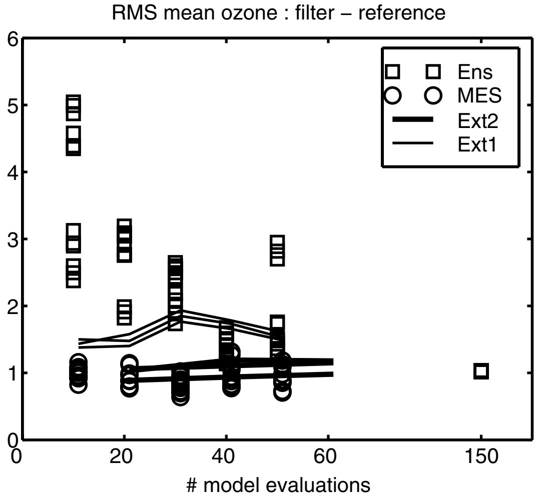


Figure 7.3: *RMS error of filter mean if compared with three different reference solutions (ENS 150). Results for EXT2 with 40 or 50 modes are not very different from the results with 30 modes, and therefore skipped. The lines connect errors computed against the same reference. Note the irregular spacing in the horizontal ax.*

by the number of model evaluations, equal to $m+1$ for EXT1 and MES, $2m+1$ for EXT2, and m for the ENS forecast.

To investigate the error made by different forecast methods, the mean ozone concentrations computed with a filter have been compared with concentrations of a reference solution:

$$\text{RMSE}(O_3) = \sqrt{\frac{1}{n_x n_y n_t} \sum_{i,j,k} \left(\hat{c}_{ij}^{filt}[k] - \hat{c}_{ij}^{ref}[k] \right)^2} \tag{7.26}$$

for all available ozone concentrations $c_{ij}[k]$ at ground level. The reference solution is obtained from an ensemble filter with very large ensemble size (150 members), since this filter is known to converge to the exact filter solution. Even for 150 members there was still some variation observed due to the use of random numbers, however; therefore, two extra reference runs have been produced in addition. The results are plot in figure 7.3. The difference between two arbitrary reference solutions is equal to about 1.0 ppb. This value is therefore an underbound for the RMS error which could be obtained with any forecast method. The errors made with the ensemble filters show a clear convergence to the reference for growing ensemble size. The only exception is an experiment with 50 ensemble members, where the error is unexpected large in comparison with all reference solutions. A large spread in the results is observed for the smaller ensemble sizes, illustrating that statistical noise dominates the results here. The error made using first order linearizations (EXT1) shows a rather unexpected growth for increasing number of modes, with a maximum for truncation to 30 modes. A possible explanation is the increased numerical inaccuracy for higher truncations, when a large number of modes are filled with almost zero numbers. Increasing the number of modes to 40 or 50 decreases the error however. The straight lines connect the errors with respect to different reference runs; the spread illustrates that the differences between the reference solutions are not negligible. For truncation to 10 or 20 modes, the error made with the EXT1 method is for any experiment smaller than an ensemble filter with comparable costs.

The error made with second order forecasts (EXT2 and MES) is much smaller than those observed for the first order forecast, fluctuating around an RMS of about 1.0 (the 'reference' value). Similar as for the EXT1 method, the error made with the EXT2 scheme increases slowly although less steep. The accuracy is always higher than obtained with ENS or EXT1 for comparable costs. The best results are obtained for the MES method. Although the MES is build using a random generator, the spread in the result is quite small and seems to be converged after truncation to 10 modes. The results hardly change for growing number of modes, and are smaller then or comparable with the other filters for almost each experiment.

The accuracy of the covariances computed by various filters can not be checked by a simple comparison with a reference solution. Correct estimation of the converged covariance is not a target, unless the computed mean has converged too. A computed covariance might be very different from a reference, but could in fact be very accurate if it describes the error in the computed mean correctly. For a fair competition, a *computed covariance* should be compared with the *true covariance* of the computed mean, that is, the covariance of the true state around the computed mean (see also the Taylor expansions in appendix C). The true covariance is hard to obtain, however, since this requires precise knowledge of the true state. In practice, there are only two ways to compare the true and computed covariance. First, one could analyze the theoretical difference between them after application of a certain forecast method. For the first and second order linearizations, this has been carried out using Taylor expansions in appendix C. Second, the difference between true and computed covariance could be analyzed in a twin experiment, in which a true state is obtained from a model run with stochastic noise input. For an experiment with data from a measurement network and a stochastic model which is probably not perfect, the true covariance is hard to obtain, however.

A better way of judging the quality of the computed covariance is to check whether computed mean and covariance are able to explain the residue between observations and filter mean. If we neglect the spatial correlations between different measurement sites, the residue scaled by its computed standard deviation should have a standard normal distribution:

$$\frac{\mathbf{h}'\hat{\mathbf{x}} - y_i^o}{\sqrt{\mathbf{h}'\mathbf{P}\mathbf{h}' + r_i^2}} \sim \mathcal{N}(0,1) \tag{7.27}$$

If over a large number of measurements the average value of this ratio is far from zero, the filter solution is seriously biased from the measurements (at least during some occasions). If the standard deviation exceeds one, the filter is too optimistic about the quality of the mean: a measurement often differs more from the mean than what could be expected based on covariance and measurement error. The statistics of ratio (7.27) are therefore an indication for the quality of the computed covariance.

Figure 7.4 shows the statistics of ratio (7.27) during the experiments. In all experiments, the average ratio shows a small but positive bias (about 0.1). Detailed investigation of the assimilated ozone time series showed that this bias is caused by a period of over estimation in Yarner Wood during the first two days, and a rise of the simulated ozone level in Sibton during the morning which is too fast in comparison with the measurements. The forecast methods do not show large differences with respect to the bias. The overall smallest bias is

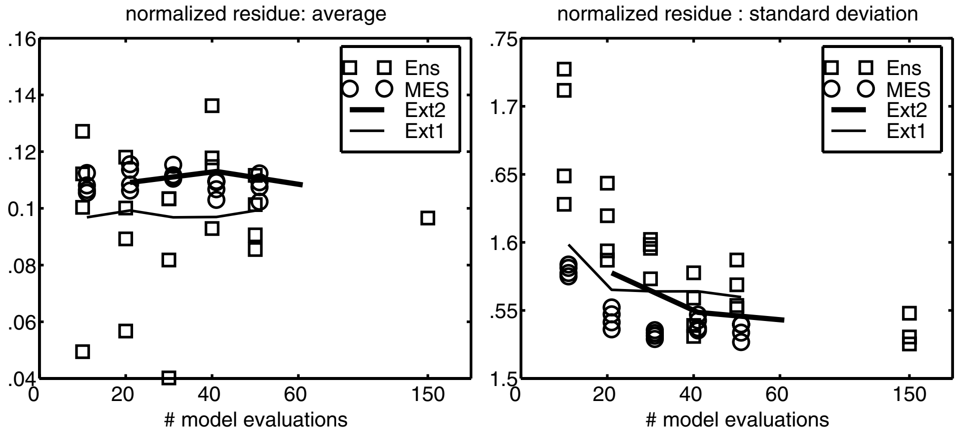


Figure 7.4: *Average and standard deviation of normalized residue (7.27) over all assimilated measurements.*

obtained with an ensemble filter, but given the large spread in the results this is only due to a lucky shot with the random generator.

The standard deviation of the normalized residue shows a clear convergence to a value of about 1.50-1.55. The difference between mean and observed ozone is therefore on average 50% larger than what could be expected given covariance matrix and measurement error. Underestimation of the standard deviation is an indication that the stochastic model is not perfect, which is not unusual during assimilation of data from an observation network (see for example (*Ménard et al., 1999*)). A standard deviation of about 1.50 seems to be the best what could be obtained for this experiment. The value of 1.50 is obtained for lowest costs with a minimal exact sample (MES). The MES method gives equal or better results for the same number of model evaluations, while the impact of the random generator is limited. The ensemble method suffers from a slow convergence; at least a double amount of model evaluations is required to obtain similar results as obtained with a MES. The methods with first (EXT1) and second order linearizations (EXT2) seem both to require at least 20 modes to converge, with the best results obtained for EXT2; both methods are not able to beat the MES method, however.

The difference between EKF2 and MES is remarkable, since both methods are based on exact propagation of second order nonlinearities. The only explanation is the different ways in which both methods introduce uncertainty. The EXT2 method introduces uncertainty in terms of unity vectors, varying one stochastic parameter at the time. Cross correlations are therefore neglected. For our stochastic model, cross correlations are important however. For example, increasing both NO_x and VOC emissions gives quite different results than increasing only one of them (see also figure 2.3). The MES method introduces uncertainties in all directions at the same time in different ratios, and the results in figure 7.4 show that this improves the results significantly.

7.5 Measuring nonlinearity

The previous described experiments with the filter around LOTOS showed that treating the nonlinearity problem with a different technique improves the results significantly. For the chosen stochastic model and measurement set, a MES forecast with 20-30 model evaluations was found to be the cheapest and most accurate technique. Although the experiments give some useful insight in the (dis)advantages of the different techniques, the results should not be interpreted as valid for other filter problems too. Even for the same filter problem but applied to another simulation period, the number of modes might to be increased, or an ensemble filter could be preferable. The only way to figure this out is to perform multiple filter experiments with very large ensembles. To avoid the time consuming procedure of multiple filter experiments, an ideal filter should be able to decide on its own whether nonlinearities should be treated with a sophisticated technique, or whether a more simplified approach is also accurate enough.

A first step to such a flexible filter was made in (*Verlaan and Heemink, 2001*). A method was proposed to analyze the nonlinearities in a filter problem in terms of single 'nonlinearity' number V. The value of V was shown to be sensitive to all aspects that define whether a nonlinear model complicates the filter problem or not. In the first place, this is the nonlinearity of the stochastic model, with respect to the basic time step; a pure transport model is much more linear than a model including chemistry. Second, the quality of the measurements is important, since assimilation is able to compensate for errors involved with the nonlinear model. And third, if high quality measurements are available, the time period between successive assimilations should be small enough. Experiments with the Lorenz model showed that for different parameter settings controlling these aspects, the error made by the filter is related to the value of V. Besides, for larger values (highly nonlinear problem), the more sophisticated forecast techniques provided more accurate results, while for small values (almost linear problem), the performance is indifferent. If the nonlinearity number is observed continuously during a filter run, an appropriate action could be undertaken if its value becomes too large.

7.5.1 Tracking the bias

The nonlinearity measure proposed in (*Verlaan and Heemink, 2001*) is based on tracking of the bias between true state and central forecast of the mean:

$$\mathbf{b}[k+1] \;=\; \mathrm{E}\left[\, \mathbf{x}^t[k+1] - \mathbf{M}(\hat{\mathbf{x}}[k]) \,\right] \tag{7.28}$$

A forecast equation for the bias is derived in appendix C.5 using Taylor expansions. Together with an analysis equation this gives the following system for tracking the bias:

$$\mathbf{b}^f[k+1] \;=\; \left.\frac{\partial \mathbf{M}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}^a[k]} \mathbf{b}^a[k] \;+\; \left[\, \hat{\mathbf{x}}^{f,\star}[k+1] \;-\; \mathbf{M}(\mathbf{x}^a[k]) \,\right] \tag{7.29a}$$

$$\mathbf{b}^a[k+1] \;=\; (\mathbf{I} - \mathbf{K}\mathbf{H}')\, \mathbf{b}^f[k+1] \tag{7.29b}$$

where $\hat{\mathbf{x}}^{f,\star}[k+1]$ is a higher order forecasts made with the EXT2, MES or ENS method. The bias is thus increased with the difference between a first and higher order forecast. For an almost linear model, this difference is close to zero, and the bias will converge to zero due to the analysis. For strongly nonlinear models, the bias increases during the forecast and will remain increasing unless the analysis is able to compensate for the growth.

The length of the bias is a suitable measure for the error in the filter due to nonlinearities. The definition used in (*Verlaan and Heemink, 2001*) computes the length of the bias relative to the covariance, which is a measure for the total error:

$$V = \|\boldsymbol{\beta}\| \qquad , \qquad \boldsymbol{\beta} = (\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'\mathbf{b} \tag{7.30}$$

The m-vector $\boldsymbol{\beta}$ contains the coefficients of the projection of $\mathbf{b}$ on the columns of $\mathbf{S}$. The measure V is not affected by linear state transformation and thus more or less independent of the exact definition of the state vector. A disadvantage of the nonlinearity number V is that it is sensitive to the number of elements in $\boldsymbol{\beta}$, however. Since this number is equal to the number of columns in $\mathbf{S}$, a growing number of modes will automatically lead to a larger value for V. To avoid this dependency, two additional measures are introduced which are related to V but independent of the number of modes. The measures compare the bias vector with the subspace in which the filter assimilates measurements (see figure 7.5). For a low-rank filter, the error between mean and true state is supposed to be a sample $\mathbf{Sw}$ for some $\mathbf{w} \sim \mathcal{N}(\mathbf{o}, \mathbf{I}_m)$. The bias should be small compared to $\mathbf{Sw}$, since otherwise the analysis is not able to account for the nonlinearity error. The first new measure compares the length of the projection vector $\boldsymbol{\beta}$ with a χ^2-distribution[2]:

$$p = \mathcal{P}\left(\chi^2_m \leq \boldsymbol{\beta}'\boldsymbol{\beta}\right) \tag{7.31}$$

If $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{o}, \mathbf{I}_m)$, then $\boldsymbol{\beta}'\boldsymbol{\beta} \sim \chi^2(m)$. Thus, if the projection of the bias is unlikely to be a sample out of $\mathcal{N}(\mathbf{o}, \mathbf{P})$, this will be visible in a large value for p. The second new measure computes the angle between the bias $\mathbf{b}$ and its projection $\mathbf{S}\boldsymbol{\beta}$ on the subspace spanned by $\mathbf{S}$:

$$\cos\theta = \frac{\|\mathbf{S}\boldsymbol{\beta}\|}{\|\mathbf{b}\|} \tag{7.32}$$

If a large part of the bias is not in the subspace spanned by the columns of $\mathbf{S}$, this will be visible in θ.

The major costs of computing the nonlinearity measures is formed by the propagation of the bias (7.29a), which requires one extra model evaluation for the finite difference approximation of (7.29a). Once the bias $\mathbf{b}$ has been analyzed, the $m \times m$ matrix $\mathbf{S}'\mathbf{S}$ and the m-vector $\mathbf{S}'\mathbf{b}$ need to be computed, to solve $\boldsymbol{\beta}$ from the matrix-vector equation $(\mathbf{S}'\mathbf{S})\boldsymbol{\beta} = (\mathbf{S}'\mathbf{b})$. Note that the matrix $\mathbf{S}'\mathbf{S}$ is rank deficient if the columns of $\mathbf{S}$ are deviations form a sample mean (MES or ENS forecast), such that the equation should be solved in least square sense. Finally, the measures V, θ, and p are to be computed from the l_2 norms over $\boldsymbol{\beta}$, $\mathbf{S}\boldsymbol{\beta}$, and $\mathbf{b}$.

7.5.2 Experiments with nonlinearity measures

The non-linearity numbers V, θ, and p have been computed during the experiments described in §7.4. The time series obtained for the reference runs (ensemble forecast with 150

---

[2] A chi-square random variable of order m is the sum $X_1^2 + \cdots + X_m^2$ of m uncorrelated $\mathcal{N}(0,1)$ distributed variables. The probability function is given by $\mathcal{P}\left(\chi^2_m \leq R^2\right) = \frac{1}{2}\int_{r=0}^{R^2}(r/2)^{m/2-1}e^{-r/2}\mathrm{d}r/\Gamma(m/2)$.

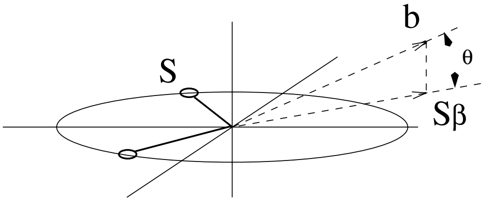


Figure 7.5: *Illustration of bias and projection on* **S**.

ensemble members, figure 7.6) are illustrative for the development of the nonlinearities. During the first four days, the length V of the bias is rather small and constant compared with the covariance, but during the following night and day, the bias seems to be rather large. The last days of the assimilation period were characterized by sun shine, leading to large ozone peaks during the afternoon and therefore large differences between day and night, making the nonlinear character much stronger. The same behavior was examined in (*Verlaan and Heemink, 2001*) for the Lorenz model, where V increased at critical points in the trajectories. The large values for V indicate again that the stochastic model is not completely able to explain the difference between model and measurements, as observed from figure 7.4 too.

Whether a value of V is 'acceptable' or 'too large' is provided by the value of p, which compares V with a χ^2 distribution (see also the dotted lines in the left panel of figure 7.6). The value of p acts as a binary switch: zero for a small bias, one for a large bias, and otherwise in between. As long as p is small enough, say less than 10^{-2}, the nonlinearity is not a major problem for the filter. The angle θ between bias and sub space spanned by the covariance is measured to be $20^\circ - 50^\circ$, but does not show a clear trend. Information contained in θ is therefore limited, and won't be discussed any more.

The trends observed in V and p for the reference run were found for the other filter experiments too, although in general the high values for V are reached sooner. Figure 7.7 shows the average values of the nonlinearity numbers during the assimilation period, for the filters based on higher order forecast techniques EXT2, MES, and ENS and different number of modes. The values of V observed for ensemble forecasts with small ensembles are extreme high; the bias seems to exceed the covariance during the complete assimilation period. However, the bias vector probably contains a large error in this case, since the ensemble mean used to compute the bias is far from converged. With growing ensemble size, the length of the bias vector shows a clear decreasing trend as expected; for 50 ensemble members, the length is on average in the interval covered by the χ^2 distribution.

The values of V observed for the second order EXT2 and MES methods show a clear increasing trend with growing number of modes. This is however not due to an decreasing accuracy, but simply the result of the growing size of the projection vector $\boldsymbol{\beta}$ from (7.30). The dependency on the size of $\boldsymbol{\beta}$ is canceled in measure p, and as the right panel of figure 7.7 shows, the impact of nonlinearities is in fact more constant under growing number of modes. About 40 model evaluations are required to reach converged values for p.

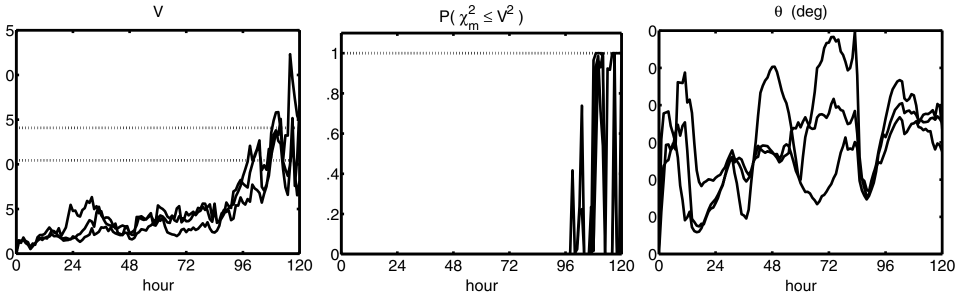


Figure 7.6: *Timeseries of non-linearity measures V, p, and θ, measured for the three reference runs with large ensembles (150 members); the three runs differ due the impact of the random generator. The dotted lines in the left panel denote the 99% confidence interval for $\|\boldsymbol{\beta}\|$ for a 150-element vector $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{o}, \mathbf{I})$; a bias with a V below this interval is small in comparison with the covariance. A close relation exists between V, the confidence interval, and p: if V is below, in, or above the interval, then p is close to zero, between zero and one, or close to one respectively.*

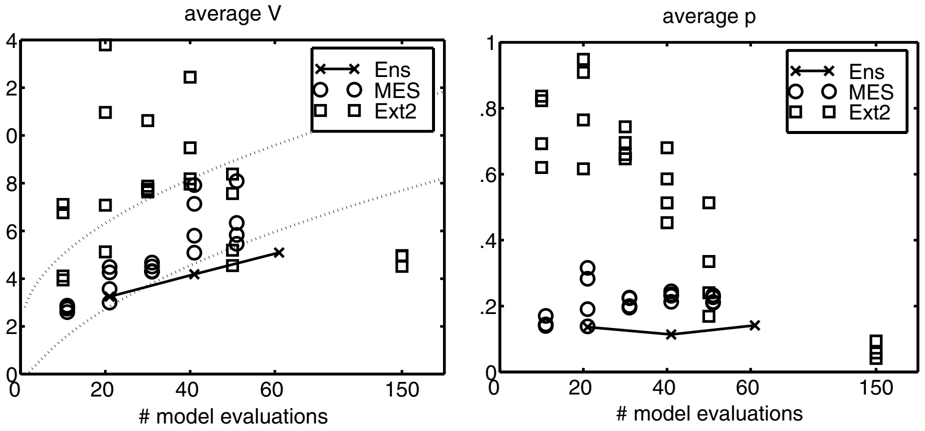


Figure 7.7: *Average nonlinearity measures V and p during assimilation experiments with* EXT2, MES, *or* ENS *forecasts. Similar as in figure 7.6, the dotted lines in the left panel denote the 99% confidence interval for a χ^2_m variable, where m is set to the number of model evaluations.*

The differences in nonlinearity measure p observed for the different forecasts techniques show a large correspondence with the plots of the RMS error in the mean (figure 7.3) and the average error plotted in the right panel of figure 7.4. For example, the nonlinearities measured for ensemble filters exceed the values for other methods for small numbers of model evaluations, but converge slowly to reference values which are less then all other. The value of p measured for the MES filter is converged for 30 or more model evaluations, and similar has the total error. A difference is that the nonlinearities measured for the MES filters exceed those measured for EXT2, while the errors are lower. Investigation of the time series learned that the values of V for both methods act quite similar, but that whenever they exceed the χ^2 interval, the extrema for MES are much higher than those for EXT2. If the probability that p exceeds a certain threshold is plotted against against the number of model evaluations, the result is almost the same as a plot of the error.

The correlation between error and observed nonlinearity might become a useful feature in online applications. Where the errors are computed from comparison with a reference run or measurement data, the nonlinearity is observed without using any of these external information. In online applications this is very useful, since a reference run is not available and measurement data might be sparse or of low quality during longer periods. By tracking and observing the bias, the filter has however insight in its own accuracy, and is able to perform appropriate action if V or p exceed critical levels. The filter might decide to increase the accuracy by increasing the ensemble size or number of modes, or to switch to another forecast scheme. This application, decision of which forecast scheme to use through observation of the bias, was an important goal in (*Verlaan and Heemink, 2001*). In our assimilation experiments, the value of the nonlinearity measure p could have been used to decide on the appropriate number of modes or ensemble members. Comparison of the p-measures in figure 7.7 with the errors in figures 7.3 and 7.4 suggest for example that p should be less than 0.3 for accurate results. If the filter is initiated with an ENS-forecast with 40 ensemble members, the value of p is about $0.4-0.8$. This indicates that 40 members is to low; the ensemble size should be increased to at least 60-80 members to obtain a $p < 0.3$. Another option is to switch to a MES or EXT2 scheme, since these provide for this particular application an appropriate p even for 40 model evaluations.

7.6 Summary and conclusions

In this chapter, approximations to the Kalman filter have been discussed, that are able to deal with a nonlinear model. Models such as the atmospheric chemistry model LOTOS are characterized by strong nonlinear dynamics. In contrast with linear models, a probability density of the state can not be propagated exactly for this kind of dynamics. The use of approximate methods in the forecast step of the Kalman filter is therefore necessary; the assimilation of observation is however not influenced by the nonlinearities however.

A comparison has been made between four nonlinear forecast methods, each of them embedded in a low-rank filter. The ensemble forecast has the advantage of a simple implementation, and produces the most accurate results if the ensemble size is large enough. Methods based on first or second order linearizations or a minimal exact sample have the disadvantage of a more complicated implementation, but limit the computational costs.

Experiments with the LOTOS model showed that the method of minimal exact sampling (MES) is the most efficient and accurate. With similar costs as made for first order linearizations (EXT1), the MES method produces more accurate results. Minimal exact sampling is therefore always preferred over first order linearizations. The method of second order linearizations (EXT2) has the same accuracy as a MES in theory, but is twice times as expensive. The experiments with the filter around the LOTOS model showed that the EXT2 method is less accurate if cross correlations in the dynamic noise are more important. The ensemble forecast (ENS) was able to provide results even more accurate than a MES forecast, but only with the cost of at least a double amount of model evaluations. Even for ensembles with 50 members, statistical noise still dominated the results, while the MES method converged for less than 30 modes. If the resources for computation time and storage are not restricted, the ensemble forecast is to be preferred for use in a Kalman filter; otherwise, minimal exact sampling is an useful alternative.

Tracking and observation of the bias is able to provide useful insight in the nonlinearity present in the filter. The ratio between bias and covariance gives an indication whether assimilation of measurements is able to compensate for errors made due to nonlinearities. The experiments with the filter around LOTOS showed that a clear correlation exists between observed bias and total error. This feature might become useful in online applications, to let the filter automatically increase the accuracy if the bias (and thus the total error) starts to increase.

Chapter 8

Parallelization of low-rank filters

For implementation on a parallel computer, the low-rank filter around the LOTOS *model has been parallelized in two ways. First, the covariance square root of the filter was decomposed over the modes, without any change to the model. This method has the advantage of an efficient parallelization of the forecast stage of the filter, but the disadvantage of a more complicated implementation of the filters matrix algebra. Second, the filter has been implemented in combination with a parallel, domain-decomposed version of the model, which leads to efficient matrix algebra but a less efficient forecast stage. The performance of the parallelization of different filter components has been analyzed in terms of speedup and total execution time. The domain-decomposition is slightly favored as the best parallelization strategy based on speedup, flexibility, and implementation.* [1]

8.1 Introduction

Application of a data-assimilation tool to large scale models puts a large demand on computing power. In practice, the demand is always ahead of the available computing devices, because both the need for assimilation has grown, and the underlying models have become more extensive. The availability of computing power therefore provides a limit to the assimilation problems which can be solved. The chosen implementation should explore the capacities of the platform as much as possible.

The growing availability of fast multiprocessor machines encouraged the application of the Kalman filter technique to large models. In fact, without the use of these machines, some filter problems can not be solved at all. Online application of filter techniques for periodic forecasts are bounded by the time period in which the problem has to be solved, and this task is often not feasible for a single processor. Besides, some applications do simply not fit in the memory which can be addressed by a single processor. An implementation with not frequently accessed entities stored on external devices could be considered here, but this will extremely slow down the application. An early example of parallel implementation of a Kalman filter is found in (*Lyster et al., 1997*). Measurements from the UARS satellite

---

[1] To appear as *Parallelization of a large scale Kalman filter: comparison between mode- and domain-decomposition* by A.J. Segers and A.W. Heemink (2002). In Wilders, P., Ecer, A., Periaux, J., Satofuka, N., and Fox, P., editors, Parallel Computational Fluid Dynamics: Recent developments and applications, Proceedings of the Parallel CFD'01 Conference. Elsevier.

were assimilated with a global 2D model by a filter with a full-rank covariance matrix, with either the model parallelized over its domain, or through processing the model on parts of the covariance matrix in parallel. In both methods, the model had to be evaluated $2n$ times, with n the number of elements in the state (here $1.3 \cdot 10^4$). The analysis of measurements turned out to be a bottleneck for the speedup of the parallel filter.

A fundamental change in parallel implementation of large scale Kalman filters was the introduction of low-rank filters (see chapter 6 for an extensive overview). Instead of operating and storing a full-rank covariance matrix, a limited ensemble of state vectors is used to describe the correlations. In all low-rank formulations, the ensemble members need to be propagated in time by the dynamic model, and this is often the computational most expensive part of the filter. In the context of the Ensemble Kalman filter, (*Evensen, 1994*) proposed to propagate the ensemble members independently by multiple processors. This strategy leaves the model intact, and could therefore be applied to any available model immediately. The approach was used in (*Keppenne, 2000*) for a large 2 layer shallow water model, assigning one ensemble member to each processor. The analysis equations are however not easily solved in this configurations, since analysis of each single measurement requires data from all ensemble members. Therefore, the analysis step was implemented using a domain-decomposition approach, where each processor analyzed the measurements in circular, overlapping regions. A similar method was used for a parallel ensemble filter around a much larger ocean general circulation model (*Keppenne and Rienecker, 2000*), where each ensemble member needed to be decomposed over 8 processors.

The rather inefficient implementation of the analysis step is now recognized as a major disadvantage of an ensemble-decomposed parallel filter, especially when large amounts of measurements are to be analyzed. In (*Houtekamer and Mitchell, 1998; Houtekamer and Mitchell, 2001*) a method is proposed for online analysis of order 10^4 measurements. The method consists of analysis of batches of observations located in disjunct areas, where spatial correlations between grid cells and observations are ignored after some distance. These batches are efficiently processed in parallel, if the ensemble is decomposed over the domain rather than over the ensemble members. Application of this approach is therefore limited to models for which a domain-decomposition is available.

The two different approaches for a parallel filter, decomposition over the modes or over the model domain, have both been implemented for the filter around the LOTOS model (chapters 4 and 5). Some general remarks about parallel computing are made in section 8.2. In section 8.3, the general equations for a low-rank filter are summarized. Then, in section 8.4, a decomposition of the filter over the modes is described, where the LOTOS model remains unchanged. Sections 8.5 and 8.6 describe a domain-decomposition of LOTOS and the filter respectively. Both techniques have been implemented and analyzed on a massive parallel machine (CRAY T3E). Finally, the two methods are compared based on computational characteristics, ease of use, and flexibility.

8.2 Parallel computing

Thanks to the large variety in computer architectures and related software libraries, a simple overview of how to solve a problem in parallel is hard to give. Often, the decision of how the problem is to be solved is not given by what is possible but more by what is available to a user.

In this research we had access to the CRAY T3E massive parallel computer of Delft University. The T3E consists of a scalable number of processors (128 at time of writing), interconnected by a fast network with 3-D torus topology (see figure 8.1). Each processor is equipped with 128 Mb local memory, and is able to access the local memory on other processors through the network. According to different classification schemes, the architecture of the T3E can be classified as:

- Multiple instruction stream, multiple data stream (MIMD).
  Each processor is able to perform different kinds of instructions on different memory items, independent of other processors.
- Logically shared, physically distributed memory.
  Each processor is able to access all memory addresses, although only a small part is stored locally.
- Non uniform memory access (NUMA).
  The access times for different parts of the memory is not uniform, but depends on the physical distance. In the 3-D torus, each node is connected directly to 6 other nodes, and access to memory at these nodes will be faster than access to memory on more remote nodes.

The MIMD/NUMA architecture is often used in modern supercomputers, but also describes the architecture of a cluster of workstations or PC's. The configuration of processor/memory nodes and interconnection network is extended rather easily, such that the machine is able to met growing needs by including an extra set of nodes. The architecture of the CRAY T3E allows the programming style of Single Program/Multiple Data (SPMD): the same compiled program is executed on all processors, but each processor is responsible for a different part of the data. SPMD does not imply that each processor at any time executes the same instruction. At run time, each of the running programs obtains a key identifying the processor on which it is running. Given the value of this key, the program might decide to execute certain statements or not. Software libraries are available to send data to or to receive data from other processors, and to synchronize the execution of the program.

The architecture of the CRAY T3E is exploited by using the logically shared, distributed memory access (SHMEM) library for the communication between the processors. SHMEM routines are based on message passing, and without major changes, the communication could be performed with the low level but more general Message Passing Interface (MPI) library too.

To judge the performance of a parallel algorithm, a number of measures is available. The **speedup** denotes how much faster a certain task is executed in parallel in comparison with sequential execution. The speedup is computed as the ratio between computation time T_1 spent on the task by a single processor, and the time T_p spent by p processors:

$$S(p) = \frac{T_1}{T_p} \tag{8.1}$$

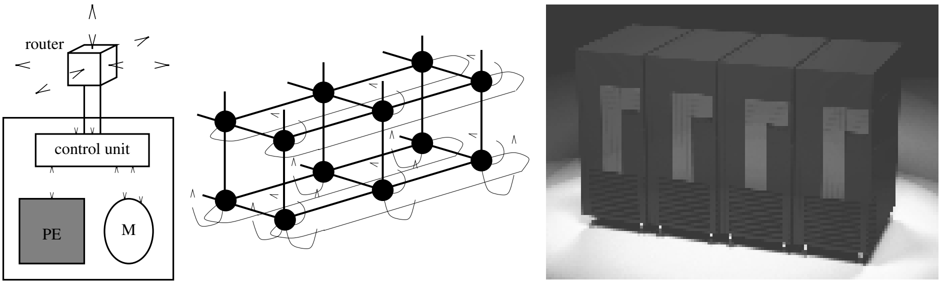


Figure 8.1: *Architecture of the* CRAY T3E. *Each node (left) consists of a processing element (*PE*), local memory (*M*), and a* CONTROL UNIT *for connection between processor, memory, and the communication network. The nodes are connected in a 3D torus topology (middle). The photo on the right shows an example of a* CRAY T3E *as installed at the center for* High Performance Applied Computing *(*HPαC*) at Delft University.*

An optimal speedup is achieved if $S(p) = p$, that is, a double number of processors performs a certain task twice as fast. If the problem requires that the processors communicate with eachother, the speedup curve will flatten for $p \longrightarrow \infty$, since a growing number of processors will lead to more communication. While the speedup curve is still increasing, a problem will be solved faster if more processors are used. If communication is not the bottleneck for the parallelization, a particular algorithm might even show a *super linear* speedup: the problem is solved faster than what could be expected based on the number of processors only. A super linear speedup is observed if the processor is able to make more efficient use of fast memory (cache), for example if the total amount of data managed by the processor is small. Occurrence of this effect is more related to the hardware/software configuration than to the algorithm.

Related to speedup is **efficiency**, which is defined as the ratio of the time spent on a problem by a single processor, and the time spent by p processors together:

$$E(p) = \frac{T_1}{p \cdot T_p} = \frac{S(p)}{p} \tag{8.2}$$

The efficiency describes which fraction of the total consumed cpu time is used to solve the actual problem (workload). What remains is overhead due to communication time or load-imbalance (processors might become unemployed if other processors have to complete their tasks first). If computing facilities are rented on a commercial base, the efficiency gives insight in how effective the money is spent.

8.3 Filter equations and covariance decomposition

The filter equations used in this chapter are based on the general formulae for low-rank filters (§6.8). The model/observation pair with a general nonlinear model is given by:

$$
\begin{aligned}
\mathbf{x}[k{+}1] &= \mathbf{M}(t[k],\mathbf{x}[k]) + \boldsymbol{\eta}[k] \quad , \quad \boldsymbol{\eta}[k] \sim \mathcal{N}(\mathbf{o},\mathbf{Q}[k]) && \text{(8.3a)}\\
\mathbf{y}^o[k] &= \mathbf{H}'[k]\,\mathbf{x}[k] + \mathbf{v}[k] \quad , \quad \mathbf{v}[k] \sim \mathcal{N}(\mathbf{o},\mathbf{R}[k]) && \text{(8.3b)}
\end{aligned}
$$

In following equations, the time identifications for $\mathbf{M}$ and $\mathbf{H}'$ will be skipped, if the time is implied by the arguments. The target of the filter is to compute a mean $\hat{\mathbf{x}}[k]$ and covariance $\mathbf{P}[k]$ for the true state given the model and the observations. The covariance matrix is parameterized with the factorization $\mathbf{P} = \mathbf{S}\mathbf{S}'$, where $\mathbf{S}$ is the low-rank covariance square root. In chapters 6 and 7, a low-rank filter based on the RRSQRT approach and a forecast using a minimal exact sample was found to be suitable for assimilation of data in LOTOS. The filter equations are summarized to:

$$
\begin{aligned}
\textbf{forecast:}\quad & \boldsymbol{\xi}_j[k] = \hat{\mathbf{x}}^a[k] + \varepsilon\,\mathbf{s}^a_j[k] \quad , \quad j = 1,\ldots,m && \text{(8.4a)}\\
& \boldsymbol{\xi}_j[k{+}1] = \mathbf{M}(\boldsymbol{\xi}_j[k]) + \boldsymbol{\eta}_j[k] \quad , \quad j = 1,\ldots,m && \text{(8.4b)}\\
& \hat{\mathbf{x}}^f[k{+}1] = \overline{\boldsymbol{\xi}_j[k{+}1]} && \text{(8.4c)}\\
& \mathbf{S}^f[k{+}1] = (\,[..,\boldsymbol{\xi}_j[k{+}1],..] - \hat{\mathbf{x}}^f[k{+}1]\,)/\varepsilon && \text{(8.4d)}\\
\textbf{analysis:}\quad & \boldsymbol{\Psi}' = \mathbf{H}'\,\mathbf{S}^f[k{+}1] && \text{(8.4e)}\\
& \boldsymbol{\Theta} = \boldsymbol{\Psi}(\boldsymbol{\Psi}'\boldsymbol{\Psi} + \mathbf{R}[k{+}1])^{-1} && \text{(8.4f)}\\
& \mathbf{a} = \boldsymbol{\Theta}\,(\,\mathbf{y}^o[k{+}1] - \mathbf{H}'\,\hat{\mathbf{x}}^f[k{+}1]\,) && \text{(8.4g)}\\
& \mathbf{B}\mathbf{B}' = \mathbf{I} - \boldsymbol{\Theta}\,\boldsymbol{\Psi}' && \text{(8.4h)}\\
& \hat{\mathbf{x}}^a[k{+}1] = \hat{\mathbf{x}}^f[k{+}1] + \mathbf{S}^f[k{+}1]\,\mathbf{a} && \text{(8.4i)}\\
\textbf{rank reduction:}\quad & \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}' = \mathbf{B}'\,(\,\mathbf{S}^f[k{+}1]'\,\mathbf{S}^f[k{+}1]\,)\,\mathbf{B} && \text{(8.4j)}\\
\textbf{transformation:}\quad & \mathbf{S}^a[k{+}1] = \mathbf{S}^f[k{+}1]\big(\mathbf{B}\tilde{\mathbf{V}}\boldsymbol{\Omega}\big) && \text{(8.4k)}\\
\textbf{diagonal:}\quad & \mathbf{d}[k{+}1] = \mathrm{diag}(\mathbf{S}^a[k{+}1]\mathbf{S}^a[k{+}1]') && \text{(8.4l)}
\end{aligned}
$$

The listed equations cover the main operations performed in other filter algorithms too; see tables 7.2 and 6.1 for the corresponding choices for $\boldsymbol{\Omega}$, $\mathbf{a}$, and $\mathbf{B}$. Transformation (8.4k) combines three matrix-matrix multiplications originally part of the analysis ($\mathbf{B}$), rank reduction ($\tilde{\mathbf{V}}$), and preparation of the forecast ensemble ($\boldsymbol{\Omega}$). The computation of the diagonal of the covariance matrix is added as an extra stage. In typical applications, at least a small part of the diagonal is used for output; to investigate the maximum costs of this operation, the complete diagonal was computed here.

Since the major workload in the filter is related to the covariance square root $\mathbf{S}$, a parallel version of the filter should distribute the tasks related to $\mathbf{S}$ over the processors. The non-uniform memory access of the CRAY T3E (and other frequently used platforms) requires thereto that each processor has quick access to that part of $\mathbf{S}$, for which tasks are to be performed. Therefore, the covariance matrix has to be decomposed and distributed over the processors (if the size of $\mathbf{S}$ is too large to fit in the local memory of a single processor, this would be necessary anyway). Figure 8.2 illustrates two options considered: a decomposition

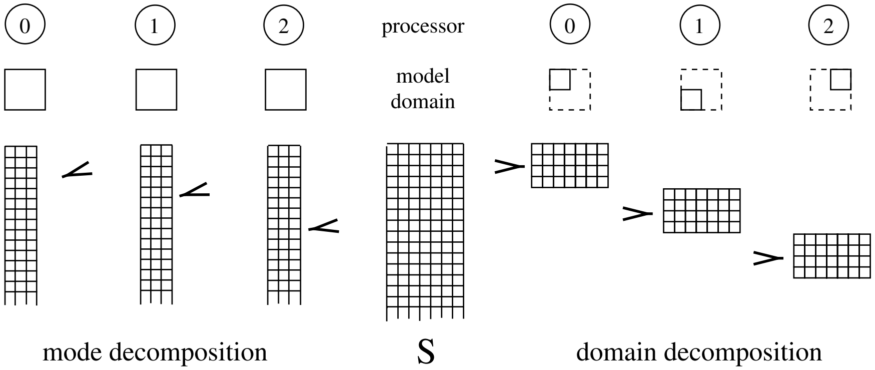


Figure 8.2: *Decomposition of the covariance square root: over the modes (column-wise) or over the domain (row-wise).*

over the columns, where each processor owns a number of modes of the covariance, and a decomposition over the rows, where each processor is responsible for a certain part of the model domain. Some smaller entities will not be decomposed or distributed, but each processor will own a complete copy. If each processor updates these entities in the same way, some work is done double, but this is often more efficient than decomposition with the cost of communication. The implementation of the two strategies for a parallel filter, mode-decomposition and domain-decomposition, are discussed in the following sections.

8.4 Parallel filter: decomposition over the modes

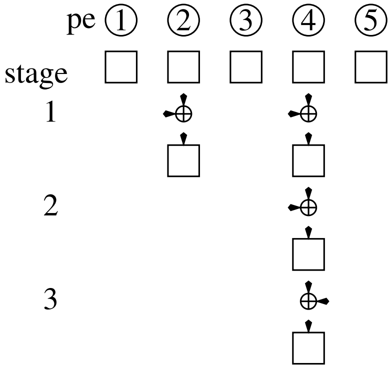
A filter based on a mode-decomposition explores the natural parallelism of the forecast. Eq. (8.4b) requires a large number of similar model propagations; these could be processed in parallel without interaction. An almost optimal speedup is expected for the forecast stage, and since this is the major time consumer, the speedup of the filter is expected to be large.

The strategy of a mode-decomposition is best interpreted in the view of the ensemble filter (section 6.6). Each processor should manage the propagation and analysis of a certain number of ensemble members. Thereto, each processor owns a complete copy of the model dynamics and model data (meteo, land use, etc). If the ensemble members are distributed equally, the time required for a single filter step is the same for each processor. In terms of a covariance square root, the decomposition of the ensemble is equal to distribution of the columns of $\mathbf{S}$ over the processors (figure 8.2). To let each processor own the same number of states, the number of modes should be a multiple of the number of processors n_{pe}, which is often a power of 2. For efficient implementation of operations (8.4a) and (8.4d) on the forecast ensemble, each processor should be able to store a copy of the mean state too.

The next paragraphs contain a description of the filter operations for a mode-decomposed covariance square root. The following notations and conventions will be used:

- a processor is identified by a key $k \in \mathcal{K}$, were $\mathcal{K}$ owns n_{pe} elements;

Figure 8.3: Adding of a set of state vectors, distributed over five processors. During each stage of the algorithm, pairs of processors cooperate in summing their states. Some processors might be unemployed for a while if it does not own a state to be summed. The total number of stages is equal to $\lceil {}^2\log(n_{pe}) \rceil$.



- the modes stored on a processor k are identified by the set $\mathcal{J}_k$; the sets $\mathcal{J}_k$ for $k = 1,..,n_{pe}$ form disjunct subsets of $\mathcal{J} = \{1,\ldots,m\}$;
- the loop '`for` $j \in \mathcal{J}_k$' denotes that j is set sequentially to an element of $\mathcal{J}_k$;
  the loop '`for all` $k \in \mathcal{K}$' denotes that the loop is performed in parallel, for each processor with a different k at the same time;
- 'receiving' and 'sending' denotes transfer of data between processors.

8.4.1 Adding distributed states

A simple operation, which often occurs in filter operations, is computation of the sum over a set of states vectors. If the set of states $\{\mathbf{x}_k\}$ is distributed over multiple processors (one state for each pe k), the following notation for this global sum will be used:

$$\textbf{sum(}\mathbf{x}_k\textbf{,}\mathcal{K}\textbf{,}k_{dest}\textbf{)} = \sum_{k \in \mathcal{K}} \mathbf{x}_k \tag{8.5}$$

where the result is stored in $\mathbf{x}_k$ on processor k_{dest}. Figure 8.3 illustrates a recursive algorithm (*Foster, 1995*) which could be used used to perform operation (8.5). The number of recursive stages is equal to $\lceil {}^2\log(n_{pe}) \rceil$. During each stage, at least one state vector is to be transfered from one processor to another. For large state vectors, the total costs of the adding is almost completely determined by the communication time required to transfer the states. Routines for computing the global sum are often provided by the communication library in optimized form.

8.4.2 Operations on the covariance square root

The decomposition of the covariance square root has important consequences for the implementation of the various filter operations. In this section we will give an overview of the implementation of the basic operations on $\mathbf{S}$.

$\boldsymbol{\Psi}' = \mathbf{H}'\mathbf{S}$

This operation is performed during the analysis in eq. (8.4e). The result is a matrix

$\boldsymbol{\Psi}'$ with the projections of the modes on the observation space. Interpolation matrix $\mathbf{H}'$ is often sparse; each row corresponds with a single observation, and contains often only one element unequal to zero. Each processor should project the locally stored columns on each availabe observation. The corresponding columns ψ'_j of $\boldsymbol{\Psi}'$ are send to the other processors:

```
Require: $\mathbf{S}$ distributed column-wise, $\mathbf{H}'$ available on each pe
  for all $k \in \mathcal{K}$ do
    for $j \in \mathcal{J}_k$ do
      $\psi'_j = \mathbf{H}'\mathbf{s}_j$
      send $\psi'_j$ to all $k \in \mathcal{K}$
    end for
  end for
Ensure: $\boldsymbol{\Psi}' = [\,\ldots\ \psi'_{\mathcal{J}}\ \ldots\,] = \mathbf{H}'\mathbf{S}$ on all pe's
```

$\mathbf{x} = \mathbf{S}\mathbf{a}$

The result of this operation is a linear combination of columns of **S**, and is for example used to analyze the mean state in eq. (8.4i). Taking the sample mean over all modes is a special form of **Sa**, with each element of **a** equal to $1/m$. Each processor could compute a part of the result for the modes stored in the local memory; the final result is computed with a global sum:

```
Require: $\mathbf{S}$ distributed column-wise, $\mathbf{a}$ available on each pe
  for all $k \in \mathcal{K}$ do
    $\mathbf{x}_k = \sum_{j \in \mathcal{J}_k} a_j \mathbf{s}_j$
  end for
  $\mathbf{x} = \mathrm{sum}(\mathbf{x}_k, k \in \mathcal{K}, k_{dest})$
Ensure: $\mathbf{x} = \mathbf{S}\mathbf{a}$ available on pe $k_{dest}$
```

The costs of this operation are determined by the communication time for the global sum. If the vector **a** contains many zero elements, the number of processors involved in the operations might be lower than n_{pe}.

$\mathbf{S} := \mathbf{S}\mathbf{C}$

This operation is a major time consumer in a filter operation, and has to be performed at least one time during the transformation in eq. (8.4k), with $\mathbf{C} = \mathbf{B}\tilde{\mathbf{V}}\boldsymbol{\Omega}$. Each column of **S** is replaced by a linear combination of all (other) columns, which makes the operation difficult for two reasons. First, a column of **S** could not be replaced by a new version, since the original value is necessary for computation of other new columns too. To avoid this, one could chose to store two copies of **S** (the old an the new one), but this is not preferable for the massive increase of memory consumption. Second, since the columns are distributed over the processors, communication can not be avoided. Two algorithms have been examined: one which is independent of the size and contents of the state vector, and one which decomposes **S** in blocks of rows.

1. State independent **SC**.

   This algorithm is based on the idea that the transformation should be independent from the exact size and shape of the state vector. If a change in the model leads to a changed state vector, this should not lead to any changes in the filter operations. All that is required is that the state is a vector, with suitable definitions for adding and scalar multiplication. The matrix **C** is first decomposed into **LU**, the product of a lower and upper triangular matrix. Note that **C** might have a rank less than m. For example, the matrix **B** from analysis eq. (8.4h) has a rank equal to the number of observations, if this number is less than the number of modes. The actual transformation **SC** is now performed through first replacing **S** by **SL** and then by $(\mathbf{SL})\mathbf{U}$. This methods avoids the storage of two covariance square roots, since each new column is formed from a linear combination of columns not transformed yet:

   ```
   Require: S distributed column-wise, C available on each pe
     for all $k \in \mathcal{K}$ do
       $\mathbf{LU} = \mathbf{C}$
       for $j = 1, \ldots, m$ do
         $\mathbf{s}_j := \mathbf{Sl}_j$
       end for
       for $j = m, \ldots, 1$ do
         $\mathbf{s}_j := \mathbf{Su}_j$
       end for
     end for
   Ensure: $\mathbf{S} := \mathbf{SC}$
   ```

   In total $2m$ operations of the form **Sa** have to be applied, where **a** is a column of **L** or **U**. The costs of one of such operations is determined by the number of state-transfer stages, with a maximum of $\lceil {}^2\log(n_{pe}) \rceil$ stages if all processors are involved. If each processor owns a set of sequential numbered columns, the number of processors involved in the transformation decreases with the number of zero elements in $\mathbf{l}_j$ or $\mathbf{u}_j$. A simple calculation shows that for a scenario with exactly one mode per processor, the operation **SL** is performed with the cost of $\phi_k = 1$, 5, 17, or 49 transfer stages for a number of processors $k = 2$, 4, 8, or 16 respectively. If the number of modes m exceeds the number of processors, the number of stages increases with about m/n_{pe}. The total communication costs of multiplication with **L** and **U** is the equivalent of $2\phi_k m/n_{pe}$ state transfers.

   Since the communication costs grow dramatically with the number of processors, the expected speedup of the state independent transformation is not very good. Many processors are unemployed if the columns $\mathbf{l}_j$ and $\mathbf{u}_j$ contain many zeros, and have to wait until other processors have finished their jobs.

2. Decomposition over the rows.

   Instead of treating the transformation **SC** as acting on the columns of **S**, the operation could also be viewed as an operation on the rows. After the transformation, a single row $\mathbf{s}_i'$ has been replaced by the vector-matrix product $\mathbf{s}_i'\mathbf{C}$. Each processor could perform a number of these multiplications; a row should be collected first, transformed, and redistributed afterwards. For communication it is more efficient to transfer a block of rows rather than a large number of single rows. The maximum number of rows in a block is about n/n_{pe}, with n the number of elements in the state; for small numbers of processors it is more efficient to use a smaller number since each block has to be stored completely in the local memory.

Let each block of rows be identified by a key $l \in \mathcal{L}$; each processor k is expected to collect, transform, and redistribute the blocks for all l in a subset $\mathcal{L}_k$. The notation $(\mathbf{S})_{kl}$ is introduced for the block l stored on processor k:

```
Require: $\mathbf{S}$ distributed column-wise, $\mathbf{C}$ available on each pe
  for all $k \in \mathcal{K}$ do
    for $l \in \mathcal{L}_k$ do
      receive $(\mathbf{S})_{kl}$ from all $k \in \mathcal{K}$ in $\Sigma_l = [\ldots, (\mathbf{S})_{kl}, \ldots]$
      $\Sigma_l := \Sigma_l \mathbf{C}$
      send all $(\mathbf{S})_{kl}$ from $\Sigma_l$ to all $k \in \mathcal{K}$
    end for
  end for
Ensure: $\mathbf{S} := \mathbf{SC}$
```

Since each row is collected and distributed once, a total amount of $2m(n_{pe} - 1)/n_{pe}$ states is transfered between the processors. If only one pair of processors is able to communicate at the same time, the communication time is the equivalent of the same number of state transfers; if each processor is able to communicate directly with all other processors at the same time, the communication time is decreased with a factor $n_{pe} - 1$. In the 3-D torus network on the T3E, the total communication time will be in between the costs of $2m/n_{pe}$ and $2(n_{pe} - 1)m/n_{pe}$ state transfers.

Comparison of the communication costs for the two algorithms shows that the row-decomposition is always cheaper than the state-independent algorithm: the factor ϕ_k for the costs of the later makes the difference. In a non-fictive configuration with about 100 modes managed by 8 processors, the operation $(\mathbf{SL})\mathbf{U}$ with the state-independent algorithm would require more than 400 transfer stages; for 16 processors, this number has grown to more than 1200. For the row-decomposition, however, the communication costs are even for 16 pe's in the range of 25-200 transfer stages only.

d =diag(SS′)

The diagonal of the covariance matrix is often the minimum of what is extracted from the covariance matrix as output; storage of the complete covariance square root each hour is often not necessary and feasible. The diagonal is equal to a global sum of all columns, squared element wise; in simplified notation:

```
Require: $\mathbf{S}$ distributed column-wise
  for all $k \in \mathcal{K}$ do
    $\mathbf{d}_k = \sum_{j \in \mathcal{J}_k} \mathbf{s}_j^2$
  end for
```

```
    $\mathbf{d} = \mathbf{sum}(\mathbf{d}_k, k \in \mathcal{K}, k_{dest})$
Ensure: $d$ =diag(SS')
```

$\mathbf{A} = \mathbf{S}'\mathbf{S}$

The matrix $\mathbf{S}'\mathbf{S}$ is used during the rank reduction of $\mathbf{S}$ in a RRSQRT filter, and for projections on the space spanned by the columns of $\mathbf{S}$ (for example used in POENK filter, see §6.7). An element a_{ij} is equal to the dot-product between modes $\mathbf{s}_i$ and $\mathbf{s}_j$. Since these might be stored on different processors, this operation requires the transfer of many state vectors:

```
Require: S distributed column-wise
  for all $k \in \mathcal{K}$ do
    for $i = 1, \ldots, m$ do
      for $j = i, \ldots, m$ do
        if $j \in \mathcal{J}_k$ then
          if $i \notin \mathcal{J}_k$ receive $\mathbf{s}_i$
          $a_{ij} = a_{ji} = \mathbf{s}_i{}'\mathbf{s}_j$
          send $a_{ij}, a_{ji}$ to all pe
        end if
      end for
    end for
  end for
Ensure: $\mathbf{A} = \mathbf{S}'\mathbf{S}$
```

If each processor owns an equal number of modes, the communication time required for this procedure is the equivalent of $(n_{pe} - 1)m/2$ state transfers; the costs grow linear with the number of processors. A less expensive algorithm is to collect rows of $\mathbf{S}$ on different processors (similar as for $\mathbf{SC}$), and to perform the operation row-wise:

$$\mathbf{S}'\mathbf{S} = \sum_{i=1}^{n} \mathbf{S}'_{(i,:)}\mathbf{S}_{(i,:)} \tag{8.6}$$

Since the rows do not have to be redistributed, the communication time of a row-wise $\mathbf{S}'\mathbf{S}$ is half the communication time of a row-wise $\mathbf{S} := \mathbf{SC}$, thus $\mathcal{O}(m)$ rather than $\mathcal{O}(m\, n_{pe})$ required for the first algorithm.

$\mathbf{a} = \mathbf{S}'\mathbf{x}$

This operation is used for projections on the subspace spanned by the columns of $\mathbf{S}$, see for example eq. (6.23d) for the POENK filter, or eq. (7.30) for the nonlinearity number. If each processor owns a copy of $\mathbf{x}$, the major part of this operation could be performed locally, and only a few elements of $\mathbf{a}$ have to be transfered; otherwise, first $\mathbf{x}$ should be copied to each processor:

```
Require: S distributed column-wise, x available on each pe
  for all $k \in \mathcal{K}$ do
    for all $j \in \mathcal{J}_k$ do
      $a_j = \mathbf{s}'_j\mathbf{x}$
```

```
            send $a_j$
        end for
    end for
Ensure: $\mathbf{a} = \mathbf{S}'\mathbf{x}$
```

8.4.3 Performance of the mode-decomposed filter

The performance of a mode-decomposed filter was tested during assimilation experiments with the LOTOS model. The experimental setup was almost similar as described in chapter 5. To be able to run the complete filter on a single processor too, the model domain was limited to an area of 32×32 grid cells. A number of 9 uncertain parameters was included in the stochastic model (NO_x and VOC emissions and deposition rates in three different area). During each time step, the number of modes was reduced to 61; from these, a forecast ensemble with 62 members is formed, and together with the filter mean and a deterministic model run for diagnose, this leads to a total number of 64 model evaluations per time step. The number of 64 evaluations ensures that the number of model evaluations could be distributed equally over the processors, if their number is equal to a power of 2. In our experiments, the filter run on either 1, 2, 4, 8, 16, or 32 processors. A number of 23 measurements was assimilated each hour. Since this number is rather limited, no special treatment of the analysis is considered. Each of the processors computes the vector **a** and the matrix **B** from equations (8.4g-8.4h) in the same way, using the algorithm in appendix B. Computation times for the filter have been measured for an assimilation experiment over 12 hours, after an initialization period to obtain a covariance square root of at least 61 modes. The parallel filter was implemented on the CRAY T3E.

To have a clear insight in the parallelization performance of the operations in the filter, the executable compiled from the source code was kept the same as much as possible, for each number of processors involved. The only difference is the number of modes stored in the local memory, which could be lower if the covariance square root is distributed over multiple processors. Therefore, an executable which is produced to run on multiple processors could have been formed with speed increasing, but memory consuming optimizations too. These are however not possible in the single processor case, and therefore not used during the experiments.

Figure 8.4 shows the speedup of the total filter and the different filter stages, measured for 2, 4, 8, 16, or 32 processors. The stages recognized here reflect equations (8.4).

As expected for a mode-decomposed filter, the best speedup is achieved for the forecast stage, when each processor performs an equal number of model integrations. The speedup is not perfect, since each processor has to spent some time on the initialization of the model. Other sources of overhead are the formation of the ensemble (8.4a) and the formation of the new modes (8.4d), since in both operations, each processor should receive a copy of the mean state. The overhead leads to a parallel forecast stage which is only 26 times faster if evaluated on 32 processors, in spite of the idealized distribution of state vectors. Note that the solid lines between the measured speedups for the forecast stage are not representative for intermediate numbers of processors. The speedup is determined by the maximum number of model steps performed on one of the processors. The true speedup will show a staircase pattern, since for example 17 processors will not perform 64 model integrations

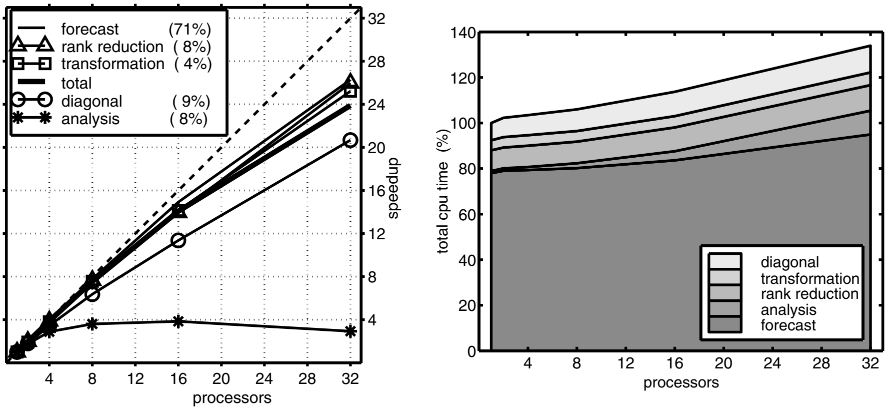


Figure 8.4: *Performance of the mode-decomposed filter around* LOTOS. *Left: speedup of total filter and different filter stages. The percentages in the legend denote the fraction of the total execution time spent on a stage, for evaluation on 32 processors. Right: total execution time (summed over all processors); the time required for the single processor case is set to 100%.*

faster than 16 can do.

The speedups measured for the rank reduction and the transformation are comparable with that of the forecast. The communicational overhead for growing number of processors seems to be compensated for by the spread of the workload. Both operations are implemented 'block' wise, which turned out to have a much better speedup than the state-independent implementations, as expected from the lower amount of communication. Although the communication is still quite large (complete states have to be transfered), the fast communication network of the T3E ensures a reasonable performance. The costs of the rank reduction are completely determined by computation of $\mathbf{S}'\mathbf{S}$; the eigenvalue-decomposition of $\mathbf{B}'(\mathbf{S}'\mathbf{S})\mathbf{B}$ into $\mathbf{L}\Lambda\mathbf{L}'$ takes less than 5% of the costs. Implementation of a parallel eigenvalue-decomposition, such as a parallel Jacobi algorithm (*Golub and van Loan, 1996,* §*8.4*), has therefore not been considered.

The speedup measured for the computation of the covariance diagonal is significant less than that of forecast, transformation, and rank reduction. The costs of the operation $\mathrm{diag}(\mathbf{S}\mathbf{S}')$ are dominated by the computation of the global sum (order $^2\log(n_{pe})$). The workload per processor is minor, and increasing the number of processors does therefore hardly decrease the computation time.

Note that the computation of $\mathbf{S}'\mathbf{S}$ (rank reduction) and the diagonal of the covariance matrix are rather expensive in comparison with the computation of $\mathbf{S}(\mathbf{B}\tilde{\mathbf{V}}\Omega)$. The later requires a similar number of flops (p. 187) as the rank reduction, but seems to be significant cheaper. Simple tests showed that the compiler/processor combination is able to compute the vector-matrix product $\mathbf{s}_i{}'\mathbf{A}$ ($2m^2$ flops in theory) about 50% faster than the vector-vector product $\mathbf{s}_i\mathbf{s}_i{}'$ (m^2 flops). Introduction of a few extra transformations in the filter algorithm would therefore hardly increase the total execution time. This is for example useful if the rank reduction is implemented including the scaling algorithm from section 6.9. The scaling requires computation of the diagonal of the covariance matrix before the actual rank reduction, and this is a simple task only if the transformation **SB** is performed immediately after the analysis.

The worst speedup is measured for the analysis stage. In the chosen implementation, hardly any parallelism is present. Each processor computes a copy of the vector **a** and matrix **B** from eq. (8.4g-8.4h), and this part of the analysis has therefore no speedup at all. Some speedup is however achieved from the interpolation of the modes to the measurement locations in (8.4e). The total costs of computing these entities is small (8% on 32 pe's), since the number of measurements is limited. For larger numbers of measurements, equations (8.4f) and (8.4h) might be solved in parallel, or a domain-decomposition strategy could be considered.

The total speedup is a sum of the speedup of the different filter stages, weighted by their importance in the total algorithm. For 8 processors, the total speedup is almost perfect (7.5), while for 16 and 32 processors, the speedup is still very good (14 and 24 respectively). The speedup curve is not flattening, thus running the filter on more than 32 pe's will still decrease the computation time. The total speedup is strongly related to the speedup of the forecast, which consumes 70-80% of the total computation time (right panel of figure 8.4). The less efficient parallelization of the analysis, and less important, of the diagonal computation, do therefore hardly hamper the total speedup, but make it only slightly smaller than the speedup of the forecast.

Concluding, the decomposition over the modes provides an efficient parallelization of the filter. The major costs are spent on the propagation of the modes, and this part of the forecast stage is to parallelize very efficient. The linear algebra operations on the covariance square root are parallelized efficient too, if the workload is spread over the processors using some form of domain-decomposition. In the chosen experiments, the filter profits from the optimal number of model evaluations, and the communication network of the CRAY T3E which allows fast transfers of complete state vectors.

8.5 Parallelization of the LOTOS model

The previous section discussed the implementation of a parallel filter around the LOTOS model, where the model is not aware of the existence of other processors. In this and the following section, the reverse situation will be discussed: a parallel model, in a filter hardly aware of the parallelism. The discussion is started with the parallelization of the LOTOS model; the description of the filter is left for §8.6.

For a parallel implementation of LOTOS, the concentration array needs to be decomposed in a number of sub arrays. Each processor owns one of the sub arrays, and needs to communicate if a concentration is required that is stored somewhere else. A few options for the decomposition of the concentration array have been considered.

chemical decomposition In this option, each processor manages the operation on a single or a group of component(s). Advection, horizontal and vertical diffusion, and the exchange between between model layers could be applied without communication. Only the chemistry operator requires transfers of concentrations stored somewhere else. This decomposition is efficient if the chemistry is rather simple, or concentrations could be divided in chemical more or less independent groups, and is therefore strongly dependend on the chemical scheme.

vertical decomposition The concentration array is now decomposed over the layers. Each processor manages the concentrations in a single layer of the model. Horizontal advection, chemistry and deposition (lowest layer only) do not require any communication in this decomposition, while vertical exchange and the changing mixing height do. Since the later are minor operations, a decomposition over the layers could be very efficient. An import drawback is however that the number of processors which can be used is bounded by the number of layers (3 in LOTOS). For a full 3D model with multiple layers this problem is less important, although such a model probably includes vertical advection requiring additional communication. Another drawback is the different computational costs associated with different layers; emissions and deposition concern the lowest layer(s) only, while in higher model layers stronger wind fields require smaller time steps for the advection. These differences are not in favor of an efficient load balance for the processors.

horizontal decomposition In a horizontal decomposition, the domain is divided in several horizontal subdomains, and each processor manages all concentrations in the corresponding grid cells. In this decomposition, only advection requires communication between the processors: concentrations at the edges of a subdomain are shared with other subdomains. Since most processes in LOTOS are cell oriented (chemistry) or column oriented (treatment of mixing layers, vertical diffusion, deposition), a horizontal decomposition promises an efficient parallelization. Examples of this approach are found in (*Owcarz and Zlatev, 2000; Barone et al., 2000*).

Since the chemical interaction between the components is strong and the number of vertical layers is limited, the horizontal or domain-decomposition is the best strategy for parallelization of LOTOS. This parallelization strategy will be discussed in two parts: first the actual decomposition, followed by a discussion of the operator splitting in relation to the speedup.

8.5.1 Decomposition of the LOTOS domain

Since the domain of the LOTOS model is very regular (rectangular, almost equidistant grid), a domain-decomposition could be made in a straight forward way. The domain selected for a certain application is divided into a number of rectangular subdomains (figure 8.5). Each subdomain is assigned to a different processor, and if each covers a similar number of cells, operations such as chemistry, vertical exchange and deposition will require a similar computation time.

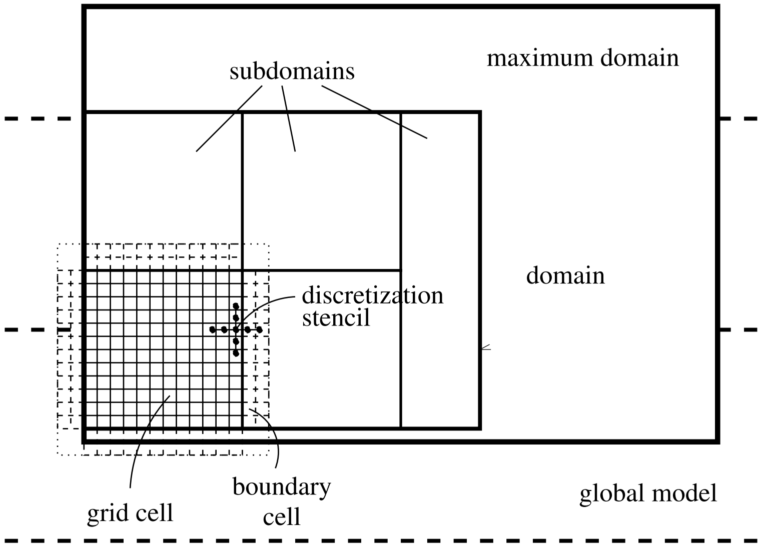


Figure 8.5: *Domain-decomposition in* LOTOS. *The maximum domain is the area for which meteo and surface data is available; the actual domain in typical applications is often smaller. The domain is decomposed into a number of rectangular subdomains, each covering a number of grid cells as equal as possible. Each subdomain is extended with two shells of boundary cells. Meteorological data for the boundaries is filled with data for the maximum domain, or extrapolated over the edge. Concentrations for the boundary cells are either copied from another subdomain or from a global model.*

Communication between the subdomains is necessary during the advection stage. The advection is implemented using κ-fluxes for the horizontal flow in combination with an Runge-Kutta scheme for the time integration (*van Loon, 1996*). The advection scheme is explicit: advected concentrations are computed from the current concentrations and wind fields only. The κ-fluxes use a 9-point discretization stencil; to update one single cell, the scheme requires concentrations up to two grid cells away. All grids involved in horizontal processes are therefore extended with 2 shells of boundary or ghost cells: concentrations and horizontal wind fields, but also the mixing height which is interpolated over the cell borders to estimate the volume. Entities in the boundary cells are never subject to physical processes, but only used to store data: either meteorological data from the maximum grid, or concentrations managed by other subdomains or the global model. The implementation of a domain-decomposition is now very simple.

1. At the begin of a time step, the meteorological fields are read for the maximum domain. These are used to fill the data arrays for each subgrid and its boundaries. For those boundary cells that are not part of the maximum domain, appropriate values are

obtained through extrapolation.

2. Before an advection step is performed, each processor fills the boundaries of its subdomain with concentrations copied from other domains. Depending on the operator splitting scheme (section 8.5.2), the exchange of boundary concentrations is performed 6-8 times for each time step of one hour.

3. At the end of a time step, output concerning the complete domain is collected from the subdomains and saved.

The parallel model hardly differs from the sequential model. Each subdomain is managed in the same way as the total domain used to be managed; the only difference is how the boundary cells are filled. Originally, boundary cells were filled from the maximum domain or the global model only; in the parallel model, they might be filled with data from other subdomains too.

A domain-decomposition implemented in this way is very straight forward, and provides a benchmark for further improvements. A simple improvement might be to synchronize only upwind boundary concentrations. This method reduces the amount of concentrations to be transfered with a factor 2, but increases the total amount of transfers (many small blocks of memory rather than a single large one); the total communication time is therefore not necessarily decreased. Another option is to use a simpler discretization scheme at the boundaries instead of the 9 point stencil used now. This reduces the amount of memory transfers, but leads to small differences between the original and the domain-decomposed model.

The straight forward domain-decomposition has been implemented in LOTOS and tested on the CRAY T3E. For a decomposition as shown in figure 8.5 the long shaped subdomain at the right hand side is a bottleneck for the speedup, since the ratio between the number of boundary and internal cells is largest here. The best strategy would be to let subdomains have a shape as equal as possible. A model domain of 40 × 40 cells has been decomposed into equally shaped subdomains of 20 × 40 (2), 20 × 20 (4), 10 × 20 (8), 10 × 10 (16), or

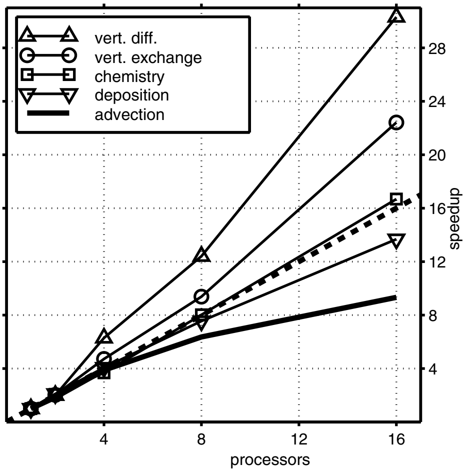


Figure 8.6: *Speedup of advection, chemistry, etc. after domain-decomposition. Measured for* LOTOS *model on* 40 × 40 *grid, decomposed in equal shaped, rectangular subdomains.*

5×10 (32). Figure 8.6 shows the speedup measured for the different model operators. Thanks to the very fast communication of the T3E, the speedup of the domain-decomposed advection is good in spite of the simple implementation. With decomposition in 8 subdomains, the speedup of the advection is about 6.3, which is reasonable in comparison with other examples of parallel Runge-Kutta schemes (*Fritsch and Möhres, 1998*). For 16 subdomains, the speedup has decreased to 9.3, since the ratio between boundary cells and cells inside the subdomain has become worse. The speedup of the other operators is in theory linear with the number of processors, since they do not require additional communication. The results in figure 8.6 show that for some operations, the speedup is even super linear. The vertical diffusion for example becomes 30 times faster when evaluated on 16 processors. An explanation is that for decomposition in smaller subdomains, the concentration array is stored in a smaller block of memory; the time spent on collecting and restoring all concentrations in a column or a single cell is smaller then. Since the vertical diffusion and exchange are rather simple operations, the time spent on memory management is quite important. For the chemistry this is however only a minor part of the costs, and the super linear speedup is limited here.

8.5.2 Operator splitting

The total speedup of the domain-decomposed model is strongly determined by the operator splitting. The operator splitting determines how many times a model operator is called during a time step. If the splitting is chosen such that processes with low speedups are avoided as much as possible, the total speedup will reach a maximum.

The time integration in LOTOS is implemented using a Strang-splitting technique (*Strang, 1968*). The concentrations at time $t+\Delta t$ are computed from the concentrations at time t through application of all subprocess in a symmetric order, for example:

$$\begin{aligned}\mathbf{c}(t+\Delta t) \;=\; & \mathcal{L}_{ade}(\Delta t/2) \circ \mathcal{L}_{dep}(\Delta t/2) \circ \mathcal{L}_{vdf}(\Delta t/2) \circ \mathcal{L}_{chem}(\Delta t) \\ & \circ \mathcal{L}_{vdf}(\Delta t/2) \circ \mathcal{L}_{dep}(\Delta t/2) \circ \mathcal{L}_{ade}(\Delta t/2)\; \mathbf{c}(t) \end{aligned} \tag{8.7}$$

where the operators denote horizontal advection/diffusion and emission, deposition, vertical diffusion, and chemistry (see §2.4). In splitting (8.7), the chemistry operator is applied only once, and concerns an integration over a period Δt; all other processes are applied twice and perform integrations over $\Delta t/2$. The maximum value of the time step Δt is determined by the chemistry, which allows a maximum time step of 15 minutes (Maarten van Loon, personal communication). To integrate the dynamics over one hour (the basic time step of the model and between two successive assimilations in the filter), the sequence of operations in (8.7) is repeated four times.

Operator splitting involves an error since it decouples subprocesses which actually interact with eachother. A theoretical analysis of the error involved in splitting is hard to make, and only possible for simplified operators (*Lanser and Verwer, 1999*). Practice has however shown that splitting works very well, almost independent of the order in which the operators are applied.

For the total computation time of a certain splitting order it is important how expensive the different operations are. As an example, figure 8.7 shows the fraction of the time spent on each model operation for a model with splitting (8.7). The advection with κ-fluxes turns out

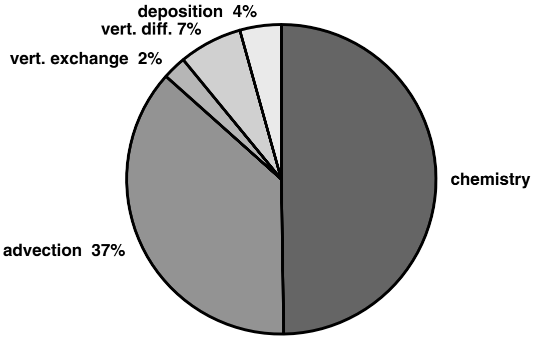


Figure 8.7: *Percentage of execution time spent on various operations in the model, in case of Strang splitting with chemistry as central operation.*

to be an expensive operation, which takes almost 40% of the total time. The fractions will be different if the splitting order is changed. Table 8.1 shows the total execution time for three different splitting schemes, measured for the LOTOS model using a 40×40 grid. The first scheme is equal to (8.7), with chemistry as the central operator. The maximum time step is limited by the chemistry (15 min.), and therefore the operator sequence is repeated four times for simulation over one hour. Whenever possible, two sequential calls to the advection operator over 7.5 minutes are replaced by a single operation over 15 minutes. With this time step, the Courrant condition is always satisfied during the assimilation period for the chosen grid. In 50% of the period, even a time step of 20 minutes is allowed. For a domain extended to the highest latitudes with smaller grid cells, a maximum time step of only 12 minutes is required during 10% of the time.) The chosen scheme has the advantage of a minimum number of calls to the chemistry operator, with the cost of an increased number

				processors	1	2	4	8	16
Strang splitting:			$n_{opp} \times \Delta t_{opp}$		execution time				
nr	central	loops	chem.	advec.	**speedup**				
1.	chem.	4	4×15.0	2×7.5,	100.0	51.3	25.5	13.8	8.2
				3×15.0	**1.0**	**1.9**	**3.9**	**7.2**	**12.2**
2.	advec.	3	6×10.0	3×20.0	89.5	45.1	23.1	11.9	6.7
					1.0	**2.0**	**3.9**	**7.5**	**13.4**
3.	advec.	4	2×7.5,	4×15.0	95.8	48.7	24.5	13.0	7.5
			3×15.0		**1.0**	**2.0**	**3.9**	**7.4**	**12.8**

Table 8.1: *Total execution time and speedup for three different Strang splittings. The central operation in the splitting is either chemistry, with always 4 loops of 15 minutes per hour, or advection, either 3 or 4 loops per hour given the Courrant condition. For the chemistry and advection operators, the number of calls and corresponding time steps are listed in the third and fourth column respectively. The total execution time is scaled with the time for splitting around chemistry, single processor case.*

of advection steps. Schemes 2 and 3 use the advection as central operator, called either 3 times with steps of 20 minutes (scheme 2), or 4 times with steps of 15 minutes (scheme 3), appropriate for different Courrant conditions. For scheme 3, two sequential chemistry steps over 7.5 minutes are replaced by a single one over 15 minutes; for the second scheme this is not possible since it would lead to an integration over 20 minutes. Schemes 2 and 3 lead to total execution time on a single processor of 89.5 % and 95.8 % of scheme 1 respectively.

The last columns of table 8.1 show the speedup measured for the different splitting schemes. Up to 4 processors, there is hardly any difference in performance between the three methods. For execution on 8 and 16 processors however, the splittings 2 and 3 with advection as the central operation show both a better speedup than splitting 1 with chemistry in the center. With a decomposition over 16 processors for example, the speedup of the advection-splittings are 12.8 and 13.4 respectively, while the chemistry-splitting has a speedup of 12.2 only. The relatively large number of advection steps in this splitting hampers the performance.

The total execution time for the advection-splittings is smaller than what could be expected from the speedup only. In splittings 2 and 3, the chemistry is integrated over rather small time steps (10.0 or 7.5 minutes instead of 15.0). The number of iterations required for convergence of the solver is often smaller in this case. For a single chemistry step the computation time is reduced with 12% for a 10 minutes timestep, and to 20% for a 7.5 minutes timestep. Thus, although the splittings around advection lead to an increased number of chemistry steps, a part of the additional computation time is compensated for by the, on average, smaller time steps.

Concluding, the splitting with advection as the central operation is preferable over the original scheme. For a model run with each hour a number of advection steps as small as possible, a speedup of about 7.5 might be reached on 8 processors, which is close to optimal. The total execution time is about 5-10% lower than what could be achieved with chemistry as the central operation, due to the lower amount of advection steps and increased speed of the chemistry solver for smaller time steps.

8.6 Parallel filter: decomposition over the domain

With the parallel LOTOS model available, implementation of a parallel low-rank filter is rather simple. The problem of exploring parallelism has been moved from the top (filter) downwards to the model; the filter is not necessarily aware of the presence of multiple processors. The data structures of the filter should follow the domain-decomposition of the model. Each processor is assigned to a single subdomain; therefore, it should own a copy of all entities of the filter that have any relation with this subdomain. For example, the covariance square root is distributed in blocks of rows (figure 8.2), such that each processors owns the rows corresponding to its subdomain. Similar, the mean state is distributed. A number of smaller entities that are of interest for all subdomains are not decomposed, but each processor owns a copy; their contents have to be synchronized.

In the following two paragraphs the implementation of the operations in the domain-decomposed filter will be described. Apart from the conventions defined at page 146, the following additional notations are introduced:

- $(x)_k$ denotes the part or the copy of the entity x stored on processor k;
- 'distribution' of an entity denotes a decomposition over the subdomains, followed by a transfer of the results to the corresponding processors.

8.6.1 Operations on domain-decomposed states

The following operations produce entities which are formed from data spread over several domains, but are of interest for all domains. The results are therefore synchronized between the processors.

$\mathbf{y} = \mathbf{H}'\mathbf{x} \ = \ \langle \mathbf{H}, \mathbf{x} \rangle_{\mathcal{K}}$

The value of a state vector evaluated at a certain measurement point is of interest for all domains during the analysis. Elements of $\mathbf{y}$ are often formed from concentrations in one or a few grid cells only. The observation matrix $\mathbf{H}'$ could therefore be decomposed over the domains, such that the actual interpolation is performed in parallel:

```
Require: $\mathbf{x}$ and $\mathbf{H}$ distributed row-wise
  for all $k \in \mathcal{K}$ do
    $(\mathbf{y})_k = (\mathbf{H})_k{}'(\mathbf{x})_k$
  end for
  send $(\mathbf{y})_k$ to all $k \in \mathcal{K}$
Ensure: $\mathbf{y} = \mathbf{H}'\mathbf{x}$
```

If each domain is involved in the analysis of a similar number of observations, the parallelization of this operation is rather efficient.

$a = \mathbf{x}_1{}'\mathbf{x}_2 \ = \ \langle \mathbf{x}_1, \mathbf{x}_2 \rangle_{\mathcal{K}}$

The dot-product between two states is not a standard operation in a filter, but used for special operations such as rank reduction and projection on the covariance. The operation is almost similar to the operation $\mathbf{y} = \mathbf{H}'\mathbf{x}$; see above for the algorithm. Since only a single number has to be exchanged between the processors, the communication time for this operation is minor.

Both operations are part of the model context rather than of the filter. A change in the model state or the observations will require changes in these operations, and their implementation should therefore be separated from the filter.

8.6.2 Operations on covariance square root

In this paragraph, all basic operations on a covariance square root $\mathbf{S}$ are described in case of a decomposition over the rows. Since each processor owns all elements in a row, operations acting on a row do not require any communication.

$\boldsymbol{\Psi}' = \mathbf{H}'\mathbf{S}$

The mapping of modes onto observation space is processed easily using the previous described mapping of a single state:

```
Require: $\mathbf{S}$ and $\mathbf{H}'$ distributed row-wise
  for all $k \in \mathcal{K}$ do
    for $j = 1, \ldots, m$ do
      $\psi'_j = \langle \mathbf{H}, \mathbf{s}_j \rangle_{\mathcal{K}}$
    end for
  end for
Ensure: $\Psi' = \mathbf{H}'\mathbf{S}$
```

$\mathbf{S} := \mathbf{SC}$

Where the discussion of the transformation $\mathbf{SC}$ took almost 2 pages for the mode-decomposed $\mathbf{S}$, the implementation becomes very simple in case of a domain-decomposition. Each row $\mathbf{s}'_i$ is replaced by the vector-matrix product $\mathbf{s}'_i\mathbf{C}$, completely independent from other rows. The expected speedup is therefore perfect, if each subdomain owns the same number of state variables.

```
Require: $\mathbf{S}$ distributed row-wise, $\mathbf{C}$ available on each pe
  for all $k \in \mathcal{K}$ do
    $(\mathbf{S})_k := (\mathbf{S})_k\mathbf{C}$
  end for
Ensure: $\mathbf{S} := \mathbf{SC}$
```

$\mathbf{d} = \mathbf{diag}(\mathbf{SS}')$

Similar as any other state vector, the diagonal $\mathbf{d}$ is domain-decomposed over the processors. Since each element of the diagonal is formed from a single row of $\mathbf{S}$, the implementation is simple in case of a domain-decomposition:

```
Require: $\mathbf{S}$ distributed row-wise
  for all $k \in \mathcal{K}$ do
    $(\mathbf{d})_k = \sum_j (\mathbf{s}_j)_k^2$
  end for
Ensure: $\mathbf{d} = \mathbf{diag}(\mathbf{SS}')$
```

$\mathbf{A} = \mathbf{S}'\mathbf{S}$

The matrix $\mathbf{S}'\mathbf{S}$ is used during the rank reduction of $\mathbf{S}$ in a RRSQRT filter, and for projections on the space spanned by the columns of $\mathbf{S}$. With the definition of the dot-product on page 161, this operation is implemented straight forward:

```
Require: $\mathbf{S}$ distributed row-wise
  for $j = 1, \ldots, m$ do
    $a_{ij} = a_{ji} = \langle \mathbf{s}_i, \mathbf{s}_j \rangle_{\mathcal{K}}$
  end for
Ensure: $\mathbf{A} = \mathbf{S}'\mathbf{S}$
```

$\mathbf{x} = \mathbf{Sa}$

The result of this operation is a linear combination of columns of $\mathbf{S}$, and is for example used to analyze the mean state. Since the result is a state vector and therefore decomposed over the domain, no communication is required:

```
Require: $\mathbf{S}$ distributed row-wise, $\mathbf{a}$ available on each pe
  for all $k \in \mathcal{K}$ do
    $(x)_k = \sum_{j=1}^{m} a_j \; (\mathbf{s}_j)_k$
  end for
Ensure: $\mathbf{x} = \mathbf{S}\mathbf{a}$
```

$\mathbf{a} = \mathbf{S}'\mathbf{x}$

This operation is used for projections on the subspace spanned by the columns of **S**, see for example eq. (6.23d) for the POENK filter, or eq. (7.30) for the nonlinearity number. Each element of the result is a dot-product between state **x** and a column of **S**:

```
Require: $\mathbf{S}$ distributed row-wise, $\mathbf{x}$ distributed
  for $j = 1, \ldots, m$ do
    $a_j = \langle \mathbf{s}_j, \mathbf{x} \rangle_{\mathcal{K}}$
  end for
Ensure: $\mathbf{a} = \mathbf{S}'\mathbf{x}$
```

8.6.3 Performance of domain-decomposed filter

The performance of the domain-decomposed filter was measured for the same assimilation experiment as described in §8.4.3. To be able to run the filter on a single processor to, the domain was limitted to 32 × 32 cells again. Figure 8.8 shows the speedup and total execution time of the different filter stages (compare with figure 8.4 for the mode-decomposed filter).

The most apparent result is the observed super linear speedup of the transformation stage. Thanks to smaller concentration arrays in the decomposed **S**, the computation of **SC** is 44 times faster if evaluated on 32 processors. Note that the super linear effect is not observed for the 2 processor configuration; in that case, the concentration arrays are still too large to explore all possibilities of the processors. The net effect of this super linear speedup is however limited, since the total computation time spent on the transformation is less than 2% of the total.

Another apparent result is the not very optimal speedup of the forecast stage. Evaluated on 16 processors, the forecast is only 10 times faster, while speedups above 13 were measured for the LOTOS model (table 8.1). The parallel LOTOS model suffered in this experiment from the smaller grid size of 32 × 32 cells, necessary for running the filter on a single processor too. For 40 × 40 cells over 16 processors, the subdomains have size 10 × 10 and ratio between boundary and inner cells is 0.8, while for 32 × 32 cells the subdomains are 8 × 8 and the ratio has grown to 1.0 . With decomposition over 32 processors, the situation is even worse. Therefore, the number of processors used in a domain-decomposed filter should be related to the size of the domain to remain sufficient speedup. Note the constant computation time for the forecast for evaluation on 2 and 4 processors; similar as observed for the transformation, the forecast seems to profit from the smaller concentration arrays here.

Figure 8.8 shows that the performance of the domain-decomposed rank reduction and computation of the covariance diagonal are comparable with that of the mode-decomposed

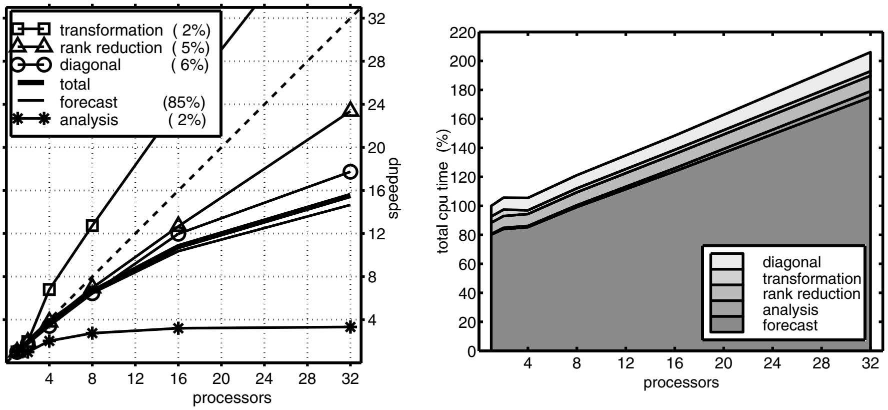


Figure 8.8: *Performance of domain-decomposed filter. Left: speedup of total filter and different filter stages. The percentages in the legend denote the fraction of the total execution time spent on certain stage, for evaluation on 32 processors. Right: total execution time (summed over all processors). The time required for the single processor case is set to 100%.*

filter. This is remarkable, since the operations on the covariance square root used in these stages require a lower amount of communication in case of domain-decomposition. This might be explained from the fast communication network in the Cray computer, where transfers of large blocks of memory are rather cheap. The costs of the communication are more determined by the number of communication events. An interesting experiment would be to compare the two strategies on a platform with relatively slow communication, such as a Beowulf cluster.

Similar as for the mode-decomposition, the worst speedup is observed for the analysis, since hardly any effort has been put in parallelization. The total time spent on this operation is however minor, and therefore hardly visible in the total speedup. The total speedup is almost equal to that of the forecast, since this stages consumes about 80-85% of the total cpu time (right panel of figure 8.4).

Concluding, the performance of the domain-decomposed filter is completely determined by the performance of the parallel model. If the size of the domain is too small for decomposition over many processors, the speedup of the filter is strongly limited. The filter operations on the covariance square root are however implemented very efficient.

8.7 Comparison between parallelization strategies

For a true comparison of the different filter strategies, both have been implemented in most optimal form for the experimental setup of chapter 5. The analysis of the different filter stages in sections 8.4.3 and 8.6.3 required that the filter was able to run on a single processor too. This limited the use of important compiler optimizations, which are able to increase

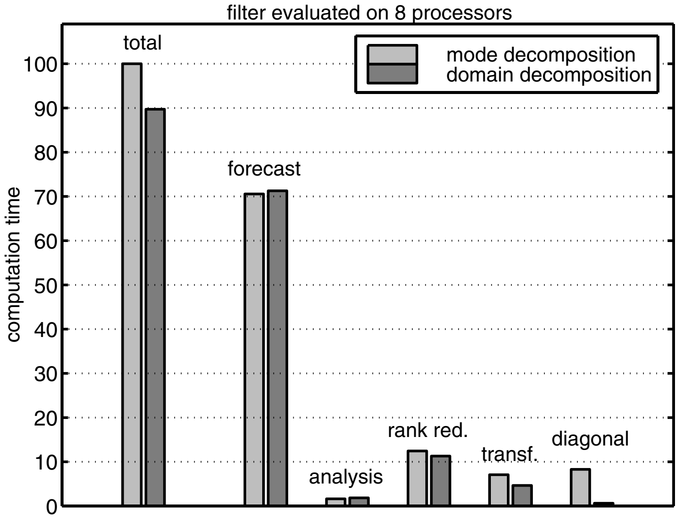


Figure 8.9: *Total computation time spent on different filter stages for mode or domain-decomposition, evaluated on 8 processors; experimental setup as described in chapter 5 (*40×40 *grid cells, 64 model evaluations). The computation time is scaled with the total time for the mode-decomposed filter.*

the speed of the computations with the cost of additional memory consumption.

The two filter codes have been compiled in full optimized form for a domain of 40×40 grid-cells, 64 model evaluations in total, and execution on 8 processors. The differences in computation time between mode- and domain-decomposition are shown in figure 8.9. The total computation time for the domain-decomposed filter is about 10% lower than for the mode-decomposed filter. A large part of the difference could be explained from the time spent on computation of the covariance diagonal, which turns out to be significant cheaper in domain-decomposed form (no communication). Also the transformation and the rank reduction are cheaper in case of domain-decomposition. The costs of the forecast stage are comparable in both parallelizations, in spite of the different communicational needs. Where the mode-decomposed filter requires the transfer of a limited amount of complete states, the domain-decomposition requires a large amount of small transfers (boundary concentrations); the costs seem to be comparable however.

The bars in figure 8.9 show that the rank reduction takes about 12% of the total computation time, for the current application. Application of a RRSQRT filter is therefore not significant more expensive than application of an Ensemble Kalman filter with the same number of model evaluations. If the reduction mechanism is able to limit the number of model evaluations, the total computation time for a RRSQRT filter is soon smaller than for a ENKF.

8.8 Discussion

After evaluating the experiments with the mode- and domain-decomposed filters, the different strategies are judged on aspects of performance, flexibility, and ease-of-use. The discussion is extended to applications or filter configurations that are not in use for the current filter around LOTOS, but might be implemented in future. Table 8.2 shows a summary of the conclusions.

speedup The two filter strategies have almost optimal speedup for either the model propagation or the filter algebra.
A mode-decomposed filter has a perfect speedup for the forecast, as long as the number of model evaluations is a multiple of the number of processors. The matrix algebra in the filter equations require however the transfer of complete state vectors, and this will hamper the speedup on platforms with relatively slow communication.
The reverse holds for the domain-decomposed filter. The speed up of the matrix algebra is almost optimal (as long as the number of state elements is divided equally over the subdomains), while the speedup of the forecast is hampered by the speedup of the parallel model. For the atmospheric chemistry models considered here, the domain-decomposed model is very efficient, if the subdomains are not too small. The number of processors should therefore be related to the size of the domain.

memory A mode-decomposed filter makes less efficient use of the available memory, since each processor owns a complete copy of the model. For an atmospheric model as considered here, the amount of meteorological, emission, and land use data is substantial, and for some models this could take more than 95% of the total storage [2].
The domain-decomposed filter is more efficient here, since each processor owns the local parameters only.

scalability For the current filter/model combination, typical applications require different shaped and sized domains. If the performance of the filter remains the same if both the grid size and the number of processors is increased, the parallelization is called scalable.
For the mode-decomposed filter, increasing the number of processors will lead to a decrease of efficiency since most of the communication is related to state transfers from one to all other processors. The number of communication events for a single processor will therefore increase. The sizes of the data arrays stored on a processor are increased too, and this is often not in favor of the computation time.
The scalability of the domain-decomposed filter is however much better. Most of the communication is required between neighboring subdomains only, thus the total number of communication events for a single processor does hardly grow if the domain is extended with a few new subdomains. The sizes of data arrays won't increase, such that the memory management on a processor is unchanged.
Note that none of the filter strategies is scalable with the number of modes, since the filter operations other than the forecast are $\mathcal{O}\left(m^2\right)$ rather than $\mathcal{O}\left(m\right)$.

---

[2] The EUROS model at the start of the STROPDAS project (*Velders et al., 2001*)

observations with local support In application with large amounts of measurements to be assimilated, for example satellite tracks, the use of a domain-decomposed filter is favored. Each processor needs to store only those measurements which are located in its own subdomain, and this is only a fraction of the total number. Besides, since the spatial correlation between measurements and computed concentrations is often limited, the implementation of an analysis with local support could be considered (*Houtekamer and Mitchell, 2001*). In a domain-decomposition, this limits the communication between the processors to neighboring subdomains only.

In a mode-decomposed filter however, each processor needs to store the information about all measurements. Projections of the modes stored on a processor onto observation space need to be send to all other processors, even in case of a gain with local support.

different treatment of modes In some formulations of the low-rank filter, the mean and/or the modes or are treated different. Examples are a filter in which the modes are propagated by a reduced instead of the original model, or a filter which uses two filter algorithms next to eachother (section 6.7). The parallelism is much better preserved in these examples in a domain-decomposed filter. The mode-decomposed filter suffers from irregular distribution of the different type of modes over the processors. For optimal speedup, each processor should own an equal number of each type of modes, and this property is hard to preserve in case of a flexible number of modes or processors. A domain-decomposed filter is more flexible it this case, since the modes are processed sequentially rather than parallel.

independence of model If the filter tool is used in combination with different models, the implementation should be flexible in incorporating another model.

The mode-decomposed filter is favored here; implementation of filter and model are almost completely independent. The only requirement for the model is a clear description of the state vector, including suitable definitions for linear operations between states (the state should be a *vector*). For efficient implementation of some filter operations it is useful if operations are available to decompose the state in blocks (see the row-based algorithms for computing $\mathbf{SC}$ and $\mathbf{S}'\mathbf{S}$ at pages 150 and 151). Given that these operations are defined, any model could be embedded in a mode-decomposed filter. A domain-decomposed filter depends on the existence of a parallel model however.

coding The code for a mode-decomposed filter is quite complex. For each operation acting on a mode, the program has to consider whether it is stored locally or not. If such a code is used exclusively on a single processor machine, the overhead of unused statements is large.

In a domain-decomposed filter, the problem of parallel implementation has been moved from the desktop of the filter programmer to that of the model develloper. The filter is hardly aware of running on multiple processors, since communication is required on the model level only. For the LOTOS model, the parallel model code hardly differs from the sequential code. The total number parallel statements in model and filter is therefore limitted.

	decomposition	
filter aspect	**mode**	**domain**
speedup	+	+
memory consumption		+
scalability		+
observations with local support		+
different treatment of modes		+
independence of model	++	
coding		++

Table 8.2: *Comparison of parallel filter strategies, judged on different aspects of application and implementation.*

8.9 Conclusions

Parallelization of a low-rank Kalman filter is efficiently performed with both a mode as well as a domain-decomposition strategy. Each of the discussed parallelization strategies has its own advantages and disadvantages, but whether these are important strongly depends on the application and the computing platform.

For the filter around LOTOS, the domain-decomposition is slightly favored. Implemented on a massive parallel machine (CRAY T3E), the speedup achieved with mode or domain-decomposition is almost indifferent. The choice for domain-decomposition is therefore based on more or less subjective criteria such as easy implementation, and potential positive effects when running on other platforms or when large numbers of measurements are to be analyzed. For the LOTOS model, building an efficient domain-decomposed model is not complicated, and the amount of communication is limited. An efficient parallel filter is easily implemented given the domain-decomposed model. The low amount of communication becomes more important if the filter is implemented on a platform with slower communication than the CRAY T3E used in our experiments. Further, the domain-decomposition gives an efficient memory management, shows better scalability, and promises an efficient analysis of large amounts of measurements.

All these advantages are however not relevant if a domain-decomposition for the model is not available, or not very efficient. Spending some time on building or improvement of the parallel model should be considered here; having an efficient parallel model is useful anyway. If this is not possible or the result is not satisfactory, the mode-decomposition is the only alternative. The parallelism inherent to the low-rank filter will ensure an useful efficiency. For efficient implementation of the matrix algebra in the filter, a rough form of a domain-decomposition is necessary. A suitable approach could be to implement a hybrid parallelization, with the forecast parallelized with a mode- and the analysis with a domain-decomposition.

Chapter 9

Discussion and conclusions

The chemical composition of the air in urbanized areas is one of the most uncertain aspects of atmospheric research. Uncertainties in emissions, meteorological conditions, and unknown interaction with vegetation all lead to variations in trace gas concentrations which are hard to describe with a model. Measurements of trace gases are able to give insight in the actual value of the variations however. The best way to improve simulations is therefore to assimilate the measurement data in the model.

This thesis describes the assimilation of measurements in an air-pollution model using a Kalman filter. The technique is applied to the LOng Term Ozone Simulation (LOTOS) model. LOTOS is a photo-oxidant model that simulates the tropospheric ozone level accross Europe. Measurements made at ground-based sites are available on a regular basis for comparison with the simulations. The main deficiencies in the LOTOS model and its inputs are believed to lie the parameterizations of emissions and deposition. Values of emission strength and deposition velocity are based on yearly totals, projected to simulation time using standard profiles. Many temporal variations present in nature are therefore not present in the model. Parameters related to the meteorological dynamics are obtained from weather forecast data, and already show large temporal variations. These are not considered as uncertain in this study, although especially the height of the mixing layer has a large impact on the ozone level.

Using a stochastic model for uncertain parameters in LOTOS, the Kalman filter is able to compute an optimal estimate of the ozone level given available measurements. Uncertainties in emission parameters are of large scientific and political interest, and therefore these are the first parameters to consider. The majority of the emissions in Europe is released from a few densely populated areas. Variations in ozone concentrations in the plumes released from these areas are smoothed due to deposition and titration by NO during the night. The variations in ozone do therefore not range much further than 300-600 km, unless an emission plume is transported over water, where deposition and titration is limited. Other uncertainties than emissions should be considered to explain the differences between model and measurements. A stochastic model based on a combination of uncertain emissions, photolysis rates, and deposition velocity was shown to be useful to assimilate ozone measurements in LOTOS. The uncertainties in these parameters are able to explain differences between model and measurements for both suburban and remote sites, for daytime as well as nighttime hours. Since the absolute differences are hard to quantify beforehand, an adaptive adjustment should be included in the filter to balance the stochastic model with the actual observed residues. In this research, the representation error was adjusted adaptively, to explain the residue between model and measurements given a fixed degree of freedom in the model parameters. The chosen stochastic model was able to represent the measurements up to a difference of 7–12 ppb in the afternoon, and up to 15 ppb during the rest of the day.

Given the large uncertainties involved in air-pollution modeling, users will always ask whether a data assimilation procedure is able to estimate model parameters. The Kalman filter is able to provide optimal estimates of uncertain model parameters such as emissions if it is implemented as a smoother. An important application of a smoother is the estimation of emissions. Assimilation experiments showed that a smoother is able to decrease the uncertainties in emitted nitrogen oxides (NO_x) and volatile organic compounds (VOC) through assimilation of ozone measurements. The tight chemical coupling between these components ensures that variations in emissions are visible in the ozone measurements. The most accurate estimations have been made for uncertain NO_x emissions during the night, when these are the only emissions acting on the ozone level. Estimation of VOC emissions from ozone measurements requires that the ratio in which the various organic components are emitted is exactly known. For more accurate or exact reconstruction of the emissions, other measurements than ozone should be assimilated too. Assimilation of other components was not considered in this research, however, since these are either inaccurately represented by the model (NO_x), or only available for urban sites (VOC). To be able to assimilate other components too, the model should be improved at two points: the vertical resolution should be increased to improve the simulation of NO_x, and measurements of VOC in urban area should be represented better, for example by nesting a high-resolution model. Profile data from soundings or satellite instruments could become useful in future to estimate emissions; especially satellite information is useful because of its large spatial coverage. Assimilation of satellite profiles requires extension of the current LOTOS model in the vertical, however, such that at least the total troposphere is covered. In addition, the vertical resolution of the satellite retrieval should be increased.

The model parameters estimated with the filter/smoother are able to improve the model simulations significantly. If the assimilation tool is used in a forecast procedure, the estimates of the ozone maxima for the coming day are 25% more accurate when based on estimated parameters than when based on analyzed concentrations only. Initial variations in concentrations have a limited lifetime in the chemical active troposphere, and cannot explain all variations observed in ozone concentrations. An assimilation procedure for a smog forecast should therefore be based on an estimation of uncertain parameters rather than on an estimation of initial concentrations.

The filter developed for LOTOS takes the form of a low-rank filter. This type of approximate Kalman filter is are suitable for application to models with large state vectors. A number of commonly used low-rank filters has been compared, and although these filters have been developed from different theoretical backgrounds, the actual implementation is quite similar. All methods use a low-rank parameterization of the covariance matrix, and propagate the covariance structure in time using 40–100 evaluations of the model. A RRSQRT filter using the forecast step of the SEIK filter was shown to be the best choice for the filter around LOTOS, combining high accuracy with low computational costs. The forecast step incorporated from the SEIK filter uses a technique of minimal exact sampling (MES), which ensures accurate propagation of the forecast error even in case of strong nonlinear dynamics. The MES method is computationally as expensive as the first-order linearizations used in the original RRSQRT filter, and their default use is therefore suggested. Ensemble techniques, which are ultimately suitable for coping with non linearities, were found to suffer from a slow convergence in multiple experiments. At least a double amount of model evaluations is required for these methods in comparison with the RRSQRT approach to obtain similar results. An ensemble filter is able to provide results up to any desired accuracy however. For experiments with simple models, where computation time is no constraint, an ensemble method is therefore the best choice.

The computational costs of the filter around LOTOS are impressive, even for the chosen RRSQRT approach. Implementation of the filter on a parallel machine is therefore necessary. Two approaches have been considered: parallelization of the filter over the modes, leaving the model unchanged, and parallelization of the model using a domain decomposition. Both methods have been implemented on a massive parallel machine (CRAY T3E) and both provided an efficient speedup. Implementation of a domain-decomposed filter turned out to be much easier than a mode decomposition, and requires in theory less communication. Besides, the domain decomposition is more efficient if large numbers of measurements are to be assimilated, for example retrieved from a satellite. If an efficient domain-decomposed model is available or easily developed, using a parallel filter based on this approach is therefore favored.

For future research, a few recommendations are made. As already mentioned, to be able to assimilate other components than ozone, the model should be improved. The vertical resolution of the model should be increased for better representation of NO_x, and horizontal resolutions should be increased at least locally for representation of VOC in urban areas. However, both improvements will increase the computational costs of the model and therefore of the filter. To reduce the costs of the filter, a simplified version of the model could be developed too. The full model is then used to propagate the filter mean, while the simplified model is used for the modes. The simplified model could be based on a coarse grid or on a parameterized chemistry, or just be a complete new (tangent linear) model for forecasting of the ozone maxima only. The parallelized filter with the full model provides a benchmark against which a simplified filter could be validated.

Appendix A

Gasphase reactions

The chemistry model used in LOTOS is a version of the *Carbon Bond Mechanism IV* described in (*Gery et al., 1989*). The basic idea in CBM-IV is to represent the organic chemistry not in terms of molecules, but in terms of reactive groups of atoms (table A.1). Only formaldehyde and ethene are represented as single molecules because of their characteristic chemical behavior. Other organic molecules should be decomposed into a sum of the characteristic groups represented by the mechanism. This method has the advantage that a large variety of molecules can be expressed in only a limited number of carbon bonds. A disadvantage is that it is difficult to compare measurements of organic molecules with CBM-IV calculations. If CBM-IV proposes a certain mixture of carbon bonds, this might be representative for a large variety of molecules.

The version of CBM-IV used in the LOTOS model is slightly different from the original version (*Kuhn et al., 1998*). Reactions involving the oxygen radicals O and $O(^1D)$ have been eliminated for example, while reactions for methane and sulfur oxides have been included.

Table A.1: Chemical components in CBM-IV version for LOTOS.

component	description	
inorganic components		
O_3	ozone	
NO	nitric oxide	
NO_2	nitrogen dioxide	
N_2O_5	dinitrogen pentoxide	
PAN	peroxyacyl nitrate	
HNO_2	nitrous acid	
SO_2	sulfar dioxide	
SO_4	sulfate	
organic components		
CO	carbon monoxide	
CH_4	methane	
ETH	ethene	CH_2CH_2
FORM	formaldehyde	$CH_2(O)$
MGLY	methylglyoxal	$CH_3C(O)C(O)H$
TOL	toluene	$C_6H_5 - CH_3$
XYL	xylene	$C_6H_4 - (CH_3)_2$
PHEN	phenol	$C_6H5 - OH$
PAR	paraffin carbon bond	$C - C$
OLE	olefinic carbon bond	$C = C$
ALD2	high molecular weight aldehydes	RCHO
radicals		
OH	hydroxyl radical	
HO_2	hydroperoxy radical	
NO_3	nitrate radical	
C2O3	peroxyacyl radical	$CH_3C(O)OO\cdot$
PHO	phenolhydroxy radicaal	
operations		
XO_2	NO-to-NO_2 operation	
XO_2N	NO-to-nitrate operation	

Table A.2: Chemical reactions in CBM-IV version for LOTOS.

inorganic chemistry			
R1	NO_2	$\xrightarrow{h\nu}$	$NO + O_3$
R3	$O_3 + NO$	$\rightarrow$	NO_2
R7	$NO_2 + O_3$	$\rightarrow$	NO_3
R8	O_3	$\xrightarrow{h\nu}$	a_1 OH + a_2 O_3
R10	$O_3 + OH$	$\rightarrow$	HO_2
R11	$O_3 + HO_2$	$\rightarrow$	OH
R12	$NO_3 + NO$	$\rightarrow$	$2\,NO_2$
R13	$NO_3 + NO_2$	$\rightarrow$	$NO + NO_2$
R14	$NO_3 + NO_2$	$\rightarrow$	N_2O_5
R15	N_2O_5	$\rightarrow$	
R16	N_2O_5	$\rightarrow$	$NO_2 + NO_3$
R17	$NO + NO_2$	$\rightarrow$	$2\,HNO_2$
R18	$HNO_2 + HNO_2$	$\rightarrow$	$NO + NO_2$
R19	HNO_2	$\xrightarrow{h\nu}$	NO + OH
R20	$NO_2 + OH$	$\rightarrow$	
R21	NO + OH	$\rightarrow$	HNO_2
R22	$NO + HO_2$	$\rightarrow$	$OH + NO_2$
R23	NO + NO	$\rightarrow$	$2\,NO_2$
R26	$HNO_2 + OH$	$\rightarrow$	NO_2
R27	NO_3	$\xrightarrow{h\nu}$	$NO_2 + O_3$
R28	NO_3	$\xrightarrow{h\nu}$	NO
R29	$HO_2 + HO_2$	$\rightarrow$	
R30	$HO_2 + HO_2$	$\rightarrow$	
R31	CO + OH	$\rightarrow$	HO_2
organic chemistry			
R32	FORM + OH	$\rightarrow$	CO + HO_2
R33	FORM	$\xrightarrow{h\nu}$	CO + 2 HO_2
R34	FORM	$\xrightarrow{h\nu}$	CO
R36	FORM + NO_3	$\rightarrow$	CO + HO_2
R38	ALD2 + OH	$\rightarrow$	C2O3
R39	ALD2 + NO_3	$\rightarrow$	C2O3
R40	ALD2	$\xrightarrow{h\nu}$	FORM + CO + 2 HO_2 + XO_2
R42	C2O3 + NO	$\rightarrow$	FORM + NO_2 + HO_2 + XO_2
R43	C2O3 + NO_2	$\rightarrow$	PAN
R44	PAN	$\rightarrow$	C2O3 + NO_2
R45	2 C2O3	$\rightarrow$	2 FORM + 2 HO_2 + 2 XO_2
R46	C2O3 + HO_2	$\rightarrow$	0.79 FORM + 0.79 OH

			+ 0.79 HO_2 + 0.79 XO_2
R47	MGLY	$\xrightarrow{h\nu}$	C2O3 + CO + HO_2
R48	MGLY + OH	$\rightarrow$	C2O3 + XO_2
R50	PAR + OH	$\rightarrow$	0.45 ALD2 - 0.75 PAR + 0.93 HO_2 + 1.49 XO_2 0.067 XO_2N
R52	OLE + OH	$\rightarrow$	FORM + ALD2 - PAR + HO_2 + XO_2
R53	OLE + O_3	$\rightarrow$	0.66 FORM + 0.50 ALD2 - PAR + 0.212 CO + 0.28 HO_2 + 0.08 OH + 0.144 XO_2
R54	OLE + NO_3	$\rightarrow$	0.91 HO_2 + 0.91 XO_2 + 0.09 XO_2N
R56	ETH + OH	$\rightarrow$	2 FORM + HO_2 + XO_2
R57	ETH + O_3	$\rightarrow$	FORM + 0.37 CO + 0.13 HO_2
R58	TOL + OH	$\rightarrow$	0.36 PHEN + 0.56 MGLY + 0.36 PAR + 1.13 FORM + 0.37 CO + HO_2 + 0.64 XO_2
R59	PHEN + NO_3	$\rightarrow$	PHO
R60	PHO + NO_2	$\rightarrow$	
R61	XYL + OH	$\rightarrow$	0.67 FORM + 1.33 MGLY + 0.28 PHEN + 0.67 CO + 0.56 PAR + HO_2 + 0.72 XO_2
methane chemistry			
R49	CH_4 OH	$\rightarrow$	FORM + HO_2 + XO_2
operator chemistry			
R67	XO_2N + NO	$\rightarrow$	
R68	XO_2 + NO	$\rightarrow$	NO_2
R69	XO_2 + HO_2	$\rightarrow$	
R70	XO_2 + C2O3	$\rightarrow$	C2O3 + XO_2 + HO_2
sulfar chemistry			
R71	SO_2 + OH	$\rightarrow$	SO_4 + HO_2
R72	SO_2	$\rightarrow$	SO_4

Appendix B

Repeated scalar update

The repeated scalar update is an algorithm to solve the analysis equations for a general low-rank filter (§6.3) based on the Potters algorithm (*Maybeck, 1979, chapter 7*). The algortihm is able to handle the analysis equation for a filter based on factorization as wel as an ensemble filter (table 6.1), for analysis with a minimal variance gain.

Given the observation equation $\mathbf{y}^o = \mathbf{H}'\mathbf{x} + \mathbf{v}$ with $\mathbf{v} \sim \mathcal{N}(\mathbf{o}, \mathbf{R})$ and $\mathbf{R}$ diagonal, the analysis of the forecasted mean $\hat{\mathbf{x}}^f$ and covariance-square-root $\mathbf{S}^f$ is given by:

$$\begin{aligned}
\boldsymbol{\Psi}' &= \mathbf{H}'\,\mathbf{S}^f && \text{(B.1a)}\\
\mathbf{K} &= \mathbf{S}^f\,\boldsymbol{\Psi}(\boldsymbol{\Psi}'\boldsymbol{\Psi} + \mathbf{R})^{-1} && \text{(B.1b)}\\
\hat{\mathbf{x}}^a &= \hat{\mathbf{x}}^f \,+\, \mathbf{K}\,(\mathbf{y}^o - \mathbf{H}'\hat{\mathbf{x}}^f) && \text{(B.1c)}\\
\mathbf{S}^a &= \mathbf{S}^f\,\left[\mathbf{I} \,-\, \boldsymbol{\Psi}(\boldsymbol{\Psi}'\boldsymbol{\Psi} + \mathbf{R})^{-1}\boldsymbol{\Psi}'\right]^{1/2} && \text{(B.1d)}
\end{aligned}$$

The Potters algorithm is able to analyze the covariance square root without matrix factorization (B.1d), in case that the observation vector $\mathbf{y}^o$ has only one element. For the vector case, the algorithm should be applied for each element one after the other. The result does not differ from analysing all measurements at once using a matrix factorization. Assimilation of a sequence of measurements with infinite small time intervals in between is not different from analysis of all measurements together, if the mean and covariance square root have not been changed.

The Potters algorithm is bassed on the observation that for a scalar measurement $y^o = \mathbf{h}'\mathbf{x} + v$, with $v \sim \mathcal{N}\left(0, r^2\right)$, the matrix to be inverted in (B.1d) is the scalar $\sigma^2 = \psi'\psi + r^2$. The matrix in square brackets becomes a Householder matrix with simple to derive factorization:

$$\mathbf{I} - \sigma^{-2}\psi\psi' \;=\; \left(\mathbf{I} - \alpha\psi\psi'\right)\left(\mathbf{I} - \alpha\psi\psi'\right)' \quad , \quad \alpha = 1/(\sigma(\sigma + r)) \qquad \text{(B.2)}$$

The complete analysis of a scalar measurement becomes:

$$\begin{aligned}
\mathbf{k} &= \mathbf{S}^f\psi/\sigma^2 && \text{(B.3a)}\\
\mathbf{u}' &= \psi'\,r/(\sigma + r) && \text{(B.3b)}\\
\hat{\mathbf{x}}^a &= \hat{\mathbf{x}}^f \,+\, \mathbf{k}\,(y^o - \mathbf{h}'\hat{\mathbf{x}}^f) && \text{(B.3c)}\\
\mathbf{S}^a &= \mathbf{S}^f\,\left[\mathbf{I} \,-\, \psi\psi'/(\sigma(\sigma + r))\right] \,=\, \mathbf{S}^f \,+\, \mathbf{k}\left(\mathbf{u}' - \psi'\right) && \text{(B.3d)}
\end{aligned}$$

Comparison of (B.3) with table 6.1 shows that the scalar analysis with Potters algorithm takes the same form as the ensemble analysis. With a different vector $\mathbf{u}'$ and an average observation error added to y^o, the equations are the same. The most expensive part of the scalar update are the computation of the gain vector $\mathbf{k}$ in (B.3a) and the replacement of the modes in (B.3d). Each requires about $2nm$ flops (page 187); if a number of n_y measurments is analyzed in this way, $4n_y mn$ flops are required in total. The memory requirements are limited, since apart from some smaller entities only the gain vector $\mathbf{k}$ needs to be stored.

Repeated application of Potters algorithm becomes expensive if the number of measurements grows. However, it is possible to combine a number of scalar analysis in a way that the mean and covariance matrix are updated only once according to:

$$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f + \mathbf{S}^f\,\mathbf{a} \tag{B.4a}$$

$$\mathbf{S}^a = \mathbf{S}^f\,\mathbf{B} \tag{B.4b}$$

The m-vector $\mathbf{a}$ and $m \times m$ matrix $\mathbf{B}$ are sequentially updated for each element of $\mathbf{y}^o$. With a small difference for factorized or ensemble kind of analysis, the *repeated scalar update* is given by:

```
$\mathbf{a}_0 = \mathbf{o}$ , $\mathbf{B}_0 = \mathbf{I}$    (B.5a)
for $l = 1, \ldots, \mathrm{size}(\mathbf{y}^o)$
    $\phi = {\mathbf{h}_l}'\hat{\mathbf{x}}^f$ , $\varphi' = {\mathbf{h}_l}'\,\mathbf{S}^f$    (B.5b)
    $\psi' = \varphi'\,\mathbf{B}_{l-1}$    (B.5c)
    $\sigma^2 = \psi'\psi + r_l^2$    (B.5d)
```

	factorized filter	ensemble filter
		$(\mathbf{v}')_{j=1,..,m} \sim \mathcal{N}\left(0, r_l^2\right)$
$\bar{\mathbf{v}} =$	$\mathbf{o}$	$\overline{(\mathbf{v}')_j}$
$\mathbf{u}' =$	$\psi' r_l/(\sigma + r_l)$	$(\mathbf{v}' - \bar{\mathbf{v}})/\sqrt{m-1}$

(B.5e)

```
    $\mathbf{a}_l = \mathbf{a}_{l-1} + \mathbf{B}_{l-1}\psi\{y_l^o + \bar{\mathbf{v}} - (\phi + \varphi'\mathbf{a}_{l-1})\}/\sigma^2$    (B.5f)
    $\mathbf{B}_l = \mathbf{B}_{l-1}\left(\mathbf{I} - \psi(\mathbf{u}' - \psi')/\sigma^2\right)$    (B.5g)
end for
$\mathbf{a} = \mathbf{a}_l$ , $\mathbf{B} = \mathbf{B}_l$    (B.5h)
```

The matrix $\mathbf{B}$ formed in this way is in fact the matrix square root of the bracketed term in eq. (B.1d), obtained with a sequence of Householder matrices. The costs of the repeated scalar update are completely determined by transformation (B.4b), approximately $2m^2n$ flops. The repeated update is therefore cheaper than sequential application of Potters algorithm if more than $m/2$ measurements are to be analyzed, or if the transformation with $\mathbf{B}$ is combined with other transformations in the filter.

Appendix C

Taylor expansions

The accuracy of the different nonlinear forecast methods described in chapter 7 are discused using Taylor expansions. The approach followed here is a generalization of the one used by in (*Julier et al., 1995*). After introduction of notations, conventions, and basic expansions, the accuracy of first and second order linearizations will be discussed in §C.2 and §C.3; for completeness, also the Taylor expansion of the ensemble forecast is given §C.4. The equation used for the bias propagation is derived in §C.5.

C.1 Notations and conventions

To facilitate simple notations of Taylor expansions, an operator D is introduced for summations over partial derivatives:

$$D^l_{\boldsymbol{\delta}\mathbf{x}} = \frac{1}{l!}\left(\boldsymbol{\delta}\mathbf{x}'\nabla\right)^l = \frac{1}{l!}\left(\sum_{k=1}^{n}\delta x_k \frac{\partial}{\partial x_k}\right)^l \qquad \text{(C.1)}$$

where ∇ denotes the vector with partial derivatives to the state elements and $\boldsymbol{\delta}\mathbf{x}$ denotes a (small) difference between two state vectors. For a vector function $\mathbf{M}$, two specific examples are:

$$D_{\delta x}\mathbf{M} = \left(\boldsymbol{\delta}\mathbf{x}'\boldsymbol{\nabla}\right)\mathbf{M} \qquad \text{(C.2a)}$$

$$D^2_{\delta x}\mathbf{M} = \tfrac{1}{2}\left(\boldsymbol{\nabla}'\boldsymbol{\delta}\mathbf{x}\boldsymbol{\delta}\mathbf{x}'\boldsymbol{\nabla}\right)\mathbf{M} \qquad \text{(C.2b)}$$

Let $\mathbf{x}^t[k]$ be the true value of the state vector at a time $t[k]$, which deviates from a user derived state $\hat{\mathbf{x}}[k]$ by $\boldsymbol{\delta}\mathbf{x}[k] = \mathbf{x}^t[k] - \hat{\mathbf{x}}[k]$. With this notation, the true state at the next time step is equal to a Taylor series of the stochastic model $\mathbf{M}$ from eq. (7.1a) around $\hat{\mathbf{x}}[k]$:

$$\mathbf{x}^t[k+1] = \mathbf{M}\left(\mathbf{x}^t[k]\right) + \boldsymbol{\eta}[k] = \mathbf{M}\left(\hat{\mathbf{x}}[k] + \boldsymbol{\delta}\mathbf{x}[k]\right) + \boldsymbol{\eta}[k] \qquad \text{(C.3a)}$$

$$= \mathbf{M}(\hat{\mathbf{x}}) + \left(D_{\delta x}\mathbf{M}\right)(\hat{\mathbf{x}}) + \left(D^2_{\delta x}\mathbf{M}\right)(\hat{\mathbf{x}}) \qquad \text{(C.3b)}$$

$$+ \left(D^3_{\delta x}\mathbf{M}\right)(\hat{\mathbf{x}}) + \ldots + \boldsymbol{\eta}[k] \qquad \text{(C.3c)}$$

We adopt the convention that the model or partial derivatives of the model are evaluated in $\hat{\mathbf{x}}$ if not mentioned otherwise. Taking expectation over (C.3) leads to an equation for the mean forecast:

$$\hat{\mathbf{x}}^f[k+1] = \mathrm{E}\left[\,\mathbf{x}^t[k+1]\,\right] \qquad \text{(C.4)}$$

$$= \mathbf{M}(\hat{\mathbf{x}}) + \mathrm{E}\left[\,D_{\delta x}\mathbf{M}\,\right] + \mathrm{E}\left[\,D^2_{\delta x}\mathbf{M}\,\right] + \mathrm{E}\left[\,D^3_{\delta x}\mathbf{M}\,\right] + \ldots \qquad \text{(C.5)}$$

If the distribution of $\boldsymbol{\delta}\mathbf{x}$ is symmetric (for example Gaussian), expectation over factors formed by an odd number of elements of $\boldsymbol{\delta}\mathbf{x}$ (odd powers of operator $D_{\delta x}$) will be zero.

The true error covariance of $\hat{\mathbf{x}}^f[k+1]$ is given by:

$$\begin{aligned}
\mathbf{P}^{f,t}[k+1] &= \mathrm{E}\left[\left(\mathbf{x}^t[k+1]-\hat{\mathbf{x}}^f[k+1]\right)\left(\mathbf{x}^t[k+1]-\hat{\mathbf{x}}^f[k+1]\right)'\right] && \text{(C.6a)}\\
&= \mathrm{E}\left[D_{\delta x}\mathbf{M}D_{\delta x}\mathbf{M}'\right] - \mathrm{E}\left[D_{\delta x}\mathbf{M}\right]\mathrm{E}\left[D_{\delta x}\mathbf{M}'\right] && \text{(C.6b)}\\
&\quad + \left(\mathrm{E}\left[D_{\delta x}\mathbf{M}D^2_{\delta x}\mathbf{M}'\right] - \mathrm{E}\left[D_{\delta x}\mathbf{M}\right]\mathrm{E}\left[D^2_{\delta x}\mathbf{M}'\right]\right) && \text{(C.6c)}\\
&\quad + \left(\mathrm{E}\left[D^2_{\delta x}\mathbf{M}D_{\delta x}\mathbf{M}'\right] - \mathrm{E}\left[D^2_{\delta x}\mathbf{M}\right]\mathrm{E}\left[D_{\delta x}\mathbf{M}'\right]\right) && \text{(C.6d)}\\
&\quad + \ldots
\end{aligned}$$

The true covariance (C.6) is an indication for the quality of our estimate $\hat{\mathbf{x}}^f$, since it describes what the difference between $\hat{\mathbf{x}}^f$ and $\mathbf{x}^t$ might be. However, even if $\mathbf{x}^t$ is available we are hardly able to compute $\mathbf{P}^{f,t}$ exactly. In practice, all what we have is a computed estimated $\mathbf{P}^{f,c}$, which should be as close to $\mathbf{P}^{f,t}$ as possible.

In the context of the low-rank Kalman filter, we usually assume that $\mathbf{x}^t[k]$ is a sample from a random distribution with mean $\mathbf{x}^a[k]$. The deviation $\boldsymbol{\delta}\mathbf{x}$ is a random variable with zero mean and covariance $\mathbf{P}^a$, parameterized by modes $\mathbf{s}_j$:

$$\mathrm{E}\left[\boldsymbol{\delta}\mathbf{x}\right] = \mathbf{o} \quad , \quad \mathrm{E}\left[\boldsymbol{\delta}\mathbf{x}\boldsymbol{\delta}\mathbf{x}'\right] = \mathbf{P}^a = \sum_{j=1}^{m}\mathbf{s}^a_j\mathbf{s}^{a\prime}_j \tag{C.7}$$

For simplicity, the index 'a' will be omitted from now on. The modes $\mathbf{s}_i$ are used to form input states for the model:

$$\mathbf{M}(\hat{\mathbf{x}}+\varepsilon\mathbf{s}_i) = \mathbf{M}(\hat{\mathbf{x}}) + \mathbf{o} + \varepsilon^2\left(D^2_{s_i}\mathbf{M}\right)(\hat{\mathbf{x}}) + \mathbf{o} + \ldots \tag{C.8}$$

These Taylor expansions will be used to analyze the theoretical difference between $\hat{\mathbf{x}}^f$ and $\hat{\mathbf{x}}^t$, and between $\mathbf{P}^{f,c}$ and $\mathbf{P}^{f,t}$ for the forecast methods discussed in chapter 7.

C.2 First order linearizations

To judge the accuracy of a forecast based on first order linearizations (§7.3.1), the mean and covariance computed with algorithm (7.12) should be compared with the true state and true covariance. Comparison of (7.12a) with (C.5) shows that the computed first order mean is equal to the first term of the Taylor series of $\mathbf{x}^t[k+1]$ in (C.3):

$$\hat{\mathbf{x}}^{f,ext1}[k+1] = \mathbf{M}(\hat{\mathbf{x}}[k]) \tag{C.9}$$

In case of a symmetric distribution of the state around $\hat{\mathbf{x}}$, it is also accurate up to the first order partial derivatives, since these are equal to zero in that case. The true covariance of this computed mean is equal to:

$$\begin{aligned}
\mathbf{P}^{f,ext1,t} &= \mathrm{E}\Big[\, \big(\mathbf{x}^{t}{}_{[k+1]} - \mathbf{x}^{f,ext1}{}_{[k+1]}\big)\big(\mathbf{x}^{t}{}_{[k+1]} - \mathbf{x}^{f,ext1}{}_{[k+1]}\big)' \,\Big] && \text{(C.10a)}\\
&= \mathrm{E}\big[\, D_{\delta x}\mathbf{M}D_{\delta x}\mathbf{M}' \,\big] - \mathbf{O} && \text{(C.10b)}\\
&\quad + \big(\mathrm{E}\big[\, D_{\delta x}\mathbf{M}D^2_{\delta x}\mathbf{M}' \,\big] - \mathbf{O}\big) + \big(\mathrm{E}\big[\, D^2_{\delta x}\mathbf{M}D_{\delta x}\mathbf{M}' \,\big] - \mathbf{O}\big) && \text{(C.10c)}\\
&\quad + \ldots
\end{aligned}$$

The actual with (7.12b) computed covariance is given by:

$$\begin{aligned}
\mathbf{P}^{f,ext1,c} &= \sum_{i=1}^{m} \mathbf{s}_i^{f,ext1}\mathbf{s}_i^{f,ext1\prime} = \mathrm{E}\big[\, D_{\delta x}\mathbf{M}D_{\delta x}\mathbf{M}' \,\big] - \mathbf{O} && \text{(C.11a)}\\
&\quad + \varepsilon\Big(\sum_{i=1}^{m} D_{s_i}\mathbf{M}D^2_{s_i}\mathbf{M}' - \mathbf{O}\Big) + \varepsilon\Big(\sum_{i=1}^{m} D^2_{s_i}\mathbf{M}D_{s_i}\mathbf{M}' - \mathbf{O}\Big) + \ldots && \text{(C.11b)}
\end{aligned}$$

where we used the relation

$$\begin{aligned}
\sum_{i=1}^{m} D_{s_i}\mathbf{M}D_{s_i}\mathbf{M}' &= \sum_{i=1}^{m} (\nabla\mathbf{M}')'\mathbf{s}_i\,\mathbf{s}_i(\nabla\mathbf{M}') = (\nabla\mathbf{M}')'\mathbf{P}(\nabla\mathbf{M}') && \text{(C.12a)}\\
&= (\nabla\mathbf{M}')'\mathrm{E}\big[\, \delta\mathbf{x}\delta\mathbf{x}' \,\big](\nabla\mathbf{M}') = \mathrm{E}\big[\, D_{\delta x}\mathbf{M}D_{\delta x}\mathbf{M}' \,\big] && \text{(C.12b)}
\end{aligned}$$

In the limit $\varepsilon \to 0$, the extended forecast becomes equal to the forecast of the Extended Kalman Filter, which uses Jacobian matrices to approximate non-linear dynamics. Using very small scalefactors has numerical disadvantages, however, such as computing differences between state vectors which hardly differ, and division by small numbers. Besides, close approximation of the EKF is not a final target of the forecast scheme. A better approach would be to set ε to a value that maximizes the (theoretical) accuracy of the forecast. The largest terms in computed mean and covariance which are influenced by the scalefactor are the third order moments in (C.11). These terms might be rewritten to a form that is close to an approximation of expectation by a sample mean:

$$\frac{1}{\varepsilon^2}\sum_{i=1}^{m} D_{\varepsilon s_i}\mathbf{M}D^2_{\varepsilon s_i}\mathbf{M}' \approx \mathrm{E}\big[\, D_{\delta x}\mathbf{M}D^2_{\delta x}\mathbf{M}' \,\big] \tag{C.13}$$

In a usual sample mean, all sample members are equivalent since there is no preference for direction or amplitude; each element in the result is an average over m equivalent elements. Therefore, each sample should be weighted by $1/m$, suggesting a $\varepsilon^2 = m$ is an appropriate choice. In the RRSQRT context, however, the 'sample members' $\mathbf{s}_i$ are not equivalent since they have been made orthogonal; each sample in (C.13) should become a unit weight, thus $\varepsilon^2 = 1$. Similar considerations about the model states fed to the model lead to the formulation of the ε-rule at page 125, also leading to scalefactors $\varepsilon = 1$. Setting ε to 1 seems to be a suitable choice for obtaining appropriate model input on forehand as well as accurate results afterwards.

C.3 Second order accurate forecast

Second order forecasts for the mean are given in equations (7.17) for the EXT2 and (7.20a) for the MES method. The Taylor series is in both cases given by:

$$\hat{\mathbf{x}}^{f,sec} = \mathbf{M}(\hat{\mathbf{x}}) + \sum_{i=1}^{\bar{m}} \frac{\mathbf{M}(\hat{\mathbf{x}}+\varepsilon\mathbf{s}_i)-\mathbf{M}(\hat{\mathbf{x}})}{\varepsilon^2} \tag{C.14a}$$

$$= \mathbf{M}(\hat{\mathbf{x}}) + \mathbf{o} + \mathrm{E}\left[\, D^2_{\delta x}\mathbf{M} \,\right] + \mathbf{o} + \varepsilon^4 \sum_{i=1}^{\bar{m}} \mathrm{E}\left[\, D^4_{s_i}\mathbf{M} \,\right] + \ldots \tag{C.14b}$$

where (C.2b) is used to derive:

$$\sum_{i=1}^{m} D^2_{s_i}\mathbf{M} = \sum_{i=1}^{\bar{m}} \frac{1}{2}\left\{\boldsymbol{\nabla}'\mathbf{s}_i\mathbf{s}_i{}'\boldsymbol{\nabla}\right\}\mathbf{M} = \frac{1}{2}\left\{\boldsymbol{\nabla}'\mathbf{P}\boldsymbol{\nabla}\right\}\mathbf{M} = \mathrm{E}\left[\, D^2_{\delta x}\mathbf{M} \,\right] \tag{C.15}$$

Comparison of (C.14b) with the optimal state estimate (C.5) shows that the second order forecasts compute the term with second order partial derivatives correctly. If the true state is distributed symmetric around $\hat{\mathbf{x}}$, the largest error occurs in the fourth order term, because the odd order terms in (C.5) vanish in that case. Otherwise, the largest error appears in the first order term, but, under the assumption that the distribution is at least close to symmetric, this error is small.

The true covariance of (C.14b) is given by:

$$\mathbf{P}^{f,sec,t}[k+1] = \mathrm{E}\Big[\, \left(\mathbf{x}^t[k+1]-\hat{\mathbf{x}}^{f,sec}[k+1]\right)\left(\mathbf{x}^t[k+1]-\hat{\mathbf{x}}^{f,sec}[k+1]\right)' \,\Big] \tag{C.16a}$$

$$= \mathrm{E}\left[\, D_{\delta x}\mathbf{M}D_{\delta x}\mathbf{M}' \,\right] - \mathbf{O} \tag{C.16b}$$

$$+ \left(\mathrm{E}\left[\, D_{\delta x}\mathbf{M}D^2_{\delta x}\mathbf{M}' \,\right] - \mathrm{E}\left[\, D_{\delta x}\mathbf{M} \,\right]\mathrm{E}\left[\, D^2_{\delta x}\mathbf{M}' \,\right]\right) \tag{C.16c}$$

$$+ \left(\mathrm{E}\left[\, D^2_{\delta x}\mathbf{M}D_{\delta x}\mathbf{M}' \,\right] - \mathrm{E}\left[\, D^2_{\delta x}\mathbf{M} \,\right]\mathrm{E}\left[\, D_{\delta x}\mathbf{M}' \,\right]\right) \tag{C.16d}$$

$$+ \left(\mathrm{E}\left[\, D_{\delta x}\mathbf{M}D^3_{\delta x}\mathbf{M}' \,\right] - \mathbf{O}\right) + \left(\mathrm{E}\left[\, D^3_{\delta x}\mathbf{M}D_{\delta x}\mathbf{M}' \,\right] - \mathbf{O}\right) \tag{C.16e}$$

$$+ \left(\mathrm{E}\left[\, D^2_{\delta x}\mathbf{M}D^2_{\delta x}\mathbf{M}' \,\right] - \mathrm{E}\left[\, D^2_{\delta x}\mathbf{M} \,\right]\mathrm{E}\left[\, D^2_{\delta x}\mathbf{M}' \,\right]\right) + \ldots \tag{C.16f}$$

For computation of the new covariance modes, three algorithms have been listed in table 7.1. Method 'a' calculates the new modes as deviations from the computed mean:

$$\mathbf{s}_i^{f,sec^a}[k+1] = \frac{\boldsymbol{\xi}_i[k+1]-\hat{\mathbf{x}}^{f,sec}[k+1]}{\varepsilon} \tag{C.17}$$

The computed covariance expanded in Taylor series becomes:

$$\mathbf{P}^{f,sec^a} = \mathrm{E}\left[\, D_{\delta x}\mathbf{M}D_{\delta x}\mathbf{M}' \,\right] - \mathbf{O} \tag{C.18a}$$

$$+ \left(\mathbf{O} - \mathbf{O}\right) + \left(\mathbf{O} - \mathbf{O}\right) \tag{C.18b}$$

$$+ \varepsilon^2\Big(\sum_{i=1}^{\bar{m}} D_{s_i}\mathbf{M}D^3_{s_i}\mathbf{M}' - \mathbf{O}\Big) + \varepsilon^2\Big(\sum_{i=1}^{\bar{m}} D^3_{s_i}\mathbf{M}D_{s_i}\mathbf{M}' - \mathbf{O}\Big) \tag{C.18c}$$

$$+ \left(\varepsilon^2 \sum_{i=1}^{\bar{m}} D^2_{s_i}\mathbf{M}D^2_{s_i}\mathbf{M}' - \left(2-\bar{m}/\varepsilon^2\right)\mathrm{E}\left[\, D^2_{\delta x}\mathbf{M} \,\right]\mathrm{E}\left[\, D^2_{\delta x}\mathbf{M}' \,\right]\right) \tag{C.18d}$$

$$+ \ldots$$

In case of a symmetric distributed state, the largest errors are introduced in the fourth order terms, because odd order terms vanish in that case.

With the choice $\varepsilon = \sqrt{\bar{m}}$, the expansion shows much resemblance with a Monte-Carlo-like approximation; the factor $2 - \bar{m}/\varepsilon^2$ in (C.18d) becomes equal to 1.0 which is correct according to the expansion of the true covariance (C.16). This choice is undesirable, however, because the input to the dynamics could become unrealistic large. Taking the ε very small is undesirable too, because this would amplify the second term in (C.18d). This effect can be explained from the fact that in case of a small scalefactor, the propagated states form a cloud around the central forecast of the mean. Algorithm C.17 computes the covariance around the second order forecast, however. If the scalefactor becomes small enough, this estimate will be outside the propagated cloud, leading to an over-estimation of the covariance. The problems with the second term in (C.18d) are avoided by computing the new modes as deviations from the central forecast of the mean (method 'b' in table 7.1):

$$\mathbf{s}_i^{f,sec^b}[k+1] \;=\; \frac{\boldsymbol{\xi}_i[k+1] - \boldsymbol{\xi}_0[k+1]}{\varepsilon} \tag{C.19}$$

It is straight forward to show that the computed covariance following from this method is equal to (C.18e), except that the second term in (C.18d) vanishes. Besides the 'a' and 'b' methods for computing the new modes, a third 'c' method was suggested in (*Voorrips et al., 1999*) for use in the EXT2 forecast (section 7.3.2):

$$\mathbf{s}_i^{f,sec^c}[k+1] \;=\; \frac{\boldsymbol{\xi}_{+i}[k+1] - \boldsymbol{\xi}_{-i}[k+1]}{2\varepsilon} \quad , \quad i = 1, \ldots, \bar{m} \tag{C.20}$$

In here, the state $\boldsymbol{\xi}_{+i}[k+1]$ and $\boldsymbol{\xi}_{-i}[k+1]$ denote states formed after propagation of a mode in positive and negative direction respectively. Method (C.20) saves memory, because the number of modes is not doubled; eq. (7.17) for computation of a second order accurate forecast of the mean can still be used, however. The computed covariance in case of method 'c' is similar to (C.18e), except that both terms in (C.18d) completely vanishes. The method has therefore the disadvantage of a small under estimation of the covariance.

C.4 Ensemble forecast

In a similar way as has been done for the first and second order linearizations, the ensemble forecast could be expanded in Taylor series too. At a time $t[k]$, the ensemble members ξ_j form a cloud around their sample mean $\hat{\mathbf{x}}$ such that the scaled deviations $\mathbf{s}_i = (\boldsymbol{\xi}_i - \hat{\mathbf{x}})/\sqrt{m-1}$ have zero mean and $\sum \mathbf{s}_j \mathbf{s}_j{}' = \mathbf{P}$. If the ensemble members are propagated, the new sample mean is given by:

$$\hat{\mathbf{x}}^{f,ens} \;=\; \frac{1}{m}\sum_{i=1}^{m} \mathbf{M}(\hat{\mathbf{x}} + \sqrt{m-1}\,\mathbf{s}_i) \tag{C.21a}$$

$$=\; \frac{1}{m}\sum_{i=1}^{m} \left\{ \mathbf{M}(\hat{\mathbf{x}}) \;+\; \sqrt{m-1}(\boldsymbol{\nabla}'\mathbf{M})\mathbf{s}_i \;+\; \tfrac{m-1}{2}\boldsymbol{\nabla}'(\mathbf{s}_i\mathbf{s}_i{}')\boldsymbol{\nabla}\mathbf{M} \;+\; \ldots \right\} \tag{C.21b}$$

$$=\; \mathbf{M}(\hat{\mathbf{x}}) \;+\; \mathbf{o} \;+\; \frac{m-1}{m}\boldsymbol{\nabla}'\mathbf{P}^t\boldsymbol{\nabla}\,\mathbf{M} \;+\; \ldots \tag{C.21c}$$

The first terms of the new mean converge to the first terms of a second order linearization with order $1/m$. For strongly nonlinear functions, the ensemble forecast becomes more accurate than the linearizations since also higher order terms are approximated correctly (for large m). In a similar way, the computed covariance (a sample covariance) can be shown to converge to the true covariance with order $1/m$ too.

C.5 Bias equation

For observation of the nonlinearity as described in §7.5, an equation for the forecast of the bias should be derived. The bias at a time $t[k{+}1]$ is defined as the expected difference between true state and central forecast of the mean:

$$\mathbf{b}[k{+}1] \;=\; \mathrm{E}\left[\,\mathbf{x}^t[k{+}1] - \mathbf{M}(\hat{\mathbf{x}}[k])\,\right] \tag{C.22}$$

With $\boldsymbol{\delta}\mathbf{x} = \mathbf{x}^t - \hat{\mathbf{x}}$ and $\mathrm{E}[\,\boldsymbol{\delta}\mathbf{x}\,] = \mathbf{b}$, and the aid of the Taylor expansion (C.3) of the true state, the forecast of the bias becomes:

$$\begin{aligned}
\mathbf{b}^f[k{+}1] \;&=\; \mathrm{E}\left[\,(D_{\delta x}\mathbf{M})(\hat{\mathbf{x}}) \;+\; \left(D^2_{\delta x}\mathbf{M}\right)(\hat{\mathbf{x}}) \;+\; \left(D^3_{\delta x}\mathbf{M}\right)(\hat{\mathbf{x}}) \;+\; \ldots\,\right] && \text{(C.23a)}\\
&=\; (D_b\mathbf{M})(\hat{\mathbf{x}}) \;+\; \mathrm{E}\left[\,\left(D^2_{\delta x}\mathbf{M}\right)(\hat{\mathbf{x}})\,\right] \;+\; \mathrm{E}\left[\,\left(D^3_{\delta x}\mathbf{M}\right)(\hat{\mathbf{x}})\,\right] \;+\; \ldots && \text{(C.23b)}\\
&=\; (\partial\mathbf{M}/\partial\mathbf{x})(\mathbf{b}) \;+\; \left[\,\hat{\mathbf{x}}^{f,\star} \;-\; \mathbf{M}(\hat{\mathbf{x}})\,\right] && \text{(C.23c)}
\end{aligned}$$

where $\hat{\mathbf{x}}^{f,\star}$ is one of the higher order forecasts EKF2 or MES expanded in (C.14), or the EKF forecast expanded in (C.21). The bias is thus propagated by the Jacobian $\partial\mathbf{M}/\partial\mathbf{x}$ and increased with the difference between a first and second order forecast.

Appendix D

Minimal Exact Sampling

The term 'minimal exact sample' (MES) was introduced in (*Pham, 1996*) for the forecast step of the SEIK filter, and is used in §7.3.2 to define a second order accurate forecast. A MES was defined as the smallest possible set of modes with zero mean and matching a given covariance.

Given a covariance $\mathbf{P} = \mathbf{S}\mathbf{S}'$ defined by the rank m square root $\mathbf{S}$, the MES consists of a set of linear combinations $\mathbf{S}\boldsymbol{\omega}_j$ of columns of $\mathbf{S}$, for $j = 1,..,\bar{m}$. In matrix notation: the set is formed by the columns of the matrix $\mathbf{S}\ \boldsymbol{\Omega}^{mes}$. To let the set have zero mean and covariance $\mathbf{P}$, the vectors $\boldsymbol{\omega}_j$ should satisfy two constraints:

$$\sum_{j=1}^{\bar{m}} \mathbf{S}\boldsymbol{\omega}_j = \mathbf{o} \qquad \Rightarrow \qquad \sum_{j=1}^{\bar{m}} \boldsymbol{\omega}_j = \mathbf{o} \tag{D.1}$$

$$\sum_{j=1}^{\bar{m}} (\mathbf{S}\boldsymbol{\omega}_j)(\mathbf{S}\boldsymbol{\omega}_j)' = \mathbf{S}\mathbf{S}' \qquad \Rightarrow \qquad \sum_{j=1}^{\bar{m}} \boldsymbol{\omega}_j\boldsymbol{\omega}_j' = \mathbf{I} \tag{D.2}$$

Both requirements are met if the columns of the following $\bar{m} \times (m+1)$ matrix $\mathbf{W}$ are orthogonal:

$$\mathbf{W} = \begin{pmatrix} \bar{m}^{-1/2}, & - & \boldsymbol{\omega}_1' & - \\ \bar{m}^{-1/2}, & - & \boldsymbol{\omega}_2' & - \\ \vdots & & \vdots & \\ \bar{m}^{-1/2}, & - & \boldsymbol{\omega}_{\bar{m}}' & - \end{pmatrix} \tag{D.3}$$

The minimal value of $\bar{m}$ for which this is possible is $m+1$. An algorithm proposed in (*Pham, 1996*) constructs a suitable matrix $\mathbf{W}$ using Householder reflections. The Householder reflection of a vector $\mathbf{z} \in I\!R^k$ is a $k \times k$ orthonormal matrix with $\mathbf{z}$ as its first column:

$$\mathbf{Ho}(\mathbf{z}) = \mathbf{I}_k - \frac{1}{1-z_1} \begin{bmatrix} z_1 - 1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix} [z_1 - 1, z_2, \cdots, z_k] \tag{D.4}$$

The algorithm to produce a suitable matrix $\mathbf{W}$ is now:

```
$\mathbf{W}_1 = \pm 1$
for $k = 2, \ldots, \bar{m}$ do
  if $k == \bar{m}$ then
    $\mathbf{z}_k = [1, \cdots, 1]' \in \mathbb{R}^k$
  else
    $\mathbf{z}_k \sim \mathcal{N}(\mathbf{o}, \mathbf{I}_k)$
  end if
  $\mathbf{H}(\mathbf{z}_k) = [\, \mathbf{Ho}(\mathbf{z}_k / \|\mathbf{z}\|) \,]_{(:,2:k)}$
  $\mathbf{W}_k = [\, \mathbf{z}_k, \mathbf{H}(\mathbf{z}_k)\, \mathbf{W}_{k-1}' \,]$
end for
$\mathbf{W} = \mathbf{W}_{\bar{m}}$
```

The subscript beneath the brackets around the Householder matrix denotes that the $k \times (k-1)$ matrix $\mathbf{H}(\mathbf{z}_k)$ is formed with the second to the last column. The $k \times k$ matrices $\mathbf{W}_k$ have $\mathbf{z}_k$ as its first column, and are orthonormal because the other columns are just a rotation of the orthonormal complement of $\mathbf{z}_k$. The special choice for $\mathbf{z}_{\bar{m}}$ ensures that $\mathbf{W}_{\bar{m}}$ is the matrix $\mathbf{W}$ from eq. (D.3). The matrix $\boldsymbol{\Omega}^{mes}$ for the minimal exact sample is formed by the transpose of the last columns of $\mathbf{W}_{\bar{m}}$:

$$\boldsymbol{\Omega}^{mes} = [\, \mathbf{W} \,]_{(:,2:)}{}' \tag{D.5}$$

Since the vectors $\mathbf{z}_k$ in the algorithm are drawn from a random generator, the number of suitable matrices $\boldsymbol{\Omega}^{mes}$ which can be formed is infinite. The use of a random generator will not lead to formation of samples with unrealistic large elements. Because $\boldsymbol{\Omega}^{mes\prime}$ has orthonormal columns, all its elements have a magnitude less or equal to one. Thus, a new mode $\mathbf{S}\boldsymbol{\omega}_j$ is build up as a summation of (positive or negative) fractions of the older modes.

It is possible to construct a 'general exact sample' (GES) with $\bar{m} > m+1$, as a generalization of the case with minimal $\bar{m}$. The algorithm above should still produce a $\bar{m} \times \bar{m}$ matrix $\mathbf{W}$; the rows of the last m columns form a suitable set of vectors $\boldsymbol{\omega}_j$. Using a GES rather than a MES provides a flexible way of generating more or less random samples of any size with given covariance.

Appendix E

Symbols and notations

E.1 Vectors and matrices

entity	description
$\mathbf{x}, \boldsymbol{\xi}$	vector (column)
$\mathbf{A}, \boldsymbol{\Gamma}$	matrix
$\mathbf{x}', \mathbf{A}'$	transpose
$\mathbf{o}$	null vector
$\mathbf{I}$	identity matrix

E.2 Mathematical accents

notation	description
x^t	true value
x^o	observed value
x^b	background value
x^f	forecasted value
x^a	analyzed value
$\hat{x}$	mean value
x^e	ensemble entitie

E.3 Definitions

term	definition
flop	(floating point operation) Defintion in (*Golub and van Loan, 1996*): one flop is the amount of work associated with an operation of the form $a = b + c$ or $a = b \times c$.

E.4 Symbols

symbol	description	see page(s)
α	auto correlation parameter	42
$\boldsymbol{\beta}$	coefficients of bias projection	135–139
$\boldsymbol{\Gamma}$	covariance of residue $\mathbf{d}$	78

γ	colored noise process	42
ε	scale-factors in finite difference approximation	125
$\boldsymbol{\eta}$	model error vector	37, 41–45
$\boldsymbol{\Theta}$	projection of gain on covariance square root	104
θ	nonlinearity measure	135–139
$\boldsymbol{\Lambda}$	eigenvalue matrix	103
λ	model parameter	43–45, 50
$\boldsymbol{\xi}$	ensemble member (sample state)	94, 99–101, 103
$\boldsymbol{\Pi}$	projection matrix	102
ρ	parameter of residue covariance $\boldsymbol{\Gamma}$	78
σ	standard deviation	42
σ_α	correlation dependent standard deviation	43
τ	auto correlation parameter	42
ϕ_k	number of transfer stages	149
$\boldsymbol{\Psi}$	projection of $\mathbf{S}$ on observation space	94, 147, 161
$\boldsymbol{\Omega}$	forecast ensemble coefficient matrix	97, 103, 185
$\boldsymbol{\omega}$	column of $\boldsymbol{\Omega}$	185
$\mathbf{A}$	linear model	33, 91, 105
$\mathbf{a}$	analysis coefficient vector for mean	103, 178
$\mathbf{B}$	analysis coefficient matrix for $\mathbf{S}$	103, 178
$\mathbf{b}$	bias	135–139, 184
$\mathbf{c}$	concentration vector (ppb)	20, 31, 41, 158
$c_p(z)$	deposition profile (ppb)	26
D	partial derivative operator	179–184
$\mathbf{d}$	residual or innovation vector	32, 36, 78
$\mathbf{E}$	ensemble covariance square root	99, 102
E	efficiency	144
$\mathbf{G}$	model error operator	45, 50
$\mathbf{H}'$	observation operator	31
$\mathbf{Ho}$	Householder matrix	186
h_{ref}	reference height for deposition profile	25
J	photolysis rate	13, 55–58, 85
$\mathcal{J}$	cost function	32–33, 39
$\mathcal{J}$	set of mode identifiers	147
$\mathbf{K}$	gain matrix	34, 35, 38
$\mathcal{K}$	set of processor identifiers	146

k_n	reaction constant	13
$\boldsymbol{\mathcal{L}}$	LOTOS model operator	20, 31, 41, 158
$\mathbf{M}$	nonlinear model	31, 45, 50
m	number of modes (columns of $\mathbf{S}$)	92
n	size of state vector	45
n_{pe}	number of processors	146
$\mathbf{P}$	state error covariance matrix	33–35
p	nonlinearity measure	135–139
$\mathbf{Q}$	model error covariance matrix	37
q	number of elements in model error noise $\mathbf{w}$	45
$\mathbf{R}$	representation error covariance matrix	33, 35
r	size of observation vector $\mathbf{y}$	31
R_a	atmospheric resistance (s/m)	25
R_b	viscous-sublayer resistance (s/m)	26
R_c	surface resistance (s/m)	25, 60
R_t	atmospheric and viscous-sublayer resistance (s/m)	60
S	speedup	144
$\mathbf{S}$	error covariance square root	92–94, 122, 145
$\mathbf{s}$	mode of error covariance, column of $\mathbf{S}$	92–94
T	execution time	144
$\mathbf{T}$	model error covariance square root	94
$\mathbf{t}$	mode of model error covariance, column of $\mathbf{T}$	122–127
t	time	20, 31
$\mathbf{U}$	representation error covariance square root	94
V	nonlinearity number	135–139
$\mathbf{V}$	eigenvector matrix	96, 103
$\mathbf{v}$	representation error vector	37
v_d	deposition velocity (m/s)	26
$\mathbf{w}$	model error noise	41, 45, 50
$\mathbf{x}$	state vector	31, 45, 50
$\mathbf{y}$	observation vector	31, 72

Acknowledgments

This work has been carried out with financial support of the European Community as a part of the research project RIFTOZ, and the Netherlands Remote Sensing Board (BCRS) as a part of the research project STROPDAS.

The following institutes are acknowledged for providing measurements:

- the Britisch *Department of the Environment, Transport and the Regions* for the ground based measurements in the *UK National Air Quality Information Archive*;
- the *Norwegian Institute for Air Research* for the ozone measurements collected from the EMEP network;
- EURAD (University of Köln, Department of Geophysiscs and Meteorology) for measurements in central Europe,
- the Belgium *Royal Meteorological Institute* (KMI) for the ozone soundings from Uccle.

The Delft University *Center for High Performance Applied Computing* (HPαC) is acknowledged for providing access to the CRAY T3E .

Bibliography

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering*. Prentice-Hall, Englewood Cliffs.

Barone, G., Ambra, P. D., di Serafino, D., Giunta, G., Montella, R., Murli, A., and Riccio, A. (2000). An operational mesoscale air quality model for the Campania region. In Barone, G., Builtjes, P. J. H., and Giunta, G., editors, *Proceedings of the 3rd GLOREAM Workshop, Ischia (Naples), september 22-24, 1999*, pages 179–189, Italy. Naval University of Naples.

Bierman, G. J. (1977). *Factorization Methods for Discrete Sequential Estimation*, volume 128 of *Mathematics in Science and Engineering*. Academic Press, New York.

Bucy, R. S. and Joseph, P. D. (1968). *Filtering for Stochastic Processes with Applications to Guidance*. Weiley-Interscience. 195 pp.

Builtjes, P. (1992). The LOTOS - Long Term Ozone Simulation-project; summary report. TNO-report TNO-MW - R 92/240, TNO, Delft, The Netherlands.

Burgers, G., van Leeuwen, P. J., and Evensen, G. (1998). Analysis scheme in the Ensemble Kalman filter. *Mon. Weather Rev*, 126:1719–1724.

Cañizares, R. (1999). *On the Application of Data Assimilation in Regional Coastal Models*. PhD thesis, Delft University of Technology / International Insitute for Infrastructural, Hydraulic and Environmental Engineering.

Cohn, S., Sivakumaran, N., and Todling, R. (1994). A fixed-lag Kalman Smoother for retrospective data assimilation. *Mon. Weather Rev*, 122:2838–2867.

Cohn, S. E. and Todling, R. (1995). Approximate data assimilation schemes for stable and unstable dynamics. *J. Meteorological Soc. of Japan*, 74(1):63–75.

Committee on Tropospheric Ozone (1991). *Rethinking the ozone problem in urban and regional air pollution*. National Research Council, National Academy Press, Washington D.C.

Daley, R. (1991). *Atmospheric Data Analysis*. Cambridge University Press, Cambridge.

Dee, D. (1995). On-line estimation of error covariance parameters for atmospheric data assimilation. *Mon. Weather Rev*, 123:1128–1145.

Elbern, H. and Schmidt, H. (2001). Ozone episode analysis by four dimensional variational chemistry data assimilation. *J. Geophys. Res.*, 106(D4):3569.

Elbern, H., Schmidt, H., and Ebel, A. (1997). Variational data assimilation for tropospheric chemistry modeling. *J. Geophys. Res.*, 102(D13):15,967–15,985.

Elbern, H., Schmidt, H., Talagrand, O., and Ebel, A. (2000). 4D-variational data assimilation with an adjoint air quality model for emission analysis. *Environmental Modelling and Software*, 15(6–7):539–548.

Encalada, O., Pérez-Correa, J., Jorquera, H., and Solar, I. (1998). Indoor carbon monoxide contamination in Santiago, Chile. In Brebbia, C., Ratto, C., and Power, H., editors, *Airpollution VI*, Advances in Air Pollution, pages 247–256, Ashurst Lodge, Ashurst, Southampton, UK. Wessex Institute of Technology, WIT Press/Computational Mechanics Publications.

Eskes, H., Piters, A., Levelt, P., Allaart, M., and Kelder, H. (1999). Variational assimilation of GOME total-column ozone satellite data in a 2d latitude-longitude tracer-transport model. *J. Atmos. Sci.*, 56:3560–3572.

Evensen, G. (1992). Using the Extended Kalman Filter with a multilayer quasi-geostrophic ocean model. *J. Geophys. Res.*, 97:17905–17924.

Evensen, G. (1993). Open boundary conditions for the Extended Kalman Filter with a quasi-geostrophic ocean model. *J. Geophys. Res.*, 98(C9):16529–16546.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.*, 99(C5):10143–10162.

Evensen, G. (1997). Advanced data assimilation for strongly nonlinear dynamics. *Mon. Weather Rev*, 125:1342–1354.

Evensen, G. and van Leeuwen, P. J. (1996). Assimilation of Geosat altimeter data for the Agulhas Current using the Ensemble Kalman Filter with a quasi-geostrophic model. *Mon. Weather Rev*, 124:85–96.

Evensen, G. and van Leeuwen, P. J. (2000). An Ensemble Kalman Smoother for nonlinear dynamics. *Mon. Weather Rev*, 128:1852–1867.

Foster, I. (1995). *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison-Wesley. 381 pp.

Fritsch, G. and Möhres, G. (1998). Multistage simulations for turbomachinery design on parallel architectures. In Emerson, D., Ecer, A., Periaux, J., Satofuka, N., and Fox, P., editors, *Proceedings of the Parallel CFD'99 Conference, Manchester, 1997*, pages 225–238. Elsevier Science B.V.

Fukumori, I. and Melanotte-Rizzoli, P. (1995). An approximate Kalman filter for ocean data assimilation; an example with an idealized Gulf Stream model. *J. Geophys. Res.*, 100:6777–6793.

Gaspari, G. and Cohn, S. E. (1999). Construction of correlation functions in two and three dimensions. *Quart. J. Roy. Meteor. Soc.*, 125:723–757.

Gauthier, P., Courtier, P., and Moll, P. (1993). Assimilation of simulated wind lidar data with a Kalman filter. *Mon. Weather Rev*, 121:1803–1820.

Gery, M., Whitten, G., Killus, J., and Dodge, M. (1989). A photochemical kinetics mechanism for urban and regional scale computer modelling. *J. Geophys. Res.*, 94(D10):12.925–12.956.

Ghil, M., Cohn, S., Tavantzis, J., Bube, K., and Isaacson, E. (1981). Applications of estimation theory to numerical weather prediction. In Bengtsson, L., Ghil, M., and Källén, E., editors, *Dynamic Meteorology: Data Assimilation Methods*, pages 139–224. Springer-Verlag.

Giering, R. and Kaminski, T. (1998). Recipies for adjoint code construction. *ACM Trans. on Math. Software*, 24(4):437–474.

Golub, G. H. and van Loan, C. F. (1996). *Matrix computations*. The Johns Hopkins University Press, London, third edition.

Graedel, T. and Crutzen, P. (1993). Global change: the last several decades. In *Atmospheric Change, an earth system approach*, chapter 13, pages 251–275. W.H. Freeman, New York.

Haagen-Smit, A., Darley, E., Zaitlin, M., Hull, H., and Noble, W. (1951). Investigation on injury to plants from air pollution in the Los Angelos basin. *Plant Physiology*, 27:18–34.

Hasselmann, K. (1988). PIP's and POP's: The reduction of complex dynamical systems using principal interaction and oscillation patterns. *J. Geophys. Res.*, 93:11,015–11,021.

Heemink, A. and Kloosterhuis, H. (1990). Data assimilation for non-linear tidal models. *Int. J. for Numerical Methods in Fluids*, 11:1097–1112.

Heemink, A. and Segers, A. (2000). Modeling and prediction of environmental data in space and time using Kalman filtering. *submitted to Stochastic Environmental Research and Risc Assesment*.

Heemink, A., Verlaan, M., and Segers, A. (2001). Variance reduced Ensemble Kalman filtering. *Mon. Weather Rev*, 129(7):1718–1728.

Heemink, A. W. (1988). Two-dimensional shallow water flow identification. *Appl. Math. Modelling*, 12:109–118.

Henriksen, R. (1980). A correction of a common error in truncated second order nonlinear filters. *Modeling, Identification and Control*, 1(3):187–193.

Houtekamer, P. and Mitchell, H. L. (2001). A sequential Ensemble Kalman filter for atmospheric data assimilation. *Mon. Weather Rev*, 129(1):123–137.

Houtekamer, P. L. and Mitchell, H. L. (1998). Data assimilation using an Ensemble Kalman Filter technique. *Mon. Weather Rev*, 126:796–811.

Houweling, S. (2000). *Global Modeling of Atmospheric Methane Sources and Sinks*. PhD thesis, University of Utrecht, The Netherlands.

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

Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F. (1995). A new approach for filtering nonlinear systems. In *Proceedings of the 1995 American Control Conference*, pages 1628–1632, Seattle, Washington.

Kalman, R. E. (1960). A new aproach to linear filter and prediction theory. *J. of Basic Enginering.*, 82D:35–45.

Keppenne, C. L. (2000). Data assimilation into a primitive-equation model with a parallel Ensemble Kalman filter. *Mon. Weather Rev*, 128(6):1971–1981.

Keppenne, C. L. and Rienecker, M. M. (2000). Assimilation of temperature data into an ocean general circulation model with a parallel Ensemble Kalman filter. In *Proceedings of Third International Symposium on Assimilation of Observations in Meteorology and Oceanography (Québec City, Canada, 7–11 June 1999)*, number WMO/TD-986 in WMO Technical Documents, pages 17–20. World Meteorological Organization.

Khattatov, B., Gille, J., Lyjak, L., Brasseur, G., Dvortsov, V., Roche, A., and Waters, J. (1999). Assimilation of photochemically active species and a case analysis of UARS data. *J. Geophys. Res.*, 104(D15):18,715–18,738.

Kley, D., Geiss, H., and Mohnen, V. (1994). Concentrations and trends of tropospheric ozone and precursor emissions in the United States and Europe. In Calvert, J., editor, *The Chemistry of the Atmosphere: Its Impact on Global Change*, IUPAC, Chemistry for the 21st Century, pages 245–259. Blackwell Scientific Publications, Oxford.

Kuhn, M., Builtjes, P., Poppe, D., Simpsons, D., Stockwell, W., Andersson-Sköld, Y., Baart, A., Das, M., Fiedler, F., Hov, O., Kirchner, F., Makar, P., Milford, J., Roemer, M., Ruhnke, R., Strand, A., Vogel, B., and Vogel, H. (1998). Intercomparison of the gas-phase chemistry in several chemistry and transport models. *Atmos. Environment*, 32(4):693–709.

Lanser, D. and Verwer, J. (1999). Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modelling. *J. Comp. Appl. Math.*, 111:201–216.

Lenhart, L., Heymann, M., and Friedrich, R. (1995). The temporal variation of emission data and the GENEMIS project. Technical report, Institute of Energy Economics and the Rational Use of Energy, Stuttgart University, Germany.

Lermusiaux, P. and Robinson, A. (1999a). Data assimilation via Error Subspace Statistic. Estimation. Part I: Theory and schemes. *Mon. Weather Rev*, 127:1385–1407.

Lermusiaux, P. and Robinson, A. (1999b). Data assimilation via Error Subspace Statistical Estimation. Part II: Middle atlantic bright shelfbreak front simulations and ESSE validation. *Mon. Weather Rev*, 127:1408–1432.

Lin, X., Trainer, M., and Liu, S. (1988). On the nonlinearity of tropospheric ozone producion. *J. Geophys. Res.*, 93:15879–15888.

Liu, S., Trainer, M., Fehnsenfeld, F., Parrish, D., Williams, E., Fahey, D., Hubler, G., and Murphy, P. (1987). Ozone production in the rural troposphere and implications for regional and global ozone distributions. *J. Geophys. Res.*, 92:4191–4207.

Lyster, P., Cohn, S., Ménard, R., Chang, L.-P., Lin, S.-J., and Olsen, R. G. (1997). Parallel implementation of a Kalman filter for constituent data assimilation. *Mon. Weather Rev*, 125(7):1674–1686.

Matthijsen, J. (1995). *Modelling of tropospheric ozone and clouds*. PhD thesis, University of Utrecht, The Netherlands.

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

Ménard, R., Chang, L.-P., and Larson, J. W. (1999). Application of a robust chi-square validation diagnostic in PSAS and Kalman filtering experiments. In *Proceedings of Third International Symposium on Assimilation of Observations in Meteorology and Oceanography (Québec City, Canada, 7–11 June 1999)*, number WMO/TD-986 in WMO Technical Documents, pages 5–8. World Meteorological Organization.

Miller, R. N., Ghil, M., and Gauthiez, F. (1994). Advanced data assimilation in strongly nonlinear dynamical systems. *J. Atmos. Sci.*, 51:1037–1056.

Mitchell, H. L. and Houtekamer, P. (2000). An Adaptive Ensemble Kalman Filter. *Mon. Weather Rev*, 128(2):416–433.

Noordijk, H. (1994). The national smog warning system in the Netherlands: a combination of measuring and modelling. In *Air Pollution '94*. Wessex Institute of Technology.

Owcarz, W. and Zlatev, Z. (2000). Running a large air pollution model on an IBM SMP computer. In Barone, G., Builtjes, P. J. H., and Giunta, G., editors, *Proceedings of the 3^{rd} GLOREAM Workshop, Ischia (Naples), september 22-24, 1999*, pages 153–177, Italy. Naval University of Naples.

Parrish, D. and Cohn, S. (1985). A Kalman filter for a two dimensional shallow-water model: Formulation and preliminary experiments. Office note 304, New York University.

Pham, D. T. (1996). A Singular Evolutive Interpolated Kalman Filter for data assimilation in oceanography. Technical Report, IDOPT project INRIA-CNRS-UJF-INGP RT 163, september 1996, Laboratoire de Modeélisation et Calcul, France, Grenoble Cédex, France.

Pham, D. T. (1997). Dimension, predictibility and reduced rank Kalman filtering in data assimilation. In *Proceeding of Third Bilateral Franco-Russian Conference*. French-Russian A.M.Liapunov Institute in Computer Science and Applied Mathematics.

Pham, D. T., Verron, J., and Roubau, M. C. (1998). A Singular Evolutive Extended Kalman filter for data assimilation in oceanography. *J. of Marine Systems*, 16(3–4):323–340.

Rijshøjgaard, L. P. and Källén, E. (1997). On the correlation between ozone and potential vorticity for large-scale Rossby waves. *J. Geophys. Res.*, 102(D7):8,793–8,804.

Robertson, L., Langner, J., and Engardt, M. (1999). An Eulerian limited-area atmospheric transport model. *J. Applied Meteorology*, 38:190–210.

Robinson, A. R., Lermusiaux, P. F., and Quincy Sloan III, N. (1998). Data assimilaton. In Brink, K. H. and Robinson, A. R., editors, *The Sea*, volume 10, chapter 20, pages 541–594. John Wiley & Sons, Inc.

Roemer, M. (1996). *Trends of tropospheric ozone over Europe*. PhD thesis, University of Utrecht.

Roemer, M. and van den Hout, K. (1992). Emissions of NMHCs and NOx and global ozone production. In *Proceedings of the 19th NATO/CCMS International technical meeting on Air Pollution Modeling and its application, September 29 – October 4, 1991, Crete, Greece*, pages 405–414, New York. Plenum Press.

Rostaing, N., Dalmas, S., and Galligo, A. (1993). Automatic differentiation in O∂yssée. *Tellus*, 45 A(5):558–568.

Segers, A. and Heemink, A. (2002). Parallelization of a large scale Kalman filter: comparison between mode and domain decomposition. In Wilders, P., Ecer, A., Periaux, J., Satofuka, N., and Fox, P., editors, *Parallel Computational Fluid Dynamics: Recent developments and applications, Proceedings of the Parallel CFD'01 Conference*. Elsevier.

Segers, A., Heemink, A., Verlaan, M., and van Loon, M. (2000a). A modified rrsqrt-filter for assimilating data in atmospheric chemistry models. *Environmental Modelling and Software*, 15(6–7):663–671.

Segers, A., Heemink, A., Verlaan, M., and van Loon, M. (2000b). Nonlinear Kalman filters for atmospheric chemistry models. In Kasibhatla, P., Heimann, M., Rayner, P., Mahowald, N., Prinn, R. G., and Hartley, D. E., editors, *Inverse Methods in Global Biogeochemical Cycles*, volume 114 of *Geophysical Monograph*, pages 139–146. American Geophysical Union.

Segers, A., van Loon, M., and Builtjes, P. (2000c). Data assimilation based on a Kalman filter approach. In Barone, G., Builtjes, P. J. H., and Giunta, G., editors, *Proceedings of the 3rd GLOREAM Workshop, Ischia (Naples), september 22-24, 1999*, pages 107–118, Italy. Naval University of Naples.

Seibert, P., Beyrich, F., Gryning, S., Joffre, S., Rasmussen, A., and Tercier, P. (2000). Review and intercomparsion of operational methods for the determination of the mixing height. *Atmos. Environment*, 34(7):1001–1027.

Strang, G. (1968). On the construction and comparison of difference schemes. *SIAM J. of Numerical Analysis*, 5:506–517.

Stull, R. B. (1988). *An introduction to boundary layer meteorology*. Amospheric Science Library. Kluwer Academic Publishers.

Talagrand, O. and Courtier, P. (1987). Variational assimilation of meteorological observations with the adjoint vorticity equation, I, theory. *Quart. J. Roy. Meteor. Soc.*, 113:1311–1328.

Thompson, A. M. and Stewart, R. W. (1991). Effect of chemical kinetics uncertainties on calculated constituents in a tropospheric photochemical model. *J. Geophys. Res.*, 96(D7):13089–13108.

Tilmes, S. (1999). *Verfahren zur Analyze vom Mesungen atmosphärischer Spurengase mit dem Ziel der Assimilation in Chemie-Transportmodellen*. PhD thesis, Mathematisch-Naturwissenschaftlichen Fakultät der Universität Köln, Deutscher Wetterdienst, Frankfurter Str. 135, 63067 Offenbach am Main.

Tilmes, S. and Zimmermann, J. (1998). Investigation on the spatial scales of the variability in measured near-ground ozone mixing ratios. *J. Geophys. Res.*, 25(20):3827–3830.

van Leeuwen, P. J. (1998). Comment on “data assimilation using an Ensemble Kalman Filter technique”. *Mon. Weather Rev*, 127:1374–1377.

van Loon, M. (1996). *Numerical Methods in Smog Prediction*. PhD thesis, University of Amsterdam, The Netherlands.

van Loon, M., Builtjes, P., and Segers, A. (2000). Data assimilation applied to LOTOS: First experiences. *Environmental Modelling and Software*, 15(6–7):603–609.

Velders, G., Matthijsen, J., v. Loon, M., van Oss, R., Sauter, F., and Segers, A. (2001). Smog forecasts with a chemistry-transport model using data assimilation; possibilities of GOME tropospheric ozone observations. NRSP-2 report 00-38, RIVM.

Verlaan, M. (1998). *Efficient Kalman Filtering Algorithms for Hydrodynamical models*. PhD thesis, Delft University of Technology.

Verlaan, M. and Heemink, A. (1996). Data assimilation schemes for non-linear shallow water flow models. In M. Rahman, C. B., editor, *Advances in Fluid Mechanics 96, New Orleans*, pages 277–286, 25 Bridge st,Billerica,MA 01821,USA. Wessex Institue of Technology, Computational Mechanics Publications.

Verlaan, M. and Heemink, A. (2001). Non-linearity in data assimilation applications: a practical method for analysis. *Mon. Weather Rev*, 129(6):1578–1589.

Verlaan, M. and Heemink, A. W. (1995). Reduced Rank Square Root Filters for large scale data assimilation problems. In *Second International Symposium on Assimilation of Observations in Meteorology and Oceanography*, pages 247–252. World Meteorological

Organization, WMO.

Verlaan, M. and Heemink, A. W. (1997). Tidal Flow Forecasting using Reduced Rank Square Root Filters. *Stoch. Hydrology and Hydraulics*, 11(5):349–368.

Verron, J., Gourdeau, L., Pham, D., Murtugudde, R., and Busalacchi, A. (1999). An Extended Kalman fiter to assimilate satellite altimeter data into a nonlinear numerical model of the tropical pacific ocean: Method and validation. *J. Geophys. Res.*, 104(C3):5441–5458.

Voorrips, A. C., Heemink, A. W., and Komen, G. J. (1999). Wave data assimilation with the Kalman filter. *J. of Marine Systems*, 18:267–291.

Vossepoel, F. C. (1999). *Sea-level data assimilation for estimating salinity variability in the tropical Pacific*. Ph.d. thesis, Delft University of Technology.

Wang, K., Lary, D., Shallcross, D., Hall, S., and Pyle, J. (2001). A review of the use of the adjoint method in four-dimensional atmospheric-chemistry data assimilation. *Quart. J. Roy. Meteor. Soc.*, 127:2181–2204.

Zhang, X.-F. (1996). *Data assimilation in air pollution modelling*. PhD thesis, Delft University of Technology.

Zhang, X.-F., Heemink, A., Janssen, L., Janssen, P., and Sauter, F. (1999). A computationally efficient Kalman smoother for the evaluation of the CH4 budget in europe. *Appl. Math. Modelling*, 23(2):109–129.

Summary

Data assimilation in atmospheric chemistry models using Kalman filtering

The problem of air pollution around urbanized area across Europe is strongly related to tropospheric ozone. Tropospheric ozone is a result of photo-chemical oxidation, and therefore an indication of the presence of pollutants. Overexposure to ozone is harmful to the health of humans, animals, and vegetation, and the concentrations are therefore measured on a regular basis to check exceedence of air-quality guidelines. Models have been developed to simulate the ozone formation, for example to make a forecast of the air quality for the coming days, or to study underlying mechanisms. Where possible, the models are validated with the measurements. A new direction in air-pollution modeling is data assimilation: merging model simulations and measurements in a single procedure. The target of a data assimilation problem is to decrease the difference between models and measurements, and with this, improvement of the simulations for which measurements are not available.

This study describes the development of a data assimilation tool for the air pollution model LOTOS, based on a Kalman filter. The LOTOS (LOng Term Ozone Simulation) model computes hourly concentrations of pollutants for the area of Europe, representative for suburban and remote sites. A detailed description of the chemistry and other operators in LOTOS is given in chapter 2. Ground-based measurements of ozone are available on a regular basis. Given stochastic models for the model error in LOTOS and the representation error of the measurements, a Kalman filter is able to compute an optimal estimate of the pollutant concentrations in terms of a mean and covariance. The background and definition of the Kalman filter and other data assimilation approaches is given in chapter 3. The core of this work is split into two parts: the application of the developed filter to LOTOS is described in chapters 4 and 5, while the actual implementation is left for chapters 6 to 8.

Chapter 4 describes the result of the Kalman filter experiments with a small-scale version of LOTOS (domain limited to England and Wales). During these experiments, a number of different model parameters has been defined as uncertain, to examine their usage in a stochastic model. Variations in ozone level due to uncertainties in emissions of nitrogen oxides (NO_x) and volatile organic compounds (VOC) were shown to be compensated for by assimilation of ozone (O_3) measurements. The tight chemical coupling between these components ensures that variations in the emissions are visible in the measurements too. Other useful parameters to consider stochastic were photolysis rates. The photolysis rates of O_3 and NO_2 have a large impact on the height of the afternoon ozone peak, and are uncertain by their deterministic value, cloud cover, and absorption of sunlight at higher altitudes. For variations in the nighttime ozone level, it was found useful to consider the deposition velocity of ozone a stochastic parameter. Together the uncertainties in all these parameters are able to explain the differences between model and measurements for both suburban and remote sites, and during daytime as well as nighttime hours.

Chapter 5 describes the application of the Kalman filter to a full-scale version of LOTOS, with a domain covering west and central Europe, and for a time period of one month. The stochastic model is based on the uncertain emissions, photolysis rates, and deposition velocities found in chapter 5. The assimilated ozone fields represent the measurement data up to an average difference of 10-15 ppb. Afternoon ozone maxima are estimated most accurately with a difference up to 7–12 ppb, which is a decrease of 5 ppb in comparison to the deterministic model. The assumed variance of the representation error was estimated adaptively, to prevent the filter from overestimating the degree of freedom in the stochastic model. A Kalman smoother has been implemented to be able to estimate the value of the uncertain parameters and to obtain insight in how the filter uses the degree of freedom in the stochastic model. The parameter estimations show that the filter especially uses the degree of freedom in deposition, for example to increase the night time simulations of ozone, which systematically underestimate the measurements. The degree of freedom in emissions is only used to explain ozone variations near the industrialized areas, where, for example, the uncertainty in NO_x emissions was used to decrease the ozone level in the morning. The model parameters estimated with the filter were shown to improve the model simulations significantly. If the filter is used to provide initial conditions for an ozone forecast, the ozone maxima for the coming day are estimated 25% more accurate when the estimated parameters are included in the initial condition.

The second part of this thesis describes the actual implementation of the Kalman filter. The filter developed for LOTOS takes the form of a low-rank filter. This type of approximate Kalman filters are suitable for application to models with large state vectors. A number of commonly used low-rank filters were discussed in chapter 6: RRSQRT, SEIK, ESSE, ENKF, and in addition the POENK filter, which combines RRSQRT and ENKF. Although these filters have been developed from different theoretical backgrounds, their actual implementation is quite similar. All methods use a low-rank parameterization of the covariance matrix, and propagate the covariance structure in time using a large number of model evaluations (40–100). A RRSQRT filter using the forecast step of the SEIK filter was shown to be the best choice for the filter around LOTOS, with a few modifications focussing on the characteristics of a chemistry model.

One of the difficulties associated with the application of a Kalman filter comes from the fact that it was originally designed for linear models. The LOTOS model is strongly nonlinear however, due to the chemistry. A number of methods for treatment of nonlinearities is discussed in chapter 7. The methods are either based on linearizations or on ensemble statistics, and all require additional model evaluations for increased accuracy. The theoretical performance of the nonlinear methods has been analyzed using Taylor expansions, and was examined in practice for filter experiments with the LOTOS model. The method of minimal exact sampling (MES) as introduced for the SEIK filter was shown to be not only the most accurate but also the cheapest method, both in theory and in practice, during the filter experiments. A second-order accurate result is obtained at almost the same cost as that of a first-order result. A higher accuracy could be reached with an ensemble forecast, the basic approach of the ENKF. Ensemble methods suffer seriously from statistical noise, however, which can only be reduced by increasing the number of model evaluations in the filter.

Chapter 8 describes the implementation of the filter on a parallel computer. Application of the developed Kalman filter to LOTOS is expensive because 40-100 model evaluations are required for propagation of the error covariance. The filter has therefore been implemented on a parallel computer in two different ways. In a mode-decomposed parallelization, each processor is equipped with a complete copy of the model. Independent model evaluations are parallelized efficiently in this way, and since these form the major costs of the filter, the reduction of the computation time is almost optimal. The linear algebra operations in the filter are less efficient since these require much communication between the processors. On a platform with fast communication as used in this study (CRAY T3E, a massive parallel machine), the decrease of efficiency is small, however. The second approach tested is based on parallelization of the model rather than parallelization of the filter. The LOTOS model is efficiently parallelized using a domain decomposition, such that the model can be evaluated multiple times very fast. Building a filter around a parallel model is rather simple, and offers the possibility of efficient assimilation of large numbers of measurements. For assimilation runs as held in this research, the domain-decomposed filter is slightly favored over the mode-decomposed filter because of the simple implementation and slightly better speedup.

Arjo Segers, December 2001

Samenvatting

Data-assimilatie in atmospheer-chemie modellen met behulp van Kalman filteren

Het probleem van luchtverontreiniging in stedelijke gebieden is sterk gerelateerd aan tropospherisch ozon. Tropospherisch ozon ontstaat als gevolg van foto-chemische afbraak, en is daarmee een indicatie voor de aanwezigheid van vervuilende stoffen. Verhoogde blootstelling aan ozon is schadelijk voor de gezondheid van mensen, dieren en planten, en de concentraties worden daarom voortdurend gemeten om overschreiding van richtlijnen voor luchtkwaliteit waar te nemen. Modellen zijn ontwikkeld om het ontstaan van ozon te simuleren, bijvoorbeeld om de luchtkwaliteit voor de komende dagen te voorspellen, of om de onderliggende reacties te bestuderen. Waar mogelijk zijn de modellen gevalideerd met de metingen. Een nieuwe trend bij modellering van luchtverontreiniging is data assimilatie: samenvoegen van simulaties en metingen in één en dezelfde procedure. Het doel van data assimilatie is verkleinen van het verschil tussen modellen en metingen, en daarmee simulaties te verbeteren voor plekken waar geen metingen beschikbaar zijn.

Deze studie beschrijft de ontwikkeling van een data-assimilatie tool rondom het luchtverontreiniging model LOTOS, gebaseerd op een Kalman filter. Het LOTOS (LOng Term Ozone Simulation) model berekent uurlijkse concentraties van vervuilende stoffen boven Europa, representatief voor verstedelijkte en afgelegen gebieden. Een gedetailleerde beschrijving van de chemische en andere operatoren in LOTOS is gegeven in hoofdstuk 2. Grond metingen van ozon zijn beschikbaar op regelmatige basis. Gegeven een stochastisch model voor de modelfout in LOTOS en de representatie fout in de metingen is het Kalman filter in staat om een optimale schatting te maken van de concentraties vervuilende stoffen, in termen van een gemiddelde en een covariantie. De achtergrond en definitie van het Kalman filter en andere data-assimilatie methoden is gegeven in hoofdstuk 3. De kern van dit proefschrift kan verdeeld worden in twee stukken: de toepassing van het ontwikkelde filter op LOTOS is beschreven in hoofdstukken 4 en 5, terwijl de daadwerkelijke implementatie is beschreven in de hoofdstukken 6 tot en met 8.

Hoofdstuk 4 beschrijft de resultaten van Kalman filter experimenten met een verkleinde versie van LOTOS (domein beperkt tot Engeland en Wales). Tijdens deze experimenten zijn aan verschillende model parameters onzekerheden toegekend, om zo hun geschiktheid voor een stochastisch model te onderzoeken. De resultaten toonden dat variaties in ozon niveaus door onzekerheden in emissies van stikstof oxiden (NO_x) en vluchtige organische stoffen (VOS) gecompenseerd kunnen worden door assimilatie van ozon metingen (O_3). De sterke chemische koppeling tussen deze componenten verzekerd dat variaties in de emissies ook zichtbaar zijn in de metingen. Een andere bruikbare groep model parameters om onzekerheden aan toe te kennen zijn fotolysesnelheden. De fotolysesnelheden van O_3 en NO_2 zijn van grote invloed op de hoogte van het ozon maximum in de middag, en zijn onzeker met betrekking tot absolute grootte, wolken bedekking, en absorptie van zonlicht hoger in de atmosfeer. Voor variaties in het nachtelijke ozon niveau is de depositie snelheid van ozon een bruikbare parameter gebleken om onzekerheden aan toe te kennen. De onzekerheden in al deze parameters te samen zijn in staat om verschillen tussen model en metingen te verklaren voor verstedelijkte en meer afgelegen gebieden, zowel overdag als tijdens de nacht.

Hoofdstuk 5 beschrijft de toepassing van het Kalman filter op een volledige versie van LOTOS, met een domein met west en midden Europa en voor een tijds periode van 1 maand. Het stochastische model is gebaseerd op onzekere emissies, fotolyse snelheden en depositie zoals verkregen in hoofdstuk 5. De geassimileerde ozon velden zijn in staat om de metingen te representeren met een gemiddeld verschil van 10-15 ppb. De ozon maxima in de middag worden het meest nauwkeurig geschat met een verschil van 7–12 ppb, wat een afname van 5 ppb is in vergelijking met het deterministische model. De variantie van de representatie fout is adaptief bepaald, om te voorkomen dat het filter de vrijheidsgraad van het stochastisch model overschat. Een Kalman smoother is geïmplementeerd om de waarde van de onzekere model parameters te kunnen schatten en inzicht te krijgen in hoe het filter de vrijheidsgraad in het stochastisch model gebruikt. De parameter schattingen tonen dat met name de vrijheidsgraad in depositie wordt gebruikt door het filter, hoofdzakelijk om de ozon simulaties tijdens de nacht te verhogen omdat deze systematisch worden onderschat door het model. De vrijheidsgraad in emissies is alleen gebruikt om ozon variaties te verklaren rondom de belangrijkste industrie gebieden, waar bijvoorbeeld de onzekerheid in NO_x is gebruikt om het ozon niveau in de ochtend te verlagen. De met het filter geschatte parameters zijn in staat om de model simulaties te verbeteren. Als het filter wordt gebruikt om begin condities voor een ozon voorspelling te optimaliseren worden de ozon maxima voor de volgende dag 25% nauwkeuriger voorspeld als schattingen van model parameters in de begin toestand worden opgenomen.

Het tweede deel van dit proefschrift beschrijft de daadwerkelijke implementatie van het Kalman filter. Het filter ontworpen voor LOTOS heeft de vorm van een lage-rang filter. Dit type Kalman filters is geschikt voor toepassing op modellen met grote toestands vectoren. Een aantal veel gebruikte lage-rang filters zijn besproken in hoofdstuk 6: RRSQRT, SEIK, ESSE, en ENKF, en tot slot het POENK filter, wat een combinatie is van RRSQRT en ENKF. Hoewel deze ontwikkeld zijn vanuit verschillende theoretische achtergronden, vertonen de uiteindelijke implementaties grote overeenkomsten. Alle methoden gebruiken een lage-rang parameterisatie van de covariantie matrix, en propageren de covariantie structuur in tijd met behulp van een groot aantal model evaluaties (40–100). Een RRSQRT filter met de propagatie stap van het SEIK filter bleek de beste keuze voor het filter rond LOTOS, met enkele aanpassingen voor specifieke eigenschappen van een chemie model.

Een moeilijkheid bij toepassing van een Kalman filter is dat het oorspronkelijk is ontworpen voor lineaire modellen. Het LOTOS model is echter sterk niet-lineair ten gevolge van de chemie. Een aantal methoden voor het behandelen van niet-lineariteiten is besproken in hoofdstuk 7. De besproken methoden zijn ofwel gebaseerd op linearisaties of op ensemble methoden, en vereisen extra model evaluaties voor extra nauwkeurigheid. De theoretische nauwkeurigheid van de niet-lineaire methodes is onderzocht met behulp van Taylor reeksen, en getest in de praktijk met filter experimenten met het LOTOS model. De methode van minimaal exacte samples (MES) zoals geïntroduceerd voor het SEIK filter bleek de meest nauwkeurige maar ook goedkoopste methode, zowel in theorie als in praktijk tijdens de filter experimenten. Een tweede orde nauwkeurig resultaat is verkregen voor bijna dezelfde kosten als vereist voor een eerste orde resultaat. Een nog hogere nauwkeurigheid zou eventueel bereikt kunnen worden met een ensemble voorspelling, het basis idee achter het ENKF. Ensemble methoden ondervinden echter in belangrijke mate last van statistische ruis, wat alleen onderdrukt kan worden door het aantal model evaluaties te verhogen.

Hoofdstuk 8 beschrijft de implementatie van het filter op een parallelle computer. Toepassing van het ontwikkelde Kalman filter op LOTOS is duur door het aantal van 40-100 model evaluaties vereist voor de propagatie van de fouten covariantie. Het filter is daarom geïmplementeerd op een parallelle computer op twee verschillende manieren. In een parallellisatie gebaseerd op een mode-decompositie bevat iedere processor een kopie van het complete model. Onafhankelijke model evaluaties worden op deze manier efficiënt geparallelliseerd, en omdat deze in het filter de belangrijkste kostenpost vormen is de reductie van de rekentijd bijna maximaal. De lineaire algebra operaties in het filter zijn minder efficiënt omdat deze veel communicatie tussen de processoren vereisen. Op een platform met snelle communicatie zoals gebruikt in deze studie (CRAY T3E, een massaal parallelle machine) is de afname in efficiëntie echter beperkt. De tweede geteste methode is gebaseerd op parallellisatie van het model in plaats van het filter. Het LOTOS model kan efficiënt worden geparallelliseerd met een domein-decompositie, zodat het model snel een groot aantal malen geëvalueerd kan worden. Een filter bouwen rondom een parallel model is betrekkelijk eenvoudig. Bovendien biedt een domein-decompositie de mogelijkheid om grote hoeveelheden data efficiënt te assimileren. Voor de assimilatie experimenten zoals gehouden in deze studie is de domein-decompositie lichtelijk favoriet boven de mode-decompositie door een betere efficiëntie en de eenvoudige implementatie.

Arjo Segers, december 2001

Dankwoord

Met het voltooien van dit proefschrift is het tijd om een aantal mensen te bedanken die van invloed zijn geweest op de totstandkoming er van. Verspreid over Delft, Apeldoorn en Bilthoven zijn dat vele collega's en vakgenoten, met wie ik vol- of deeltijds heb mogen samenwerken. De stelling dat 'afwisseling van werkplek tot creatiever onderzoek leidt' (F. Vossepoel, 1999) is gedurende de afgelopen jaren uitermate verdedigbaar gebleken.

De eersten om te bedanken zijn uiteraard de promotoren Arnold Heemink en Peter Builtjes, voor het op de rails zetten van en richting geven aan het onderzoek. De combinatie van techniek (Arnold) en toepassing (Peter) was uitermate inspirerend om iets te ontwikkelen dat zowel wiskundig als praktisch interessant is. Waarmee maar weer bewezen is dat dit niet per definitie onmogelijk is. Het uitgebreide netwerk van beiden heeft bovendien geleid tot vele contacten met vakgenoten, op al dan niet exotische plaatsen.

Van alle collega's aan de TU wil ik als eerste Martin Verlaan bedanken voor de begeleiding in de wondere wereld van het Kalman filter. Zijn aanstekelijke enthousiasme voor nieuwe mogelijkheden van verbetering, uitbreiding, en toepassing hebben me doen inzien dat de techniek na 40 jaar nog lang niet uitgeëvolueerd is. Als enige kamergenoot heeft hij het bovendien de volledige vier jaar uitgehouden (in deeltijd weliswaar, maar toch). Een rijke schare aan andere kamergenoten is daarnaast nog de revue gepasseerd: Karin (wier aanstekelijke enthousiasme toch een belangrijk argument was om het aio-schap te aanvaarden), Jan (die het wel meer dan 2 jaar uithield), Duncan (slechts 2 weken), Liedwien, en Fahmi. Verder nog een woord van dank aan alle collega's van de voorheen zevende/achtste verdieping voor het koffie- dan wel lunchleuten over de bijzaken des levens (naast werk). Ik zou hier nog een collega met name kunnen noemen, maar iedereen verwacht dat ik dat later nog doe, dus dat doe ik hier maar niet. Tot slot, Mirjam Nieman bedankt voor de engelse correctie, die de leesbaarheid van dit proefschrift aanmerkelijk vergroot heeft.

Alleen al vanwege de wekelijkse treinreis over de Veluwe was het een waar genoegen om met TNO in Apeldoorn te mogen samenwerken. In het bijzonder Maarten van Loon bedankt voor het toegankelijk maken van LOTOS (sorry voor het kompleet overhoop halen van de code; dat leer ik nooit af, ben ik bang). De discussies over waarom iets op een bepaalde manier, en waarom eigenlijk überhaupt gemodelleerd is, vormden een nuttige cursus. Daarnaast nog een woord van dank aan Michiel Roemer, die bij iedere rare uitkomst wel weer een reactie mechanisme wist te bedenken waardoor alles in eens weer heel logisch werd; in atmosfeerchemie is alles verklaarbaar.

De wekelijkse bezoeken aan het RIVM gedurende het laatste jaar waren qua reis uiteraard minder inspirerend dan die aan TNO, maar de indrukwekende toegangscontrole maakte veel goed. Werken met het EUROS model gaf bovendien het geruststellende gevoel dat LOTOS eigenlijk helemaal zo gek nog niet was. Guus Velders bedankt voor de nu reeds legendarische project naam (al zullen weinigen weten waar STROPDAS ook al weer de afkorting van was), Ferd Sauter voor de scriptjes en landkaartjes, en natuurlijk Jan Matthijsen voor de fotolyse cursus en de analyse van de Bilthovense huizenmarkt (nog maar even in Delft blijven wonen). Afscheid nemen van het RIVM hoeft gelukkig niet echt, want via Remus zal ik ook de komende jaren nog wel enigzins kijk houden op het wel en wee bij LLO.

Na de TU, TNO, en het RIVM is er natuurlijk nog maar één instituut waar je ooit geweest moet zijn: het KNMI! De Koninklijke wordt bij deze bedankt voor de pre-doctorale opvang; ik kijk nu al uit naar de post-doctorale periode. Collega ASers: bedankt voor het niet al te nadrukkelijk vragen 'of het al af is'. Het moment waarop je dit volmondig bevestigend kan beantwoorden is overigens wel een opluchting.

Nog meer mensen? Tuurlijk! Ouders, broer, vrienden en anderen die voor de nodige afleiding gezorgd hebben. En gelukkig ook al niet al te veel gezeurd hebben 'of die som nou al af is'. De paranimfen, Jorg Benningshof en Suzanne van Dalen, voor de bereidwilligheid om aan de protocolaire zaken mede deel te nemen (dan ben ik in ieder geval niet de enige). Tot slot, Suzanne natuurlijk nog even speciaal. De afgelopen jaren waren in veel opzichten bijzonder, en in ieder geval anders dan we in het begin vermoed hadden kunnen hebben. Het heeft je er niet van weerhouden om voor korte tijd collega te worden; de indruk die ik je gegeven heb van het aio-bestaan was blijkbaar positief genoeg. Bedankt voor je geduld bij het afronden; nu zit het er toch echt op. Tijd voor andere dingen!

Arjo Segers, december 2001

Curriculum Vitae

Arjo Segers werd geboren op 19 juli 1974 in Gouda, maar groeide op in het onbetwiste dieptepunt van Nederland: Nieuwerkerk aan de IJssel. Van 1986 tot en met 1992 volgde hij het atheneum aan het Emmauscollege in Rotterdam. Aan de Technische Universiteit Delft volgde hij daarna de studie Technische Wiskunde, eind 1996 met lof afgesloten bij de vakgroep Toegepaste Analyse. Voor het afstuderen werd onderzoek gedaan aan een model voor lichtverstrooiïng in biologische weefsels.

Van 1997 tot en met januari 2001 was hij als assistent in opleiding verbonden aan de Technische Universiteit Delft. Bij de leerstoel Wiskundige Analyse van Grootschalige Modellen van prof. A.W. Heemink verrichte hij onderzoek gedaan aan assimilatie van metingen in luchtverontreinigingsmodellen. Bij dit onderzoek werd nauw samengewerkt met de afdeling Milieukwaliteit en Analyse van TNO-MEP in Apeldoorn, en het Laboratorium voor Luchtonderzoek van het RIVM te Bilthoven. De resultaten van het onderzoek zijn beschreven in dit proefschrift.

Vanaf februari 2001 is hij in dienst van het KNMI in De Bilt als onderzoeker op het gebied van assimilatie van satellietmetingen in globale modellen.